An Open-source High Speed C++/ MEX Framework for the Detection and Delineation of Long-duration Ambulatory Holter ECG Events: HSEDF

Article · April 2012

CITATIONS
0

READS
124

3 authors, including:

Milad Khazraee
University of Waterloo
12 PUBLICATIONS **110** CITATIONS
SEE PROFILE

Moein Khazraee
University of California, San Diego
12 PUBLICATIONS **69** CITATIONS
SEE PROFILE

An Open-source High Speed C++/ MEX Framework for the Detection and Delineation of Long-duration Ambulatory Holter ECG Events: HSEDF

Mohammad. R. Homaeinezhad[1,2,3], Milad Khazraee[1,2], Moein Khazraee[2,4]

[1]Department of Mechanical Engineering, K. N. Toosi University of Technology, Tehran, Iran

[2]CardioVascular Research Group (CVRG), K. N. Toosi University of Technology, Tehran, Iran

[3]Non-invasive Cardiac Electrophysiology Laboratory (NICEL), DAY General Hospital, Tehran, Iran

[4]Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran

Abstract-**The major focus of this study is to describe the construction procedure of an open-source C++/Mex solution for the detection and delineation of long-duration ambulatory holter electrocardiogram (ECG) individual events. To this end, detailed flow-chart of the proposed ECG detection-delineation algorithm as well as the corresponding lines in the C++/Mex code are explained. In addition, two methodologies for generating the detection-delineation decision statistic (DS) named as Area-Curve Length Method (ACLM) and Geometrical Index Method (GIM) and their appropriate programming implementations are explained. The presented code has been evolved via application of it to several databases such as MIT-BIH Arrhythmia Database, QT Database, and T-Wave Alternans Database. As a result, the average values of sensitivity and positive predictability, Se = 99.96% and P+ = 99.96% were obtained for the detection of QRS complexes, with the average maximum delineation error of 5.7 msec, 3.8 msec and 6.1 msec for P-wave, QRS complex and T-wave respectively. In addition, the code was applied to DAY general hospital high resolution holter data (more than 1,500,000 beats including Bundle Branch Blocks-BBB, Premature Ventricular Complex-PVC and Premature Atrial Complex-PAC) and average values of Se=99.98% and P+=99.98% were obtained for QRS detection. Marginal performance improvement of ECG events detection-delineation process in a widespread range of signal to noise ratio (SNR), reliable robustness against strong noise, artifacts, probable severe arrhythmia(s) of high resolution holter data and the processing speed 185,000 samples/sec can be mentioned as important merits and capabilities of the presented C++/Mex solution.**

Keywords-ECG Detection-delineation; Discrete Wavelet Transform; C++/mex Programming; Detection-delineation Decision Statistic; Curve Length; Variance; First-order Derivative; Second-order Derivative.

I. INTRODUCTION

Heart is a special electro-mechanical organ, the constitutive cells (myocytes) of which possess two important characteristics - nervous excitability and mechanical tension with force feedback. The superposition of all myocytes electrical activity on the skin surface results in a detectable potential difference that its detection and registration is called electrocardiography [1]. If incidents happen, electro-mechanical function of a region of myocytes encounters with failure, the corresponding abnormal effects would appear in the ECG signal and in the heart of hemodynamic performance. Statistical analyses of ECG parameters in long-term occasions can yield successful and prompt solutions for diagnosis of some certain phenomena such as T-Wave Alternans (TWA) [2,3], Atrial Fibrillation (AF) [4,5] and QT-prolongation [6]. Furthermore, proper delineation of ECG waveforms can help to achieve more accurate results in applications such as pattern recognition or arrhythmia clustering and classification [7,8]. Therefore, parameterization and detection of ECG signal events using a reliable algorithm are the first stage of the computer analysis of the ECG signal. Numerous approaches have yet been developed to detect the ECG events including mathematical models [9] , Hilbert transform and the first derivative [10,12], statistical higher order moments [13], second order derivative [14], wavelet transform and the filter banks [15-17], soft computing (Neuro-fuzzy, genetic algorithm) [18] and Hidden Markov Models (HMM) application [19]. The performance of QRS detection algorithms can easily be verified by utilizing the standard databases such as MIT-BIH Arrhythmia Database [20]. However, validation of a proposed algorithm for the detection-delineation of P and T-waves has turned to a difficult problem due to the lack of a gold standard as universal reference [15]. The algorithms which were already developed in this area such as [15-17] accomplish satisfactory results in the area of QRS detection, localization of J and fiducial points, the beginning of P-wave as well as the peak and end of T-wave. Applying some modifications to these methods, more innovative and more accurate approaches can be developed in the area of wave detection and delineation. As it will be shown in this study, by adding some innovations and modifications to previous methods, it would be possible to apply them to more challenging data including ambulatory holter ECG containing high-level noise and strong motion artifacts as well as severe arrhythmia with abnormal morphologies such as PVCs, PACs, a combination of ectopies and multifocal PVCs with complicated morphologies. The corrections which are considered to be added to the previous methods will make them more safe and robust in these cases [16].

The presented detection-delineation framework was validated by utilizing the clinical manual annotations of different databases such as MIT-BIH Arrhythmia Database (F_s=360Hz) [20], QT (F_s=250Hz) [21], and TWA Challenge 2008 Database (F_s=500Hz) [23] as well as high resolution holter data (MEDSET®-1000Hz, 3-Channel, 32-bits) [16-17]. As a result, the average values of Se = 99.96% and P+ = 99.96% were obtained for the detection of QRS complexes with the average maximum delineation error of 5.7 msec, 3.8 msec and 6.1 msec for P-wave, QRS complex and T-wave respectively. In addition, the proposed method was applied to DAY general hospital high resolution holter data (including BBB, PVC and PAC) and average values of Se=99.98% and P+=99.98% were obtained for sensitivity and positive predictability respectively. The organization of this article is

arranged as follows. In section B, DWT pre-processing via à trous method; furthermore, the ACLM and GIM structures and calculation of corresponding elements are described. In section C, technical information of the utilized databases, results obtained from application of detection-delineation algorithm, and verification procedure of the ACLM -based detection-delineation method are demonstrated. Finally, several conclusions obtained during the completion of this study are presented in section D.

II. MATERIALS AND METHODS

In this research, the structure of the long-duration holter ECG waves detection-delineation code is described. According to the block-diagram shown in Fig. 1, first, the holter extracted signal is pre-processed via implementation of resampling, band-pass digital FIR filtering and à trous discrete wavelet transform algorithms. It should be noted that almost all parameters of the proposed detection delineation algorithms [11,13,16,17] are highly dependent to the sampling frequency of the holter systems. For instance, sampling frequencies 128 Hz, 250 Hz, 500 Hz, 750 Hz, 1 kHz, 2 kHz and 10 kHz can be seen among holter based databases [35]. Thus, to introduce a unified ECG individual events detection framework which is applicable for all sampling frequencies, after pre-processing of the original ECG, the resulted signal in the core sampling frequency is mapped to a new trend with target sampling frequency 1000 Hz.. By this operation, once the parameters of the algorithm are properly regulated for the target sampling frequency, the algorithm can be implemented to the holter data sampled at any rate.

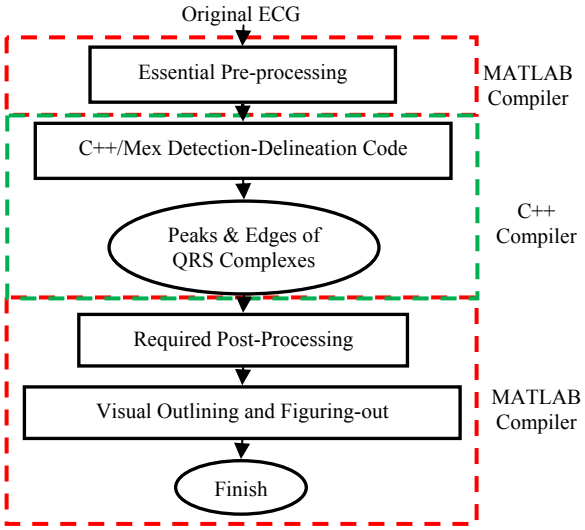


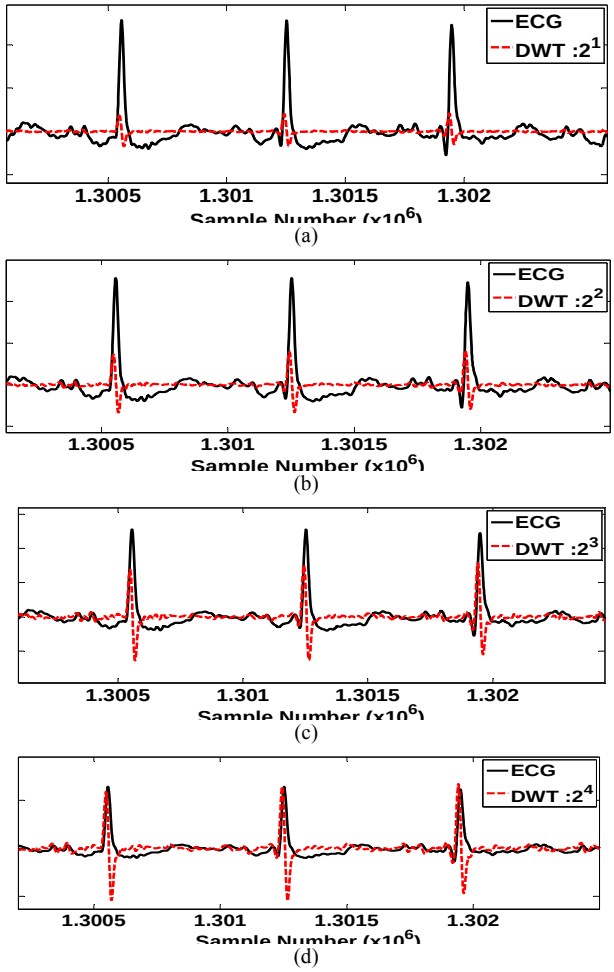
Figure 1. General block diagram of the HSEDF code designed for detection-delineation of the long-duration holter ECG events.

A. Discrete Wavelet Transform Using à trous Method

The wavelet transform has three appropriate and useful properties. First, by employing this approach, the original signal can be described in different time scales. Therefore, different spatial resolutions would be achieved. For instance, in the scale 2^1, high energy waves (such as QRS complexes) can be easily distinguished from other waves while in the scale 2^4 or 2^5, weak or very weak waves (such as T-wave P-wave or probable U-waves) can be detected [16]. Thus, by applying a multi-step algorithm it would be possible to detect strong, weak and very weak waves. This feature should be noted as one of the most significant characteristics of the wavelet transform which can be implemented to obtain more accurate results. Second, contaminating factors such as noise, artifacts, and baseline wandering can be distinguished from heart electrical activity based on their specific frequency contents which lead to better performance for the detection algorithm. Third, the wavelet transform can be easily implemented in practical cases due to the fact that it is a cascade of sequential short- length unit impulse response of digital FIR filters.

Generally, it can be stated that the wavelet transform is a quasi-convolution of the hypothetical signal $x(t)$ and the wavelet function $\psi(t)$ with the dilation parameter a and translation parameter b . The parameter a can be used to adjust the wideness of the basis function. Therefore, the transformation can be adjusted in temporal resolutions. [13, 15].

In Fig. 2, the DWT of an arbitrary holter ECG in different dyadic scales 2^{λ} (λ=1,2,…,6) are illustrated. It should be noted that by utilizing this approach, the original signal can be described in different time scales and therefore different spatial resolutions would be achieved. For instance, in the scale 2^1, high energy waves (such as QRS complexes) can be easily distinguished from other waves while in the scale 2^4 or 2^5, weak or very weak waves (such as T-wave P-wave or probable U-waves) can be detected [16]. Thus, by employing a multi-step algorithm, it would be possible to detect strong, weak and very weak waves.



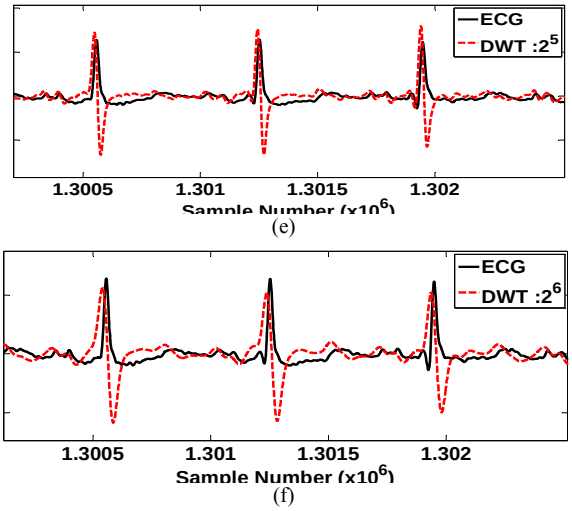


Figure 2. Illustration of the different à trous DWT dyadic scales to choose the best transformation . (a) scale 21, (b) scale 22, (c) scale 23, (d) scale 24, (e) scale 25, (f) scale 26. According to the conducted explorations, scale 24 possesses the best (max – min)/duration ratio.

B. Structure of the Geometrical Index (GI) Metric

1) Summation of Absolute First-Order Derivative

First and second order derivatives, curve length, area and second order statistical moment (variance) are the basic constitutive elements of the GI metric. Below, some justifications for the selection of each measure are presented.

If signal $x(t)$ is sampled from the continuous waveform with sampling frequency F_s, subsequently, the summation of absolute first order derivative of signal $x(t)$ in the analysis window is obtained as

$$M_{dfI}(k) = F_s \sum_{t=k}^{k+W_L-1} \left[\left| x(t+1) - x(t) \right| \right] \quad (1)$$

This measure clearly detects the activity extent of the high-frequency components of the original signal. In other words, in weak P or T-waves in which their amplitude and area are not large enough, this quantity shows more accurate sensitivity to occurrence of such phenomena.

2) Summation of the Absolute Second-Order Derivative

Summation of the absolute second-order derivative of signal $x(t)$ with the sampling frequency F_s, in the k-th slid of the analysis window can be obtained as

$$M_{dfII}(k) = F_s^2 \sum_{t=k}^{k+W_L-2} \left[\left| x(t+2) - 2x(t+1) + x(t) \right| \right] \quad (2)$$

This measure indicates the ascend/descend rate or kurtosis of the signal x(t) detecting the activity period of the signal generation source. The measure $M_{dfII}(k)$ shows remarkable fluctuations during activity of the heart individual events. A large value of M_{dfI} and M_{dfII} indicates a sharp transition from low to high or from high to low values in the excerpted segment. Consequently, these measures detect the probable edges of the signal in the analysis window thus making the GI metric sensitive to behavior of the signal in the edges.

3) Curve Length

Curve length of signal $x(t)$ in the k-th slid of the analysis window can be obtained approximately as [16, 17]

$$M_{CL}(k) \approx \frac{1}{F_s} \sum_{t=k}^{k+W_L-1} \sqrt{1 + \left[\left(x(t+1) - x(t) \right) F_s \right]^2} \quad (3)$$

The curve length is a suitable quantity to measure the duration of the signal $x(t)$ events, either being strong or weak.

4) Area under Curve

The approximate area under curve $x(t)$ in the k-th slid of the analysis window is obtained from the following equation

$$M_{AR}(k) \approx \frac{1}{F_s} \sum_{t=k}^{k+W_L} \left| x(t) \right| \quad (4)$$

A large M_{CL} value of a signal points out a sharp ascending or descending trend. Consequently, this quantity makes the GI metric sensitive to the high-slope parts of the signal in the analysis window, either ascending or descending. Generally, the measure M_{CL} indicates the extent of flatness (smoothness or impulsive peaks) of samples in the analysis window. This measure allows the detection of sharp ascending/descending regimes occurred in the excerpted segment, [31, 32].

5) Centralized Mean Square Value

An estimate of the centralized mean square value of the signal $x(t)$ excerpted segment can be obtained as

$$M_{MS}(k) = \frac{1}{W_L} \sum_{t=k}^{k+W_L} \left[x(t) - \mu_k \right]^2 \quad (5)$$

Where μ_k is the sample mean of the $x(t)$ in the analysis window and can be obtained from the following summation

$$\mu_k = \frac{1}{W_L} \sum_{t=k}^{k+W_L} x(t) \quad (6)$$

The physical meaning of the $M_{MS}(k)$ is the average power of the events while this quantity graphically shows the dispersion of the samples around the mean value [13]. The $M_{MS}(k)$ indicates the difference between absolute maximum and minimum values of an excerpted segment. This difference may not be seen via mean value because it is possible that the mean of a segment is a small value whilst the difference between its maximum and minimum values is large.

C. Area-Curve Length Method (ACLM)

If the discrete wavelet transform of the ECG signal is calculated according to the appropriate block-diagram [13]. Consequently some dyadic scales will be obtained for scale values of 2^{λ}, $\lambda = 1, 2, 3, 4, 5$. The proposed metric is developed based on simple mathematical calculations. Suppose a window with the length of L samples sliding sample to sample on the signal. Therefore, the signal in k-th window can be obtained as follows

$$\mathbf{Y}_k = \mathbf{W}_{2^{\lambda}}[k : k+L] \quad (7)$$

Where Y_K is a vector including the elements k to k+L of the scale 2^{λ}. To define a new measure, the area under the absolute value of the curve Y_K and the curve length of Y_K is obtained as

$$\begin{aligned} Area(k) &= \int_{t_{0k}}^{t_{fk}} \left| y_k(t) \right| dt \\ Curve(k) &= \int_{t_{0k}}^{t_{fk}} \sqrt{1 + \dot{y}_k^2}\, dt \end{aligned} \quad (8)$$

Where t_{0k} and y_{fk} are the start and end points of the vector Y_K respectively and the variable $Y_{K(t)}$ represents the samples existing in the vector Y_K. Accordingly, the new measure named Area-Curve Length (ACL) is defined as follows

$$ACL(k) = Area(k) \times Curve(k) \quad (9)$$

More details about the ACL metric specifications can be seen in [16]. The most significant reason for the definition of the ACL measure according to Eq. 9 is its capability in the detection of ECG wave edges (onset and offset locations). ACL reaches its minimum value when both the value of signal $\mathbf{Y}_k$ and the corresponding derivative in the window reach their minimum values. Therefore, minimum value of ACL is an indicator of minimum amplitude and minimum slope events. This can be observed in wavelet transform scales (For more details see [16]).

III. DESCRIPTION OF THE HSEDF CODE STRUCTURE

In Fig. 3, general flow-chart of the HSEDF code is illustrated. In this flow-chart, the overall structure of the QRS detection-delineation algorithm is described. In the first major operation of the HSEDF code, the pre-processed signal is utilized for detection of the QRS complexes as well as primary onset-offset identification for each detected complex. In the second main operation of the code, the precision of each detected QRS onset and offset locations is improved via implementation of rchecker and lchecker functions. In the third chief operation of the HSEDF code, the QRS peak enhancement and post-processing are performed. In other words, the proposed ECG detection-delineation algorithm includes three major operations and three minor stages. In the first minor stage(or pre-processing stage), original ECG signal is pre-processed by resampling algorithm as well as implementation of low-pass and high-pass FIR filters to remove baseline wander (low-frequency disturbance) and measurement distortions (high-frequency components). The pre-processing stage is prolonged by taking the DWT from the filtered signal to obtain several dyadic scales. The second minor stage of the detection-delineation code is the C++/Mex part compiled by the Microsoft Visual Studio 2008 in the MATLAB environment. The third minor stage of the code is the post-processing and visual presentation of the obtained results from C++/Mex subroutine. It should be noted that in order to detect and delineate individual events of the long-duration holter ECG, an appropriate computational environment should be chosen in order to achieve acceptable processing speed as well as to access efficient graphical interface. To this end, in the presented study, C++/Mex environment of MATLAB with Microsoft Visual Studio 2008 compiler are chosen in which the speed of low-level C++ programming environment is accessible while efficient utilities of MATLAB can be employed to post-process the output of the high-speed detection-delineation subroutine.

A. Embedded Functions in the HSEDF Code Structure

In this section, a summarized descriptions for the mathematical functions embedded in the HSEDF code are presented.

mean(). This function calculates the sample mean of an excerpted segment of signal x(t) as follows (Appendix A Lines 9 to 15).

$$\mu = \frac{1}{fi - st + 1} \sum_{j=st}^{fi} x(j) \quad (10)$$

Where st and fi represent the start and finish sample numbers, respectively.

stdi(). This function determines the sample standard deviation in the analysis window as follows (Appendix A Lines 18 to 26).

$$\sigma = \sqrt{\frac{1}{fi - st} \sum_{j=st}^{fi} \left(x(j) - \mu\right)^2} \quad (11)$$

integral(). This function calculates the finite numerical integration of an excerpted segment starting from st and ending to fi samples (Appendix A Lines 29 to 35).

$$I_x \approx \sum_{j=st}^{fi-1} 0.5 \times \left(x(j) + x(j+1) \right) \quad (12)$$

integralabs(). This function calculates the finite numerical integral of an excerpted segment absolute value as follows (Appendix A Lines 38 to 44).

$$I_{|x|} \approx \sum_{j=st}^{fi-1} 0.5 \times \left(\left| x(j) \right| + \left| x(j+1) \right| \right) \quad (13)$$

maxfinder(). This function determines numerically the absolute maximum of an excerpted segment in the interval [st,fi] (Appendix A Lines 64 to 70).

$$k_x = \left\{ k \,\middle|\, k \in (st, fi),\ x(k) > x(k-1)\ \&\ x(k) > x(k+1) \right\} \quad (14)$$

maxabsfinder(). This function determines numerically the absolute extremum of an excerpted segment in the absolute value of interval [st,fi] samples (Appendix A Lines 73 to 79).

$$k_{|x|} = \left\{ k \,\middle|\, \begin{array}{l} k \in (st, fi),\ \left| x(k) - l \right| > \\ \left\| x(k-1) - l \right| \ \&\ \left| x(k) - l \right| > \left| x(k+1) - l \right| \end{array} \right\} \quad (15)$$

Where l is the baseline value relative to it, the maximum value of the absolute difference should be determined. It should be mentioned that in this function the absolute extremum is the point that has the maximum difference (distance) with l (the baseline value). Base line value in each local search analysis window is the mean of onset and offset of that analysis window.

regre(). The duty of this function is to introduce a standard to determine approximately the noise level of the excerpted segment of x(t) in the [st,fi] segment using the following equation (Appendix A Lines 82 to 100).

$$r = \left| \frac{\frac{1}{fi - st + 1} \sum_{j=st}^{fi} x(j)\, y(j) - \mu_x \mu_y}{\sigma_x \sigma_y} \right| \quad (16)$$

Where x(j) is the sample of the understudy signal, μ is sample mean, σ is sample standard deviation, $y(j) = \alpha\ (j - st)$ is a linear line with the slope of the line connecting two positions on the DS starting from st and ending to fi, i.e.,

$$\alpha = \frac{x(fi) - x(st)}{fi - st + 1} \quad (17)$$

Comprehensive studies indicate that Eq. 17 with the defined value for $\alpha = (x(fi) - x(st)) / (fi - st + 1)$, can reveal the similarity level of the excerpted segment and a

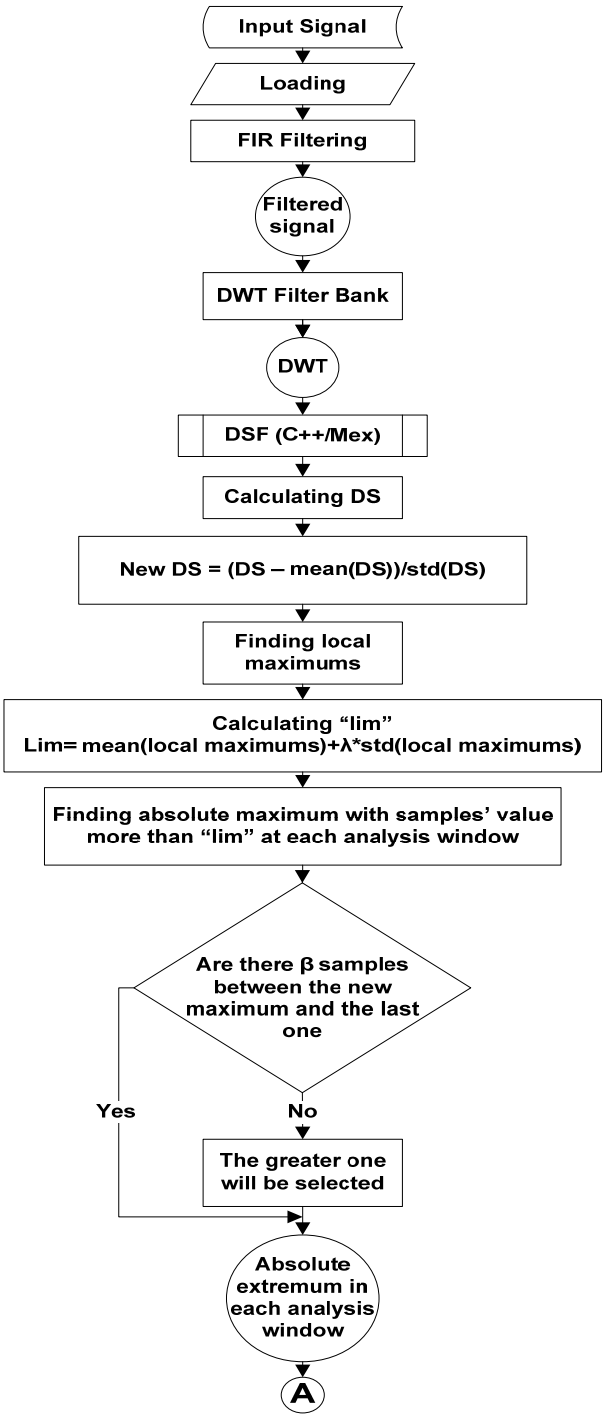


Figure 3-a. General flow-chart of the HSEDF code describing the structure of the proposed detection-delineation algorithm. QRS detection subroutine of the code (First and second minor stages and first part of first major operati on).

linear line in the [st,fi] interval, therefore, the level of noise can be lowered.

noisechecker(). This function gets the location of an input point and determines the noise level of the specific neighborhood (window) of that point with the given length using the regre() function as follows (Appendix A Lines 103 to 110).

$$noise\ checke\ r(k) = \begin{cases} 0 & regre(k) < LB \\ 1 & LB \le regre(k) < UB \\ 2 & regre(k) \ge UB \end{cases} \quad (18)$$

Where numbers 0 ,1 and 2 show high, medium and low levels of noise respectively. In the section C.4.1. , the empirical procedure for determining the lower and upper bounds (LB and UB) is explained.

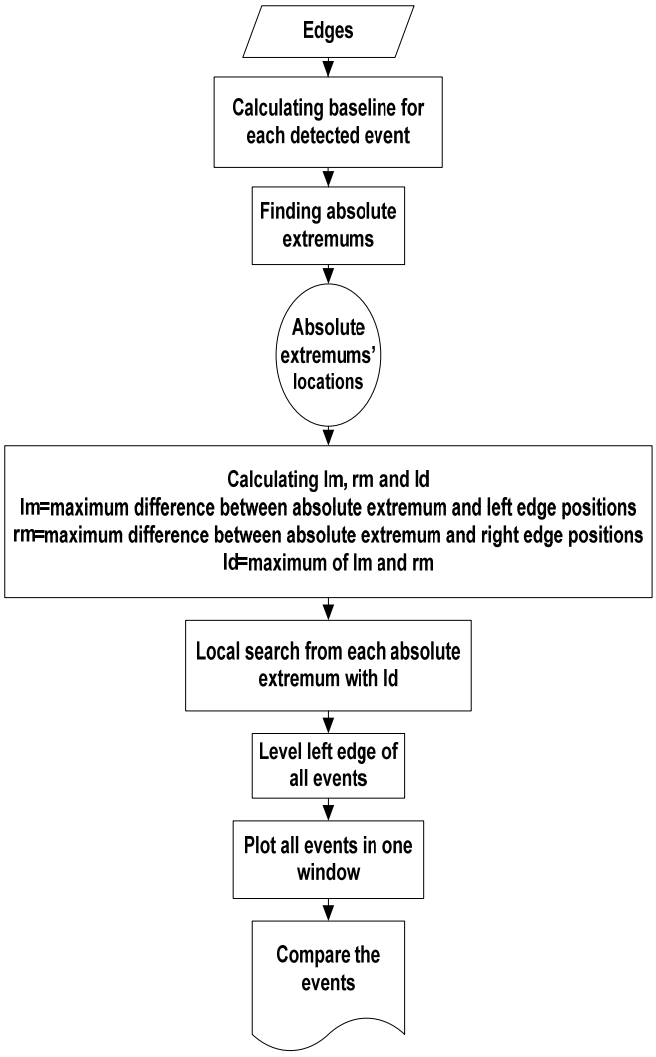


Figure 3-d. General flow-chart of the HSEDF code describing the structure of the proposed detection-delineation algorithm. Removing of probable R-peak eccentricity by conducting a new local search on the interval starting from onset and ending to offset points by means of the maxabsfinder function (Third major operation then third minor stages).

rchecker(). The Duty of this function is to improve the location of the identified offset point by updating the search process with respect to noise level. In this function, it is primarily checked that whether slop of the candidate for the new offset point is lower than the slope threshold (The slope threshold is equal to slope coefficient (η_s) multiplied with maximum slope in the detected right part of that QRS complex. (The slope is defined as the difference between two continuous samples) and then it is controlled regardless that the elevation of the candidate for the new offset point is close enough to the baseline of the DS when compared to the previous founded right edge or not. It should be closer to the baseline of the DS at least equal to difference parameter (η_d). If these two conditions are satisfied, the noise level of the specific neighborhood (window) of the candidate for the new offset point is estimated using the noisechecker function. If the noise level of the candidate for the new offset point is low, then that point is the new offset point. If it is concluded that

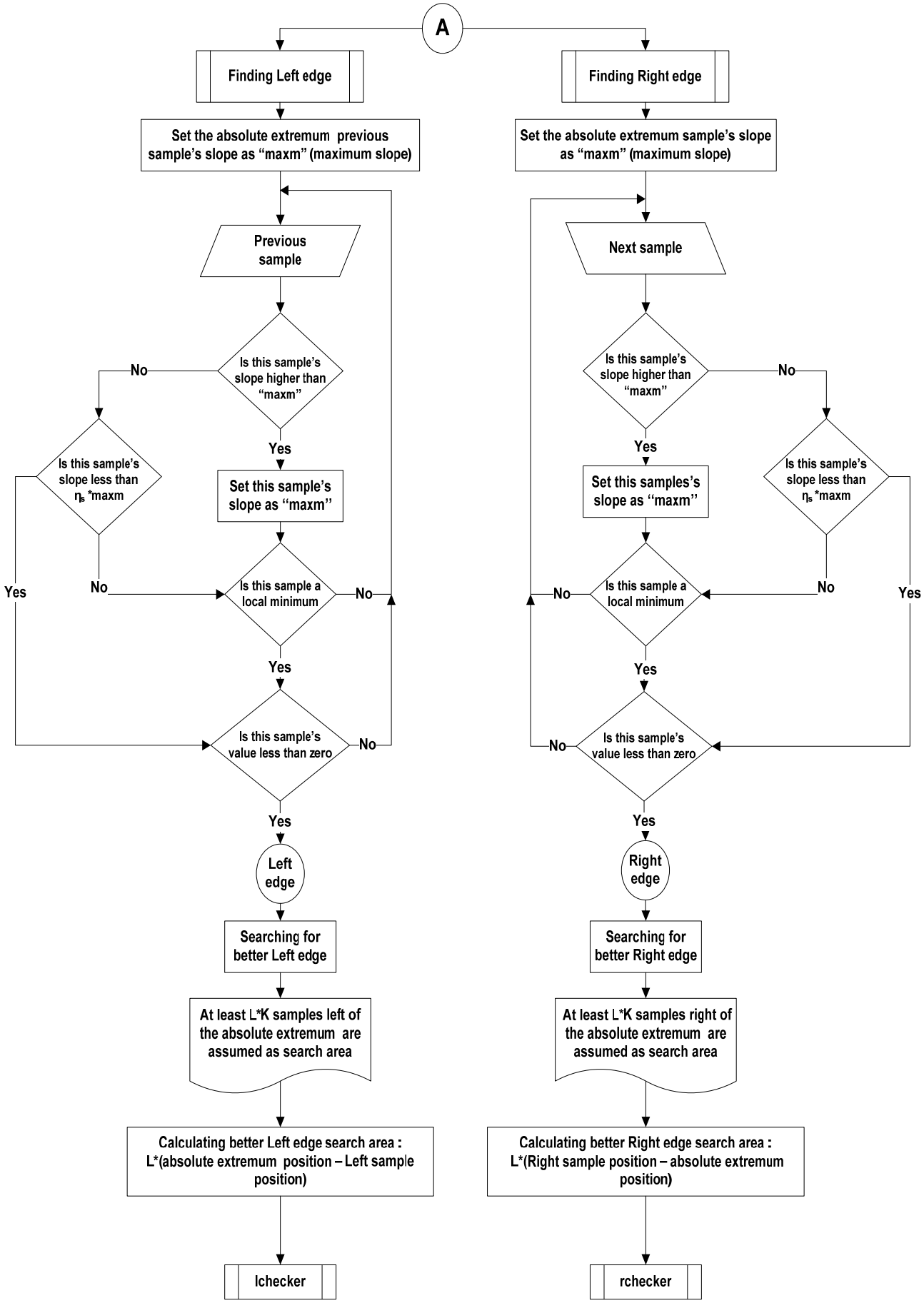


Figure 3-b. General flow-chart of the HSEDF code describing the structure of the proposed detection-delineation algorithm. Primer QRS delineation (Second part of the first major operation).

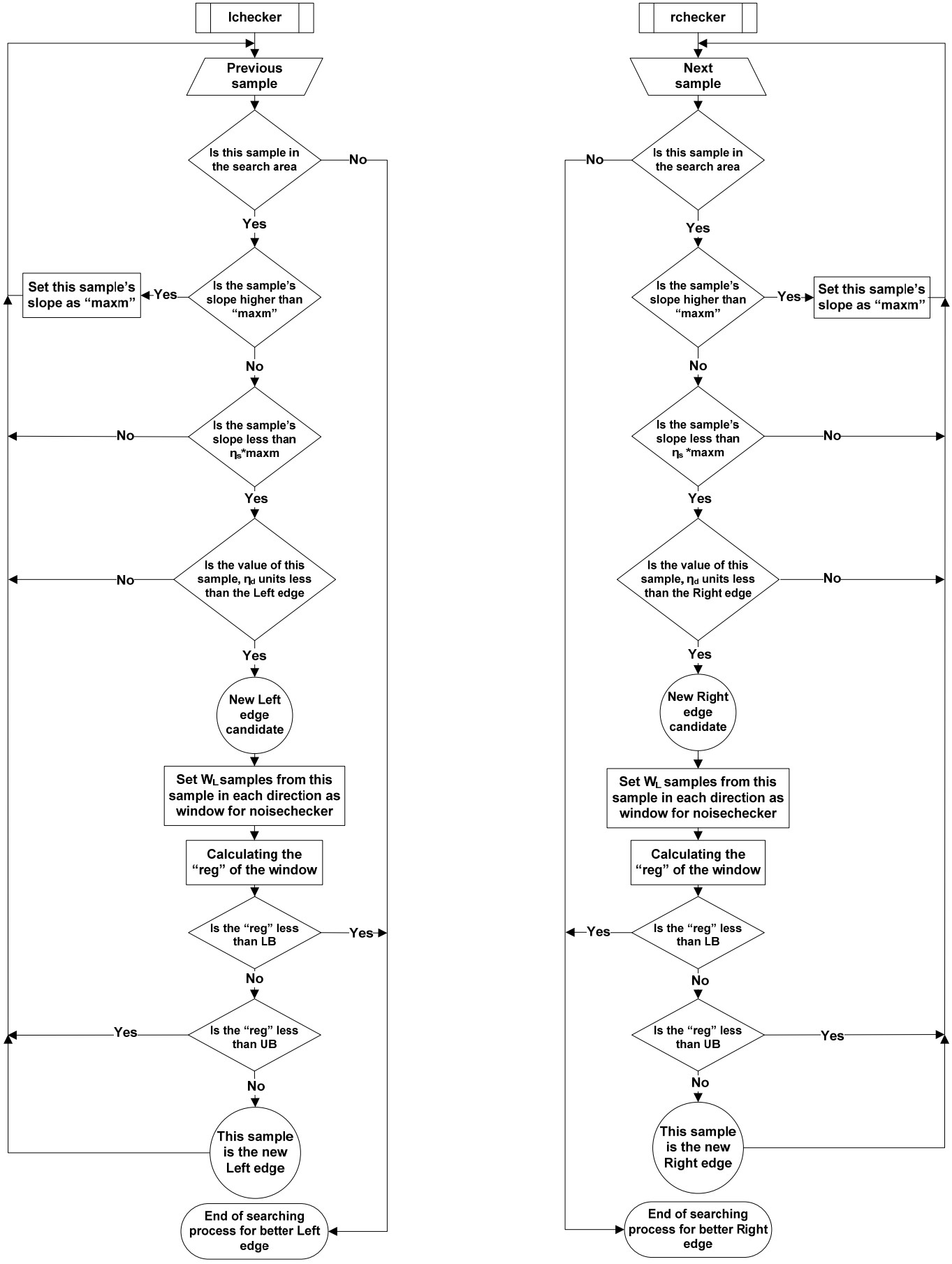


Figure 3-c. General flow-chart of the HSEDF code describing the structure of the proposed detection-delineation algorithm. Correction of primitive identified QRS onset-offset locations via checking the level of noise of the primer point via lchecker and rchecker functions (Second major operation).

the noise level of the candidate for the new offset point is medium, consequently the search for finding a better location for the new offset point continues. Otherwise, if the noise level is high, in order to prevent probable delineation errors due to rapid and strong fluctuations of noisy samples, further search for identification of better offset point is cancelled (Appendix A Lines 113 to 132).

lchecker(). This function acts similar to the rchecker function. Nevertheless, the difference is that in rchecker function, the direction of search is backward, while in this function the backward search is conducted to improve the location of the onset point (Appendix A Lines 135 to 154).

B. *Detection of the QRS Complexes as Well as Primary Onset-Offset Identification for Each Detected Complex and Improving Them via Implementation of Rchecker and Lchecker Functions*

In order to identify primer locations of each QRS complex, the strongest absolute maximums in the decision statistic trend (for instance ACLM or GIM) are primarily detected in each analysis window by maxfinder() function. Analysis windows are parts of DS sample with lower threshold $\tau = \mu + \lambda\sigma$, where μ is the sample mean, λ is an empirical parameter, and σ is the sample standard deviation of the excerpted segment. If the sample number difference between 2 continuous peaks is less than 600, the one with lower value is ignored from our peak points list. The input pre-processed signal frequency is 1000 Hz, and the period between two heart beats is more than 600 msec. Hence two continuous R-Peaks in an ECG wave must have a sample number difference more than 600. Afterward, the detected peak is chosen as the search origin and two forward and backward local searches from this origin are conducted. In order to detect the edges (onset-offset) of each detected QRS complex, three conditions should be checked. The third condition and at least one of the first or second conditions should be satisfied in order to accept the candidate sample as a primary edge point:

- The value of the decision statistic sample should be lower than the value of its previous sample and its value should be lower than or equal to the next sample value.
- The difference of the candidate location value and value of one sample before should be smaller than a threshold. (The slope of the candidate sample should be lower than the aforementioned slope threshold).
- The value of the DS at the candidate sample should be negative.

After making a detection of the strongest absolute extremums of the DS and primary edges (onset-offset), each detected peak is considered as the origin of two other forward and backward searches in a region which is from primary edges(onset-offset) with L*K samples length (in backward and forward). These searches are performed by means of rchecker() and lchecker() functions in order to find better onset and offset locations (points).

The value of K can be obtained from the median value of the RR-tachogram, i.e.,

$$K \approx \frac{1}{10} median(\mathrm{RR}) < samples > \qquad (19)$$

Where RR is the vector containing $rr_k = R_k - R_{k-1}$ components and R_k indicates the k-th peak of the DS (each absolute extremum of DS is equivalent to a strong peak in the QRS complex).

Where K is defined as the length of sample numbers between R-Peak and found primary onset point for lchecker() function and between R-Peak and found primary offset point for rchecker() function. Sometimes primary edges are not far enough from the R-Peak. Hence, K and search region are smaller than what they should be for more satisfactory edges. Therefore, to prevent this error, the parameter K whose value is less than 100 is replaced with 100 as its new value (The "lower suitable K" is defined as 100).

C. *Improvement of R-Peak Locations*

Computerized studies show that after detection and delineation of the QRS complexes applying appropriate decision statistic (in this study CLAM or GIM), absolute extremums of the DS might not exactly coincide with the R-peaks of QRS complexes and some deferens may occur. To remove this eccentricity error, a new local search between detected onset and offset locations utilizing the maxabsfinder function is conducted. Further QRS peaks either upward or downward can be exactly identified accordingly.

In Fig. 4, the effect of a new local search for R-peak position enhancement is shown. It should be noted that in arrhythmic cases in which the wideness of the ventricular beats is larger than the normal duration, the peak of the DS might not coincide with the strongest peak of the detected QRS. Therefore, to find the best location for R-peak, a direct local search on the original ECG by means of the maxabsfinder function is conducted.

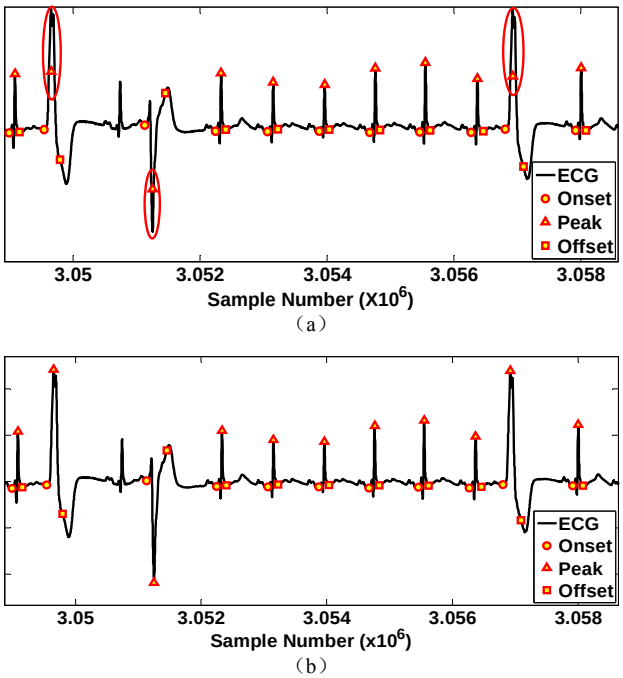


Figure 4. Application of a new local search between detected onset and offset locations by means of the maxabsfinder function to find the beast R-peak position. (a) Primary R-peak, (b) Corrected positions.

D. *Empirical Approach for the HSEDF Code Parameters Adjustment*

E. *Regulation of the Embedded Functions Parameters*

Table 1, explain the parameters which are introduced and used in this research to modify the results. It should be noted that the empirical parameters, which are used at HSEDF Code,

are obtained empirically after examination of more than 400,000 beats extracted from several subjects' holter recordings. These parameters are:

1. τ, λ : Parameters used for "maxfinder() function analysis windows lower threshold". In order to find an appropriate value for λ various values starting from 0.25 and ending to 1.0 for many types of holter extracted beats have been tested. After this comprehensive examination, it is seen that 0.75 is an appropriate value.

2. Slope coefficient (η_s): Parameter used at second condition of primary edges finding and for lchecker() and rchecker() functions. In order to determine an appropriate value for slope coefficient (η_s), various values from 0.01 to 0.000001 for many types of holter extracted beats have been tested. After this comprehensive examination, it is seen that the values in the 10^{-4} order (for instance 0.0001) are appropriate values for the slope coefficient (η_s).

In Fig. 5 the effect of examining different slope coefficients (η_s) for decrements from 0.01 to 0.000001 is shown. As illustrated, by decreasing coefficient η_s , the performance of the HSEDF code is improved by further reducing η_s value from 0.0001, while no improvement takes place in the QRS detection-delineation performance.

3. L×K : Parameter used for "rchecker() and lchecker() functions search region" . In order to find an appropriate value for L and "lower suitable K" various values from 1 to 3 for L and various values from 1 to 300 for "lower suitable K" have been tested. After this comprehensive examination, it is seen that 2.5 is an appropriate value for L, and 100 is an appropriate value for "lower suitable K".

In Fig. 6, the effect of search region length of L×K (sample number) of rchecker and lchecker functions is studied. The assessment is started by choosing L=2.5 and K=250 (samples) to investigate the effect of this windowing region. Various testing of the HSEDF code show that L=2.5 and K=100 (samples) yields the best result among other region sample numbers.

4. Difference parameter (η_d) : Parameter utilized at second condition of lchecker() and rchecker() functions. In order to determine an appropriate value for difference parameter (η_d), various values starting from 0.00005 and ending to 1.0 for many types of holter extracted beats have been tested. After this comprehensive examination, it is evident that the values in the 10^{-3} order (for instance 0.005) are appropriate values for the difference parameter (η_d), validated by cardiologist.

In Fig. 7 the effect of examining various difference parameter (η_d) for values from 0.5 to 0.00005 is shown. As illustrated, by decreasing parameter η_d , the performance of the HSEDF code is improved which by further reducing the parameter η_d value from 0.005 no improvement takes place in the QRS detection-delineation performance.

5. Specific neighborhood window of noisechecker() function (W_L) : In order to find an appropriate value for "specific neighborhood window of noisechecker() function (W_L)", various window length sample number between 1 to 100 samples were applied to functions and obtained results were studied. After several examinations on several holter extracted beats, with different sample numbers for window length, it was concluded that if the window length is chosen to be between 18 to 35 samples, the acceptable performance would be achieved form implementation of noisechecker() function.

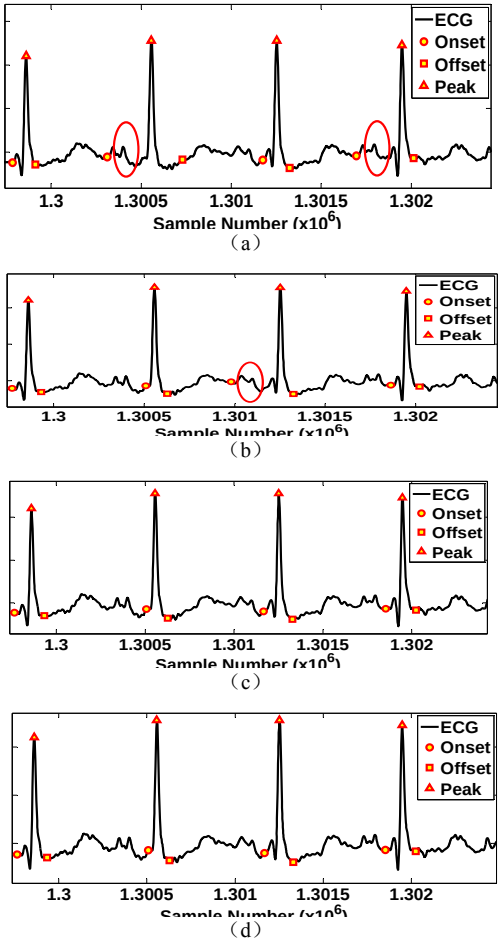
In Fig.8, the effect of noisechecker specific neighborhood window (W_L) length is tested. By choosing the length of the specific neighborhood window (W_L) a small integer, the noisechecker function cannot help lchecker() and rchecker() functions to find a suitable location for the edges(onset and offset points). On the other hand, if this parameter is selected large enough (Fig.9-d), the noisechecker function can perform better position for onset location. Although values more than 35 mislead noisechecker() function and cause other errors.

TABLE 1. PARAMETERS INTRODUCED AND USED IN THIS RESEARCH TO MODIFY THE RESULTS

Name	Application fields	Valuation interval	Accepted value	Description
	maxfinder	0.25 to 1	0.75	
	rchecker lchecker	10^{-6} to 10^{-2}	order 10^{-4} (used value: 0.0001)	By decreasing from 10^{-2} to 10^{-4}performance increases, but there is no noticeable difference for less than 10^{-4}
L	Specifying search intervals for rchecker lchecker	1 to 3	2.5	If there is more than 100 data between first edges and extremum point, K is set to this value, otherwise k is set to 100 by default
K		1 to 300	100	
	rchecker lchecker	1 to 5×10^{-5}	order 10^{-3} (used value: 0.005)	
	noisechecker	1 to 100	18 to 35 (used value: 20)	W_L is referred to length of window used to analyze noise level. If it is less than 18, it doesn't help function's performance. And if it is more than 35, causes error to function's performance
LB	noisechecker	0.001 to 0.3	0.003	First UB is set to 0.5 and 0.003 for LB is found by this consumption. After that appropriate value between 0.2 and 0.7 was found for UB.
UB		0.001 to 0.9	0.4	

6. Lower and Upper Bounds (LB and UB) : Parameters utilized at noisechecker() function and classify outputs of regre() function for employing as inputs of noisechecker() function. In order to determine appropriate values for upper and lower bounds of regre() function, diverse examinations on both bounds using several holter signals were conducted. First, the upper bound was primarily set on 0.50 and the lower bound was changed from 0.001 to 0.30 with a suitable increment. Examinations on holter beats show that 0.003 can be chosen as an appropriate value for the lower bound of the regre function. Similarly, the lower bound of the test regre function was set fixed on 0.003 and the upper bound was increased from 0.01 to 0.90. Studies and cardiologist validations show that the appropriate value for the upper bound of the regre function can be chosen from [0.20,0.70] interval.

In Figs. 9-10, the procedure for determining upper and lower bounds for the regre function is graphically described. To determine these limits, first the upper bound is set fixed and the best value for the lower bound is obtained by testing diverse values on several records. Similarly, the obtained lower bound value is set for regre() function and different values are tested to find an appropriate upper bounds. In this study, after examination of many types of the holter recorded ECG signals, the best value for the lower bound was estimated to be a point about 0.003 while an interval was estimated as a good range for the upper bound ([0.2,0.7]). Investigations reveal that lower bound equal to 0.003 is an appropriate value.



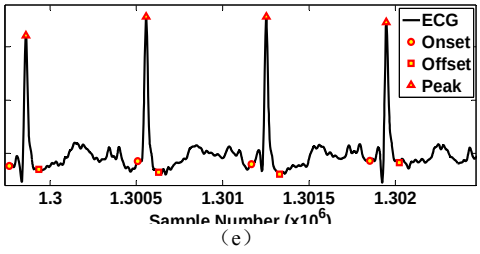


Figure 5. Finding appropriate value for the slope coefficient (ηs). (a) ηs =0.01, (b) ηs =0.001, (c) ηs =0.0001 (d) ηs =0.00001, (e) ηs =0.000001. Assessments show that by choosing theηs value lower than 0.0001, no further delineation improvement appear. (Ellipse: acceptable delineation region)

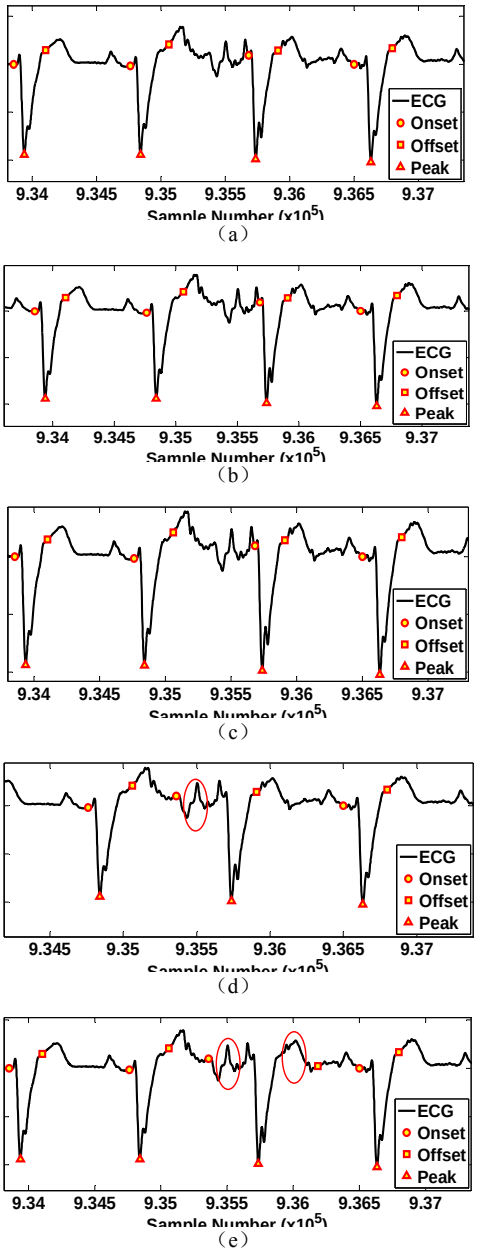
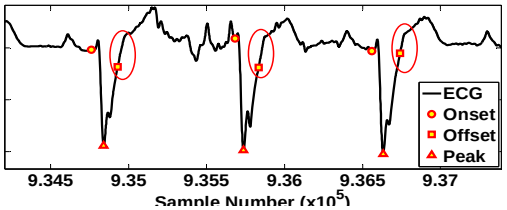
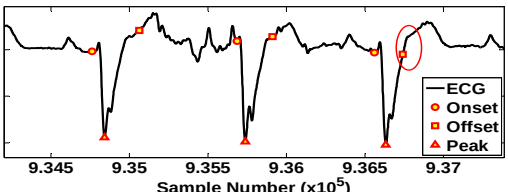


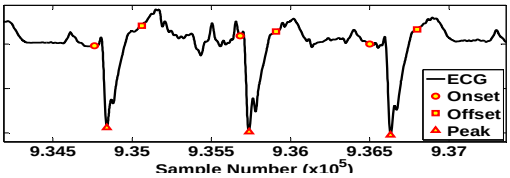
Figure 6. Assessing the effect of search region length L×K for edges (onset, offset) detection. It was concluded that L=2.5 and K=100 samples would yield the best primary edges identification; (a) L=2.5 and K=50(samples), (b) L=2.5 and K=100(samples), (c) L=2.5 and K=150(samples), (d) L=2.5 and K=200(samples), (e) L=2.5 and K=250 (samples). (Ellipse: acceptable delineation region)



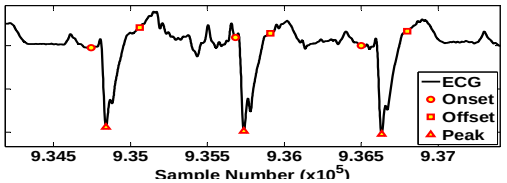
(a)



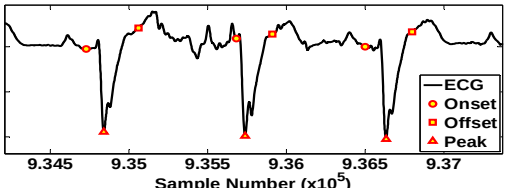
(b)



(c)

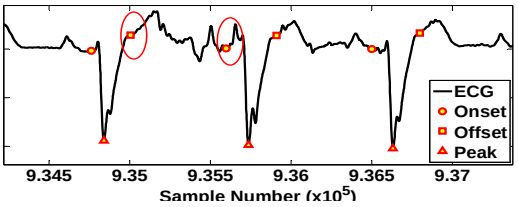


(d)

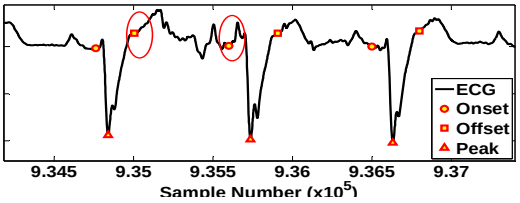


(e)

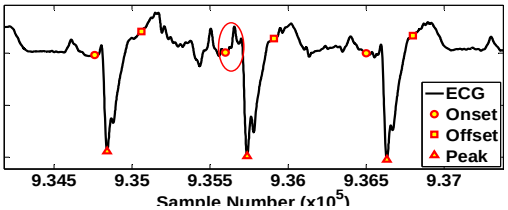
Figure 7. Finding appropriate value for the difference parameter (η_d). (a) η_d =0.5, (b) η_d =0.05, (c) η_d =0.005 (d) η_d =0.0005, (e) η_d =0.00005. Assessments show that by choosing thenη_d value lower than 0.005, no further delineation improvement appear. (Ellipse: acceptable delineation region)



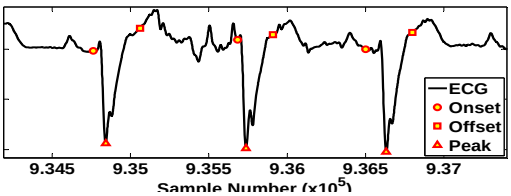
(a)



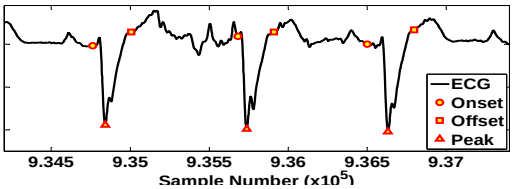
(b)



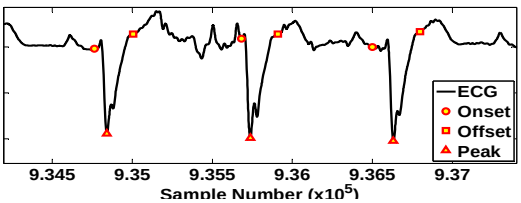
(c)



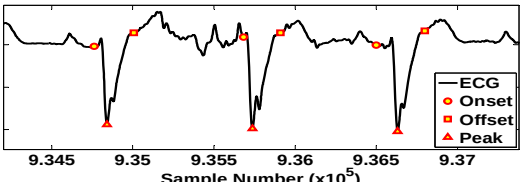
(d)



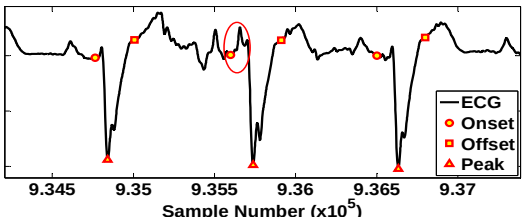
(e)



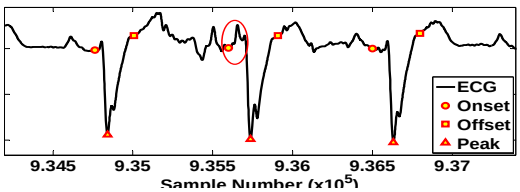
(f)



(g)



(h)



(i)

Figure 8. Assessing the effects of Specific neighborhood window (WL) of noisechecker() function on the delineation error. (a) WL=5, (b) WL=10, (c) WL=15, (d) WL=20, (e) WL=25, (f) WL=30, (g) WL=35, (h) WL=40 and (i) WL=80 (samples). Assessments show that an efficient value for Specific neighborhood window (WL) length of noisechecker function can be chosen from [18,35] (samples) interval.

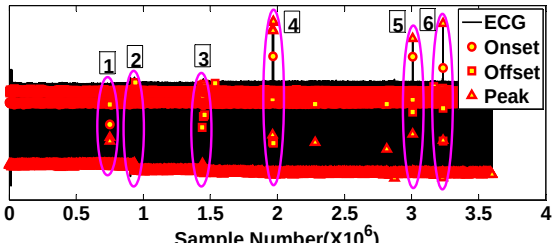
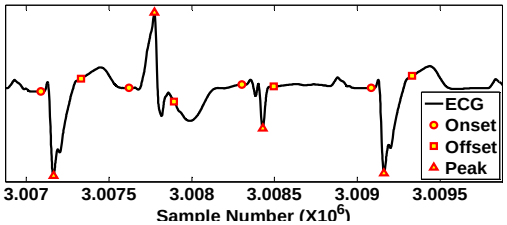
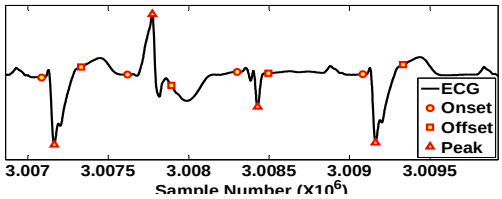


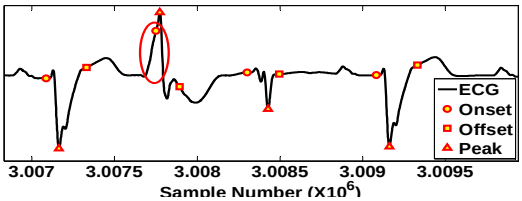
Figure 9. Some selected critical beats of an arbitrary 1-hour 1000 Hz holter ECG for regulation of the lower and upper bounds of the regre() function.



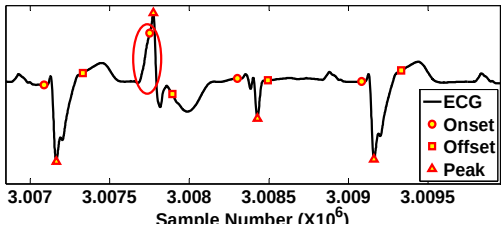
(a)



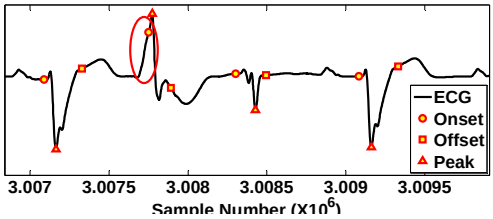
(b)



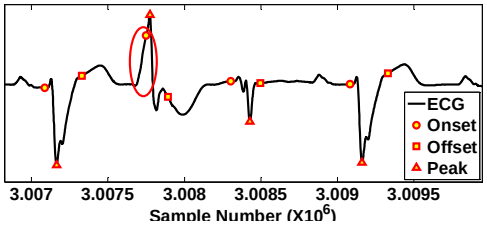
(c)



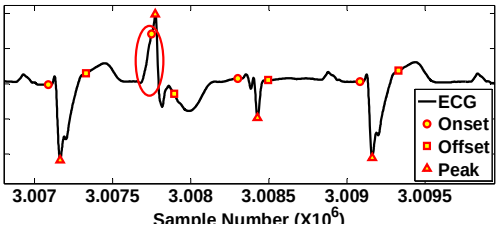
(d)



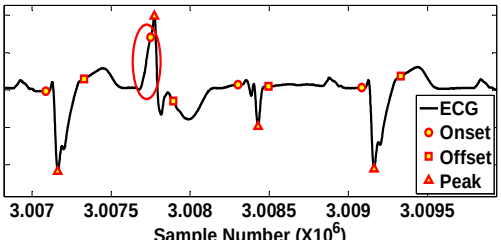
(e)



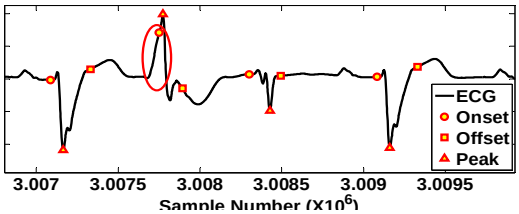
(f)



(g)

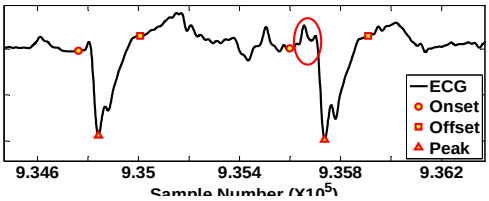


(h)

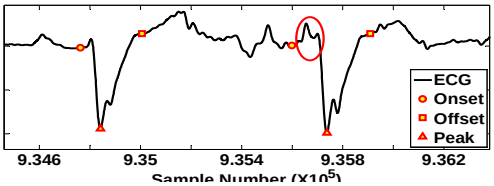


(i)

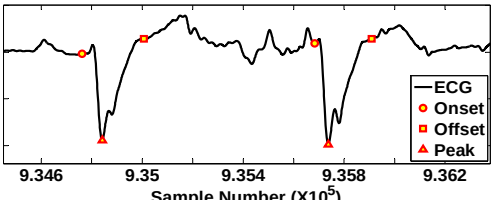
Figure 10-a. Determination of appropriate values for outputs of regre() function upper and lower bounds. Panel (A): Testing of several lower bounds for the upper bound set fixed on 0.50 : (a) [0.001,0. 50], (b) [0.003,0. 50], (c) [0.005,0. 50], (d) [0.01,0. 50], (e) [0.05,0. 50], (f) [0.10,0. 50], (g) [0.20,0. 50], (h) [0.30,0. 50], (i) [0.40,0.50]. ([Lower Bound, Upper Bound])



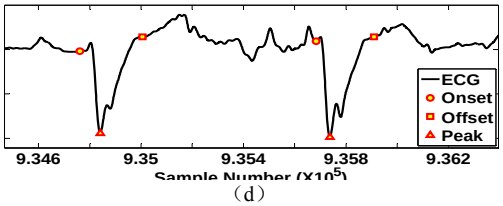
(a)



(b)



(c)



(d)

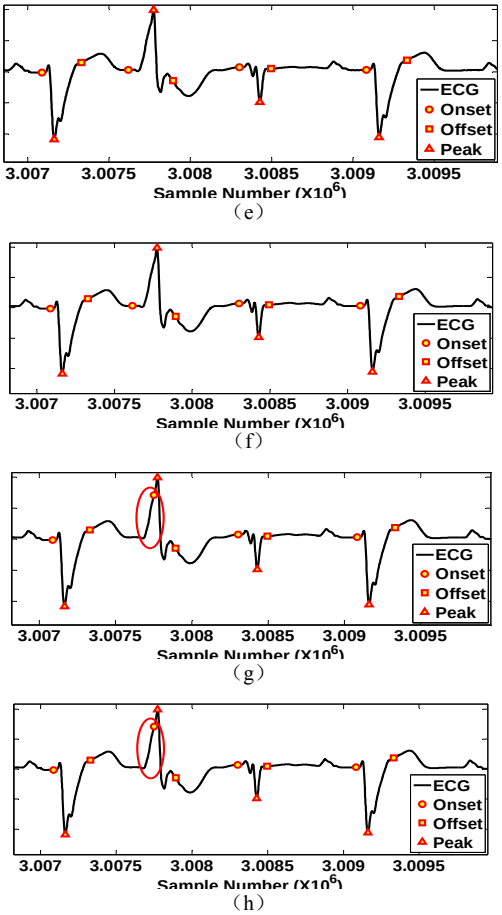


Figure 10-b. Determination of appropriate values for outputs of regre() function upper and lower bounds. Panel (B): Testing of several upper bounds (in two different critical beats) given the set fixed on 0.003 lower bound . (a) [0.003,0.01], (b) [0.003,0.10], (c) [0.003,0.20], (d) [0.003,0.30], (e) [0.003,0.60], (f) [0.003,0.70], (g) [0.003,0.80], (h) [0.003,0.9]. Investigations show that an appropriate value for the upper bound to yield an efficient delineation can be chosen from [0.20,0.70] interval. ([Lower Bound, Upper Bound])

IV. POST-PROCESSING AND VALIDATION

A. A Proposed Method for Detection and Delineation of P and T Waves (Normal, Biphasic, Inverted)

In order to detect P and T waves, a local search for two local maxima is conducted between two successive extremum values in the DS signal. The local maximum close to the right R-wave is specified as P-wave index and the one close to the left R-wave are specified as T-wave index of the preceding beat. Subsequently, in order to determine the onset and offset of P and T-waves, a segment of the signal DS between two consequent QRS complexes is chosen and a local search is conducted to determine four local minima as follows:

a) A search between the end of the preceding QRS complex and T-wave peak (beginning of the preceding T-wave)

b) A search between the T-wave peak and half of RR interval (end of the preceding T-wave)

c) A search between the half of the RR interval and the P-wave peak (beginning of P-wave)

d) A search between the P-wave peak and the beginning of the next QRS complex (the end of P-wave)

Generally, detection and delineation of T-wave is more difficult than P-wave. Therefore, the most significant part of the error corresponds to wave's detection which is related to T-wave. The block diagram of the algorithm for the detection and delineation of P-wave and T-wave is illustrated in Fig. 11.

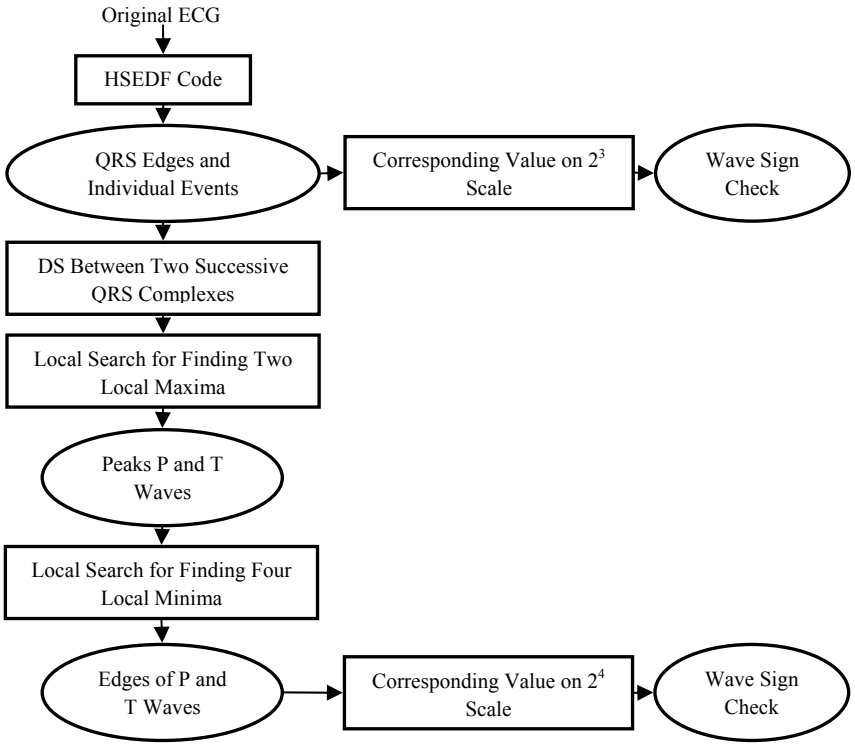


Figure 11. The block diagram of the P and T waves detection-delineation algorithm.

Detection and delineation of P and T-waves are described in more details. The two successive QRS complexes are primarily detected and the corresponding edges are delineated utilizing the DS measure. Next, the distance between the DS

of the corresponding wavelet transform of the right edge of the first complex and the left edge of the second complex is determined. The RR interval is then divided into two sections. Due to the fact that the DS measure is strictly positive, one dominant maximum will be determined in each of these two intervals. Finally, a local search is conducted to the left and right of each of these maxima, and the position where DS slope is less than 1/15 to 1/20 of the maximum slope in the window is identified as wave edge. The following conditions are observed:

- If a T-wave and a P-wave exists in the RR interval, they will be detected regardless of their sign, i. e., positive, negative or biphasic. (Waves sign is determined based on the sign of the corresponding wavelet transform)

- If there is only a T-wave in the RR interval, the left edge and the maximum amplitude of this wave will be easily detected. However, there may be some problems in finding the right edge. In this case, a new algorithm is required for the determination of the P-wave power.

- If there is no P or T wave in the RR interval, it will be accurately announced by the algorithm.

As it is mentioned above, one of the merits of the presented algorithm is that the sign of P or T-wave and their morphology will not affect the performance of the algorithm.

B. Validation of the Presented Algorithm

Numerous databases with different sampling frequencies and signal to noise ratio are used in this study to validate the performance of the proposed detection algorithm. To validate the QRS detection and delineation algorithm, MITDB (F_s=360Hz), TWADB (F_s=500Hz), and QTDB (F_s=250Hz) which contain annotation files that are applied (CHECK#0). It should be noticed that in challenging cases, results were delivered to the cardiologist and the detection algorithm was re-validated (CHECK#1) accordingly. In cases of QRS with very abnormal morphologies, the results were further checked by some residents (CHECK#2).

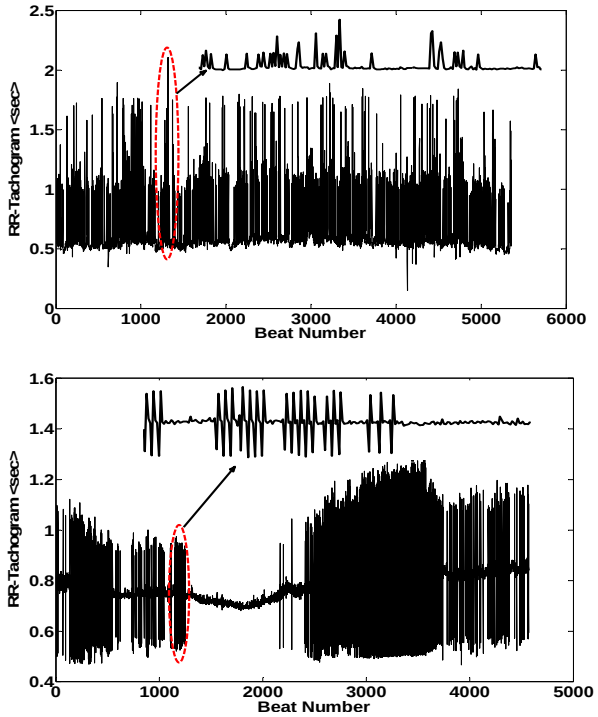


Figure 12. The detected RR-tachogram to estimate the number of FP and FN type errors; (a) this trend shows a FP error during the PAC-induced heart rate turbulence, (b) a trend with no FP and FN errors during the PVC-BBB induced heart rate turbulence.

Many approaches have yet been developed in the area of wave detection which are all applied only to the Lead I. Thus, in order to validate the performance of the proposed detection algorithm, it was applied to Lead I. If so, it would be possible to compare the presented algorithm with other researches. MITDB, QTDB and TWADB include 48, 105 and 100 subjects respectively. All these data were converted to MAT-files using the WFDB Software [24].The presented detection algorithm was validated in a sequential order in the following steps: detection-delineation of QRS complexes, P and T-waves [13,16].

TABLE 2. PERFORMANCE CHARACTERISTICS OF SOME PROPOSED DETECTION-DELINEATION ALGORITHMS TO BE COMPARED WITH EACH OTHER, (NA: NOT APPLICABLE). IN ALL CASES, A COMPUTER SYSTEM WITH FOLLOWING SPECIFICATIONS WAS USED: CPU SPEED: 2.4 GHZ, RAM: 2 GB AND CASH: 4 MB.

Detection Algorithm	Development Environment	Speed Samples/sec	Detection/Delineation	Dependency of Parameters to Sampling Frequency	Maximum Delineation Error msec (RMS)	Se /P+ (%)
Conventional Hilbert Transform [12]	MATLAB	52,710	Yes/No	Yes	NA	99.61/99.42
Modified Hilbert Transform [11]	MATLAB	43,830	Yes/No	Yes	NA	99.70/99.75
Conventional Discrete Wavelet Transform [15]	MATLAB	47,934	Yes/Yes	Yes	12.33	99.80/99.86
DWT-based Area Curve Length Method [16]	C++/MEX (MATLAB)	101,701	Yes/Yes	Yes	7.26	99.91/99.88
DWT-based Multiple Higher Order Moments Method [13]	C++/MEX (MATLAB)	148,943	Yes/Yes	Yes	6.14	99.94/99.91
GIM (This Study)	C++/MEX (MATLAB)	185,401	Yes/Yes	No	5.29	99.96/99.96

As the final step of the accuracy performance checking, in each case the trend of RR-tachogram is obtained and plotted. It should be noted that if RR-interval is remarkably less than the mean value, probable false positive (FP) error may exist. On the other hand, if RR-interval is significantly greater than the mean value, the false negative error may probably exist. In Fig. 12, two examples of RR-tachogram obtained from holter data of two patients of hospital including PVC and PAC beats are shown.

In Table 2, some specifications of the ECG events detection-delineation are shown. According to this table, the presented detection-delineation algorithm possesses the best performance characteristics namely as speed of processing, accuracy and robustness against noise and motion artifacts.

V. CONCLUSION

In this study the design procedure of a C++/Mex solution for the detection and delineation of long-duration ambulatory holter ECG individual events are described. To achieve this objective, detailed flow-chart of the proposed ECG detection delineation algorithm as well as the corresponding lines in the C++/Mex code are explained. In addition, two methodologies for generating the detection-delineation DS named as ACLM and GIM and their appropriate programming implementations are explained. The presented code has evolved via its application to several databases such as MIT-BIH Arrhythmia Database, QT Database, and T-Wave Alternans Database and as a result, the average values of Se = 99.96% and P+ = 99.96% are obtained for the detection of QRS complexes, with the average maximum delineation error of 5.7 msec, 3.8 msec and 6.1 msec for P-wave, QRS complex and T-wave respectively. In addition, the code is applied to DAY general hospital high resolution holter data (more than 1,500,000 beats including BBB, PVC and PAC) and average values of Se=99.98% and P+=99.98% are obtained for QRS detection. In summary, marginal performance improvement of ECG events detection-delineation process in a widespread values of SNR, reliable robustness against strong noise, artifacts and probable severe arrhythmia(s) of high resolution holter data and the processing speed 185,000 samples/sec can be mentioned as important merits and capabilities of the presented C++/Mex solution.

REFERENCES

[1]. Ghaffari, M. R. Homaeinezhad, "Fading Parameters of Sodium, Potassium and Leakage Ionic Channels the Best Linear Unbiased Sequentially Estimation (BLUE) Via Voltage Clamp Technique Noisy Measurement," 16th. Annual (International) Conference on Mechanical Engineering-ISME 2008 May 14-16, 2008, Shahid Bahonar University of Kerman, Iran.

[2]. Ghaffari, M. R. Homaeinezhad, M. Atarod, Y. Ahmady, R. Rahmani, "Detecting and Quantifying T-wave Alternans Using the Correlation Method and Comparison with the FFT-based Method," 34th Annual Conference of Computers in Cardiology (CinC), September 14-17 2008, Bologna, Italy.

[3]. Ghaffari, M. R. Homaeinezhad, M. Akraminia, M. Atarod, and M. Daevaeiha, "Detecting and Quantifying T-Wave Alternans in Patients with Heart Failure and Non-Ischemic Cardiomyopathy via Modified Spectral Method," 35th Annual Conference of Computers in Cardiology (CinC), September 13-16 2009, Lake City-Utah, USA.

[4]. Ghaffari, M. R. Homaeinezhad, M. Akraminia, M. Atarod, and M. Daevaeiha, "Detecting and Discriminating Premature Atrial and Ventricular Contractions: Application to Prediction of Paroxysmal Atrial Fibrillation," 35th Annual Conference of Computers in Cardiology (CinC), September 13-16 2009, Lake City-Utah, USA.

[5]. F. Chiarugi, M. Varanini, F. Cantini, F. Conforti, G. Vrouchos, "Noninvasive ECG as a Tool for Predicting Termination of Paroxysmal Atrial Fibrillation," IEEE Transactions on Biomedical Engineering, Vol. 54, No. 8, pp. 1399-1406, August 2007.

[6]. Christov, I. Simova, "Q-onset and T-end delineation: assessment of the performance of an automated method with the use of a reference database," Physiological Measurement, 28, 213–221, 2007.

[7]. F. A. A. Minhas, M. Arif "Robust electrocardiogram (ECG) beat classification using discrete wavelet transform," Physiological Measurement, Vol. 29, pp. 555-570, 2008.

[8]. H. Lin, Y. C. Du, T. Chen, "Adaptive wavelet network for multiple cardiac arrhythmias recognition," Expert Systems with Applications, 34, 2601–2611, 2008.

[9]. Sayadi, M. B. Shamsollahi, "A model-based Bayesian framework for ECG beat Segmentation," Physiological Measurement, 30, 335–352, 2009.

[10]. M. Arzeno Natalia, Zhi-De Deng, and Chi-Sang Poon, "Analysis of First-Derivative Based QRS Detection Algorithms," IEEE Transactions on Biomedical Engineering, Vol. 55, No. 2, pp. 478-484, February 2008.

[11]. Ghaffari, M. R. Homaeinezhad, M. Akraminia, and M. Atarod, "A Methodology for Prediction of Acute Hypotensive Episodes in ICU via a Risk Scoring Model including Analysis of ST-Segment Variations," Cardiovascular Engineering, In-Press, 2010.

[12]. Benitez, P. A. Gaydecki, A. Zaidi, A. P. Fitzpatrick, "The use of the Hilbert transform in ECG signal analysis," Computers in Biology and Medicine, Vol. 31 pp. 399–406, 2001.

[13]. Ghaffari, M. R. Homaeinezhad, M. Khazraee, M. Daevaeiha, "Segmentation of HolterECG Waves via Analysis of a Discrete Wavelet-Derived Multiple Skewness-Kurtosis Based Metric," Annals of Biomedical Engineering, Springer Publishing, In-Press, 2010.

[14]. M. Mitra, S. Mitra, "A Software Based Approach for Detection of QRS Vector of ECG Signal," IFMBE Proceedings, 15, pp. 348-351, 2007.

[15]. Martinez J. P., R. Almeida, S. Olmos, A. P. Rocha, P. Laguna, "A Wavelet-Based ECG Delineator: Evaluation on Standard Databases," IEEE Transactions on Biomedical Engineering, Vol. 51, No. 4, pp.570-581, 2004.

[16]. Ghaffari, M. R. Homaeinezhad, M. Akraminia, M. Atarod, and M. Daevaeiha, "A Robust Wavelet-based Multi-Lead Electrocardiogram Delineation Algorithm," Medical Engineering & Physics, 31(10):1219–1227, 2009.

[17]. Ghaffari, M. R. Homaeinezhad, M. Akraminia, and M. Daevaeiha, "Finding Events of Electrocardiogram and Arterial Blood Pressure Signals Via Discrete Wavelet Transform with Modified Scales," Imeche Proceedings, Part H: Engineering in Medicine, In-Press, 2009.

[18]. N. Kannathal, C.M. Lim, U. R. Acharya, P. K. Sadasivan, "Cardiac state diagnosis using adaptive neuro fuzzy technique," Medical Engineering & Physics, 28, 809–815, 2006.

[19]. G. de Lannoy, B. Frenay, M. Verleysen, J. Delbeke, "Supervised ECG Delineation Using the Wavelet Transform and Hidden Markov Models," The Proceedings of IFMBE, Vol. 22, pp. 22-25, 2008.

[20]. G. B. Moody, R. G. Mark, "The MIT-BIH Arrhythmia Databaseon CD-Rom and Software for it," The Proseeding of Computers in Cardiology, pp. 185-188, 1990.

[21]. P. Laguna, R. Mark, A. Goldenberger, G. B. Moody, "A Database for Evaluation of Algorithms for Measurement of QT and Other Waveform intervals in ECG," The Proceeding of Computers in Cardiology, pp. 673-676, 1997.

[22]. Taddei, G. Distante, M. Emdin, P. Pisani, G. B. Moody, C. Zeelenberg, C. Marchesi, "The European ST-T Database: Standards for Evaluating Systems for the Analysis of ST-T Changes in Ambulatory Electrocardiography," European Heart Journal, Vol. 13, pp. 1164-1172, 1992.

[23]. G. B. Moody, "The PhysioNet/Computers in Cardiology Challenge 2008: T-Wave Alternans," The Proceeding of Computers in Cardiology 2008, Proceeding of the Computers in Cardiology, 2008; 35: 505−508.

[24]. G. B. Moody, "WFDB Applications Guide," Tenth Edition, Harvard-MIT Division of Health Sciences and Technology, 2006. (http://www.physionet.org/physiotools/wag/)

[25]. Li, C. Zheng, C. Tai, "Detection of ECG Characteristic Points using Wavelet Transforms," IEEE Transactions on Biomedical Engineering, Vol. 42, pp.21-28, 1995.

[26]. P. S. Hamilton, W. Tompkins, "Quantitative Investigation of QRS Detection Rules using the MIT/BIH Arrhythmia Database," IEEE Transactions on Biomedical Engineering, Vol. 33, pp. 1157-1165, 1986.

[27]. J. Pan, W. J. Tompkins, "A Real-Time QRS Detection Algorithm," IEEE Transactions on Biomedical Engineering, Vol. 32, pp. 230-236, 1985.

[28]. G. B. Moody, R. G. Mark, "Development and Evaluation of a 2-Lead ECG Analysis Program," The Proceeding of Computers in Cardiology, pp. 39-44, 1982.

[29]. P. Laguna, R. Jane, P. Caminal, "Automatic Detection of Wave Boundaries in Multi Lead ECG Signals: Validation with the CSE Database," Computers and Biomedical Research, Vol. 27, No.1, pp. 45-60, 1994.

[30]. J. Vila, Y. Gang, J. Presedo, M. Fernandez-Delgado, M. Malik, "A New Approach for TU Complex Characterization," IEEE Transactions on Biomedical Engineering, Vol. 47, pp. 764-772, 2000.

[31]. Douglas C. Montgomery, George C. Runger, "Applied Statistics and Probability for Engineers," Third Edition, John Wiley & Sons, 2003.

[32]. Christopher M. Bishop, "Pattern Recognition and Machine Learning," Springer Publishing, 2006.

[33]. Steven M. Kay, " Fundamentals of Statistical Signal Processing: Estimation Theory", Prentice- Hall Inc. 1979.

[34]. H. Vincent Poor, "An Introduction to Signal Detection and Estimation," Second Edition, Springer-Verlag, 1994, Chapters 1-3.

[35]. Physionet.org,http://www.physionet.org/physiobank/database/.

APPENDIX A (C++/Mex code)

```
#include "mex.h"
#include "string.h"
#include "math.h"
#include "new.h"

double *ddxx;

//Function for calculating average value of array v
double mean (double *v, int l){
   int j;
   double k=0;
   for(j=0;j<l;j++)
      k=k+v[j];
   return (k/double(l));
}

//Function for calculating standard deviation value of array v
double stdi (double *v,int l, double me){
   double k=0;
   int j;
   for (j=0;j<l;j++)
      k=k+(v[j]-me)*(v[j]-me);
   k=k/(double(l)-1);
   k=sqrt(k);
   return k;
}

//Function for calculating finite integral of array v
double integral (double *v, int st, int fi){
   int j;
   double k=0;
   for (j=st;j<fi;j++)
      k=k+(0.5*(v[j]+ v[j+1]));
   return k;
}

//Function for calculating finite integral of absolute value of array v
double integralabs (double *v, int st, int fi){
   int j;
   double k=0;
   for (j=st;j<fi;j++)
      k=k+(0.5*(fabs(v[j])+ fabs(v[j+1])));
   return k;
}

//Function for calculation of detection-delineation decision statistic
//ACL Curve Length Method (ACLM)
void ACL (double *v,int l, double *out,int wl, double * dif){
   double area,curve;
   int j,st,fi;
   for (j=0;j<l;j++){
   st = j - wl;
   fi = j + wl;
   if (st < 0) st = 0;
   if (fi > l-1)fi = l-1;
   area = integralabs(v,st,fi);
   curve = integral(dif,st,fi-1);
   out[j] = area*curve;
   }
   return;
}

//Function for finding maximum of array v
int maxfinder (double *v,int st,int fi){
   int k=st;
   int j;
   for (j=st+1;j<=fi;j++)
      if (v[j]>v[k]) k=j;
   return k;
}

//Function for finding maximum of absolute value of of array v, lli is a reference for difference
int maxabsfinder (double *v,int st,int fi,double lli){
   int k=st;
   int j;
   for (j=st+1;j<=fi;j++)
      if (fabs(v[j]-lli)>fabs(v[k]-lli)) k=j;
   return k;
}

//Function for calculating the noise level for the specefic neighberhood (window) defined in noisechecker() function
double regre (double *xx,int st,int fi,int wxx){
   int i;
   double *n,*xs,ym,xm,yst,xst,zig=0,k;
   n= new double [2*wxx+1];
   xs= new double [2*wxx+1];
   for (i=st;i<=fi;i++){
      n[i-st]=xx[i];
      xs[i-st]=4*i;
      zig=zig+(xx[i]*4*i);
   }
   ym=mean(n,2*wxx+1);
   xm=mean(xs,2*wxx+1);
   yst=stdi(n,2*wxx+1,ym);
   xst=stdi(xs,2*wxx+1,xm);
   k=((zig/(2*wxx+1))-(xm*ym))/(xst*yst);
   delete [] n;
   delete [] xs;
   return k;
}

//Function for determining noise level by using regre() function
int noisechecker (int edg){
   int wxx=20;
   double reg;
reg=fabs(regre(ddxx,edg-wxx,edg+wxx,wxx));
   if ((reg<0.003)) return 0;
   else if ((reg>=0.003)&&(reg<0.5))return 1;
   else return 2;
}

//Function for improving right edge location in array v
double rchecker(double *v,double * ms,int st,int fi,double edg,double maxm){
   int i,noi;
   bool cm;
   double ch;
   i=st;
   while (i<=fi){
      if (ms[i]>maxm) {
            maxm=ms[i];
            cm=0;}
      else cm=(ms[i]<=0.0001*maxm);
   ch=v[int(edg)]-0.005;
```

```
    if (cm&&(v[i]<ch)){
      noi=noisechecker(i);
      if (noi==2) edg=i;
      else if (noi==0) i=fi;
    }
    i++;
    }
    return edg;
}

//Function for improving left edge location in array v
double lchecker(double *v,double * ms,int st,int fi,double edg,double maxm){
    int i,noi;
    bool cm;
    double ch;
    i=st;
    while (i>=fi){
      if (ms[i]>maxm) {
          maxm=ms[i];
          cm=0;}
      else cm=(ms[i]<=0.0001*maxm);
    ch=v[int(edg)]-0.005;
    if (cm&&(v[i]<ch)){
      noi=noisechecker(i);
      if (noi==2) edg=i;
      else if (noi==0) i=fi;
    }
    i--;
    }
    return edg;
}

//MATLAB C++/MEX function
void mexFunction( int nlhs, mxArray *plhs[],int nrhs, const mxArray *prhs[] ){

    //Definig variables
    mxArray *m,*mxx;
    mwSize  len;
    double *dd,*d,*dif,*f,*fxx,mm,stdd,*rm,lim,maxm,*redge,*ledge,*edgenum,*ld,*maxs,*maxxs;
    int i,j,k,rmnum,*maxp,*maxx,maxplen,sc,maxt,lm,rmm,md;
    bool c1,c2,c3,c4;

    //DWT signal
    len = mxGetNumberOfElements(prhs[0]);
    m=mxDuplicateArray(prhs[0]);
    dd=mxGetPr(m);

    mxx=mxDuplicateArray(prhs[1]);
    ddxx=mxGetPr(mxx);

    plhs[0]=mxCreateDoubleMatrix(len,1,mxREAL);
    d = mxGetPr(plhs[0]);

    f=new double [len];
    fxx=new double [len];
    rm=new double [len];
    dif=new double[len];

    for(i=0;i<len-1;i++){
      dif[i]=dd[i+1]-dd[i];
      dif[i]=sqrt(1+(dif[i]*dif[i]));}


    ACL(dd,len,d,20,dif);

    mm=mean(d,len);
    stdd=stdi(d,len,mm);

    for(i=0;i<len;i++)
      d[i]=(d[i]-mm)/stdd;

    for(i=0;i<len-1;i++){
      f[i]=fabs(d[i+1]-d[i]);
    }

    rmnum=0;
    for(i=0;i<len-2;i++)
      if ((d[i]<d[i+1])&&(d[i+1]>d[i+2])){
        rm[rmnum]=d[i+1];
        rmnum++;}

    mm=mean(rm,rmnum);
    stdd=stdi(rm,rmnum,mm);
    lim=mm+0.75*stdd;

    maxp=new int [rmnum];

    i=0;
    maxplen=1; maxp[0]=0;

    while (i<len-1){
      if((d[i]>=lim)&&(d[i]<=d[i+1])){
        j=1;
while ((i+j<len)&&(d[i+j]>=lim))
          j++;
        maxt=maxfinder(d,i,i+j);
        if((maxt-maxp[maxplen-1])>600) {
          maxp[maxplen]=maxt;
          maxplen++;}
        else if (d[maxp[maxplen-1]]<d[maxt])
          maxp[maxplen-1]=maxt;
        i=i+j+10;
      }
      else i++;
    }

    plhs[1]=mxCreateDoubleMatrix(maxplen,1,mxREAL);
    ledge = mxGetPr(plhs[1]);
    plhs[2]=mxCreateDoubleMatrix(maxplen,1,mxREAL);
    redge = mxGetPr(plhs[2]);
    plhs[3]=mxCreateDoubleMatrix(maxplen,1,mxREAL);
    maxs = mxGetPr(plhs[3]);
    plhs[4]=mxCreateDoubleMatrix(1,1,mxREAL);
    edgenum=mxGetPr(plhs[4]);
    edgenum[0]=maxplen;
    plhs[5]=mxCreateDoubleMatrix(maxplen,1,mxREAL);
    maxxs=mxGetPr(plhs[5]);
    plhs[6]=mxCreateDoubleMatrix(1,1,mxREAL);
    ld=mxGetPr(plhs[6]);

    for(i=0;i<maxplen;i++)
      maxs[i]=maxp[i]+1;

    for (i=1;i<maxplen-1;i++){
      sc=maxp[i];
      j=0;
      maxm=f[sc-1];
      do{
      j++;
      c1=(d[sc-j]<d[sc-j-1])||(d[sc-j-1]>d[sc-j-2]);
      if (f[sc-j-1]>maxm) {
        maxm=f[sc-j-1];
        c2=1;}
      else c2=(f[sc-j-1]>=0.0001*maxm);
      c3=d[sc-j-1]>=0;
      }while((c1&&c2)||c3);

      if(j<100) k=100; else k=j;
      ledge[i]=lchecker(d,f,sc-j,sc-int(2.5*k),sc-j-1,maxm)+1;

      j=0;
      maxm=f[sc];
      do{
      j++;

      c1=(d[sc+j]<d[sc+j+1])||(d[sc+j+1]>d[sc+j+2]);
      if (f[sc+j+1]>maxm) {
        maxm=f[sc+j+1];
        c2=1;}
      else c2=(f[sc+j+1]>=0.0001*maxm);
```

```
        c3=d[sc+j+1]>=0;
        }while((c1&&c2)||c3);

        if(j<100) k=100; else k=j;
        redge[i]=rchecker(d,f,sc+j,sc+int(2.5*k),sc+j+1,maxm)+1;

    }

    maxx=new int [maxplen];

    for (i=1;i<maxplen-1;i++)
        maxx[i]=maxabsfinder(ddxx,ledge[i],redge[i],(ddxx[int(ledge[i])]+
ddxx[int(redge[i])])/2);

    for(i=0;i<maxplen;i++)
        maxxs[i]=maxx[i]+1;

    lm=maxp[1]-ledge[1];
    rmm=redge[1]-maxp[1];
    for (i=2;i<maxplen-1;i++){
        if (lm<(maxp[i]-ledge[i])) lm=(maxp[i]-ledge[i]);
        if (rmm<(redge[i]-maxp[i])) rmm=(redge[i]-maxp[i]);
    }

    if (lm>rmm) md=lm+10;
    else md=rmm+10;
    if (md>600) md=600;
    ld[0]=md;

    Delete [] dif;
    Delete [] f;
    Delete [] rm;
    Delete [] maxp;

    return;
}
```

APPENDIX B (MATLAB CODE FOR COMPILING THE C++ WAVE DELINEATION)

```
% Clearing used memory and closing other open projects
tic
clc;clear all;close all,

% Loading needed the signal source files
load part1;
load UIR;
load UIR1;

% Compiling the c++ code to generate ACLcalc
mex ACLcalc.cpp;

% Choosing data
data = double(data);
ECG = data(:,1);
tt1=toc;

% High-pass and low-pass filtering to remove noise and motion
artifacts.
hh = floor(length(h) / 2);
hh1 = floor(length(h1) / 2);

ECG = conv(h,ECG);
ECG = ECG(hh : end - hh);

ECG = conv(h1,ECG);
ECG = ECG(hh1 : end - hh1);

% Removing DC value of the signal
lx = length(ECG);
ECG = ECG - mean(ECG);
tt2=toc;

% a'trous discrete wavelet transform at DWT-scale
an = analyze111(ECG,4);
DWT = an(lx + 1 : 2*lx);
tt3=toc;

% ACLcalc generates required outputs
[ACL,ledges,redges,maxps,num,maxxs,ld]=ACLcalc(DWT,ECG);
tt4=toc;

% Calculating each wave position matrix by adding and deducing
maximum
% Difference between edge and top point to top point position and
putting
% This intervals together
QRS = zeros(2*ld+1,num-2);
for i = 2 : num-1
    QRS(:,i-1)=ECG((maxxs(i)-ld) : (maxxs(i)+ld))- ECG(ledges(i));
end

tt5=toc;

% Plot of obtained results
figure(1);
plot(ECG,'k-','LineWidth',4);
hold on;
plot(ledges(2:num-1),ECG(ledges(2:num-
1)),'o','MarkerEdgeColor','r','MarkerFaceColor','y','MarkerSize',12);
hold on;
plot(maxxs(2:num-1),ECG(maxxs(2:num-
1)),'^','MarkerEdgeColor','r','MarkerFaceColor','y','MarkerSize',12);
hold on;
plot(redges(2:num-1),ECG(redges(2:num-
1)),'s','MarkerEdgeColor','r','MarkerFaceColor','y','MarkerSize',12);
xlabel('Sample Number')
legend ('ECG','Onset','Peak','Offset',1)

figure(2);
plot(DWT,'k-','LineWidth',4);
hold on;
plot(ledges(2:num-1),DWT(ledges(2:num-
1)),'o','MarkerEdgeColor','r','MarkerFaceColor','y','MarkerSize',12);
hold on;
plot(redges(2:num-1),DWT(redges(2:num-
1)),'s','MarkerEdgeColor','r','MarkerFaceColor','y','MarkerSize',12);
xlabel('Sample Number')
legend ('DWT','Onset','Offset',1)

figure(3);
plot(ACL,'k-','LineWidth',4);
hold on;
plot(ledges(2:num-1),ACL(ledges(2:num-
1)),'o','MarkerEdgeColor','r','MarkerFaceColor','y','MarkerSize',12);
hold on;
plot(maxps,ACL(maxps),'^','MarkerEdgeColor','r','MarkerFaceColor'
,'y','MarkerSize',12);
hold on;
plot(redges(2:num-1),ACL(redges(2:num-
1)),'s','MarkerEdgeColor','r','MarkerFaceColor','y','MarkerSize',12);
xlabel('Sample Number')
legend ('ACL','Onset','ACL Peak','Offset',1)

figure (4);
plot (QRS,'k-','LineWidth',4)

tt6=toc;

% Calculating each part time
load_time=tt1
firstfilter_time=tt2-tt1
secondfilter_time=tt3-tt2
whole_Load_Fllter_time=tt3
acl_time=tt4-tt3
QRS_time=tt5-tt4

whole_Runing_time=tt5

plot_time=tt6-tt5
```

LIST O ACRONYMS

HSEDF: high-speed ECG delineation framework
DS: decision statistic
DSF: decision statistic function
BBB: bundle branch block
TP: true positive
P+: positive predictability (%)
Se: sensitivity (%)
UB: upper bond
LB: lower bond
SMF: smoothing function
FIR: finite-duration impulse response
LE: location error
ACLM: area curve-length method
GIM: geometrical index method
ECG: electrocardiogram
DWT: discrete wavelet transform
SNR: signal to noise ratio
QTDB: QT Database
MITDB: MIT-BIH Arrhythmia Database
TWADB: T-wave alternans Database
FP: false positive
FN: false negative
PVC: premature ventricular contraction
PAC: premature atrial contraction
CHECK#0: procedure of evaluating obtained results using MIT-BIH annotation files
CHECK#1: procedure of evaluating obtained results consulting with a control cardiologist
CHECK#2: procedure of evaluating obtained results consulting with a control cardiologist and also at least with 3 residents