Proceedings of the 28th IEEE
EMBS Annual International Conference
New York City, USA, Aug 30-Sept 3, 2006

Automated adaptive heart interbeat time extraction from long term noisy and variable ECG signals.

Jorge Torres-Solis, Christopher Chan, Tom Chau

Abstract— **We describe the development of an automated, adaptive method to obtain the time interval between successive heart beats from noisy and highly variable electrocardiography signals. These interbeat time series are critical to the fractal characterization of cardiac health. When the biophysical measurement is severely tainted with noise from multiple sources (i.e. [1], [2]), there is a need for algorithms to robustly extract the important patterns from the signal in question. The proposed method yields interevent times that are in close agreement with those obtained by manual extraction, while significantly reducing the requisite processing time. The algorithm can be extended to other applications where simple threshold-based extraction methods fall short.**

I. INTRODUCTION

There are many instances where biophysical measurements become contaminated with noise, and oftentimes these measurements possess important information about an experiment or a patient. In recent years, the advent of portable biomedical devices (such as Holter heart monitors) has increased our capacity to obtain long-term measurements of biophysical parameters away from the hospital. In these long term recordings, transducers often deviate from their original positions and exhibit conductivity problems as the patient performs tasks of everyday living, away from the hospital. These issues tend to severely corrupt measurements ([1], [2], [3]). It would be extremely time consuming and costly to repeat measurements every time we encountered such noise contaminations. The data can be manually categorized and analyzed by an expert in the field, but this is time consuming even for short data records, and for large data sets, manual extraction represents an insurmountable quantity of work. Therefore, we propose to create an automated pattern extraction algorithm, capable of extracting interbeat intervals from noisy ECG data.

II. PROBLEM STATEMENT, THE NATURE OF THE DATA

The data of interest arise from electrocardiography measurements. This paper describes a method to automatically extract the interbeat intervals from these signals, to facilitate fractal analyses [3], [5]. The signal is said to be quasi-periodic because there are long term natural fluctuations between heartbeats [5]. Additionally, the average periodicity of the heart beats naturally change with the level of activity of the subject. In this particular case, a Holter heart monitor (Nasiff Associates Inc., Model KCI X5) was used to obtain the data [6], using three recording channels and a sampling frequency of 128 samples per second.

One of the most prevalent sources of noise in the empirically collected data is due to poor electrical conduction associated with movement of the electrodes. However, we also have intrinsic electrical noise from the device components, noise induced by the body, first by its own electrical currents, and also by working as an antenna for electromagnetic radiation. Hence, the noise captured by the device, especially when the conductivity of the electrodes is poor, is a combination of different sources. Therefore, this additive noise from multiple sources is considered to have a Gaussian distribution [7]. Some parts of the proposed algorithm are based on this assumption.

The ideal signal from a Holter device is a sequence of pulses in time, where each peak should correspond to a R wave. If the recorded data is not corrupted by noise, the event of a heart beat can be easily detected by a rejection threshold. However, in the presence of noise, this simple technique would yield many erroneous event times.

III. RELATED WORK

Several methods have been proposed for the extraction of features from an ECG signal. For a review, please see [4]. Most of these methods rely on a threshold or combinations of thresholds to remove undesired components of the signal and to detect the salient features. In our particular case, where the signal is severely affected by noise, obtaining the position of the desired feature using thresholds would yield a large quantity of spurious and missed events. An example of this situation can be seen in Figure 1, which shows differences in the number of events extracted by a widely used wavelet-based feature extraction algorithm [4] and manual extraction. Clearly, a simple threshold does not suffice in such instances.

Previous research has shown that automatic probabilistic extraction agrees closely with manual extraction for noisy two-state signals [1]. This motivated us to create an algorithm to extract interbeat intervals from analog cardiac signals.

IV. PROPOSED METHOD

A. General Considerations

The proposed method is designed to detect the R waves of the ECG signal. Rather than using a threshold, the method tries to embody the following domain knowledge.

1) The normal heart beat rate for an adult at rest ranges from 60 to 100 heart beats per minute [8], with slightly lower rates for physically fit persons.
2) In typical situations, the ECG is quasi-periodic.
3) In some pathological situations, localized, aberrant fluctuations (arrhythmias) may occur.

Exploiting these observations, a human analyst may look for patterns of peaks emerging from the distorted signal, with a

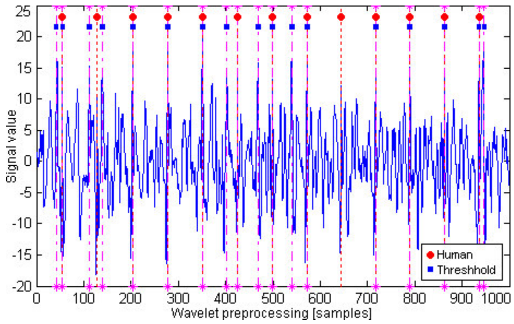

Fig. 1. Signal processed through wavelet thresholding and manual extraction approaches. Note the multiple instances where spurious events are detected by the threshold method.

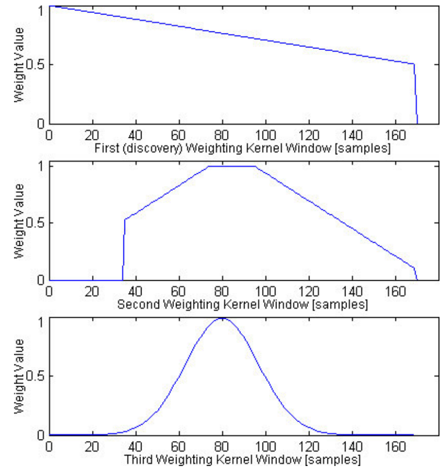

Fig. 2. First (top), second (middle) and third (bottom) proposed weighting kernels

period similar to that expected for normal heart beat rates. Additionally, the most probable location of the next beat will likely depend on the recent history of heart beats.

B. Analysis on a per-window basis

1) Window selection: From literature, we know the minimum expected time between successive heart beats. Hence, the first thing that we did was to extract windows X from the data collection D, of size N samples where at least one heart beat should appear. N is easily adjustable for each patient. The window is defined as:

$$X(i) = D(p + i); i = 1, 2, ..., N \tag{1}$$

where p is the index of the data where the last beat was found. Differences, Y,

$$Y(i) = X(i) - \overline{X} \tag{2}$$

around the mean were found within the window.

2) Signal Detrending: To counteract the effects of rising or falling trends in the signal, we calculated the corresponding linear regression values of these differences, denoted by Y_R. We obtained a set of detrended differences, Y_N, by subtracting the regression values from the trended data.

$$Y_N(i) = Y(i) - Y_R(i) \tag{3}$$

3) Probabilistic Weighting: For the purpose of finding the beat with highest probability within a window, it makes sense to assign more weight to times where the next expected beat might occur. Conversely, samples only milliseconds away or many seconds removed from the previous event would not likely be another event, but either spurious artifacts of the previous or subsequent beats. Low weights should be given to these zones. The proposed model uses three weighting kernels of the same size N as the window selected, reflecting probabilistic models of event occurrence.

The first weighting kernel is called the discovery weighting kernel, and assigns linearly decreasing weights to positions further away from the beginning time. This kernel is deployed in seeking the first possible heart beat event time. The other two kernels (truncated trapezoid and Gaussian) are 'beat seeking' kernels, in that these kernels favour the region where a typical heart beat should appear, while discouraging the selection of unexpected beats. For the Gaussian weighting factor more weight is given to the values of the window around its mean. The value for the standard deviation is chosen such that the distance between the samples where the distribution assumes half its maximum value, corresponds to 40 heart beats per minute. This technique is similar to the 3 dB rejection zone for filter design. The mean is variable, depending on a learning adaptive algorithm. The shapes of the suggested weighting factors are shown in Figure 2. It is important to note that the second weighting kernel must be less specific than the third in terms of rejection of the values of the window. The selection of the weighting kernel to be used, and the parameters to generate it are determined by the adaptive scheme explained later.

Weighted differences Y_W,

$$Y_W(i) = Y_N(i) \times W_X(i) \tag{4}$$

are calculated from the detrended values obtained by multiplying each value of Y_N by the corresponding kernel weight, W_X. The location of the selected beat in the window is given by the maximum value of the weighted result.

4) Summary of Intra-window processing: The processing within the values of a window is summarized below.

1) Obtain a set of detrended differences Y_N by subtracting the regression values from the trended data as in equation (3).
2) Define three weighting kernels, where the first is meant to discover the nearest peak of the signal, and the other two are intended to give more weight to a specific region of the window.
3) Calculate the weighted differences as given by equation (4). The highest peak is the most probable location of the desired beat.

The intra-window processing method is invoked by the *inter-window* adaptive scheme described below.

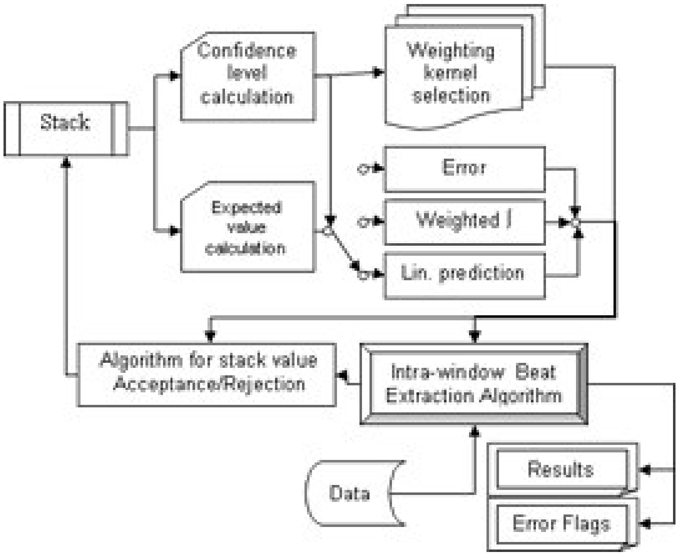

Fig. 3. Adaptive scheme block diagram

C. Inter-window Processing and Adaptive Stages

The following outline describes the interwindow and adaptive processing summarized in Figure 3.

1) Initialize a stack with zero values (No knowledge state).
2) Define a confidence level based on the number of zero values contained in the stack
   a) Low level of confidence: the stack is full of zeros or just a small quantity of the values (under a threshold) contained in it are non-zero.
   b) Medium level of confidence: the stack contains a number of non-zero values in excess of the threshold for case a but below the threshold for case c. We further check if the most recent entries into the stack are zeroes, then the confidence level is lowered to case a.
   c) High level of confidence: the stack contains a large number of non-zero values (in excess of some threshold). As above, we also check if the most recent entries are zeroes, in which case the confidence level would be lowered to that of case b.
3) Define the weighting kernel for the confidence level chosen: first kernel for confidence level a, second kernel for confidence level b, and third kernel for level c, respectively.
4) Define the next window (starting at the data value right after the previous beat found, or at the beginning of the data on initialization) and look for a beat within the window using the intra-beat processing method described above.
5) Store the result obtained in a vector of results.
6) If we are in confidence level b or c, calculate an expected value of the position of the beat in the next window from the values in the stack. For case b we use a weighted integration and for case c we perform a linear prediction. For confidence level a, we calculate the expected value from the normal heart rate at rest.
7) Test for acceptance or rejection of the beat time found. First, we test that the amplitude value of the beat found is not part of the population of the surrounding noise. If affirmative, then the beat is accepted as valid. Next, we check if the pulse appears in an expected position within the window, using the expected value just obtained in the previous step as the mean of the population, and the variance calculated from the stack for confidence levels b and c and a predefined value for case a.
8) If the beat is accepted, store its value in the stack. Otherwise, store a zero instead. Also store flags in vectors to indicate the reason for rejection of the beat (amplitude not different from surrounding noise or event time not within expected window), and the confidence level used.
9) Repeat the process from step 2 throughout the data set.

As we can see, the proposed method accounts for different contexts of the data being processed. The algorithm is thus adaptive.

V. Results

Twenty-four hour ECG were recorded from 2 male patients (aged 42 and 28) and one female patient (aged 52). All patients had kidney failure. None had symptomatic coronary artery disease or heart failure. Patients spent a few hours in the hospital and the remainder of time out in the community and at home. As in [1], we assess the algorithm by comparing the events obtained through the automatic algorithm against those derived by a human analyst. Sensitivity (*Se*) and Positive Predictivity (*+P*), as defined below, are often used to quantify extraction performance [4].

$$Se = \frac{TP}{TP+FN} \quad , \quad +P = \frac{TP}{TP+FP} \tag{5}$$

where TP is the number of true positive detected beats, FP refers to the number false positive detections, and FN accounts for the number of false negative detections. Following Chau and Rizvi[1], we also compare the absolute relative differences (Δ) between the mean interbeat intervals extracted by automatic and manual procedures.

$$\Delta = \frac{|\langle T^M \rangle - \langle T^A \rangle|}{\langle T^M \rangle} \times 100\% \tag{6}$$

where $\langle T^M \rangle$ and $\langle T^A \rangle$ denote the mean of the interbeat interval series T^M and T^A, extracted by manual and automatic methods, respectively.

Three-hundred and fifty (350) heart beats were extracted by manual extraction from noisy signals from 3 different patients. The proposed decision algorithm was applied to the signals without preprocessing (raw signal), to the first derivatives of the signals, and finally to Dmey wavelet filtered versions of the signals. We also performed threshold-based detection of heart beats over the same preprocessed

TABLE I

RESULTS COMPARING MANUAL EXTRACTION OF 350 HEART BEATS AGAINST TRADITIONAL THRESHOLD-BASED CLASSIFICATION APPROACHES AND THE PROPOSED AUTOMATIC DETECTION ALGORITHM USING DIFFERENT PREPROCESSING TECHNIQUES

Patient 1						
Conventional methods	TP	FN	FP	Se[%]	+P[%]	Δ[%]
Derivative	258	92	99	73.7	72.3	2.23
Wavelet	318	32	40	90.7	88.8	2.96
Proposed Method	TP	FN	FP	Se[%]	+P[%]	Δ[%]
Raw Data	305	45	28	87.1	91.6	1.40
Derivative	323	27	23	92.3	93.4	0.47
Wavelet	325	25	19	92.7	94.5	0.39
Patient 2						
Conventional Methods	TP	FN	FP	Se[%]	+P[%]	Δ[%]
Derivative	337	13	18	96.3	94.9	0.69
Wavelet	293	57	49	83.7	85.7	0.18
Proposed Method	TP	FN	FP	Se[%]	+P[%]	Δ[%]
Raw Data	320	28	31	92	91.2	0.05
Derivative	340	10	10	97.1	97.1	0.13
Wavelet	297	53	49	84.9	85.8	1.02
Patient 3						
Conventional Methods	TP	FN	FP	Se[%]	+P[%]	Δ[%]
Derivative	315	35	34	90	90.3	4.13
Wavelet	266	84	96	76	73.5	2.19
Proposed Method	TP	FN	FP	Se[%]	+P[%]	Δ[%]
Raw Data	295	55	54	84.3	84.5	0.46
Derivative	334	16	18	95.4	94.9	0.38
Wavelet	305	45	42	87.1	87.9	0.49

signals (derivative and wavelet) for comparison purposes. The threshold was chosen to give a balance between false positive and false negative results. The results for these tests are presented in table I.

From Table I, we see that the proposed method when used with signal preprocessing improves upon existing threshold methods in terms of sensitivity, positive predictivity and absolute relative differences, in all 3 patients. In terms of extraction effort, manual extraction of 350 heart beats took approximately 2 hours and 27 minutes per data set, while the proposed classification algorithm for the same region of the signal containing the 350 extracted beats took approximately 1.5 seconds on an average computer (1.8 GHz, non overloaded), a savings of over 3 orders of magnitude.

VI. CONCLUSIONS

This document has proposed a probabilistic approach for automated extraction of interbeat time intevals from long-term, noisy ECG signals. The results indicate that the algorithm is robust to severe noise contamination.

ACKNOWLEDGMENTS

The authors would like to acknowledge the support of the Bloorview Research Institute, Natural Sciences and Engineering Research Council of Candada and Conacyt.

REFERENCES

[1] T. Chau, S. Rizvi, *Automatic interval extraction from long, highly variable and noisy gait timing signals.* Human Movement Science, vol. 21, 2002. Elsevier Science B.V., pp. 495-513

[2] CH. Chu, EJ. Delp, *Impulsive noise suppression and background normalization of electrocardiogram signals using morphological operators,* IEEE Trans. Biomed. Eng., vol. 36, pp. 262-273, 1989.

[3] P. Ch. Ivanov, Z. Chen, K. Hu and E. Stanley. *Multiscale aspects of cardiac control.* Physica A, vol 344, 2004. Elsevier B.V., pp. 685-704

[4] B.-U. Khler, C. Henning, and R. Orglmeister, *The principles of software QRS detection,* IEEE Eng. Med. Biol. Mag., vol. 21, no. 1, pp. 4257, Jan.-Feb. 2002.

[5] A. L. Goldberger , L. N. Amaral , J. M. Hausdorff , P. Ch. Ivanov , C.-K. Peng, E. Stanley. *Fractal dynamics in physiology: Alterations with disease and aging*. PNAS, Feb. 2002, vol. 99, Sup. 1, pp. 2466-2472

[6] K. Payne. *Ambulatory Electrocardiogram.* WebMD Health. [Online resource]. Last update January 26, 2004. Available: http://www.webmd.com/hw/heart_disease/aa10253.asp

[7] J. G. Proakis, *Digital communications.* 3rd ed., Ed. McGrawHill, 1964, pp. 11, 61-62.

[8] Berkow, Robert. *The Merck Manual of Medical Information.* Second home edition. Merck, section 3, chapter 27 ”Abnormal Heart Rhythms”.