Searching for Similarities in Nearly Periodic Signals With Application to ECG Data Compression

J. Henriques, M. Brito, P. Gil, P. Carvalho
Centre for Informatics and Systems, University of Coimbra, Coimbra, Portugal
M. Antunes
Centre of Cardio-Thoracic Surgery of the University Hospital of Coimbra, Coimbra, Portugal

Abstract

This paper proposes a new methodology to identify and correlate patterns on nearly periodic signal, based on signal simplification and clustering approaches. Using cubic Bezier curves some significant signal samples (control points), enabling to segment adequately the original signal, are extracted in a first step. Next, given the correlation among extracted control points, the detection of similarities within the overall signal is then performed through a clustering technique.

Although the approach is useful for many types of signals, the compression of Electrocardiogram (ECG) signals is here investigated. Results with standard MIT-BIH databases show promising compression ratios, in particular, high compression ratios are found for long duration signals, when the signal presents strong regularities.

1. Introduction

Although digital storage media is currently almost inexpensive and computational power has exponentially increased in last few years, effective compression techniques are still very attractive and useful. Besides the increased storage capacity for archival purposes, compression methods allow real-time transmission over band-limited networks. Among all the areas of medicine, cardiology is one of the branches requiring the largest amount of data acquisition. In particular, the ECG, a biological signal reflecting the heart activity is of major importance: in the context of dedicated monitoring systems running 24 hours/day, such as signal tele-monitoring, intensive-case units or arrhythmia detection system, ECG compression is not only desirable and useful but also necessary. In the past, several methods to analyze and compress ECG signals have been proposed, e.g. [1, 2, 3].

The strong regularity in signals suggests that data compression techniques based on finding coding repetitions in data are likely to be effective. This is specially true for an ECG signal. In fact, two normal ECG cycles usually present a high degree of waveform similarity. It is then natural, in data compression context, to split the signal into beats, and viewing these as standard basic patterns. However, this idea have not yet been exploited (or only partially) by traditional compression methods [4, 5]. Actually, almost compression algorithms that takes advantage of this idea are based on the segmentation of the RR interval, since QRS complex detection is reasonably well understood and there are available several robust and fast algorithms [6, 7]. However, these methods are only viable when applied to normal ECG signals, failing in the presence of abnormal cases, like ventricular tachycardia or ventricular fibrillation.

This work presents a method for identifying and correlate patterns, mainly with application to quasi periodic signals. Although the approach can be applied for several types of signals, the application to ECG compression is here investigated. The central idea consists in finding beat to beat similarities in the ECG signal, without using any king of algorithms for clinical ECG segmentation (such as QRS detection). The proposed scheme consists of two main steps: signal simplification where a cubic Bezier curve is used for piecewise nonlinear interpolation, originating a reduced number of control points, enabling to approximate the original signal; detection of similarities within the simplified signal, by means of a clustering approach. The idea aims at exploiting redundancy through selection of a set of characteristic points. A loss compression method is then applied using a dictionary and exploiting the fact that referencing a dictionary entry takes fewer bytes than encoding the repeating sequence.

The remaining of the paper is organized as follows. In section 2 cubic Bezier curves applied to signal segmentation is presented, as well as the compression method based on the dictionary scheme. In section 3 some experimental results are shown and finally, in section 4, some conclusions are drawn.

2. Methodology

3.1 Signal Segmentation

A general Bezier curve [8] is defined by

$$y(t) = \sum_{i=0}^{L} B_{i,L} \cdot P_i \qquad (1)$$

where $t \in [0,1]$, $P_i \in \Re^n$ $i = 0,...,L$ are control points and $B_{i,L}$ are blending functions given by (2) (polynomials with one degree less than the number of control points).

$$B_{i,L} = \frac{L!}{i!\,(L-i)!}(1-t)^{L-i}t^i \qquad (2)$$

To define a 2-D cubic Bézier curve $y(t)$, four control points $(L = 4)$ are nedded, as shown in Figure 1.

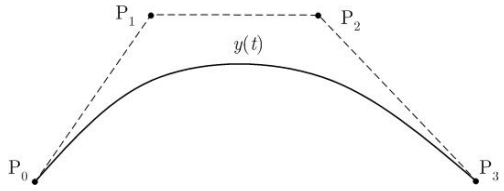

Figure 1. Cubic Bezier curve.

The curve in general does not pass through the control points $P_i(x,y) \in \Re^2$ $i = 0,..,3$ except the first and last points (P_0 and P_3). The Bezier curves have interesting properties: they always are contained within the convex hull of the control points and never oscillates wildly away from the control points [8]. Given their properties, they are commonly used to smoothly interpolate between control points.

Given a discrete time signal $x(k)$, $k = 1,..N$, a cubic Bezier curve can be used to interpolate the original signal provided the four control points are appropriately chosen. Instead of specifying freely the four control points, it is assumed that $x(1) = y(1) = P_0(x,y)$, $x(N) = y(N) = P_3(x,y)$ and the abscis of control points P_0 and P_1 as well as P_2 and P_3. Therefore, only two parameters have to be determined, the ordinates of the control points P_1 and P_2. In particular, from equation (1) and (2), a discrete time cubic Bezier curve can be defined as

$$y(k) = (1-t)^3 \cdot P_0 + 3t(1-t)^2 \cdot P_1 + 3t^2(1-t) \cdot P_2 + t^3 \cdot P_3 \qquad (3)$$

By rewriting the last equation as

$$y(k) = (1-t)^3 \cdot P_0 + t^3 \cdot P_3 + [3t(1-t)^2 \;\; 3t^2(1-t)] \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} \qquad (4)$$

or, in compact form, by

$$Y = B + M \cdot w \qquad (5)$$

$Y \in \Re^N$ is a vector consisting of the values to be approximated, $B \in \Re^N$ and $M \in \Re^{N \times 2}$ are matrices composed of known values and $w \in \Re^2$ ordinates of the control points to be evaluated. Thus, w can be easily obtained, in the least square sense, by

$$w = pinv(M) * (Y - B) \qquad (6)$$

When applied to ECG signals the algorithm is implemented iteratively, so that the most significant points are evaluated. The obtained set of control points stand for the reduced form of representation of the original signal, within a pre-specified threshold ($|x(k) - y(k)| < \varepsilon$). The following figure illustrates its application to the segmentation of an ECG (MIT-BIH#1031, sample 850 to 1200).

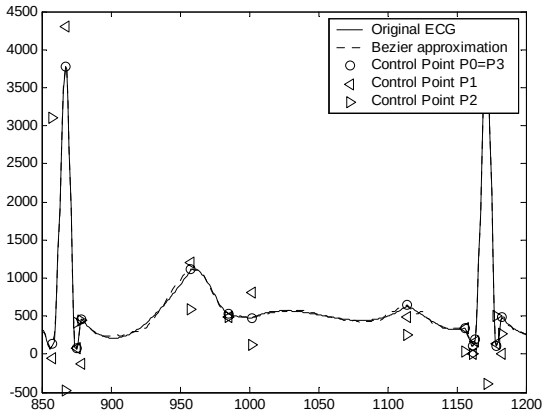

Figure 2. ECG segmenation with Bezier control points.

As can be observed, apart from the abcissa of each control point $I(k)$, three parameters for each segment of the signal (pattern) are needed to describe the piecewise nonlinear interpolation: the values P_0, P_1 and P_2. Obviously, if the algorithm stops here the obtained compression rate is given by $CR = N/(4 \cdot M)$, being N the length of the original signal and *M* the number of extracted control points.

3.2 Searching for Similarities

The compression algorithm proposed here takes into consideration that ECG beats tend to be very similar, although not exactly the same. Thus, in order to carry out data compression, the use of a dictionary consisting of the extracted patterns have been used. If sequences of similar segments occur often in ECG being compressed, theses sequences will be stored in the dictionary according to a certain criterion (error

margin). Compression is achieved since referencing a dictionary entry takes fewer bytes than encoding the repeating sequence. For practical implementation each pattern, depicted in Figure 3, is defined as:

$$\begin{bmatrix} dP & dP_1 & dP_2 \end{bmatrix} = \begin{bmatrix} P_3 - P_0 & P_1 - P_0 & P_2 - P_0 \end{bmatrix} \quad (7)$$

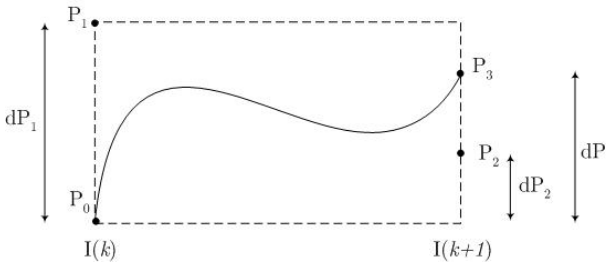

Figure 3. ECG pattern.

Actually, two dictionaries have been used: one for coding of dP values and other for $\begin{bmatrix} dP_1 & dP_2 \end{bmatrix}$ values. The patterns incorporated into the dictionary can be determined by means of a clustering technique, such as k-means or subtractive clustering [9]. Here a modified k-means version has been used, ensuring that all the patterns will belong to a data center, within a pre-specified error. For coding abcissas $I(k)$ a lossless strategy has been applied (delta coding). The amount of compression achieved by the algorithm, defined by the ratio between the number of bits necessary to describe the original data and the number of bits necessary to describe the compressed data is given by

$$CR = \frac{N}{c(I) + I_P + I_{P12} + D_P + 2D_{P12}} \quad (8)$$

Where N is the number of bits necessary to represent the original signal and $c(I)$ the lossless compression of abcissas (*I*). The parameters I_P and I_{P12} define, respectively, the entries for the two dictionaries *dP* and $\begin{bmatrix} dP_1 & dP_2 \end{bmatrix}$ being D_P and $2D_{P12}$ the number of bits needed to store both dictionaries.

3. Experimental Results

ECG records taken from MIT-BIH Arrhythmia [10] and MIT-BIH Malign Arrhythmia Databases [11] were used to experimentally assess the performance of the proposed method. The algorithm was tested using a variety of signals, from normal ECG's to sustained ventricular fibrillation, in order to investigate the amount of necessary patterns and consequently the achieved compression ratio (8), as well as the error induced by the compression process. The *percentage root mean difference* (PRD), equation (9), was used as a distortion coefficient, where signals $x(k)$ and $\tilde{x}(k)$ represent, respectively, the original and compressed signals and $\bar{x}$ defines the original signal average.

$$PRD = \sqrt{\frac{\sum_{k=1}^{N} \left(x(k) - \tilde{x}(k) \right)^2}{\sum_{k=1}^{N} x(k)^2 - \bar{x}}} \quad (9)$$

Table 1 shows the compression ratios and the respective PRD ratios for several ECG signals. The presented values reflect a value of a PRD ratio such that the compressed signal is acceptable (obtained by visual inspection). Moreover, the performance of the algorithm was tested using only the first 5 seconds of each record and the first 400 seconds (digitalized with a sampling rate of 250 Hz and 12 bits representation).

	5 seconds		400 seconds	
MIT[10]	PRD	CPR	PRD	CPR
1031	5,7	9,3	7,1	30,7
1051	4,0	8,5	7,2	25,7
2081	3,9	8,6	7,8	22,6
1191	6,6	13,7	7,6	33,4
2021	4,6	13,4	7,9	40,2
MIT[11]	PRD	CPR	PRD	CPR
418	4,3	7,9	7,4	19,8
420	7,0	9,0	7,9	41,0
602	6,1	7,8	7,4	17,5

Table 1. ECG compression results

Figures 4 and 5 allow visual assessment of the quality of two reconstructed signals. Figure 4 presents a normal ECG signal, in particular the record #1051, and the reconstructed signal for a PRD equal to 4.0.

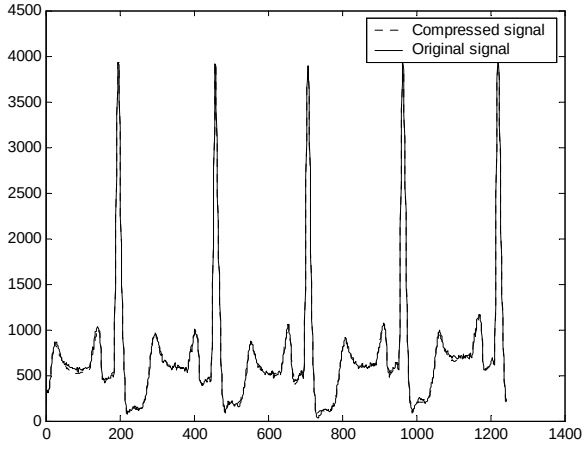

Figure 4. Compression results for the #1051 signal.

Figure 5 shows the performance of the compression method in the presence of a tachycardia/fibrillation, namely record #418, and the reconstructed signal for PRD equal to 4.3.

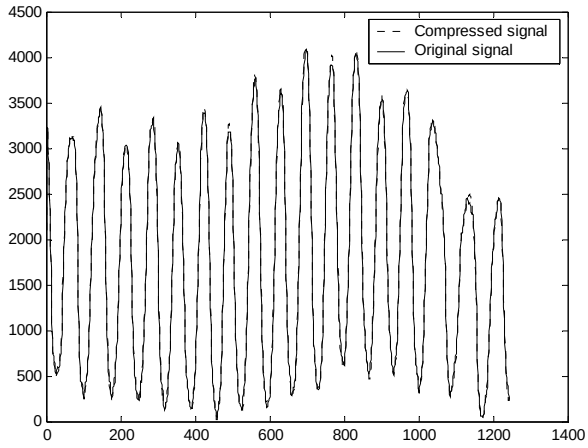

Figure 5. Compression results for the #418 signal.

Experimental results show that the proposed method achieves a good compression ratio with effectively reconstruction quality, namely an excellent preservation of QRS complexes and other important signal features. Furthermore, experiments with ECG records compares favorably with various classical and state-of the-art ECG compressors.

Computationally, the algorithm is very simple and is viable in scenarios where limited computing power is available. The decompression process is extremely lightweight while the compression stage, although more computationally intensive than the decompressor, is relatively lightweight as well, since the process to determine the model coefficients is a least squares algorithm. Furthermore, a clustering procedure is involved.

The proposed method is specially interesting when signals present strong regularity leading, as expected, to high rate of compression ratios.

4. Conclusions

A methodology able to detect similarities in time domain signal with application to data compression has been proposed. The method is based on a Bezier curve interpolation and on a pattern recognition phase founded on a clustering technique. Experimental results have shown compression performances in the range of the state-of-the-art methods. The main features of this method are: *i*) the proposed method provides a data segmentation without any help of a pre-segmentation algorithm, exploiting QRS detection (R-R interval); *ii*) the complexity of the method is low, which means it can be used for real-time purposes; *iii*) although the method is a general-purpose, it is more efficient for long quasi periodic signals; *iv*) the method can implicitly deal with different types of wave forms, namely ECG normal sinus rate, ventricular tachycardia and ventricular fibrillation.

Although a compression case-study has been shown, other practical applications concerning feature detection are currently under study and development. In fact, in cardiology context where several biosignals exhibit strong periodicities, there is a great interest in detecting anomalies or dysfunctions, such as harmful ECG morphologies and arrhythmias.

Acknowledgements. MyHeart EU project (IST-2002-507816) and CISUC (Center for Informatics and Systems of University of Coimbra).

References

[1] S. Aase, R. Nygaard, J. Husøy and D. Haugland, *Optimized time and frequency domain methods for ECG signal compression*, Internal Report, 1998.

[2] R. Istepanian, L. Hadjileontiadis and S. Panas, "ECG data compression using wavelets and higher order statistics methods", *IEEE Transactions on Information Technology in Biomedicine*, 2, 108-115, 2001.

[3] S. Jalaleddine, C. Hutchens and R Strattan, "ECG data compression techniques: a unified approach", *IEEE Trans. Biomed. Eng*., 37, 329-343, 1990.

[4] Chia-Chun Sun, *ECG Compression algorithms utilizing the interbeat correlation*, PhD. Thesis, July 2005.

[5] C. Giurcaneanu, I. Tabus and S. Mereuta, "Using contexts and R-R interval estimation in lossless ECG compression", *Computer Methods Programs Biomed*, 67(3):177-86, 2002.

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

[7] K. Hennig and R. Orglmeister, "The principles of software QRS detection. Review and comparing algorithms for detecting this important ECG waveform", *IEEE Eng in Med. and Biol*, 21, pp. 42-57, 2002.

[8] J. Foley, A. van-Dam, S. Feiner, J. Hughes and R. Phillips, *Introduction to Computer Graphics*, Addison Wesley, 1997.

[9] Matlab, *Fuzzy Logic Toolbox*, The MathWorks Inc., 2004.

[10] http://www.physionet.org/physiobank/database/html /mitdbdir/mitdbdir.htm

[11] http://www.physionet.org/physiobank/database/vfdb/