AN ECG COMPRESSION APPROACH BASED ON A SEGMENT DICTIONARY AND BEZIER APPROXIMATIONS

†CISUC, Centre for Informatics and Systems, University of Coimbra
‡CCT-HUC, Centre of Cardio-thoracic Surgery, University Hospital of Coimbra
\{mbrito, jh, carvalho, bribeiro\}@dei.uc.pt, antunes.cct.huc@sapo.pt

M. Brito†, J. Henriques†, P. Carvalho†, B. Ribeiro†, M. Antunes‡

ABSTRACT

This paper proposes a methodology for ECG (electrocardiograms) data compression based on R-R segmentation. An ECG can be seen as a quasi-periodic signal, where it is possible to find many similarities between heart beats. These similarities are explored by the proposed compression scheme through the use of a segment dictionary combined with an efficient form of progressive error codification. The dictionary is able to incorporate new patterns, in order to assure the algorithm adapts to changes in signal morphology.

Experimental results reveal that high compression ratios are possible for highly regular signals, with irregular signals still achieving acceptable results.

1. INTRODUCTION

With the growing adoption of clinical information systems and electronic patient records, paper is slowly being abandoned as a means for archiving clinical data. This technological shift brings both patients and physicians many advantages, allowing for remote data access and even remote consultation through the use of telemedicine systems. These systems, however, come at a price in terms of bandwidth (eg. real-time streaming of vital signals) and storage space (eg. 24 hour/day signal acquisition), making very desirable the existence of mechanisms which reduce application demands on these two fronts. In the context of cardiology, it is often necessary to perform 24-hour/day data acquisition from several leads, so the possibility to perform ECG compression is a clear advantage.

The problematic of ECG data compression is not new and has been a research topic for decades. The AZTEC algorithm [1] is among the earlier lossy ECG compression proposals. It involves approximating the ECG through the codification of sequences of slopes and plateaus, with slopes being predominantly used on the QRS complex.

Predictive approaches based on linear models [2] are also known to provide reasonable compression ratios at a reduced computational cost. These solutions typically involve using a first order regression model to predict the value of each sample based on the \( n \) past samples and encoding the predictive error with an entropy encoder.

Dictionary based compression methods, such as Lempel-Ziv [3], are very popular for lossless data compression. A version of LZ77 was adapted to support approximate matching [4] in order to achieve high compression ratios for 8-bit ECG signals.

In recent years, the Wavelet Transform sub-band decomposition has been gaining increased popularity among transform-based methods applied to ECG compression. A well known method is the very efficient SPIHT (Set Partitioning In Hierarchical Trees) wavelet compression algorithm, which has been shown to provide high compression ratios with reduced signal degradation [5].

Among the several approaches found in literature, most of these fail to recognize the ECG as a quasi-periodic signal and as such do not take advantage of inter-beat correlations. Such methods generally work with a fixed-size window, with no concern with ECG segmentation, or work with adjacent samples only and are thus only able to explore local contexts. It has been shown that methods which explore the beat similarity between heart cycles are able to achieve high compression ratios. In [6] singular value decomposition was used to extract the principal components of the ECG signal and reach very high compression ratios. In [7] ECG segmentation and normalization were used in conjunction with an adapted version of JPEG2000 provided a good solution capable of exploiting the similarity between adjacent heartbeats.

The objective of this work is to further investigate approaches that explore the existing similarities among heart beats. A new algorithm is proposed which takes advantage of the inter-beat similarities of the ECG signal through the use of an adaptive dictionary. Two approaches are combined, which allow switching between a high-quality mode using Bezier curve decomposition, and a high-compression mode where beat similarities are encoded.

The paper is organized as follows: in section 2, the proposed method will be described, and implementation choices will be presented and discussed. In section 3, results will be presented, and there will be a discussion regarding the type of signals that can be expected to work well with this methodology, as well as expected compression ratios. Finally, in section 4, a summary will be presented.
2. PROPOSED ALGORITHM

The proposed algorithm can be divided on several stages, which are described in the following diagram.

![Functional diagram of the proposed algorithm](image)

This section will discuss each phase of the algorithm as described in Figure 1 as well as relevant implementation decisions. The irregular algorithm step is beyond the scope of this work.

2.1 Peak Detection and segment analysis

A QRS detector based on the work of Pan and Tompkins [9] algorithm was used for QRS detection, since its low complexity makes it adequate for real-time applications. This algorithm involves the following operations:

- Bandpass filter: reduces the influence of muscle noise, 60Hz power line interference, baseline wander and T-wave interference;
- Signal Differentiation: provides QRS slope information;
- Squaring Function: non-linear amplification of the derivative output, amplifying the QRS components;
- Moving Window Integration;
- Thresholding.

With R-peak information, it is possible to have an idea if the signal is regular by analysing the variance of the heart-beat duration. In the current implementation, we compute the second derivative of a vector R, containing the R-peaks and calculate its variance. The condition $\text{VAR} (\partial R) \leq \text{TH}$ (where TH is a previously selected threshold) is used as a simple selector. If the condition is false, we switch to an irregular approach (eg. SPIHT algorithm or a predictive approach) and encode a fixed-size window otherwise.

2.2 Segment Normalization and Dictionary Search

Each segment, defined as the points contained in the interval $[R_i, R_{i+1}]$, is normalized so that different segments can be compared. A typical segment has a U-like shape, as can be observed in Figure 2. Let D denote the dictionary used by both algorithm strategies, which will hold at any given time n patterns $D = \{D_1, D_2, D_3, ..., D_n\}$, where a pattern is defined as a RR segment. Furthermore, let S be the segment to be encoded, which is defined as a vector containing k values $S = \{S(1), S(2), S(3), ..., S(k)\}$. In order to determine the most similar pattern from the dictionary, a vector V is calculated where the correlation between the segment S and each dictionary entry is computed, such that $V_i = \text{corr} (D_i, S)$.

Entry $D_i$ is selected as the most similar pattern where $\text{corr} (D_i, S) = \max(V)$.

There are two main parameters that need normalization: i) segment length and ii) segment amplitude. In our implementation, segment length was normalized using Matlab’s resample function, where a FIR filter is projected internally to remove any aliasing resulting from the scaling process. The endpoints are aligned with zero through line subtraction and then restored to the original amplitude after the resample operation, in order to avoid aliasing when resampling the segment. The dictionary stores normalized segments with 250 samples, which is typically smaller than the typical segment when the signal is sampled at 360Hz (as is the case of MIT-BIH signals) and results in the loss of certain high-frequency features.

Several schemes of amplitude normalization were experimented with. Storing $< \min(S), \max(S) >$ is wasteful and provides no significant information, since the amplitude of the segment depends strongly on the QRS amplitudes, which may vary. The amplitude of the signal between the S wave of the first QRS and the Q wave of the second will be, in most circumstances, largely unchanged and these will comprise more than 90% of the segment.

Segment baseline wander approximation through line fitting was also considered. To fit the line $y = mx + b$ to a
2.3 Segment approximation with Cubic Bezier curves

The decomposition of a segment \( S \) into a set of Bezier curves is performed in order to reach a very accurate approximation of the segment, while achieving a comparatively acceptable compression rate, somewhere between 5:1 and 15:1 in most cases. This will be the “high-quality” stage of the algorithm and will mainly be used to populate the dictionary. The reason for this will become apparent later on.

This stage of the algorithm works in the following way: let \( B \) be a list of Bezier control points \( P \) defined as \( P_0, P_1, P_2, P_3 \). Each \( P \) contains the coefficients of a cubic Bezier curve, which is defined by

\[
y(i) = (1-t)^3 P_0 + 3(1-t)^2 t P_1 + 3(1-t) t^2 P_2 + t^3 P_3,
\]

with \( t \in [0,1] \) and \( i = 1..j \) being the length of the curve. A segment \( S \) can be approximated by a set of Bezier curves by initially setting \( P_0 = S(1) \) and \( P_1 = S(k) \) and determining \( P_2 \) and \( P_3 \) in a way that the error between \( S \) and the Bezier curve is minimized in the least squares sense. A pseudo-inverse is used for this purpose.

If the determined curve does not approximate the signal well enough according to an error criterion a new control point is added at the point \( S(e) \) where the absolute error is a maximum, dividing the curve into two, the first spanning interval \([S(1), S(e)]\) and the second interval \([S(e), S(k)]\).

The process is repeated iteratively until a stop condition is reached. In this case, we used the PRD, defined by

\[
PRD = \frac{1}{N} \sum_{i=1}^{N} (x(i) - \hat{x}(i))^2 / \sum_{i=1}^{N} (x(i) - \bar{x})^2
\]

as an error criterion, stopping the decomposition process when \( PRD \leq 6 \).

Signals reconstructed with \( PRD \leq 10 \) are rated as having a good clinical quality in recent literature. However, Bezier decomposition stops when \( PRD \leq 6 \) to assure the dictionary incorporates very detailed patterns.

After the decomposition, the set of curves are organized into three vectors: an abscissa vector, an amplitude vector and a matrix with values \( P_i \) and \( P_j \), which approximate the derivative of each curve. The abscissa vector stored with differential encoding [8], the amplitude vector is encoded against the amplitude of the most similar segment in the dictionary and derivative approximations are encoded as differences \( P_i - P_j \) and \( P_j - P_i \). All residual values are then stored with a Golomb-Rice encoder. [8]

2.4 Progressive Error Encoding

In this stage, the strategy consists on directly exploring the similarity between a previously encountered R-R segment and \( S \). However, it must be stressed that this previously encountered segment no longer exists; only its approximation \( D_i \) is present on the dictionary, which was encoded with a non-null error \( e_s \).

It stands to reason that the more inaccurate this approximation (ie, the higher \( e_s \)), the higher the total error will be. Therefore, the total error signal \( e \) which we want to minimize can be defined as \( e = D_i - S + e_s \).

In order to keep the amplitude of \( e \) as low as possible, two conditions must be met:

- \( i \) must be selected so that \( D_i - S \) provide the lowest amplitude error possible. Indeed this is the case, since \( D_i \) is selected due to its high correlation with \( S \), thus being the best possible choice in the dictionary.
- A compromise must be found in order to keep error \( e_s \) as low as possible while minimizing the impact on the overall compression ratio. This is the main objective of the Bezier encoding as described in 2.3.

We initialize this stage by denormalizing \( D_i \) to the length and mean of \( S \) and measuring the PRD between \( D_i \) and \( S \). If \( PRD \leq 10 \) then we need only send denormalization information and a dictionary reference, thus attaining the highest possible compression ratio. The exact value depends on the length of \( S \), but it is possible to achieve compression ratios for a segment \( S \) exceeding 70:1 in certain circumstances. If \( PRD > 10 \) then the error \( e \) is encoded progressively at various detail levels. At the end of each detail level, the PRD of the reconstructed segment will be recalculated, and the encoding process will stop when the exit condition is reached (\( PRD \leq 10 \)).

This progressive encoding scheme shares certain similarities with the SPIHT algorithm, although differences exist. The SPIHT algorithm explores the hierarchical structure resulting from the Wavelet decomposition, and is able to mark an entire coefficient hierarchy as irrelevant with a single bit. Such a hierarchical structure does not exist in the error signal, so a different encoding scheme is used. Let \( Q \) be a scalar quantizer applied to the error signal using the operation \( q_j = \lfloor e / Q \rfloor \), where the brackets represent a rounding operation to the nearest integer. A sorted list of indices \( I \) is composed, containing the positions \( j \) where \( q_j \neq 0 \), and from this list a difference encoded vector, \( dl \), is built. Since \( dl \) contains only positive numbers, with no predictable statistical distribution (each heartbeat is a statistically independent event), a Golomb-Rice encoder is used to efficiently store the values in \( dl \). In order to accommodate two extra codes (end of detail level and end of codification codes), we set \( dl = dl + 1 \), so that the lowest value found in \( dl \) is 2. It is then possible to allocate values 1 and 0 to the codes mentioned above. A bit array containing the error signal (positive or negative) for each value where \( q_j \neq 0 \) is also stored.
After quantization of a detail level we perform the re-
construction of $D_j + q_j Q$. This operation has the side ef-
fect of introducing high-frequency components into the signal,
resulting in a noisy reconstruction, but this can be corrected by
applying a low-pass digital filter to the reconstructed
segment. In our implementation, a 3rd order low-pass Butter-
worth filter is used, with a cutoff frequency of 60Hz.

To decide if further detail levels are necessary, the PRD
between the filtered $D_j + q_j Q$ is calculated against the origi-
nal signal segment $S$. Until a stop condition is reached ( PRD $\leq 10$ ), the quantizer is updated by calculating
$Q=Q \div Q_{\text{appr}}$ and a new detail level encoded for the remaining
error.

The choice of $Q$ and $Q_{\text{appr}}$ is very important for the suc-
cess of this encoding strategy. In the SPIHT algorithm, coef-
icient bits are coded in a planar fashion. A level $m$ is selected to be $m = \lfloor \log_2 \max(C) \rfloor$, where $C$ represents the $n$
sub-bands resulting from Wavelet decomposition. The initial $Q$ is
defined as $Q = 2^n$, and since $m$ decreases in each iteration of
the SPIHT algorithm, one can conclude that $Q_{\text{appr}} = 2^m$.

These values of $Q$ and $Q_{\text{appr}}$ will not work very well in
our case, unfortunately, since there is no hierarchical struc-
ture to explore in the error signal and storing and index will
be more costly than one bit. However, we can explore the
property of Golomb-Rice encoding, where values closer to
zero need less bits to be encoded.

If the initial quantizer threshold is lower than $m$, we will
be able to capture more error values. What happens in prac-
tice is that it is possible to find some error values to be clus-
tered together, due to low-frequency variations of the error
signal, resulting in the generation of values closer to zero
when difference encoding is applied and $dI$ is constructed
from $I$. This results in an improvement both in compression
ratio when using the Golomb encoder and an improvement in
error approximation.

However, lower values of $Q$ can cause problems. Since
$\sum_{i=0}^{m} 2^i = 2^{m+1} - 1$, if $\max(\epsilon) > 2^{m+1}$ it will not be possible to
approximate the error signal adequately with $Q_{\text{appr}} = 2^m$. This
means that it would be desirable to have both $2^{m+1} < \max(\epsilon)$
and $Q_{\text{appr}} < 2^m$. The actual values for $Q$ and $Q_{\text{appr}}$ used in our
implementation were determined experimentally, using as evalua-
tion criteria the expected absolute error amplitude and
the number of points in $I$ packed close together at each detail
level. These criteria were used to evaluate in a simple way
the resulting compression ratio, for a similar degradation
level, on a number of ECG signals. From these tests, the
values $Q = 64$ and $Q_{\text{appr}} = 1.5$ seemed to provide good re-
results, and were therefore used in our tests.

2.5 Output Selection

Both strategies described in 2.3 and 2.4 are executed in paral-
lel and the output that provides the higher compression ratio
is transmitted to the receiver. If the Bezier approximation is
selected, it means that $D_j$, the best entry on the dictionary, is
too different from $S$, since too many error values need to be
encoded. In this case, the transmitted Bezier approximation is
added to the dictionary.

If the reconstruction resulting from the error encoding
stage is chosen, no entry is added to the dictionary, since the
patterns currently in the dictionary seem to represent the sig-
nal well enough. Furthermore, this reconstruction is less ac-
curate than the Bezier approximation (higher PRD threshold
as stop condition).

3. RESULTS AND DISCUSSION

The proposed methodology, being based on a dictionary ap-
proach, will naturally provide better results in cases where a
regular ECG signal is presented. The used dictionary ap-
proach is similar to LZ78, since it is able to hold previously
encountered patterns indefinitely.

However, unlike LZ78 where new dictionary entries are
referenced as soon as they are needed, we found that the two-
strategy algorithm will work better if a minimal dictionary is
built with the first few coded heartbeats. More specifically,
the first 32 R-R segments encountered are always Bezier
encoded and added to the dictionary. The following segments
are then encoded by executing both strategies and selecting
the output that provides the higher compression ratio.

For algorithm validation, datasets from the MIT-BIH da-
base were used. These datasets are sampled at 360Hz with
11-bit precision and contain regular and irregular signals.
Twenty datasets were selected from this database for testing.
The testing array includes datasets 100 through 105, 107, 111
through 119 and 121 through 124, making a total of 20 data-
sets. The first two minutes of each signal were segmented in
R-R intervals and compressed with the described algorithm.
The results are summarized in Figure 3, where Compression
Ratio (CR) and Percent Root Mean Square Difference (PRD)
are presented.

![Figure 3 – CR (circles) and PRD values (crosses) for the tested datasets. Mean CR = 23.91, mean PRD = 7.48](image-url)
In Figure 4 it is possible to visually inspect the quality of two reconstructed signals. It is important to stress that the resulting error is very small and no discontinuities exist between segments.

It is worth pointing out, however, that these results can be improved with the simple use of a 3rd order low-pass Butterworth filter, with a 60Hz cutoff frequency, applied to the entire signal prior to the compression process. This filter is used to remove noise components, and the general morphology of the signal isn’t affected in any way.

We now present Figure 5, which contains the compression ratios and PRD values for compression of the filtered signals. It can be argued that the results based on the use of this filter are less than fair, since the PRD in this second graph is measured between the reconstructed signal and the filtered signal, and there is also a non-zero PRD value between the original and filtered signals to begin with. However, since the filter only removes high-frequency noise and the clinical features are not distorted in any way, it does not seem an unreasonable approach.

The method copes with irregularities in the signal resulting from noisy conditions or cardiac pathologies. However, in these cases there will be a noticeable impact on compression ratio, since the signal will present substantial differences from the patterns stored in the dictionary and accurate representation of the pattern will be preferred over high compression.

4. CONCLUSION

In this work an algorithm for ECG compression based on the use of a segment dictionary and on Bezier curves approximation was presented. The dictionary contains low-error approximations of previously encountered R-R segments, which are representative of the signal morphology and are used as base patterns for the codification of new segments.

The main advantage of this approach is that higher compression ratios are not dependent on a higher signal quantization, which would result on a higher reconstruction error. Unlike other approaches, the compression ratio resulting from our approach is dependent on the exploited signal redundancy only, providing a reconstruction error between the thresholds defined for the two encoding strategies. The main limitation is that the encoder must run two strategies simultaneously, which comes at a computational cost.

Further improvements are possible, however. It is clear that there is still a significant overhead on the Bezier codification stage. There is still some redundancy on this approach, which should be minimized. The normalization stage should also be a focus of further research, since a more efficient normalization should reduce the number of patterns in the dictionary and therefore improve the results.

In any case, the results so far are very optimistic and we expect to be able to improve the compression ratio while maintaining a quality of $PRD \leq 10$.

REFERENCES