CHAPTER 4

OPTIMIZED TIME-DOMAIN ECG DATA COMPRESSION METHOD

4.1 Introduction

Electrocardiogram (ECG) signals are recorded from patients for both monitoring and diagnostic purposes. If the signals are to be stored or transmitted in digital form, an efficient algorithm for data compression is appropriate. Many data compression techniques for ECG waveforms have been presented. Roughly, they can be classified into two categories:

1. Dedicated techniques: These are mainly time domain techniques dedicated to compression of ECG signal. They include the AZTEC, CORTES, Turning Point, and FAN algorithms. The more recent CCSP technique, based on a Cardinality Constrained Shortest Path algorithm and guaranteeing a minimum distortion for a given sample reduction ratio, also fits into this framework.

2. General techniques: These techniques, which were developed for the compression of speech and image/video compression, are having a sound mathematical foundation. They include Differential Pulse Code Modulation (DPCM), Sub-band and Transform Coding, and Vector Quantization (VQ).

This chapter presents new results relating the performance of optimum time domain coding using linear interpolation. Perhaps equally important is that, a fair and authoritative comparison of the coding capabilities of dedicated time domain ECG coders has been presented. The comparison is not straightforward, since the dedicated methods typically refer to the sample reduction ratio, i.e., the total number of samples in the original signal divided by the number of samples in the compressed signal.
number of original signal samples, divided by the number of signal samples kept.
In contrast, general compression techniques refer to the average number of bits
used for representation of each reconstructed signal sample.

4.2 Dedicated time domain methods

Coding by time domain methods is based on the idea of extracting a subset
of significant signal samples to represent the signal. The key to a successful
algorithm is a good rule for determining the most significant samples. Decoding is
based on interpolation in this set and distinguishing between traditional heuristic
methods, of which several variants have been available for some time, and recently
developed optimization approaches. Heuristic time domain algorithms are usually
fast generators of compressed code satisfying certain reproduction requirements.
Typically, the absolute error of the decoded signal is guaranteed to be below a
prescribed bound. A frequently used time-domain technique is the FAN algorithm.

The basic idea of this algorithm is to identify signal segments where a
straight line serves as a close approximation, and to discard all but the terminal
points along this line. When significant deviations from linearity are detected, the
corresponding samples are included in the extracted signal samples. The FAN
algorithm accomplishes the above idea by initially accepting the very first sample
point. Next it computes a range within which succeeding samples must be found if
they are to be fit by a straight line. This range depends on the absolute error bound
(input to FAN), but becomes more and more narrow as more samples are
processed. Whenever a sample falls outside the range, its predecessor is accepted
as a significant sample, and the procedure above is repeated from this point on.
This method offers control of the absolute error, while the number of extracted
signal samples is beyond management. Numerous variations in how to make the eliminate-or-keep decision on signal samples, have been suggested in diverse time-domain coders. These include the original AZTEC.

4.2.1 Optimization methods

Despite the incorporation of intelligent sample selection rules, all heuristics suffer from lack of ability to extract signal samples in a manner that guarantees the smallest reconstruction error possible. However, by a rigorous mathematical model of the compression problem, and by a corresponding solution algorithm, the minimum set of samples may be achieved[3],[29],[31]. In the following section, given the number of retained samples a method guaranteeing the smallest possible distortion among all techniques using linear interpolation has been developed. The Euclidean norm has been chosen as error measure[28],[29].

If it is assumed that $x(1), x(2), \ldots, x(N)$ are the samples taken from an ECG signal at constant interval; $M$, denoting the bound on the number of extracted samples and $S$ denote the sample set $S = \{x(1), x(2), \ldots, x(N)\}$; appropriate compression set $C = \{n_1, n_2, \ldots, n_M\} \subseteq \{1, 2, \ldots, N\}$ are the corresponding sample values; by assuming $n_1 = 1$ and $n_M = N$, the approximation is then given by

$$
\hat{x}(n) = x(n) \text{ if } n \in C \quad \text{and} \quad \hat{x}_n = x(n_m) + \frac{x(n_{m+1}) - x(n_m)}{n_{m+1} - n_m} (n - n_m) \text{ when } n_m < n < n_{m+1}.
$$

This results in a continuous piecewise linear function which interpolates $\{(n, x(n)) \mid n \in C\}$. Suppose, the directed graph $G = (V, A)$ whose vertex set $V = \{1, 2, \ldots, N\}$ and arc set $A$ contains node pairs $(i, j)$ where $i, j \in V$ and $i < j$. If $n_1, n_M \in V$, the set $(n_1, n_2, \ldots, n_M)$ is said to be a path from $n_1$ to $n_M$.  

UNIVERSITY LIBRARY
JNT UNIVERSITY, VITAMULLY,
HYDERABAD 500 072.
Call No. 62138 GIR/0
P. No. 31
in $G$ if $n_1, ..., n_M \in V$ are distinct vertices and $n_1 < n_2 < ... < n_M$. Suppose, $P_n$ denote the path from node 1 up to node $n$. Then length of each arc $(i, j)$ in $A$ is given as the contribution to the total reconstruction error by eliminating all nodes between $i$ and $j$. This can be expressed as $c_{ij}^2 = \sum_{n_{i+1}}^{j-1} (\hat{x}(n) - x(n))^2$. The length of $P_n$ will thus be the sum of the length of all arcs included in the path up to node $n$. Each arc $(i, j)$ in $A$ represents the possibility of letting $i$ and $j$ be consecutive members of $C$. The arc length for a linear interpolation case is illustrated in Figure 4.1.

![Figure 4.1 Arc length in the graph.](image)

Here $\epsilon(n) - \hat{x}(n) - x(n)$ and the length of the arc connecting nodes $i$ and $j$ are thus given by $c_{ij}^2 = \sum_{n_{i+1}}^{j-1} \epsilon(n^2)$. The path length of a given path $P$ is given by $\|P\| = \sum_{(i,j) \in P} c_{ij}^2$. This has the following problem: Minimize the length of $P_n$ under the constraint that $P_n$ contains no more than $M$ vertices. The problem is an instance of the resource-constrained shortest path problem[9][14]. The resource in
question is the number of vertices on the path[51]. Unlike the general version of the problem, this model contains only one resource constraint. Omitting the resource constraint, results in the frequently studied shortest path problem. Because of the particular choice of resource in this case, the problem is termed as the cardinality constrained shortest path problem (CCSP). In the next section, a recursive formulation of CCSP is derived, and based on this an algorithm providing the exact solution will be given.

4.2.2 Optimization algorithm

In order to establish an efficient solution scheme, the following precise problem formulation is proposed defining $P_{j,m}$ as the shortest path to $j$ visiting exactly $m$ vertices, and let $f(j, m)$ denote the length of $P_{j,m}$. A path is actually searched for a $1 < m^* \leq M$ and the corresponding $P_{N,m}$ for which $f(N, m^*) = \min_{1 \leq m \leq M} f(N, m)$. In this search, all such paths $P_{N,m}$ are computed in the order given by increasing values of $m$. The CCSP problem is hence solved when these quantities become available. Considering the path $P_{j,m+1}$, the second last vertex in $P_{j,m+1}$ by $i$ is denoted. Obviously, $i < j$ and $i \leq m$. Hence $P_{j,m+1}$ contains a sub path through $m$ vertices to $i$. But this sub path has to be $P_{i,m}$, otherwise a shorter path could found through $m$ vertices to $i$. Augmenting this with vertex $j$ yields a path shorter than $P_{j,m+1}$ (through $m + 1$ vertices) ending at $j$ contradicting the fact that $P_{j,m+1}$ is the shortest of all such paths. Hence it is shown that $f(j,m+1) = f(i, m) + c_j$ for some $i = m, m+1, ..., j-1$. Furthermore, it is clear that $i$ must be the vertex minimizing the right hand side in this equation. Supplying
the obvious condition that \( f(j, 2) = c_{ij} \), the following recursive equations are obtained:

\[
f(j, 2) = c_{ij}
\] ................................. (4.1)

\[
f(j, m + 1) = \min \{ f(i, m) + c_y \mid i = m, \ldots, j - 1 \} \quad \ldots \ldots \quad (4.2)
\]

when \( j = 2, \ldots, n \) and \( m = 2, \ldots, M - 1 \) are inserted in (4.1)-(4.2), \( f(N, M) \) is uniquely determined. Equations (4.1)-(4.2) constitute a dynamic programming formulation of CCSP. Similar formulations have been proposed for various constrained shortest path problems. The formulation above resembles the one given in [14], but in (4.2) it is exploited the fact that \((i, j) \in A\) only if \( j > i \). Furthermore, it is disregarded \( f(i, m) \) for all \( i < m \) since all paths terminating at \( i \) have at most \( i \) vertices.

From the above formulation, the algorithm in Figure 4.2 suggests itself that \( p(j, m) \) signifies the predecessor of \( j \) in \((P_{j,m})\). The compression set can thus be recorded by letting \( n_m = N \) and \( n_{m-1} = p(n_m, m) \) \((m = m^*, m^* - 1, \ldots, 2)\). This produces the interpolation points \((n_1, x(n_1)), (n_2, x(n_2)), \ldots, (n_m, x(n_m))\). It is easily seen that when all arc lengths are available, the computations above involve \( O(MN^2) \) arithmetic operations. Computation of all \( c_y \) values can be shown to require no more than a total of \( O(N^2) \) operations.

Algorithm for CCSP

for \( j = 2, \ldots, N \)

begin

\[
f(j, 2) = c_y \quad // \text{Length of two-vertex path}
\]

\[
p(j, 2) = 1 \quad // \text{from 1 to } j
\]

end
\(m^* = 2\) // Assume the two-vertex path to \(N\) is optimal

for \(m = 2, \ldots, M - 1\) // Find \(m + 1\)-vertex paths
begin
for \(j = m + 1, \ldots, N\) // Find the path to \(j\)
begin
\(p(j, m + 1) = m\) // Assume \(P_{j,m+1} = \{1,2,\ldots,m,j\}\)
\(f(j, m + 1) = f(m, m) + c_{m,j}\) // The length of this path
for \(i = m + 1, m + 2, \ldots, j - 1\) // \(P_{j,m+1}\) may equal \(P_{i,m} \cup \{j\}\)
begin
if \(f(i, m) + c_{i} < f(j, m + 1)\) // Shorter!
begin
\(f(j, m + 1) = f(i, m) + c_{i}\) // Update the shortest length
\(p(j, m + 1) = I\) // Record the last step
end
end
if \(f(N, m + 1) < f(N, m^*)\) // Shortest path to \(N\) so far
begin
\(m^* = m + 1\) // Optimal number of vertices in path to \(N\)
end
end
end
end

Figure 4.2 Algorithm for the CCSP problem

4.2.3 Coding scheme

The performance of time domain compression methods are often evaluated as a chosen distortion measure as a function of sample reduction ratio, defined as the number of samples in the original signal per retained sample. When it comes to other techniques, such as sub-band and transform coding, the performance is often
evaluated as a chosen distortion measure as a function of \textit{bit rate}, i.e. the average number of bits necessary to represent one sample of the signal.

In order to be able to compare the results from time domain methods to other methods in a fully justified way, encoding of the extracted samples will have to take place. Reconstruction of a signal encoded by a linear interpolation time domain method requires one sample amplitude and the distance from the previous sample, referred to as \textit{run}, for each segment of the signal to be reconstructed. The signal samples extracted by the time domain coder are denoted by $x(n_k)$, $k = 1, \ldots, M$, and denote the run associated with each extracted signal sample as $r_k = n_{k+1} - n_k - 1$. It results in pairs of $(r_k, x(n_k))$, $k = 1, \ldots, M$ to be encoded. The symbols to be encoded are termed \textit{source symbols} in this context and have at least two possibilities when choosing between coding strategies:

1. Coding of amplitudes and runs by two separate coders.

2. Coding of the concatenated symbols $(r_k, x(n_k))$ by one single coder.

Alternative 2 implies a high number of possible source symbols. The original bit representation for the test signals is 12 bits for the amplitudes, and if it is assumed that no run is longer than 256, then it results in $2^{12} \times 2^8 = 2^{20}$ possible different source symbols. This will lead to a huge table of source words which is impractical to cope with. For this reason alternative 1 is chosen and encoded the amplitudes and runs by two separate coders. However, that alternative 2 is viable in sub-band coders[1],[2]. The structure of the encoding system is illustrated in Figure 4.3. An entropy coder is applied in compression of the extracted signal samples. A record adaptive coder, is selected and accounted for the overhead
necessary due to side information. The results from the complete coding system are
presented and discussed.

4.3 Coding experiments and discussion

For evaluation of the reconstructed signals a commonly used performance
measure is the percentage root mean square difference (PRD)[52]:

\[
PRD = \frac{\sum_{n=1}^{N} (x(n) - \bar{x}(n))^2}{\sum_{n=1}^{N} (x(n) - \bar{x})^2} \times 100 \% \quad (4.3)
\]

where \(x(n)\) and \(\hat{x}(n)\) denote the original and reconstructed signals, \(\bar{x}\) is the mean
value of \(x(n)\), and \(N\) is the original signal length. Although useful for testing the
relative performance of coding techniques within a narrow family, the PRD hardly
qualifies as an authoritative yardstick for inter-method comparisons. Each
compression method has its own distortion characteristic, and “objective”
measures like the PRD, or the signal to noise ratio, should be supplemented by
visual inspection of the reconstructed waveforms.
4.3.1 Coding evaluation

All records used in this article are taken from the MIT/BIH Arrhythmia CD-ROM database[45],[46] second edition. The sampling frequency is 360 Hz with 12 bits per sample. In order to test the robustness of the coding systems presented earlier, a varied set of test signals is used. The first 15 seconds of each test signal are plotted in Figure 4.4. Note that only the first record, mit100_1000, is a normal heartbeat signal. The other records contain various abnormal rhythms.

Figure 4.4: Original ECG's (the first 15 seconds) used in the experiments. "mit xxx yyyy" denotes record no. xxx starting at time yyyy. Each record length is in total 10 minutes, corresponding to 216000 samples.
4.3.2 Comparison based on the PRD measure

The test signals shown in figure 4.4 are coded using the two time-domain algorithms CCSP and FAN. For the time domain coders the bit representation of the retained signal samples is described. Figure 4.5 represents obtained PRDs for the two coders and test signals for bit rates between 0.2 and 1.4 bits per sample (bps). The optimality of the CCSP algorithm is clearly demonstrated when compared to the heuristic FAN algorithm. For all test signals the CCSP outperforms the FAN algorithm with a large margin over the range of bit rates shown. Especially at low bit rates (around 0.6 bps) the difference varies between 3 and 12% PRD. At higher rates (around 1.0 bps) the difference is smaller, but still significant. The CCSP and FAN algorithms perform best on signals without too much oscillations. To illustrate this, the results for mit203_0100 and mit203_1100 with the results for the other test signals are compared.

Figure 4.5: Coding performance for varying input signals. The time domain algorithm are: CCSP (solid line with x-marks), and FAN (solid line with diamonds).
This is reasonable as oscillating signals demand retaining more samples in order to achieve the same PRD as a slowly varying signal.

4.3.3 Comparison based on visual inspection

As mentioned in the beginning of this section, inter-method comparisons should also include visual inspection of the reconstructed signals. The scope of this

![Original

![CCSP

![FAN

Figure 4.6(a) : Reconstructed signal segment (taken from mit100_1000) at 1.0 bit per sample.
investigation is to show coding artifacts as they appear using the presented compression methods for a "typical" ECG signal. A short segment taken from mit100_1000 representing a regular heartbeat has been chosen. The reconstructed signal segment is shown in Figures 4.6(a) and 4.6(b) at two different bit rates: 1.0 bps and 0.5 bps, when coding the whole 10 minute record.

![Figure 4.6(a): Reconstructed signal segment (taken from mit100_1000) at 1.0 bit per sample.](image)

![Figure 4.6(b): Reconstructed signal segment (taken from mit100_1000) at 0.5 bit per sample.](image)
The original signal segment is also included. At 1.0 bps all coders smooth out some of the details in the original signal. This is particularly evident for the time-domain coders[3].

4.4 Summary and conclusions

In this chapter it has been presented an overview of the two important time-domain methods for compression of ECG signal. In both the methods the compressed signal is represented by retained signal samples. Coding experiments demonstrated that time-domain method based on linear interpolation in CCCP is better than the FAN algorithm. This optimized time-domain algorithm performed dramatically better than the well known FAN algorithm. This is shown in terms of both PRD and visual inspection.