ECG Compression using Multiscale Recurrent Patterns with Period Normalization

Eddie B. L. Filho*†, Eduardo A. B. da Silva†, Waldir S. S. Júnior‡, Murilo B. de Carvalho§
*Genius Institute of Technology, Manaus, Amazonas, Brazil, 69075-904
†Universidade Federal do Rio de Janeiro, Rio de Janeiro, Rio de Janeiro, Brazil, 21945-970
‡FUCAPI, Manaus, Amazonas, Brazil, 69075-351
§Universidade Federal Fluminense, Rio de Janeiro, Rio de Janeiro, Brazil, 24210-240
Emails: efilho@genius.org.br, eduardo@lps.ufrj.br, waldirjr@fucapi.br, murilo@telecom.uff.br

***Abstract*— Recently, the Multidimensional Multiscale Parser (MMP), an algorithm based on multiscale recurrent patterns, has been used to successfully compress data from ECG signals. Their quasi-periodic nature makes them natural candidates for the use of recurrent patterns. However, as many diagnostic relevant signals are far from periodic, the characteristics of MMP are not fully exploited. We have dealt with this problem by interpolating the ECG signal so that the interval between successive heart beats becomes constant. Following that, the algorithm subtracts the average period from the interpolated signal. Simulation results show that these modifications increase the rate-distortion performance.**

I. Introduction

The ECG is a very important tool for the diagnosis of arrhythmias, like angina or ischemia. During an exam, the volume of data produced can be very large, which claims for an efficient method for ECG storage or transmission.

By using an efficient compression method, even a complete ECG holter exam, for instance, could be transmitted through telephone lines, allowing the doctor to carry through a diagnosis without the presence of the patient.

In this work, we propose an extension to a recently developed technique to compress ECG signals [1], based on the Multidimensional Multiscale Parser (MMP) algorithm [2], [8], which is a universal lossy compression method built upon the *multiscale recurrent pattern matching* concept. In it, two vectors $\mathbf{u}$ and $\mathbf{v}$ with different lengths $(\ell(u) \neq \ell(v))$ can be matched. This is possible through the use of a *scale transformation* $T^N(\mathbf{x}) : \mathbb{R}^{\ell(\mathbf{x})} \mapsto \mathbb{R}^N$, which is implemented with classical sampling-rate change operations [2].

In order to make the comparisons to other algorithms easier, we use ECG signals taken from the MIT/BIH arrhythmia database. This database contains parts of ECG exams of 48 subjects, with two derivations each. One of these signals is illustrated in Fig. 1. The results are assessed using the PRD (*Percent Root-mean-square Difference*) distortion metric and the CR (*Compression Ratio*) rate, defined as:

$$PRD = 100\sqrt{\frac{\sum_{n=0}^{N-1}(x(n)-\hat{x}(n))^2}{\sum_{n=0}^{N-1}(x(n)-\mu)^2}} \quad (1)$$

$$CR = \frac{B_o}{B_c} \quad (2)$$

where μ is the baseline value of the analog-to-digital conversion used for the acquisition of the data $x(n)$ (in the MIT/BIH arrhythmia database $\mu = 1024$), B_o is the total number of bits of the original image and B_c is the total number of bits spent in the compressed representation of $x(n)$.

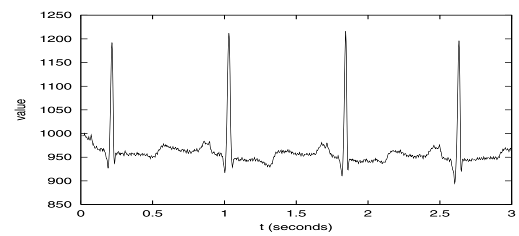

Fig. 1. A typical ECG signal.

II. The MMP algorithm applied to ECG signals

The MMP algorithm is a universal lossy compression method based on multiscale recurrent pattern matching, which is an extension of the ordinary recurrent pattern matching [2]. In the former, vectors with different lengths can be matched.

To compress ECG signals, some adaptations were performed in the base algorithm [1]. All of them aim to take advantage of the characteristics of ECG signals, such as the quasi-periodic behavior and the structure based on well defined sub-waves (P, QRS and T).

A. The base MMP algorithm

The MMP has a dictionary $\mathcal{D} = \{\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_{L-1}\}$ with L vectors $\mathbf{v}_i$ of fixed lengths, which are used to encode segments of the input signal $\mathbf{X}^0 = \begin{pmatrix} x(0) & x(1) & \ldots & x(N-1) \end{pmatrix}$, whose length N is a power of 2. When attempting to encode the input segment $\mathbf{X}^0$, the MMP searches in the dictionary $\mathcal{D}$ for the best vector $\mathbf{v}_{i_0}$ to replace $\mathbf{X}^0$. The choice of the best vector is based on the minimization of the squared error $\xi_0 = \|\mathbf{X}^0 - \mathbf{v}_{i_0}\|^2$ and, if its value is smaller than or equal to a predefined distortion threshold d^*, then the encoding of $\mathbf{X}^0$ is carried through and the MMP outputs a bit flag $'1'$, followed by the dictionary index i_0. If the distortion threshold is not reached ($\xi_0 > d^*$), the MMP splits the input segment in other two segments, $\mathbf{X}^1 = \begin{pmatrix} x(0) & x(1) & \ldots & x(N/2-1) \end{pmatrix}$ and $\mathbf{X}^2 = \begin{pmatrix} x(N/2) & x(N/2+1) & \ldots & x(N-1) \end{pmatrix}$, outputting a bit flag $'0'$ and repeating the encoding procedure for $\mathbf{X}^1$. If a match occurs, the algorithm outputs a bit flag $'1'$, followed by the dictionary index i_1, and then attempts to encode the second segment, $\mathbf{X}^2$. If the matching attempt fails, the MMP outputs a bit flag $'0'$ and splits $\mathbf{X}^1$ in other two segments, $\mathbf{X}^3 = \begin{pmatrix} x(0) & x(1) & \ldots & x(N/4-1) \end{pmatrix}$ and $\mathbf{X}^4 = \begin{pmatrix} x(N/4) & x(N/4+1) & \ldots & x(N/2-1) \end{pmatrix}$, before attempting to encode $\mathbf{X}^2$. The segmentation procedure is recursively repeated until a matching attempt is successful or the resulting segments have length $\ell(X^j) = 1$. Fig. 2 shows the segmentation procedure. In this example, the output generated by MMP would be the sequence $0, 0, 1, i_3, 0, 1, i_9, 1, i_{10}, 1, i_2$.

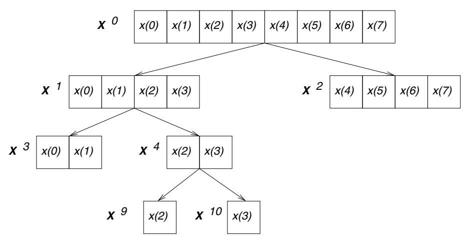

Fig. 2. Segmentation of an input vector $\mathbf{X}^0$.

The segmentation of the input segment $\mathbf{X}^0$ is represented by means of a segmentation tree $\mathcal{S}$, as shown in Fig. 3.

The matching of fixed-length dictionary elements to variable-length segments is possible trough a scale transformation $T_N^M : \mathbb{R}^N \mapsto \mathbb{R}^M$, which adjusts the lengths of the vectors before a matching attempt [2].

The MMP dictionary is updated as follows: whenever the reconstructed segments $\hat{\mathbf{X}}^{2j+1}$ and $\hat{\mathbf{X}}^{2j+2}$ associated to the children nodes n_{2j+1} and n_{2j+2} are available, the MMP concatenates them and generates $\hat{\mathbf{X}}^j$ as the reconstructed segment associated to the parent node n_j. In Fig. 2, when $\hat{\mathbf{X}}^9$ and $\hat{\mathbf{X}}^{10}$ are available, it is possible to concatenate them and generate the reconstructed element $\hat{\mathbf{X}}^4$. This segment may then be included in the dictionary and reused in the future.

The segmentation tree $\mathcal{S}$ may be optimized in a rate-distortion sense, allowing a distribution of the bits available for encoding that takes into account the global needs of the input segment. The basic optimization procedure begins with a full segmentation tree and continues from the leaf nodes to the root, pruning the children nodes n_{2j+1} and n_{2j+2} whenever the Lagrangian cost [2] of the tree containing them is greater than the cost of the segmentation tree without n_{2j+1} and n_{2j+2}. The Lagrangian cost of the segmentation tree $\mathcal{S}$ is defined as $J(\mathcal{S}) = D(\mathcal{S}) + \lambda R(\mathcal{S})$, in which $D(\mathcal{S})$ is the distortion obtained when using $\mathcal{S}$ and $R(\mathcal{S})$ is the rate.

The computational complexity of the MMP algorithm is higher than the one normally presented by the standard *vector quantization* technique (VQ) and is suitably assessed in [1].

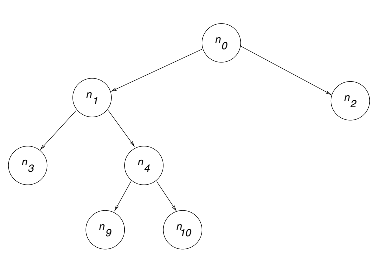

Fig. 3. Segmentation tree $\mathcal{S}$ of the input vector $\mathbf{X}^0$ of Fig. 2.

B. The prune-join optimization strategy

In [6], it is presented an algorithm to extend the segmentation options of a binary tree, called the *prune-join* algorithm. In this method, an input vector is first segmented using a binary tree, according to some rate-distortion criterion [5], and each resulting segment is matched to a polynomial function. This first step is the well known *prune step*. The second one, the *join step*, is performed through the test of all neighbor nodes which do not have the same parent node, to verify if their joint encoding reduce the global Lagrangian cost. Then, the coefficients of the polynomials associated to the remaining nodes are quantized and encoded. In [6], it was proved that the described algorithm outperforms the one using the optimization based only on the *prune step*, providing a nearly optimum rate-distortion performance.

To incorporate the *prune-join* concept to MMP, the original segmentation is applied to an input vector $\mathbf{X}^0$, obtaining a segmentation tree optimized in a rate-distortion sense $\mathcal{S}$. The optimization procedure described in section II-A is used to perform the *prune step* of the algorithm. After that, an analysis is carried out to verify if any two neighbor nodes not sharing the same parent node can be joined together to lower the Lagrangian cost. Fig. 4(a) illustrates an example of a segmentation tree after the pruning step. Fig. 4(b) illustrates a possible join operation.

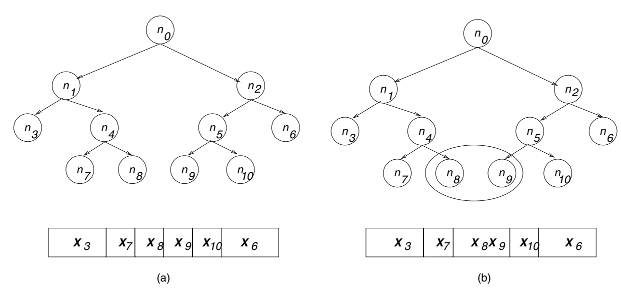

Fig. 4. The *join step*.

C. The inter-block continuity criterion

The representation $\hat{\mathbf{X}}^0$ which MMP generates can present severe discontinuities at the boundaries of the segments $\hat{\mathbf{X}}^j$ generated by the segmentation procedure, even if the original vector $\mathbf{X}^0$ is smooth, as depicted in Fig. 5.

This happens because the calculated distortion, used to evaluate the cost of the node n_j, is independent of the representations chosen for the other neighbor segments. In [7], it was proposed an efficient method for controlling the smoothness of representations produced by vector quantization schemes, called *side-match vector quantization* (SM-VQ). The SM-VQ can be incorporated to MMP to improve its performance for smooth input signals [8]. The concept is based in choosing a subset of the dictionary $\mathcal{D}$, called *state dictionary* $\mathcal{D}_S$, composed of the N_S best vectors $\mathbf{v}_k \in \mathcal{D}$ given by the adopted continuity criterion. This can be best understood with the aid of the definitions below.

The position of the first sample of $\mathbf{X}^j$ inside $\mathbf{X}^0$ is given by:

$$Fp(j) = N\left((j+1)\,2^{-\lfloor \log_2(j+1) \rfloor} - 1\right) \quad (3)$$

in which N is the length of $\mathbf{X}^0$.

The length N^j of $\mathbf{X}^j$ can be evaluated as:

$$N^j = N 2^{-\lfloor \log_2(j+1) \rfloor} \quad (4)$$

The *left neighbor* of $\mathbf{X}^j$ is given by:

$$\mathbf{L}^j = \begin{pmatrix} \hat{x}(Fp(j) - N^j) & \dots & \hat{x}(Fp(j) - 1) \end{pmatrix}, \; Fp(j) \geq 1 \quad (5)$$

To quantify the smoothness of the transitions or edges across the boundaries of the segments, we used the *rugosity* metric [1], defined as:

$$R(\mathbf{v}_k^s, j) = \Bigg| \left| L^j(N^j - 3) - L^j(N^j - 1) + v_k^s(0) - v_k^s(2) \right| - \left\lfloor \frac{4}{3} \left| L^j(N^j - 2) - v_k^s(1) \right| \right\rfloor \Bigg| \quad (6)$$

Following this concept, the MMP scales all the vectors of $\mathcal{D}$ to the same length and builds the state dictionary $\mathcal{D}_S$, which contains the N_S "least rugose" vectors according to (6).

D. The displacement dictionary

As can be seen in Fig. 1, a typical ECG signal is nearly periodic. For a strictly periodic signal, if the length of the MMP segments is equal to its period, the MMP algorithm quickly learns the period pattern and spends very few bits to encode the rest of the signal. However, given that the length of the segments may not be multiples of the period, in order to achieve efficient encoding of a periodic signal it is also necessary to learn the various displaced versions of one period. Therefore, to improve the MMP performance for nearly periodic signals, we can use a *displacement dictionary* $\mathcal{D}^D$, which contains displaced versions of the approximations for previously encoded segments. This dictionary is implemented keeping the M last samples of the reconstructed signal in a vector $\mathbf{V}_D^j$.

III. The Period Normalization

It is known that many diagnostic relevant ECG signals present significant variations in their behavior along an exam. That means the period of the heart beating and the format

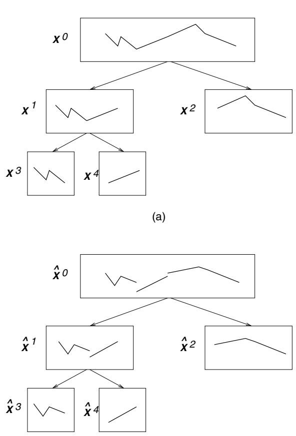

Fig. 5. Discontinuities in MMP: (a) Original signal; (b) Reconstructed signal.

of the sub-waves change along the signal and the MMP algorithm has to learn different patterns for each period. This characteristic does not allow MMP to use its dictionary in an efficient way, because the patterns learned up to a certain point in the signal are not frequently used in the subsequent periods.

To properly exploit the interbeat dependencies, we incorporate a period normalization step to MMP, as described in [3]. The procedure begins with the segmentation of the ECG signal, in which each period of the heart beating is identified and separated, as illustrated in Fig. 6. The algorithms used for the period detection were the ones available in [10]. Since each ECG period can have a different duration, we normalize them to the same length, using the method described in [3]. Following this concept, an original ECG segment $X = [x(0)\ x(1)\ \dots\ x(N_o - 1)]$ can be converted in a normalized segment $X_n = [x_n(0)\ x_n(1)\ \dots\ x_n(N_n - 1)]$, which is computed using:

$$\begin{aligned} X_n(m) &= \hat{X}(h^*) \\ h^* &= \frac{m.(N_o - 1)}{(N_n - 1)} \end{aligned} \quad (7)$$

in which $\hat{X}(h^*)$ is the interpolated version of $X(n)$, N_o is the original period length, N_n is the normalized period length and $m = 0, 1, \dots N_n - 1$.

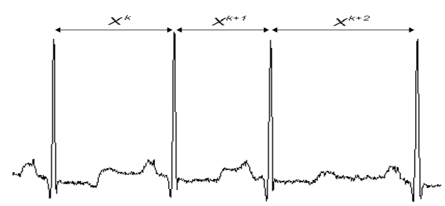

Fig. 6. Period detection.

The interpolation is computed using the cubic-spline interpolation methods described in [9]. When reconstructing the original signal, the decoder needs the original periods as side information, which are differentially and arithmetically encoded and added to the header of the compressed file. The original periods can then be recovered using the same transformation described in (7).

N_n is chosen as the integer part of the average of all periods detected. We have done so in order to keep the normalized signal with a length less than or equal to the original one.

It is worth noticing that the interpolation already introduces some distortion and its choice is decisive for a good performance. Generally, cubic splines are preferred because of their ability to fit C2-continuous cubic kernels [9]. A possible improvement to the interpolation method could be to use a point distribution based on the signal characteristics instead of the fixed linear distribution presented in (7).

The performance of the MMP algorithm is sensitive to the dynamic range of the signal being encoded, in the sense that a dynamic range reduction can lead to an improvement in performance. Since we already have all segments with the same length, one can easily reduce their dynamic range by first computing the *average normalized period segment* as:

$$X_{n_{avg}}(m) = \frac{1}{M} \sum_{k=0}^{M-1} X_n^k(m) \quad (8)$$

in which M is the number of detected periods, X_n^k is the kth normalized period and $m = 0, 1, \ldots N_n - 1$.

Each *normalized reduced-dynamic-range segment* is computed as the *normalized period segment* subtracted by the *average normalized period segment*:

$$X_{n_{rdr}}^k(m) = X_n^k(m) - X_{n_{avg}}(m) \quad (9)$$

where $m = 0, 1, \ldots N_n - 1$.

The *average normalized period segment* is then also differentially and arithmetically encoded and added to the header of the compressed file. The signal processed by MMP is then composed by normalized and reduced-dynamic-range segments of the original ECG record.

All the explained steps are illustrated for the channel 0 of record 100 of the MIT/BIH arrhythmia data base in Fig. 7.

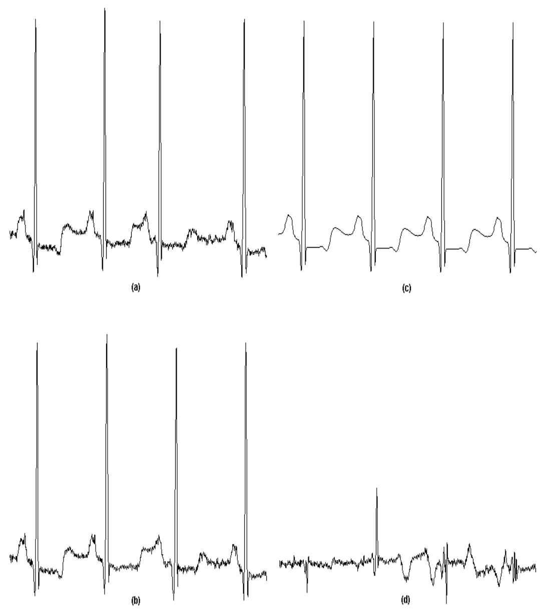

Fig. 7. Steps of processing: (a) Original signal; (b) Normalized signal; (c) Average normalized period segments; (d) Normalized reduced-dynamic-range signal.

IV. Experimental results

We have implemented MMP in software and applied it to compress ECG data from the MIT/BIH arrhythmia database. In order to make comparisons to the works in [1], [3] and [4] easier, we used the first 10 minutes of both channels of the following 11 records: 100, 101, 102, 103, 107, 109, 111, 115, 117, 118 and 119. The results are summarized in Table I; they were obtained by averaging the actual PRD values obtained for both channels of the 11 records at each CR. From this table, one can see that the proposed modifications led to an improvement in MMP performance at all data rates. In addition, the proposed method also outperforms SPIHT [4], and is competitive with the JPEG2000-based method in [3], one of the best performing methods reported in the literature.

TABLE I
Performance comparison of MMP, JPEG2000 and SPIHT, first 10 minutes (PRD).

CR	$JPEG$2000 [3]	MMP	MMP [1]	$SPIHT$ [4]
4:1	0.78	1.02	1.05	1.19
8:1	1.52	1.83	1.96	2.46
10:1	1.86	2.14	2.34	2.96
16:1	2.74	2.95	3.29	4.85
20:1	3.26	3.41	3.86	6.49

V. Conclusion

In this work, we present new developments to a recently proposed lossy ECG compression algorithm based on MMP, which is built upon the approximate multiscale recurrent pattern matching concept, an extension of the ordinary recurrent pattern matching. The MMP algorithm uses a dictionary of patterns, which is adaptively built while the data is being encoded, and a simple segmentation procedure, which can be trivially extended to operate on multidimensional data. We have incorporated a period normalization with dynamic range reduction step to the base algorithm, obtaining relevant improvements in performance. Originally used as an image data encoder, the MMP performed very well to encode ECG data, having performance as good as the best encoders known. It also preserves most of the diagnostically useful information, even at low rates [1]. Therefore, we have that MMP is an effective method for ECG compression and presents new directions for the development of ECG encoders.

References

[1] E. B. L. Filho, E. A. B. da Silva, M. B. Carvalho, W. S. S. Júnior and J. Koiller, "Electrocardiographic Signal Compression using Multiscale Recurrent Patterns", *IEEE Transactions on Circuits and Systems–I: Regular Papers*, Vol. 52, No. 12, pp. 2739-2753, December 2005.

[2] M. B. de Carvalho, E. A. B. da Silva and W. A. Finamore, "Multidimensional signal compression using multiscale recurrent patterns", *Signal Processing: Image and Video Coding beyond Standards*, No. 82, pp. 1559-1580, Elsevier Science B. V., 2002.

[3] A. Bilgin, M. W. Marcellin and M. I. Altbach, "Compression of Electrocardiogram Signals using JPEG2000", *IEEE Transactions on Consumer Electronics*, Vol. 49, No. 4, pp. 833-840, November 2003.

[4] Z. Lu, D. Y. Kim and W. A. Pearlman, "Wavelet compression of ECG signals by the set partitioning in hierarchical trees algorithm", *IEEE Transactions on Biomedical Engineering*, Vol. 47, No. 7, pp. 849-856, July 2000.

[5] G. J. Sullivan and R. L. Baker, "Efficient quadtree coding of images and video", *IEEE Transactions on Image Processing*, Vol. 3, No. 3, pp. 327-331, May 1994.

[6] R. Shukla, P. L. Dragotti, M. N. Do and M. Vetterli, "Rate-Distortion Optimized Tree-Structured Compression Algorithms for Piecewise Polynomial Images", *IEEE Transactions on Image Processing*, Vol. 14, No. 3, pp. 343-359, March 2005.

[7] T. Kim, "Side Match and Overlap Match Vector Quantizers for Images", *IEEE Transactions on Image Processing*, Vol. 1, No. 2, pp. 170-185, February 1992.

[8] E. B. L. Filho, M. B. Carvalho and E. A. B. da Silva, "Multidimensional signal compression using multi-scale recurrent patterns with smooth side-match criterion", *IEEE International Conference on Image Processing*, Singapore, October 2004.

[9] T. M. Lehman, C. Gonner and K. Spitzer, "Survey: Interpolation Methods in Medical Image Processing", *IEEE Transactions on Medical Imaging*, Vol. 18, No. 11, pp. 1049-1075, November 1999.

[10] M. Aboy, C. Crespo, J. McNames, J. Bassale, L. Jenkins and B. Goldsteins, "A Biomedical Signal Processing Toolbox", in *Procedures of Biosignal 2002*, June 2002