14th European Signal Processing Conference (EUSIPCO 2006), Florence, Italy, September 4-8, 2006, copyright by EURASIP

EMBEDDED WAVELET PACKETS–BASED ALGORITHM FOR ECG COMPRESSION

Manuel Blanco–Velasco †, *Fernando Cruz–Roldán* †, *Juan I. Godino–Llorente* ‡ *and Kenneth E. Barner* §

†Dep. Teoría de la Señal y Comunicaciones, Escuela Politécnica, Universidad de Alcalá
Alcalá de Henares (Madrid), Spain
phone: +34 91 885 67 08, fax: +34 91 885 66 99, email: manuel.blanco@uah.es
web: `http://msc.tsc.uah.es/`

‡ Dep. Ingeniería de Circuitos y Sistemas, Universidad Politécnica de Madrid
Madrid, Spain

§Dep. of Electrical and Computer Engineering, University of Delaware
Newark, DE 19716 USA

ABSTRACT

The conventional Embedded Zerotree Wavelet (EZW) algorithm takes advantage of the hierarchical relationship among subband coefficients of the pyramidal wavelet decomposition. Nevertheless, it performs worse when used with Wavelet Packets as the hierarchy becomes more complex. In order to address this problem we propose a new technique that considers no relationship among coefficients, and is therefore suitable for use with Wavelet Packets. So in this work, an embedded ECG compression scheme is presented using Wavelet Packets that shows better ECG compression performance than the conventional EZW.

Keywords: Electrocardiogram (ECG), ECG compression, Embedded Zerotree Wavelet, Wavelet Packets (WP), channel bank filter, filtering theory, filter bank.

1. INTRODUCTION

The design of electrocardiogram (ECG) compression techniques has been widely studied in the last few years. An outline of the most common techniques can be seen in [1], where a classification in three categories was proposed: *direct methods, transform methods and parameter extraction methods*. Since the early 90s, there have been many contributions among the *transform methods* due to the use of the Wavelet Transform, which has allowed the improvement of the compression ratios reported by the prior *transform methods*.

The Embedded Zerotree Wavelet (EZW) algorithm was specifically designed to use the Discrete Wavelet Transform (DWT) [2] in image coding applications. This method demonstrated good performance and was quickly applied to other types of signals, such as ECG [3] and myoelectric [4] signals. In the DWT decomposition algorithm, every coefficient at any scale is related with two other coefficients at the immediate lower scale. This correspondence is iterated through scale giving the temporal orientation tree. An example is illustrated in Fig. 1. The set of a coefficient and its descendents is called zerotree. In the encoding process, the whole set of coefficient of a zerotree can be pointed by its root which is the first coefficient of the temporal orientation tree at the lower scale. In the encoding–decoding process, a coefficient is called significant if its amplitude is greater than a given threshold value ε. Therefore, depending on the magnitude of a coefficient related to ε, i.e., its significance, it can be encoded as a symbol of a reduced alphabet to obtain a significance map. The EZW algorithm takes into account the hierarchy of the DWT coefficients among different subbands to efficiently encode the significance map and use an alphabet of four symbols [2]:{`POS, NEG, IZ, ZTR`}. Symbols {`POS`} and {`NEG`} indicate the sign of a significant coefficient. A non significant coefficient is encoded with the symbol {`ZTR`} if it is the root of a zerotree, i.e., if all the coefficients of the zerotree are also non significant. Conversely, the non significant coefficient is encode as an isolate zero with the symbol {`IZ`}.

In the ECG compression case, a modified version of the EZW algorithm is reported in [3] that uses Wavelet Packets (WP), but the resulting algorithm performed worse than the DWT–based algorithm. The reason for the poor performance in the WP case is that the best basis decomposition often splits the signal into a number of smaller hierarchies that cannot be efficiently encoded by zerotrees.

The motivation of this work has been the development of an EZW-based algorithm to be used with WP. To do so, the hierarchical relationships among coefficients has not been taken into consideration. In this sense, the {`ZTR`} [2] symbol that identifies the root of a zerotree is withdrawn from the alphabet so that only three symbols ({`POS, NEG, IZ`}) encode the significance map.

In this paper we present a versatile embedded encoding scheme to be used with WP —Embedded Wavelet Packets (EWP) algorithm. Simulations results are provided demonstrating the improvement in performance of the proposed encoder over the original EZW DWT–based algorithm. Finally, we want to emphasize that although this work focus on ECG, other kinds of signals such as myoelectric signals or images (processed using linear–phase filter banks) can be also compressed with the proposed algorithm.

2. WAVELETS PACKETS

The Discrete Wavelet Transform (DWT) decomposes a signal $f(t)$ as a successive approximation in several scales as follows [5]

$$f(t) = \sum_k c_{j_0}(k) 2^{j/2} \varphi\left(2^j t - k\right) + \sum_k \sum_{j=j_0}^{\infty} d_j(k) 2^{j/2} \psi\left(2^j t - k\right), \tag{1}$$

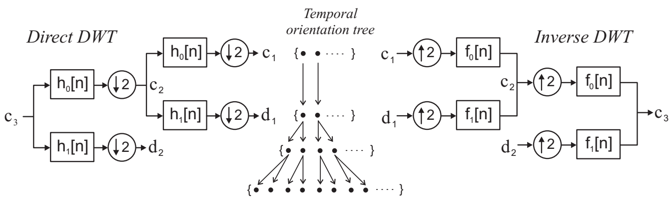

Figure 1: Algorithm for the computation of DWT. A temporal orientation tree scheme is depicted showing the relationships among coefficients through scale.

where bases functions are dilated and translated versions of the wavelet function $\psi(t)$ as well as the scaling function $\varphi(t)$.

The coarse details of $f(t)$ are represented by the scaling coefficients $c_{j0}(k)$ while the finer details are represented by the wavelets coefficients $d_j(k)$. An efficient way to compute the wavelet transform is by means of a 2-channel perfect reconstruction filter bank applied iteratively to the low pass channel as shown in Fig. 1, where the number of layers or levels of the resulting filter bank depends on the desired resolution scale. The inverse transform is carried out with the corresponding synthesis filter bank.

WP theory is the generalization of DWT. The input signal is decomposed applying the 2-channel perfect reconstruction filter bank at both the low and high pass branch. The resulting binary tree is considered as a library of bases of which only one would be needed to represent the incoming signal. The number of bases A_n for an n–layered WP can be recursively calculated as

$$A_n = 1 + {A_{n-1}}^2, \tag{2}$$

where A_{n-1} is the number of bases of a $(n-1)$–layered WP, being $A_0 = 1$. Therefore, WP can be utilized adaptively by selecting the best basis, that basically consists of pruning the tree according to a cost function. The best basis selection algorithm used in this work is the proposed in [6]. An example of how the best basis selection of an incoming signal for the case of a 4–layered WP is shown in Fig. 2. Basically, the whole binary tree is first obtained and subsequently pruned according to the Shannon entropy as proposed in [6]. The broken lines in Fig. 2 correspond to the rejected branches, while the others give the filter bank for processing the incoming signal. Accordingly, different filter banks are used to process when the input signal is split in blocks.

3. EMBEDDED WAVELET PACKETS (EWP) ALGORITHM

This compressor does not need any signal preprocessing as QRS complex detection and no *a priori* signal knowledge is required. It works over non–overlapped blocks of N samples each of the incoming signal as follows:

1. Decompose every input block using WP.
2. Encode the coefficients with an EZW–based embedded algorithm.
3. Entropy-code the significance map.

The embedded algorithm is carried out as a successive approximations that are applied to each group of N coefficients obtained from the corresponding N incoming samples.

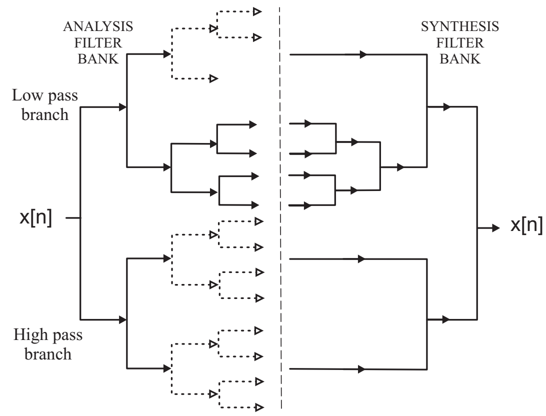

Figure 2: Example of WP for a depht of four layers. The broken lines correspond with the pruned branches.

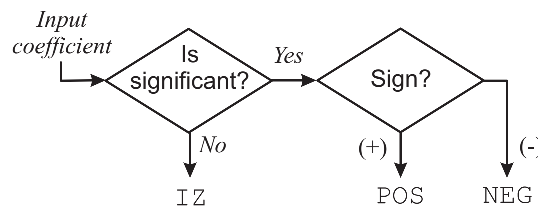

Figure 3: Alphabet for the significance map.

Basically, it consists of applying a sequence of thresholds $\varepsilon_0, \varepsilon_1, \cdots, \varepsilon_{L-1}$ in successive steps to obtain the corresponding significance maps associated to each threshold. At each iteration, the threshold is successively updated at the half of the prior value: $\varepsilon_i = \frac{\varepsilon_{i-1}}{2}$. Let $\{c_i\}, \forall i = 1, \cdots, N$ be the set of WP coefficients; the first threshold value $\varepsilon_0 = 2^p, p \in \mathbb{Z}$ is chosen in such a way that

$$p = \left\lfloor \log_2 \left(\max_{c \in \{c_i\}} \{|c_i|\} \right) \right\rfloor, \tag{3}$$

where $\lfloor \cdot \rfloor$ denotes rounding to the next smaller integer. Given p, the following threshold is $\varepsilon_1 = 2^{p-1}$ and so on. Two lists must be maintained while the encoding (and decoding) process proceeds: The dominant list (DL) contains all the coefficients found significant to the current and prior thresholds, and the subordinate list (SL) contains its magnitudes. Initially, DL equals the coefficients resulting of the transform of the corresponding incoming block, and SL is empty. These lists are updated at every iteration.

For every iteration, i.e., for every successive thresholds, the dominant pass and the subordinate pass are accomplished. During the dominant pass, coefficients in DL are compared with the threshold, e.g. $\varepsilon_0 = 2^p$ for the first iteration. Then, the significance map is encoded as explained in Fig. 3 with an alphabet made of three symbols: {`POS`, `NEG`, `IZ`}. The magnitude of the significant coefficients (encoded either as {`POS`} o {`NEG`}) is included in SL. Subsequently, the significant coefficients are zeroed in DL to avoid being significant at the following iteration. The reconstructed magnitude of a significant coefficient to the threshold 2^p on the decoder side is $|\hat{c}_i| = 2^p + 2^{p-1}$. The sign is taken from the corresponding code in the significance map.

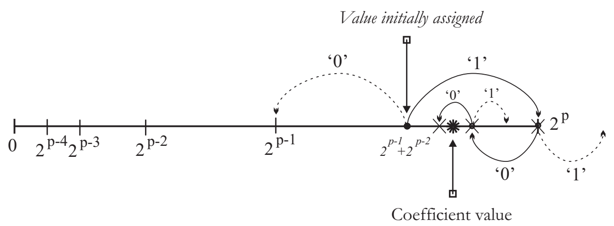

Figure 4: Example of successive refinement of a coefficient. The curved solid line arrows point at the final values once the refinement bit is successively assigned

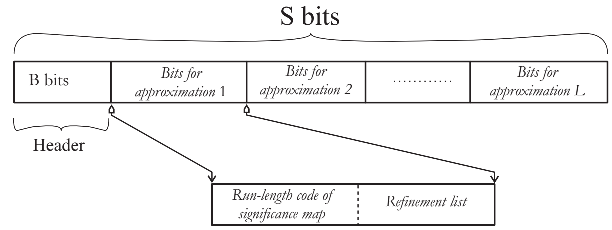

Figure 5: Coding of a block.

Once the dominant pass is finished, the subordinate pass is carried out. The aim of the subordinate pass is to improve the accuracy of all the previous significant coefficients (those included in LS) by means of a refinement bit. Figure 4 shows the subordinate pass applied successively three times to a coefficient, which is significant to the threshold of the second iteration (for $p-1$). Its initial value is $|\hat{c}_{i,1}| = 2^{p-1} + 2^{p-2}$. For this current threshold, there is an interval of uncertainty whose whidth is 2^{p-2}. The actual coefficient value is in the upper half of the uncertainty interval, so a '1' is assigned as refinement bit the first time that the subordinate pass is applied and the reconstructed coefficient is $|\hat{c}_{i,2}| = |\hat{c}_{i,1}| + 2^{p-2}$. The updated value in the refinement pass is pointed by the curved solid line arrow in Fig. 4. In the following iteration (for $p = p-2$), the width of the uncertainty interval is 2^{p-3} and a '0' is the assigned refinement bit as the actual coefficient value is in the lower half, yielding the reconstructed coefficient $|\hat{c}_{i,3}| = |\hat{c}_{i,2}| + 2^{p-3}$. Once again, for the following iteration, the refinement bit is '0' so the coefficient takes value $|\hat{c}_{i,4}| = |\hat{c}_{i,3}| + 2^{p-4}$. All the coefficients found in the SL are refined as above in each iteration. To do so, the encoder generates the refinement list (RL), which contains the refinement bits to be used by the decoder.

The encoding and decoding process is summarized following. From the encoder side, let $\{c_i\}, \forall i = 1, \cdots, N$ be the set of WP coefficients:

Step 1 Output $p = \left\lfloor \log_2 \left(\max\limits_{c \in \{c_i\}} \{|c_i|\} \right) \right\rfloor$

Step 2 *Initialization of lists:*
- (a) $DL = \{c_i\}, \forall i = 1, \cdots, N$.
- (b) $SL = \{\phi\}$.

Step 3 *Dominant pass*:
- (a) $DL(i), \forall i = 1, \cdots, N$, *is encoded as in Fig. 3.*
- (b) *If* $DL(i)$ *is significant, its magnitude is included in SL and* $DL(i) = 0$ *is done.*

Step 4 *Subordinate pass: The refinement list is generated.*

Step 5 $p = p-1$ *and go to* Step 3.

Conversely, the decoder performs as follows:

Step 1 *Initialization of lists:*
- (a) $DL = \{c_i\}, \forall i = 1, \cdots, N$.
- (b) $SL = \{\phi\}$.

Step 2 *Initial threshold* p *is received.*

Step 3 *Dominant pass:*
- (a) $\{\texttt{POS}\}$: $DL(i) = 2^p + 2^{p-1}$ *and its magnitude is included in SL.*
- (b) $\{\texttt{NEG}\}$: $DL(i) = -(2^p + 2^{p-1})$ *and its magnitude is included in SL.*
- (c) $\{\texttt{IZ}\}$: *nothing is done.*

Step 4 *Subordinate pass: coefficients in* SL *are refined after receiving* RL.

Step 5 $p = p-1$ *and go to* Step 3.

To encode the significance map, a bit '1' marks a significant coefficient so that the following indicates the sign: '11' is used for $\{\texttt{POS}\}$ and '10' for $\{\texttt{NEG}\}$. Non-significant coefficients corresponding to the symbol $\{\texttt{IZ}\}$ are marked by '0' and are run-length encoded. Every time $\{\texttt{IZ}\}$ appears, the next B_1 bits are used to encode the number of consecutive $\{\texttt{IZ}\}$ symbols. In case of overflow, B_2 bits more are used. Therefore, $2^{B_1} + 2^{B_2} - 1$ consecutive symbols can be encoded. In this work, the corresponding values are $B_1 = 5$ and $B_2 = \log_2 N$, where N is the total number of samples in the block.

Giving the above, the stream for every incoming block consists of a header followed by groups of bits with the number of groups equal to the number of approximations made for the corresponding segment, as is shown in Fig. 5, where it is supposed that L approximations have been achieved. The resulting run–length coding of the significance map must be enclosed followed by the stream of refinement bits for every iteration. The header must contain the initial threshold and a word indicating the corresponding basis decomposition. By maintaining a table with the decomposition bases, the length of the word will depend on the amount of possible bases A_n given by Eq. (2). Thus, the number of bits H to represent the word that indicate the WP filter bank can be calculated as

$$H = \lceil \log_2(A_n) \rceil, \tag{4}$$

where $\lceil \cdot \rceil$ denotes rounding to the next larger integer. At the end, both the encoding and the decoding processes finish once the compression ration (CR) is reached.

4. RESULTS

4.1 Reference algorithm

In order to show the performance of the proposed algorithm, we implemented the conventional EZW compression algorithm [2][3] to compare. In this case, the alphabet of the significance map has four symbols, so 2 bits are used for each symbol. Both $\{\texttt{ZTR}\}$ and $\{\texttt{IZ}\}$ symbols are run–length encoded as explained before, where B_1 and B_2 are 2 and 8 respectively. The bit stream for every incoming block is as in Fig. 5, but as EZW algorithm utilizes the DWT, no word indicating the basis has to be included in the header.

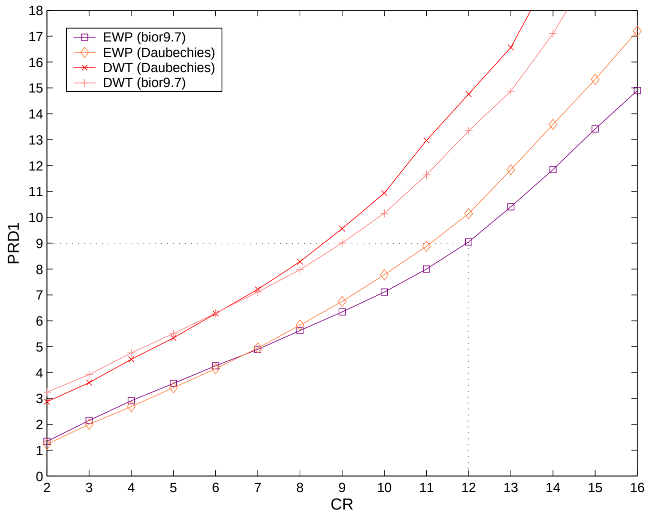

Figure 6: Results from the experiment showing PRD1 against CR.

4.2 Specifications of decomposition method

WP was studied for the design of a thresholding–based ECG compressor in [7]. Therefore, to accomplish the evaluation of the method proposed in this work, the design of the decomposition method is based on that reported in [7], which is as follows:

- The number of layers for DWT and WP are up to 4.
- Based on the good results reported by other authors [3, 8], we utilize the Cohen–Daubechies–Feauveau 9/7 (bior9.7). Results are also shown with the 14–tap Daubechies orthogonal wavelet

Moreover, the number of samples N per block of incoming signal is 1024 which is the size most commonly used by other authors [3][8][9].

4.3 Performance measurement and database

The quality of the retrieved signal is measured using the Percentage Root–mean–square Difference (PRD):

$$PRD = \sqrt{\frac{\sum_{n=1}^{N}\left(x\left[n\right]-\hat{x}\left[n\right]\right)^2}{\sum_{n=1}^{N}\left(x\left[n\right]\right)^2}} \times 100, \tag{5}$$

where $x[n]$ is the original signal and $\hat{x}[n]$ the reconstructed signal. This parameter depends on the mean value of the original signal, so it is thus strongly recommended that the following criteria be used [10]:

$$PRD1 = \sqrt{\frac{\sum_{n=1}^{N}\left(x\left[n\right]-\hat{x}\left[n\right]\right)^2}{\sum_{n=1}^{N}\left(x\left[n\right]-\overline{x}\left[n\right]\right)^2}} \times 100, \tag{6}$$

where $\overline{x}[n]$ is the mean value of the signal.

Table 1: Comparison of the proposed algorithm with other methods.

Method	*Signal*	*CR*	*PRD*[a]	*PRD1*	*PRDcc*
Proposed method (EWP)	117	8	1.5070	5.4681	0.2889
		11	2.0588	7.4703	0.3947
	232	7	5.8204	11.3139	0.2855
		9	7.7282	15.0223	0.3791
Djohan[11]	117	8	3.9	—	—
Hilton[3]	117	8	2.6	—	—
Lu[8]	117	8	1.18	—	—
Rajoub[12]	117	10.7996	—	—	0.4808
	232	4.3141	—	—	0.3005
Benzid[9]	117	16.24	2.55	—	—
	232	9.04	—	—	0.2981

[a]PRD has been obtained with (5) after removing the 1024–baseline; PRD1 has been obtained with (6) and PRD_{cc} with (5) but with the corresponding baseline included.

As the incoming signal is split in segments of 1024 samples ($N = 1024$), the compression ratio can be calculated as

$$CR = \frac{N \times 11}{S}, \tag{7}$$

where S is the bit stream for every input block (Fig. 5).

The tests utilize the MIT-BIH Arrhythmia Database. Every file from that database holds two leads sampled at 360 Hz with a resolution of 11 bits per sample. A baseline of 1024 has been added to each ECG for storage purposes that is removed before processing.

4.4 Experiment results

The experiment is carried out over both 10-minutes long leads extracted from records 100, 101, 102, 103, 107, 109, 111, 115, 117, 118 and 119 from the MIT–BIH Arrhythmia Database. This dataset was proposed in [8] and it consists of a variety of signals with different rhythms, QRS complex morphologies and ectopic beats. Figure 6 shows the performance of the proposed compressors compared with the conventional EZW. The CR is in the horizontal axis because is the target parameter and can be considered as the independent variable. As can be seen, the WP–based proposed compressor yields improved performance over the conventional EZW technique in the full range of CR values.

The comparison of the proposed method with other works is given in Table 1. The results have been obtained over the first 1–minute long lead of the corresponding records. Several cells in Table 1 are empty because not all the authors utilize all the measurement parameters, so the results of other authors are placed in the cell corresponding to the measure they have used to test. The proposed algorithm clearly improves the method presented in [12].

5. CONCLUSIONS

An ECG quantization algorithm based on the EZW for use with WP is presented. Since the hierarchical relationship among WP coefficients is difficult, zerotrees are not encoded reducing the alphabet of the significance map to three symbols. Thus these new algorithm become easier than the conventional EZW and make it possible to used to other kind of decomposition structures like conventional filter banks. Furthermore, it reports good compression ratios compared to other methods.

Acknowledgment

This work has been supported in part under grants PR–2005–0131 from the Spanish Ministry of Science and Education, UAH PI2005/074 from University of Alcalá and CAM-UAH2005/021 from both the Autonomous Community of Madrid and the University of Alcalá.

References

[1] S. M. S. Jalaleddine, C. G. Hutchens, R. D. Stranttan, and W. A. Coberly, "ECG data compression techniques. A unified approach," *IEEE Transactions on Biomedical Engineering*, vol. 37, no. 4, pp. 329–343, April 1990.

[2] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3445–3462, December 1993.

[3] M. L. Hilton, "Wavelet and wavelet packets compression of electrocardiogram," *IEEE Transactions on Biomedical Engineering*, vol. 44, no. 5, pp. 394–402, May 1997.

[4] J. A. Norris, K. B. Englehart, and D. F. Lovely, "Myoelectric signal compression using zerotrees of wavelets coefficients," *Medical Engineering and Physics*, vol. 25, no. 9, pp. 739–746, November 2003.

[5] G. Strang and T. Q. Nguyen, *Wavelets and Filter Banks*. Wellesley MA: Wellesley–Cambridge, 1996.

[6] R. R. Coifman and M. V. Winckerhauser, "Entropy–based algorithms for best basis selection," *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 713–718, March 1992.

[7] M. Blanco–Velasco, F. Cruz–Roldán, F. López, A. Bravo–Santos, and D. Martínez–Muñoz, "A low computational complexity algorithm for ECG signal compression," *Medical Engineering and Physics*, vol. 26, no. 7, pp. 553–568, September 2004.

[8] 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.

[9] R. Benzid, F. Marir, A. Boussaad, M. Benyoucef, and D. Arar, "Fixed percentage of wavelet coefficients to be zeroed for ECG compression," *Electronics Letters*, vol. 39, no. 11, pp. 830–831, May 2003.

[10] Y. Zigel, A. Cohen, and A. Katz, "The Weighted Diagnostic Distortion (WDD) measure for ECG signal compression," *IEEE Transactions on Biomedical Engineering*, vol. 47, no. 11, pp. 1422–1430, November 2000.

[11] A. Djohan, T. Q. Nguyen, and W. J. Tompkins, "ECG compression using discrete symmetric wavelet transform," in *Proc. of the 17th Annual International on Conference of the IEEE Engineering in Medicine and Biology*, vol. 1, 1995, pp. 167–168.

[12] B. A. Rajoub, "An efficient coding algorithm for the compression of ECG signals using the wavelet transform," *IEEE Transactions on Biomedical Engineering*, vol. 49, no. 4, pp. 355–362, April 2002.