Efficient Electrocardiogram (ECG) Lossy Compression Scheme For Real Time e-Health Monitoring

Hatim Anas [1]. Rachid Latif [2]. Mounir Arioua [3]

Abstract E-health monitoring is adopted to solve multiple problems such as: difficult access to hospitals, health monitoring of old patients ... Several operations slow down the e health systems, the most important one is the signal compression / decompression step. In this paper we present a new algorithm for compression / decompression of the ECG vital signal. The complexity of the proposed algorithm is very low and uses simple mathematical operations. In a hardware point of view, this property makes it suitable for real-time e-health monitoring. The algorithm's kernel is based on the delta coding technique. We introduced two coding categories low and high and we defined a new frame format. This allows us to minimize the total amount of bits of the compressed signal. Three variants of the algorithm are designed and tested using the MIT-BIH physionet and PTB Diagnostic data bases. We used several signals with different cardiac pathologies for test. We reach a maximum compression ratio (CR) of 47 with a PRD of 0,073%. Our algorithm outperforms the state of the art techniques.

Keywords Compression, ECG, Delta coding, compression ratio , Biomedical signal processing

[1] (✉) LGEMS National School of Applied Science Agadir (ENSA Agadir), Ibn Zohr University Agadir, Morocco hatim.anas@gmail.com

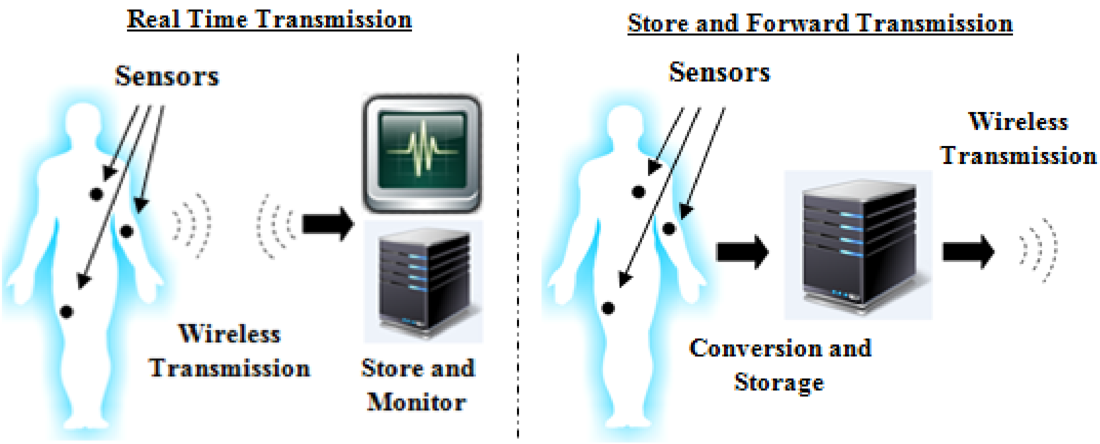
[2] LISTI, National School of Applied Sciences (ENSA Agadir) Ibn Zohr University ,Agadir, Morocco

[3] National School of Applied Sciences (ENSA Tetouan), Abdelmalek Essaadi University, Tetouan, Morocco

I INTRODUCTION

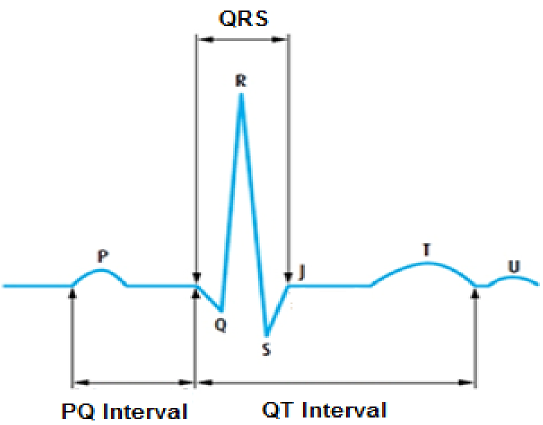
E-health monitoring is one of the most attractive research fields [1]. It takes more and more places in health care applications. Thanks to the biomedical engineering and embedded systems continuous development. Many projects and finical grants are attributed to this field [2–4].E health monitoring consists of transmitting vital signals (ECG, EEG, EMG , Blood pressure…) to a specified destination using a wireless network. In this paper our main concern is the Electrocardiogram signal. The transmission of ECG signal can be done based on two strategies (Fig 1). The first is store and forward manner. This strategy is based on storing the ECG signal in a database and with a certain frequency the signal will be transmitted to the doctors and specialists. The drawback of this strategy is that the patient is not real time monitored. The second strategy is the real time ECG transmission. This strategy allows making actions in emergency cases in the right time. The monitoring system receives in real time way the transmitted ECG signals. Complex real-time ECG monitoring systems are born recently [5,6]. The e-health real time system faces multiples technical difficulties. The evolving ECG e-health monitoring makes the real time processing task very difficult. An important number of patients can be monitored by a doctor in the same time using one base station. It needs high throughput physical architectures and high data rates. In a hardware point of view the complexity of the signal compression/ decompression is a key factor of building a system with high performances. In addition the efficiency of the compression algorithms improve the data rates considerably. This is why the compression algorithms are necessary to reduce the amount of data to be transmitted. Our aim in this paper is to propose a fast and efficient algorithm for ECG compression/decompression for real time monitoring.

Fig. 1 ECG signal Transmission policies



Many strategies are used to compress an ECG signal. It can be compressed by transformation methods, direct compression techniques, or parameters extraction techniques. The direct ECG compression techniques [7,8] usually apply preprocessing stages which can be region specific. The principal behind the transformation based compression techniques is to eliminate the redundancies based on the spectral and energy distribution analysis [9, 10]. In the state of the art, the well-known transformations are used such as: Discrete cosine transforms [13], Wavelet transformation [11–12], orthogonal transforms, fast Fourier transform [14] were applied for compressing the ECG signal. Parameter extraction algorithms [15] are based on predictions. Multiple techniques are used like splin approximations pattern matching methods…all of these methods can be lossy or lossless. A typical ECG signal is illustrated in figure 2. In ECG signals multiple intervals are involved.

Fig.2 A typical ECG signal



The P-R Interval represents the sequential activation of the right and left atria. The QRS interval is the most important one. The QRS represents the simultaneous activation of the right and left ventricles, although most of the QRS waveform is derived from the larger left ventricular musculature. The QT Interval is the time between the beginning of depolarization of the ventricular myocardium (beginning of the QRS complex) and the end of the repolarization (end of the T wave).

In this paper a lossy compression technique is presented and evaluated with different signals with different pathologies. The highlights of the proposed method are summarized as follows:

- Based on the ECG intervals variation we defined two coding categories, low and high.
- We defined a new frame format. This allows us to compress an important number of successive differences in one packet.
- We designed three variants of the proposed algorithm. The first is static low coding based, the second is dynamic low coding based and the third is based on a minimum window size.

The manuscript is organized as follows, after an introduction and a state of the art in chapter 1 and 2, the new compression scheme is presented in chapter 3. The chapter 4 presents a study of the low coding interval. The chapter 5 presents the compression algorithm with the minimal window size. In chapter 6 the testing results are presented and a conclusion comes right after.

II. REALTED WORKS

The work in [16] proposed a variant of the delta coding. It uses 8 values on each packet at the same time. The forward and the inverse encoding were developed. Each difference is coded using 8 bits. The forward or inverse encoding tries to establish a combination between two successive differences to pack them on one word of 8 bits. This work reaches a compression ratio of 15.72 and a percentage root mean squared difference (PRD) of 7.89%. The work in [17] relays on the work presented in [16]. It offers a compression depending on the intervals of the ECG signal. The QRS interval is very important for medical diagnosis, so its compression will be lossless therefore a tolerable lossy compression will be applied to the other fields. This method achieves a compression ratio of 22.47 against 7.5% PRD. Another algorithm was presented in [18]. This compression scheme is composed of two stages, a preprocessing and a compression stages. The preprocessing stage consists of filtering with 6-degree savitzky-Golay filter (SGF) using a 17-points constant window. The next step consists in ECG local minima and maxima extracting and rounding before quantification. This algorithm achieves a compression ratio of 18.27 and a 1.27% PRD. In [19] a non-recursive 1-D discrete periodical wavelet transform (1-D NRDPWT) is introduced. The main advantage of this method is to limit error propagation and word-length-growth which is the main disadvantage of wavelets ECG signal compression. This method can reach a compression ratio of 21.6 and a PRD of 4.02%. An algorithm presented in [20] presents an ECG signal based on the quad level vector (QLV) processing method. The ECG compression consists of the preprocessing flow and the classification flow. QLV is then applied for both flows with low-computation complexity to achieve better performances. It reaches 16.9 compression ratio and 0.64% PRD. In [21] is a combination of delta coding and run length encoding. In addition a quantization factor was introduced to make the differences between 0 and 99. The performance of this method reaches a compression ratio of 17 and PRD of 4.6%. The work in [22] presents an algorithm EDLZW based on the Lempel-Ziv Welch (LZW) technique. The EDLZW algorithm uses the channel separation function (CSF) which separates the ECG data into each n channels ($n \leq 12$) and uses the data control function (DCF) to compresses the separated data. Results actually showed that the EDLZW compression ratio is 8.66 without any distortion. The work in [23] presents an algorithm based on singular value decomposition (SVD), and embedded zero tree wavelet (EZW) techniques. The proposed method utilizes the low rank matrix for initial compression on two dimensional ECG data array using SVD, and then EZW is initiated for final compression. Here, three different beat segmentation approaches have been exploited for 2-D array construction using segmented beat alignment with exploitation of beat correlation. The method achieves a compression ratio of 24,25 with a PRD of 1.89%.The work in [24] is another SVD ECG compression technique. It ensures a high compression ratio by exploiting both intra-beat and inter-lead correlations. A new thresholding technique based on multiscale root fractional energy contribution was developed. The method achieves a compression ratio of 19.34 with a PRD equal to 3.05%. Another technique is presented in [25]. The algorithm is based on the compressed sensing technique. The sparsity of dimension-reduced eigenspace multichannel ECG signals is exploited to apply CS. Principal component analysis (PCA) is applied over the multichannel ECG data to retain diagnostically important ECG features in a few principal eigenspace signals. The compressed measurements are quantized using a uniform quantizer and encoded by a lossless Huffman encoder. The signal recovery is carried out by an orthogonal matching pursuit (OMP) algorithm. The method achieves a compression ratio of 17,76 with a PRD of 5,46%.In [26] a low-complexity compression scheme of electrocardiogram (ECG) signals based on the Haar wavelet transform (HWT) for use on mobile devices is introduced. The proposed scheme achieves an average a percent root mean square difference (PRD) of 9.77 along with a compression ratio (CR) of 24.95.

III Novel ECG compression algorithm

The proposed compression algorithm is based on the delta coding technique. The algorithm aims to maximize the profit of the successive differences. It's a significant improvement of delta encoding variants. The total bits of the compressed signal depend obviously on the number of bits used to code the differences, typically 8 bits. Several algorithms try to make mixtures and associations to be able to code 1 or 2 components within a single byte.

The principle of the proposed scheme is to encode the values of the differences using the minimum possible bits. We studied the computed differences ranges. This leads us to the fact that coding categories must be created. In ECG signals, the PQ and ST intervals are generally without a great variation therefore the variations in the QRS interval are important. Here we defined two encoding categories: low encoding and high encoding. The low encoding uses n1 bit to encode the differences and the high encoding uses n2 bits. In order to differentiate between the two coding categories the data must be structured as a data frame with specific fields. The coding categories and the new frame format will be the key for efficient compression. We will be able to compress an important number of successive signal values in one packet instead of 8. The constructed frames has no dependences with the protocol used for the transmission and will be considered at this point as data to be transmitted. The delta coding technique principal is to consider a first value and to compute the differences starting from it on an 8 data window with the formula below:

$$\text{Delta}(1)= \text{Input}(1)$$
$$\text{Delta}(i)= \text{Input}(i)-\text{Input}(i-1)\ ,\ 1 \leq i \leq 8$$

Some arrangements are applied to optimize the bits needed. The compression scheme begins with a pretreatment.

A. The pretreatment step

The proposed algorithm applied first a pretreatment to the input signal. The pretreatment begins with a conversion of the ADC output signal to its real value based on gain and offset signal parameters (Eq 1, 2).

$$\mathrm{Y} = \mathrm{Inputs} - \mathrm{offset} \quad (Eq\ 1)$$

$$\mathrm{Conditionned} = \frac{\mathrm{Y}}{\mathrm{gain}} \qquad (Eq\ 2)$$

Then, the converted signal is multiplied by 100 and rounded. An error is introduced while rounding by the pretreatment, which is the only one introduced by the proposed algorithm. It allows us to reduce the number of the bits needed to code the difference signal and will improve greatly the compression efficiency. In table 1 an example is introduced, considering a signal mean value of 1024 and a gain of 200. The maximum error then can occur in this example is 1.

Table 1 Example of preconditioning operations

Forward Operations	Inputs	*957*	*956*	*953*	*957*
	conditioned	*-0.335*	*-0.340*	*-0.355*	*-0.335*
	X'=X× 100	*-33.500*	*-34.000*	*-35.500*	*-33.500*
	X''=Round (X')	*-34*	*-34*	*-36*	*-34*
Proposed delta Coding/Decoding					
Inverse Operations	X'''= X'' / 100	*-0.34*	*-0.34*	*-0.36*	*-0.34*
	Inputs'	*956*	*956*	*952*	*956*
	|Input-Input'|	*1*	*0*	*1*	*1*

We will evaluate the general expression of the error introduced in this step.

Let *x* be the ADC value, and X is the conditioned *x*.
Using Eq1and Eq2 *X* is described as follow:

$$X = \frac{x - \mathrm{offset}}{\mathrm{gain}} \qquad (Eq\ 3)$$

We define *X'* as the result of multiplying *X* by 100, *X'* will be as follow:

$$X' = X \times 100 \ = (+/-)\ 0, X_1X_2X_3 \ldots X_n \ \times 100 \\ = \ (+/-)\ X_1X_2\ , X_3 \ldots X_n \qquad (Eq\ 4)$$

We define *X"* as follow :

$$X" = \mathrm{round}(X') \\ = \begin{cases} (+/-)\ \ X_1(X_2+1) & if\ \ X_3 \geq 5 \\ (+/-)\ \ X_1X_2 & if\ \ X_3 < 5 \end{cases} \qquad (Eq\ 5)$$

Let ε be the error introduced between X and X''',

$$\varepsilon \ = \ X \ - \ X''' \\ = \begin{cases} (+/-)\ \big(0.X1X2X3 \ldots Xn - 0.X_1(X_2+1)\big)\ if\ \ X_3 \geq 5 \\ (+/-)\ \ \ (0.X1X2X3 \ldots Xn - 0.X_1X_2) \qquad if\ \ X_3 < 5 \end{cases} \\ = \begin{cases} (+/-)\ (0.00X3 \ldots Xn - 0.01)\ if\ \ X_3 \geq 5 \\ \ (+/-)\ \ \ 0.00X3 \ldots Xn \qquad if\ \ X_3 < 5 \end{cases} \quad (Eq\ 6)$$

We can easily see that the maximum ε is 5.10^{-3}. Since the real values of the ECG signal are in the mV scale, the error introduced is in µV scale.

Now we will evaluate the error introduced at the ADC values. At the reception side the inverse operations must be done, *X"* is divided by 100 and then the preconditioning inverse operations are done.

Let *x'* be the reconstructed value, *x'* is defined as follow:

$$x' = (X" * \mathrm{gain}) - \mathrm{offset} \qquad (Eq\ 7\ )$$

The error introduced is:

$$\Delta = |x - x'| = |x - [X" * gain - offset)]| \\ = |x - [(X' - \varepsilon) * gain - offset)]| \\ = \left| x - [(\frac{x - offset}{gain} - \varepsilon) * gain - offset)] \right| \\ = |x - [x - offset - \varepsilon * gain - offset)]| \\ \Delta \quad = |\varepsilon \times gain| \qquad\qquad (Eq\ 8)$$

We can evaluate the maximum value of Δ with *Eq 6*.

$$\Delta_{\max} \ = 5 \times \mathrm{gain} \times 10^{-3} \qquad (Eq\ 9)$$

In the example of table 1, the gain is 200. The maximum error then is 1 as stated before:

$$\Delta_{\max} \ = 5 \times 200 \times 10^{-3} = 1 \qquad (Eq\ 10)$$

B. The proposed compression algorithm

After the rounding step, the enhanced delta encoding begins. The compression algorithm is presented in fig 4. First of all an "Anchor value" is defined. At the beginning of compression, the "Anchor value" is initialized with the first value of the signal. After this step the first difference is calculated. If the calculated difference absolute value is under the low category step, the old coding is initialized by Low otherwise it's initialized by High. After coding the first difference the initialization step is done. The next step begins by reading a new signal value. The difference between the new signal value and the old one is computed and followed by a test. The test is necessary to determine to which category belongs the difference. If the absolute value of the difference is under the low category threshold it will be considered for low coding otherwise it will be considered for high coding. Another test follows these operations. If the coding type of the computed difference remains the same as the old coding type we proceed to encoding and a counter is incremented. Otherwise if the coding type of the computed difference is not the same as the old coding type, the difference is neglected and the counter stops. The construction of the current frame is ended. We proceed to the frame composition and transmission. The red signal value is then considered as a new anchor to define a new frame starting from the initialization phase. The data frame formed is based on the following format **(**fig3). The frame format presented in figure 3 is absolutely necessary to decompress the encoded signal. The data frame is composed of 4 fields. The fields are: The anchor value (9 bits), the window size (9 bits), the coding type (1bit) and the *k*_differences (n1/n2bits). The anchor value is the key for recovering the real values of the compressed signal. The window size is the number of successive differences coded by a same type and its equal to the counter value. The k_ differences are coded using n1 bits for low coding and n2 bits for high coding. The number of the differences is equal to the window size. The thresholds and coding categories will be defined in the next sub sections. Different strategies were investigated. We defined a static low coding category, dynamic low coding category and dynamic low coding with a minimum window size.

Fig.3 Data frame composition and fields

Anchor value *(9 bits)*	Window Size *(9 bits)*	Coding type *(1 bit)*	1_difference *(n1/n2 bits)*	…	N_difference *(n1/n2 bits)*

Anchor value : 9 bits *(1 sign bit + 8bits);*
Window size : 9 bits *; is the number of the encoded differences with the same coding type (Low or High) Max 512*
Coding type: 1 bit *, if 1 Low coding , if 0 High coding;*
k_difference: n1 *bits for low coding or* ***n2*** *bits for High coding*

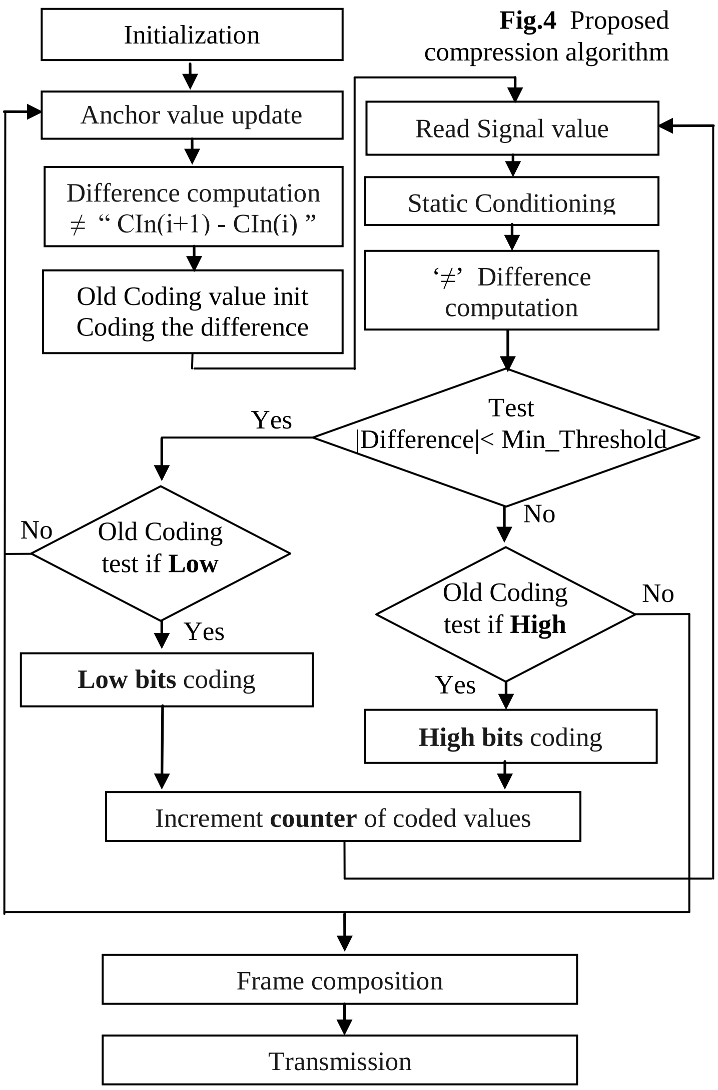


Fig.4 Proposed compression algorithm

IV. LOW CODING CATEGORY THRESHOLD STUDY

IV.a STATIC LOW CODING CATEGORY

To complete the compression scheme, a study of the coding category threshold must be done. This step is particularly important because the n1 and n2 bit influences the compression signal total size. In this sub section we will define a fixed low coding category. We plotted using MATLAB the difference signal Vs different steps and their opposite. The first step is 3 (2 bits), the second is 7 (3bits) …and the fifth is 63 (6 bits). A category is defined by [-step, step]. Several signals were used for testing and study the introduced algorithm.

The ECG data files under test are chosen from MIT-BIH and PTB diagnosis ECG database available on Physionet. We used 32 ECG test signals from Physionet with different pathologies and cardiac problems. Figure 5 presents some plotted signals vs the different steps. Analyzing the results most of the values of ECG signal can be coded with low coding. This will lead to important compression efficiency. Between successive peaks, the number of difference values is important. This will make the window size important instead of 8 usually used in the state of the art technique. This will allow us to achieve an important improvement with the proposed compression scheme. We conclude that the efficient Min step is 3 and the

Low category is [-3, 3] while the High category is [-63, 63] with n1= 3bits and n2 = 7 bits.

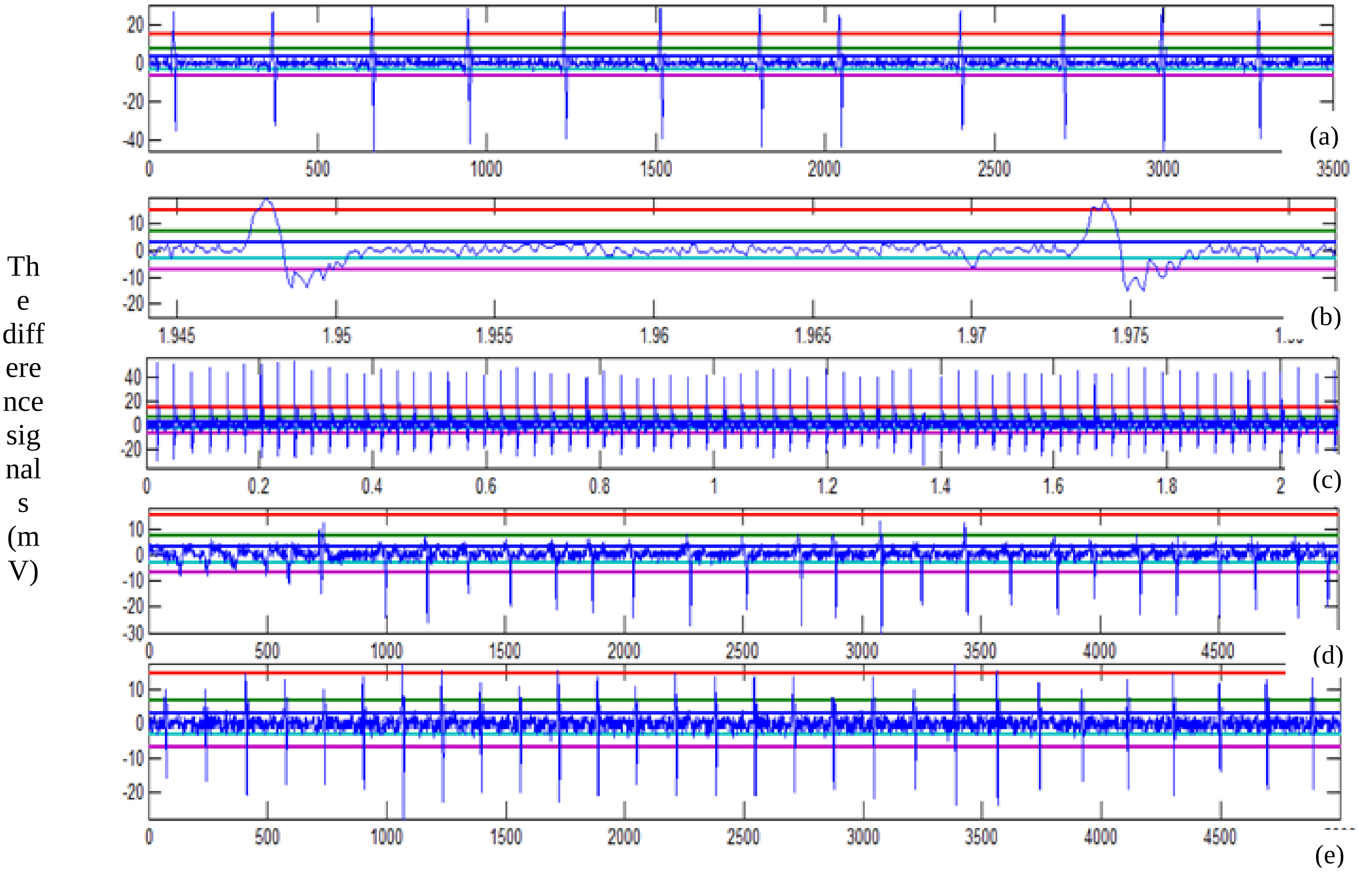


Fig.5 The difference signal Vs various coding Steps. (a) ‘100m.mat’, (b) zoom in (a), (c) ‘105m.mat’, (d) ‘231m.mat’, (e)’1442_01m.mat’.

IV.b DYNAMIC LOW CODING CATEGORY

The low coding category is the key stone of the proposed algorithm. If we change the n1 parameter the compression ratio will increase or may decrease depending on the signal properties. We evaluated the compression ratio in function of the different low coding category for multiple signals. Figure 6 presents the compression ratio Vs different low coding categories of some physionet signals. Two cases can be defined. The signals in the left, presents a maximum CR for a low coding category equal to [-3,3]. The signals in the right illustrates some cases were the optimum compression ratio is not obtained using the static low coding category [-3,3]. It’s obvious that the static low coding is not efficient to compress all the ECG signals with the different pathologies. We introduced another variant of the proposed algorithm with a dynamic low coding category. The principal behind is to evaluate at the transmitting side the compression ratio for different steps of some periods of the signal. With certain frequency, the compression ratio is evaluated and the low coding category with the highest compression ratio will be used for compression. The low coding category must be sent before the compressed signal.

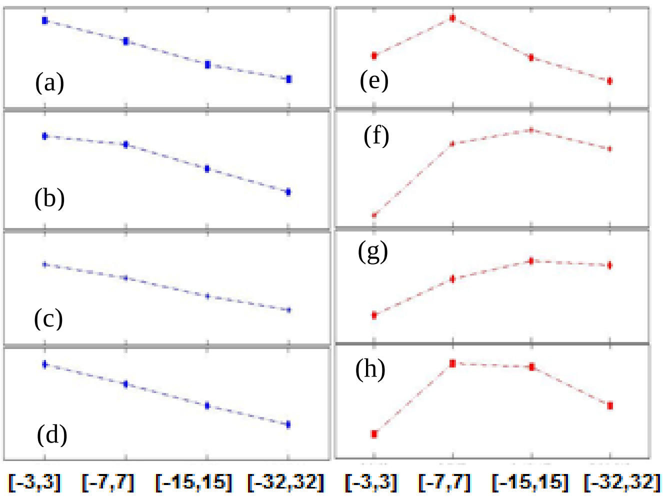


Fig 6 Compression ratio evolution Vs Low coding categories

(a) ‘100m.mat’, (b) ‘105m.mat’, (c) ‘231m.mat’, (d) ’1442_01m.mat’, (e) ‘119m.mat’, (f) ‘203m.mat’, (g) ‘217m.mat’, (h) ’ 08730_04m.mat’

IV.c NOVEL COMPRESSION ALGORITHM WITH A MINIMUM WINDOW SIZE

Using the frame format (fig 3), 19 bits are appended to the payload composed of the computed differences. This 19 bits penalty is the key of decompression phase. The 19 bits are composed by the anchor value (9bits), the window size (9 bits) and the coding type (1 bit). Since the data frames are formed when a change occurs in the coding type, an optimization can be made to reduce the number of times when it can be needed. Let's consider the following example in table 2. Assuming that low coding category is defined as [-3, +3]. In this example, the window size of the first and second data frames is equal to 1. It means that just two values of the signal are coded (anchor + one difference) on the data frame. The 19 bits penalty is spent in the second data frame to code just two values also.

Table 2 First optimisation case study for the proposed algorithm

Input Signal		113	114	139	151	152	150	149
Coding Steps	**1**	113 (Anchor)	+1(Low)	+ 25 (High)	There is a change in the coding category "CC"			
	2	113 (Anchor)	+1(Low)	139 (Anchor)	+12 (High)	+1 (Low)	Change in CC	
	3	113 (Anchor)	+1(Low)	139 (Anchor)	+12 (High)	152 (Anchor)	-2(Low)	-1(Low)
Cost		19bits + 1×3 bits		19bits + 1×7 bits		19bits + 2×3 bits		
Total cost		**73 bits**						

In this case , we can reconsider the coding type of the first data frame to remove the 19 bits penalty and replace it with lower number of bits. If we code the first data frame using the high coding type, the cost will be 65 bits instead of 73 bit (Table 3) with a savings of 8 bits. We can conclude that the window size of the encoded data frame must be over than a minimum value. In that case the switching between the coding type will be efficient. If a change on the coding type from "Low coding" to "High coding" occurs while the window size is under the min value , the change must be neglected and the differences must be coded using the high coding category. The differences are integrated with the next data and no penalty is added.

Table 3 Second optimisation case study for the proposed algorithm

Input Signal	113	114	139	151	152	150	149
Coding	113 (Anchor)	+1(High)	+25 (High)	+12 (High)	152 (Anchor)	-2 (Low)	-1(Low)
Cost	19bits + 3 × 7 bits				19bits + 2 × 3 bits		
Total cost	**65 bits**						

To define the optimum minimum window size we must express the cost function of the part to be compressed in both cases, normal and optimized algorithm.

Let N be the window size variable and k the number of bits needed to code the Low coding category . k may be 3, 4 ,5 or 6 as a maximum value. In the first case " Normal algorithm" when the High to Low coding type occures , 19 bits are added including the anchor value. After that N × k bits are added to code the differences then an other 19 bits are added including the new achor of the next data fram. We can define the cost of this part by Cost1 (N) function as follow:

$$\mathrm{Cost1}\ (\mathrm{N}) = 19 + \mathrm{N} \times \mathrm{k} + 19 \quad , \ \mathrm{N} \in \{1,2,3,..\}$$
$$= \mathrm{N} \times \mathrm{k} + 38 \quad , \ \mathrm{N} \in \{1,2,3,..\} \quad (Eq\ 12)$$

In the other side , the optimized algorithm will convert the N differences from Low coded value to High coded value , it means that no changing in the coding type occurs so the cost of this part is defined by Cost2(N) function as follow:

$$\mathrm{Cost2}\ (\mathrm{N}) = 7 + \mathrm{N} \times 7 + 7$$
$$= 7\mathrm{N} + 14 \quad , \ \mathrm{N} \in \{1,2,3,..\} \quad (Eq13)$$

The optimum window size , can be found easily by resolving analytically the folowing inequality for different values of k.

$$Cost2\ (N) \leq Cost1\ (N)\ ; \ \forall\ N \epsilon \{1,2,3 \ldots\}$$

The optimum min window siz is equal to 6, 8 and 12 repectively for k equal to 3, 4 and 5. The maximum saving that can be made is 22 for N =1 and k =5 and the minimum is 0 bits. Since the dynamique low coding changes the threshold to define the low coding category information. The threashold must be send when a change occurs. Now the window size is always over the minimum introduced before , so we choose a window size equal to 1 to send the low coding category . This frame will be called low coding category definition data frame. In that case the Anchor field and the coding type will be neglected in the reception side. The low coding category is 5 bits coded using one hot codage . The codes are 00001, 00010, 00100 and 01000 respectively for [-3,3],[-7,7],[-15,15] and [-32,32] intervals.

IV.d THE PROPOSED DECOMPRESSION ALGORITHM

With the same inverse philosophy the decompression algorithm is made with respect to the frame format (fig 3). The decompression is done frame by frame. First of all to decompress the data frame, we begin by the extraction of the compression parameters (anchor value, window size and coding type). To extract the anchor value (9 bits) a counter Cpt is initialized with 9, at every iteration a bit is red and the Cpt is decreased by 1 until Cpt reaches 0 then the Anchor value is formed. With the same manner the window size (9 bits) is formed. In this step we must test if data frame is for low coding definition or for signal compression. If the Window size is equal to 1 this means that, the low category information is being received. The next bit is red and neglected. Next we read the next 5 bits and the low category coding is defined for future data frame decompression. In that case the data frame decoding is finished and the decoder is ready for another data frame. If the data send is not low coding information, we proceed to decompress the differences. The differences decompression process begins with the initialization of a counter with the window size. Since the received data is serialized, at each iteration a bit is red. The decoding process begins by reading the difference. The difference must be deserialized with respect to the type coding. If the coding is low, a variable called word_size is initialized by n1 and if it's high it's initialized by 7. Two counters Cpt_Word_S, and Cpt_Window_S are initialized by the word size and the window size. The Cpt_Word_S is decremented until it reaches 0 and the difference is formed in parallel. The difference is then added to the old reconstructed signal. This operation is repeated until Cpt_Window_S reach 0. This means that all the differences coded by the current frame are uncompressed. The inverse of pretreatment is done to recover the original signal. An example of decompression with dynamic low coding is presented in figure 7. The decompression algorithm is presented in figure 8.

Compressed signal:

Fig 7 Decompression example

Hex: 0000410581A8FC084806304ED8540FBBAD0

Bin : 0000 0000 0000 0000 0100 0001 0000 0101 1000 0001 1010 1000 1111 1100 0000 1000 0100 1000 0000 0110 0011 0000 0100 1110 1101 1000 0101 0100 0000 1111 1011 1011 1010 1110 1101 0000

Decoding based on the frame format:

Anchor	*Wind size*	*CType*	*coded differences*
9bits	9bits	1bit	(n1/n2)

(Binary):

000000000 → 000000001 → 0 → 00001 **1st Frame** (When Window size=1,Anchor and Ctype are neglected, This frame determines the low coding category Low coding category = [-3, 3] → n1= (3bits))

000001011	000000110	1	010 001 111 110 000 001	**2nd Frame**
000010010	000000011	0	0011000 0010011 1011011	**3nd Frame**
000010101	000000111	1	101 110 111 010 111 011 010	**4fth Frame**

(Decimal):

The first data frame was for low coding category definition, so the anchor and type coding are neglected. The next data frames will be considered for decompression with a low coding category equal to [-3,3].

11	6 *coded differences*	1 (low)	+2 +1 -1 -2 0 +1
18	3 *coded differences*	0 (high)	+24 +19 -37
21	7 *coded differences*	1 (low)	-3 -2 -1 +2 -1 +3 +2

Decompressed values

11 13 14 13 11 11 12 18 42 61 24 21 18 16 15 17 16 19 21

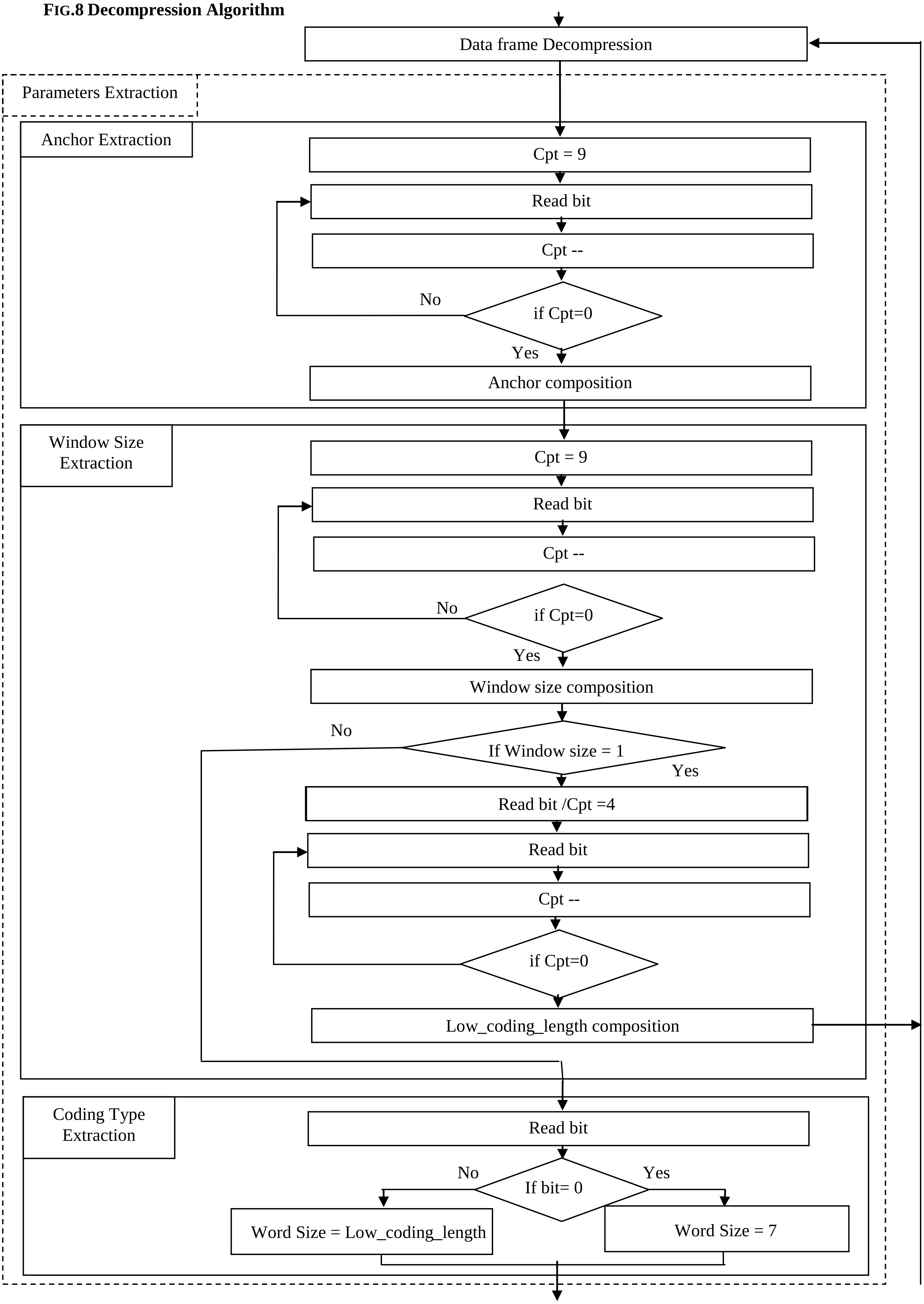
Inverse conditioned signal

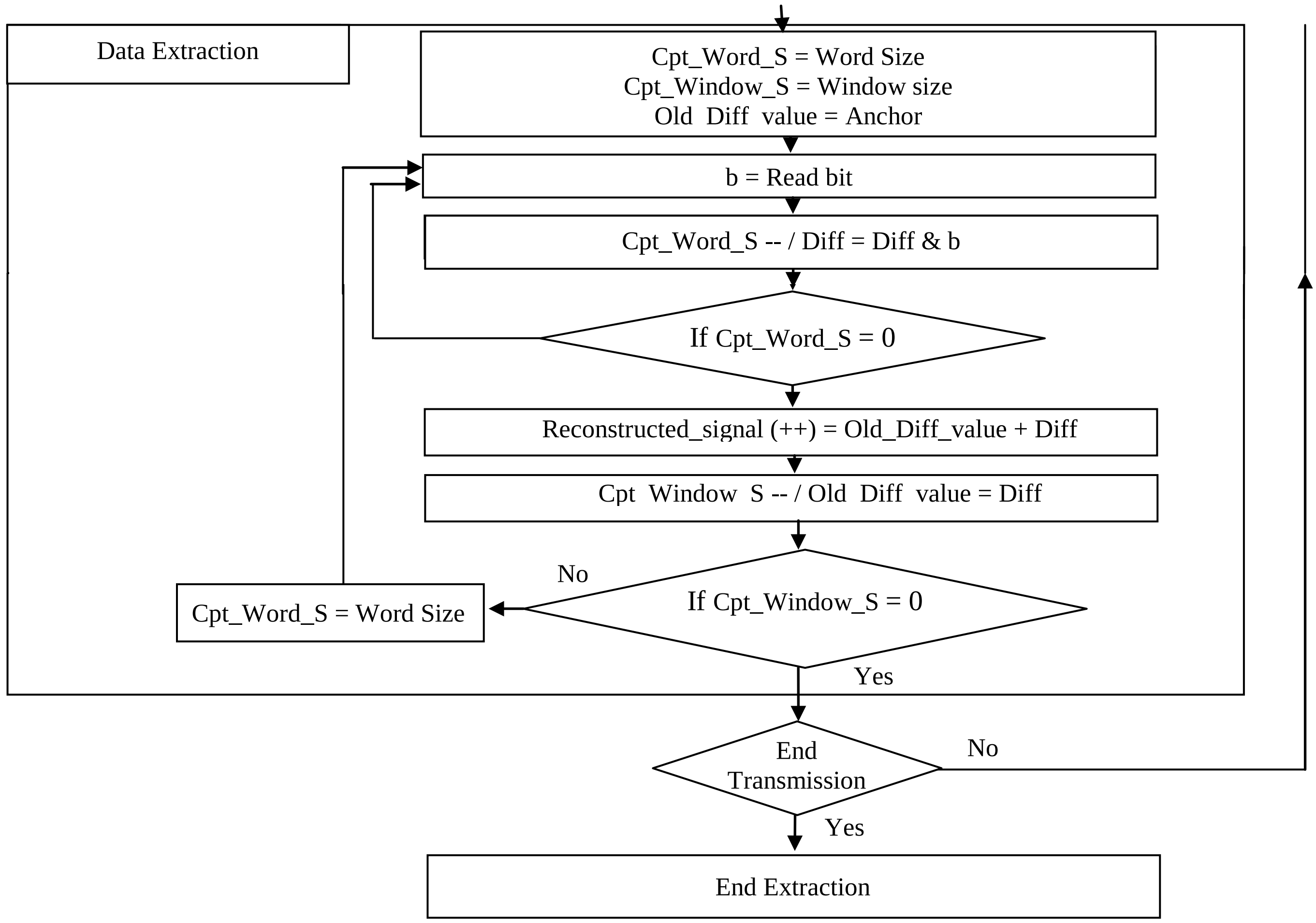
+0.11 + 0.13 +0.14 +0.13 +0.11 +0.12 +0.18 +0.42 +0.61 +0.24 +0.21 +0.18 +0.16 +0.15 +0.17 +0.16 +0.19 +0.21

Reconstructed unconditioned signal (Gain =200, Offset =1024)

1072 1052 1050 1046 1048 1060 1108 1146 1072 1066 1060 1056 1054 1058 1056 1062 1066

FIG.8 Decompression Algorithm





IV.e TESTING RESULTS

The proofs of concept are based on MATLAB software. The efficiency of the introduced compression scheme is measured by the following performance metrics: compression ratio (CR), percentage root mean squared difference (PRD). They are defined as follows:

$$CR = \frac{\text{Input data file size}}{\text{Compressed data file size}} \quad (Eq\ 14)$$

$$PRD(\%) = \sqrt{\frac{\sum(x(n)-\hat{x}(n))^2}{\sum \hat{x}(n)^2}} \times 100 \quad (Eq\ 15)$$

Where $\hat{x}(n)$ is the reconstructed signal

Multiple signals from MIT-BIH and PTB diagnostic databases from physionet with different pathologies are used for testing. The compression ratios CR and PRD obtained using the static and dynamic variants are presented in table 4. The proposed compression scheme achieves important compression ratios with a very low PRD rates. It reaches a mean compression ratio of 37.46 and 40.82 respectively for the static and dynamic low coding. The variant with the minimum window size achieves a mean compression ratio of 42,13. The mean PRD is the same for the different variants and is equal to 0.5%. Table 5 presents a comparison with different state of the art techniques. It’s obvious that our compression algorithm outperforms the state of the art techniques in term of accuracy and efficiency. Figure 9 presents some original and reconstructed signals plotted in the same figure. We notice that no significant distortion is introduced between the original and reconstructed signals. The proposed ECG compression/decompression algorithm is simple but very efficient. Its complexity is very low and uses basic operations like subtractions and tests. In addition there no need of huge memory resources. In a hardware point of view this elements are the key to build high performance architectures. In contrary other techniques like transformation based ECG compression algorithms use complex operations

like multiplications and divisions. These operations slow down the systems and generally are found in the worst paths**.**
All these advantages make the proposed algorithm very efficient and adapted for wireless e-health monitoring with high real time constraints.

Table 4 Performance analysis of the proposed static and dynamic variants of the proposed algorithm

Signal	***S1***				***S2***			
	Low coding threshold			PRD%	**Low coding threshold**			PRD%
	Static	***Dynamic***			***Static***	***Dynamic***		
	CR	***CR***	***Low coding category***		***CR***	***CR***	***Low coding category***	
100m.mat	46	46	[-3,3]	0.075	47	47	[-3,3]	0.073
105m.mat	41	41	[-3,3]	0.071	43	43	[-3,3]	0.068
119m.mat	35	38	[-7,7]	0.082	39	39	[-3,3]	0.076
122mmat	33	38	[-7,7]	0.083	11	30	[-7,7]	0.076
203m.mat	23	29	[-15,15]	0.071	25	35	[-7,7]	0.069
217m.mat	29	34	[-7,7]	0.071	35	38	[-7,7]	0.068
231m.mat	42	42	[-3,3]	0.071	45	45	[-3,3]	0.068
233m.mat	30	35	[-7,7]	0.071	33	35	[-7,7]	0.069
08730_01m.mat	40	40	[-3,3]	1.7	47	47	[-3,3]	1.7
08730_04m.mat	23	35	[-15,15]	0.57	44	44	[-3,3]	1.2
08730_06m.mat	38	40	[-7,7]	1.1	47	47	[-3,3]	1.1
11442_01m.mat	43	43	[-3,3]	1.9	47	47	[-3,3]	1.13
11950_03m.mat	45	45	[-3,3]	1.2	44	44	[-3,3]	1.4
12247_01m.mat	30	34	[-7,7]	0.667	36	37	[-7,7]	0.086
12247_02m.mat	35	35	[-3,3]	0.739	40	40	[-3,3]	0.110
13005_01m.mat	40	40	[-3,3]	0.981	43	43	[-3,3]	0.124
Mean Static Variant		**CR = 37.46**			**PRD% = 0.5%**			
Mean Dynamic Variant		**CR = 40.82**						

Table 5 Comparison with state of the art techniques

SIGNAL		**CR**	**PRD%**
Mukh *et al* [16]		15.72	7,89
Mukh *et al* [17]		22.47	7.58
Fira *et al*[18]		18.27	1,17
Ku *et al* [19]		21.60	4,02
Kim *et al*[20]		16.90	0,64
R. Gupta *et al* [21]		17.03	4,60
R. Kumar *et al* [23]		24.25	1,89
S.Padhy *et al* [24]		19.34	3,05
A.Singh *et al* [25]		17.76	5,60
Y.S.Ding *et al* [26]		24.95	9,77
		22.00	0,90
		29.10	3,20
Proposed Algorithm	**Static Low coding**	**37. 46**	**0.50**
	Dynamic Low coding	**40,82**	
	Dynamic with minimum window size	**42.13**	

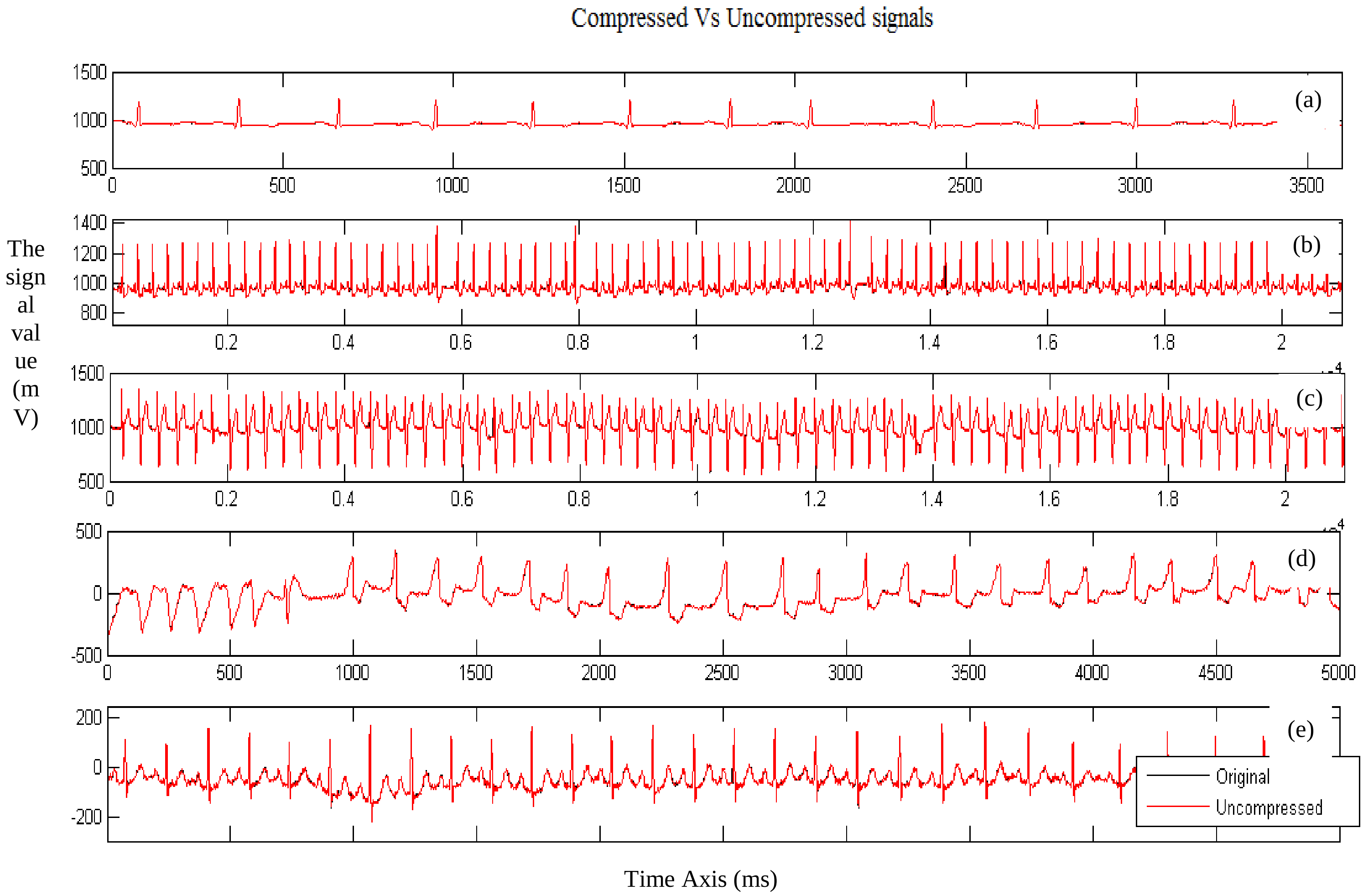


Fig.9 Compressed Vs decompressed of some test signals. (a) '100m.mat', (b) '105m.mat', (c) '231m.mat', (d) '1442_01m.mat', (e) '12247_01m.mat'

Acknowledgements

Conflict of Interest

The authors declare that they have no conflict of interest.

Ethical approval

This article does not contain any studies with human participants or animals performed by any of the authors

V CONCLUSION

A new ECG signal compression algorithm based on delta coding is introduced in this paper. An efficient technique with three variants is developed to minimize the delta differences coding bits. Category coding was developed (Low and High). Successive differences are calculated based on an anchor value while they can be coded with the same category coding. This allows optimizing the bits needed to compress the signal. We defined a new data frame format with different fields. While a change between the two categories a new frame is composed, and costs 19 bits in addition to the compressed data. The number of successive data is coded by the window size field and can reach 512. A static preprocessing stage is added before compressing. The first variant of the algorithm uses a fixed low coding category, the second uses dynamic low coding category and the third uses a window size with a minimum value. All these features make the new algorithm very efficient. It can reach high compression ratios with a minimal PRD. The algorithm has been verified using different normal and pathological types of cardiac signals from MIT-BIH and PTB diagnostic data bases from physionet. The algorithm can reach a mean compression ratio of 37.46 , 40.82 and 42.13 respectively for the static , dynamic and minimal window size algorithm variants with a PRD of 0.5%. The algorithm is suitable for real time e-health monitoring.

References

[1] M. L. Hilton. Wavelet and wavelet packet compression of electrocardiograms. IEEE Trans. Biomed. Eng., vol. 44, no. 5, pp. 394–402, May 1997.

[2] R. S. H. Istepanian and A. A. Petrosian. Optimal zonal wavelet-based ECG data compression for a mobile telecardiology system. IEEE Trans.Inf. Technol. Biomed. vol. 4, no. 3, pp. 200–211, Sep. 2000.

[3] C. C. Gutiérrez Rodríguez, M.Riveill. e-Health monitoring applications: What about Data Quality? . Proceedings of the Health Ambiant Information Systems Workshop, HamIS 2011, Gramado, Brazil, May 17, 2011.

[4] M.Srijani; D.Koustabh; D.Soumya Kanti. Patient health management system using e-health monitoring architecture. IACC 2014, 4th IEEE International Advance Computing Conference, February 21 22, 2014, Gurgaon, India.

[5] Yang, Z., Zhou, Q., Lei, L. et al. An IoT-cloud Based Wearable ECG Monitoring System for Smart Healthcare . J Med Syst (2016) 40: 286. doi:10.1007/s10916-016-0644-9.

[6] Raeiatibanadkooki, M., Quchani, S.R., KhalilZade, M. et al. Compression and Encryption of ECG Signal Using Wavelet and Chaotically Huffman Code in Telemedicine Application . J Med Syst (2016) 40: 73. doi:10.1007/s10916-016-0433-5.

[7] Anas Hatim, Said Belkouch, Mohamed El Aakif, Moha M'rabet Hassani & Noureddine Chabini. Design optimization of the quantization and a pipelined 2D-DCT for real-time applications. Multimed Tools Appl (2013) 67:667-685 DOI 10.1007/s11042-012-1043-y

[8] Anas Hatim, Said Belkouch, Abderrahim Benslimane, Moha M'Rabet Hassani & Tayeb Sadiki. Efficient architecture for direct 8 × 8 2D DCT computations with earlier zigzag ordering. Multimed Tools Appl DOI 10.1007/s11042-015-2562-0

[9] Alvaro Alesanco, Salvador Olmos. Enhanced Real-Time ECG Coder for PacketizedTelecardiology Applications. IEEE TRANSACTIONS ON INFORMATION TECHNOLOGY IN BIOMEDICINE, VOL. 10, NO. 2, APRIL 2006.

[10] W.C. Mueller . Arrhythmia detection software for an ambulatory ECG monitor. Biomedical Sciences Instrumentation 14 (1978) 81–85.

[11] C.T. Ku, H.S. Wang, K.C. Hung, Y.S. Hung. A novel ECG data compression method based on non-recursive discrete periodized wavelet transform. IEEE Transactions on Biomedical Engineering 53 (12) (2006) 2577–2583.

[12] S.G. Miaou, H.L. Yen, C.L. Lin. Wavelet-based ECG compression using dynamic vector quantization with tree codevectors in single codebook. IEEE Transactions on Biomedical Engineering 49 (7) (2002) 671–680.

[13] L.V. Batista, E.U.K. Melcher, L.C. Carvalho. Compression of ECG signals by optimized quantization of discrete cosine transform coefficients. Medical Engineering & Physics 23 (2001) 127–134.

[14] H.A.M. Al-Nashash. A dynamic Fourier series for the compression of ECG using FFT and adaptive coefficient estimation. Medical Engineering and Physics 17 (3) (1995) 197–203.

[15] Lin, KH. & Wu, JH. Low-Cost High-Accuracy QRS Detection for Body Area Network Applications. J. Med. Biol. Eng. (2016) 36: 810. doi:10.1007/s40846-016-0189-x

[16] S.K. Mukhopadhyay, S. Mitra and M. Mitra. ECG signal compression using ASCII character encoding and transmission via SMS. Biomed Sig. Proc Control, Vol. 8, No. 4, pp. 354–363, July 2013

[17] S. K. Mukhopadhyay, S. Mitra, and M. Mitra. An ECG signal compression technique using ASCII character encoding. Measurement, vol. 45, 2012 , pp. 1651–1660.

[18] C.T. Ku, K.C. Hung, H.S.Wang and Y.S. Hung. High efficient ECG compression based on reversible round off non-recursive 1-D discretevperiodized wavelet transform. Med Eng Physics, Vol. 29, No. 10,pp. 1149-1166, December 2010.

[19] H. Kim, R.F.Yazicioglu, P. Merken, C. Van Hoof and Y. H. Jun. ECG signal compression and classification algorithm with quad level vector for ECG holter system. IEEE Trans. on Information Tech in Biomed, Vol. 14 , No. 1, pp. 93 – 100, Jan 2010

[20] Catalina Monica Fira_ and Liviu Goras. An ECG Signals Compression Method and ItsValidation Using NNs. IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING, VOL. 55, NO. 4, APRIL 2008

[21] R. Gupta and M.Mitra. Wireless Electrocardiogram transmission in ISM band: an approach towards telecardiology. Journal of Medical Systems (Springer), Vol. 38, No. 10, Article 90, October 2014, pp. 1-14.

[22] Cho, GY., Lee, SJ. & Lee, TR. An optimized compression algorithm for real-time ECG datatransmission in wireless network of medical information systems. Gyoun J Med Syst (2015) 39: 161. doi:10.1007/s10916-014-0161-7

[23] R. Kumar, A. Kumar, G.K. Singh (2016). Hybrid Method based on Singular Value Decomposition and Embedded Zero Tree Wavelet Technique for ECG Signal Compression, Computer Methods and Programs in Biomedicine (2016), http://dx.doi.org/10.1016/j.cmpb.2016.01.006

[24] Sibasankar Padhy, L.N. Sharma, S. Dandapat (2015). Multilead ECG data compression using SVD in multiresolution domain. Biomedical Signal Processing and Control http://dx.doi.org/10.1016/j.bspc.2015.06.012

[25] Anurag Singh, L.N. Sharma and S. Dandapat (2016). Multi-channel ECG data compression using compressed sensing in eigenspace, Computers in Biology and Medicine, http://dx.doi.org/10.1016/j.compbiomed.2016.03.021

[26] Yi (Steven) Ding, Zeljko Zilic (2016). ECG compression for mobile sensor platforms. IEEE 13th International Conference on Wearable and Implantable Body Sensor Networks (BSN), DOI 10.1109/BSN.2016.7516240

[27] Fathi, A. & Faraji-kheirabadi, F. SIViP (2016) ECG compression method based on adaptive quantization of main wavelet packet subbands 10: 1433. doi:10.1007/s11760-016-0944-z

[28] Rajankar, S. & Talbar, S. SIViP (2016) A quality-on-demand electrocardiogram signal compression using modified set partitioning in hierarchical tree 10: 1559. doi:10.1007/s11760-016-0971-9.