CrossMark

A quality-on-demand electrocardiogram signal compression using modified set partitioning in hierarchical tree

Supriya Rajankar[1] · Sanjay Talbar[2]

Abstract Biomedical signals enfold much crucial clinical information. Cardiac imperfection includes information on the morphology of its electrical signals. These signals are classically recorded over a considerable period, so the size of data file becomes bulky and hence compression is essential. This paper focuses on the implementation of electrocardiogram signal compression using wavelet-based progressive coding such as set partitioning in hierarchical tree and its modified version to achieve improvement in the speed at low bit rate. We obtained compression ratio up to 22:1 for MIT-BIH arrhythmia database record number 117 with a percent mean square difference of 0.9 and 0.73 % using orthogonal and biorthogonal wavelets, respectively. The coders accomplish bit rate control and produce a bit stream that is progressive in quality. It facilitates the user to trim the bit stream at desired point and make required quality restoration for the reduced file size with user-defined compression ratio or bit rate.

1 Introduction

An electrocardiogram is the main physiological signal that shows graphically the cardiovascular activity of the heart. ECG acts as a tool to diagnose cardiac arrhythmias or disorders of the heart. Cardiac arrhythmia is the disorders or

✉ Supriya Rajankar
supriya.rajankar@gmail.com

1 Sinhgad College of Engineering, Pune 411041, India

2 Shri Guru Gobind Singhji Institute of Engineering and Technology, Nanded 431606, India

changes in morphological features of an ECG. Through pattern recognition, cardiologists extract information from the ECG and make the diagnosis. For investigation of patients' heart condition, ECG needs to record over an extended period. This procedure gathers enough information that increases the volume of ECG data [1]. The size of ECG data grows with the increase in channels, sampling frequency, recording time and sample resolution. For example, 24-h recording of ECG signal with 360-Hz sampling frequency, and 11-bits sample resolution produce a data of nearly 40.8 Mbytes per channel. Thus, it requires large storage capacity and transmission bandwidth. For many applications such as ambulatory recording systems and telemedicine, an efficient signal compression technique is required.

Lossless and lossy are the two types of data compression methods. It is possible to achieve perfect reconstruction of the signal with lossless methods, but the compression ratio achieved is low, so it is not suitable for the increasing task of compression and transmission of the signals. Frequently lossy compression techniques are in use with non-noticeable degradation in the signal quality to avoid false diagnosis. Thus, the primary goal of ECG compression is to remove redundancy in the signal while preserving the features required for diagnosis. The broad categories of ECG signal compression methods are direct data compression (DDC) and transform domain techniques. DDC methods utilize inter-sample correlation to eliminate signal redundancy. It is achieved using prediction or interpolation or variable length coders. Some of the techniques presented in [2] are Amplitude Zone Time Epoch Coding (AZTEC), Co-ordinate Reduction Time Encoding System (CORTES), Turning Point (TP) and Scan along Polynomial Approximation SAPA/FAN [3]. The transform domain methods use different orthogonal transforms and provide lossy ECG compression [4,5]. The energy compaction and de-

🍀 Springer

correlation are the main properties of any transform. The design of a transform for compression should reduce the redundancy between the samples or pixels. Discrete Fourier transform (DFT) [7], Karhunen Loeve transform (KLT) [8], discrete cosine transform (DCT) [9,10], the subband coding (SBC) [12], discrete wavelet transform (DWT) [11] and discrete wavelet packet transform (DPWT) [6,13] are some of the transforms used for the data compression. Due to energy compaction and good localization in the time-frequency domain, wavelet received considerable attention as an efficient transform [4,15]. Wavelet transform shows excellent compaction of the signal energy in the low-frequency subband that helps in the encoding of the wavelet coefficients with the progressive (embedded) coding method. Embedded coding methods encode the wavelet transform coefficients for progressive transmission [16]. The primary goal of such transmission is to send the most vital information contained in the signal first. Embedded zero-tree wavelet (EZW) [17] and set partitioning in hierarchical trees (SPIHT) [18,19] are the most popular progressive coding algorithms.

In the case of still image compression, SPIHT has shown significant achievement. It exploits inter-bit correlation and correlation among wavelet coefficients in different subbands. SPIHT algorithm proposed by Said and Pearlman shows a very efficient method for lossy to lossless image coding. The wavelet transformed coefficients show an arrangement in a hierarchical manner providing parent–child relationship through subbands [18].

The 2-D SPIHT techniques enlightened in [16,20–22] compress the ECG signal by converting it into the 2-D data array. 2-D conversion requires preprocessing steps such as mean removal, period and, amplitude normalization, period sorting. Also, it requires accurate QRS detection for cutting, aligning of ECG signal. The compression ratio (CR) and corresponding quality achieved are better with 2-D SPIHT, but, at the same time, it introduces computational overheads. Hence, to avoid this Lu et al. [23] suggested the modification of 2-D to 1-D SPIHT algorithm.

This paper presents the modified version of the 1-D SPIHT algorithm (mSPIHT). During SPIHT coding, only the most significant bits in transform coefficients are outputted for later decoding. It helps to decide bit rate or compression with required quality. The mSPIHT algorithm uses the concept of a number of error bits [(truncating error(μ_e)] and absolute zero tree. Hence, at low bit rate to increase speed lower bit planes can be truncated by forming zero trees. It is not required to store the coordinates of these coefficients in the list of insignificant sets (LISs). Also, the decoder is made faster by reducing the multiplication operations in sorting and refinement pass.

The sections in the paper are planned as Sect. 2 talks about 1-D SPIHT algorithm followed by the modified version of SPIHT in Sect. 3. Section 4 describes coding algorithm of

Fig. 1 1-D temporal orientation tree
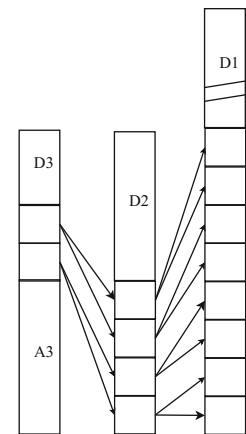
mSPIHT. Section 5 explains the mSPIHT bitstream generation with an illustrative example. The experimentation and results of the designed algorithm on MIT-BIH arrhythmia database are discussed in Sect. 6. Finally, Sect. 7 concludes the paper.

2 1-D SPIHT algorithm

SPIHT is the technique originally designed to encode the images using wavelet transform [18]. 1-D SPIHT proposed by Lu.et al. [23] avoids computational overheads of converting the signal to 2-D. It makes use of spatial similarities between the subbands. In wavelet decomposition, each coefficient in jth level corresponds to two coefficients in the $(j-1)th$ level. It reveals a parent–child relationship between the subbands. The temporal orientation tree structure in Fig. 1 shows this for different subbands with nodes having two children or no children. The maximum energy of the signal is concentrated at low-frequency subband. So, maximum coefficients with higher magnitudes attain the position at roots of the tree. The encoding process of SPIHT algorithm enlightens as follows.

In SPIHT encoding process employs three lists and sets to keep the track of significant coefficients [23].

1. List of insignificant points (LIPs): This is the list of insignificant coefficients.
2. List of the insignificant set (LIS): This list contains a set of coefficients with insignificant tree structures. These coefficients are having magnitudes smaller than the threshold.
3. List of significant points (LSPs): This list contains significant coefficients at the particular threshold in the sorting process.

Following is the tree representations for the sets:
$O(C_i)$: A set of indices of all offspring (children) of the node (i).

$D(C_i)$: A set of indices of all descendants (all nodes that are below) of the node (i). It represents a Type-A entry in LIS. $L(C_i)$: A set indices of grandchildren. This set contains the tree except children. It is simply $L(C_i) = D(C_i) - O(C_i)$. It represents a Type-B entry in LIS.

The SPIHT algorithm uses a recursive sequence of thresholds. The initial threshold is computed by Eq. (1)

$$T_{\max} = 2^{\lfloor \log_2(|C_{\max}|) \rfloor} \tag{1}$$

where C_{max} represents the maximum value of wavelet coefficient from the set of C_i. The significance of the particular coefficient depends on the threshold T as given by equation (2)

$$s_i = \begin{cases} 1, & \text{if } \max_{i \in N} |C_i| \geq T \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

The major steps in SPIHT algorithm are

– sorting pass in LIP,
– sorting pass in LIS,
– refinement pass.

3 Modified SPIHT (mSPIHT)

SPIHT for 2-D and modified SPIHT for 1-D are capable of offering bitstream that provides high CR along with required reconstruction quality of the signal. However, theoretical study and experimental results show that there is still scope for enhancement of it. Considering the morphology of ECG signal sampled at 360-Hz frequency, we can observe P and T waves in A_5 subband and QRS complex in D_5 subband. D_1 and D_2 subbands contain most of the noise. Taking the advantage of this fact when the bit budget is low SPIHT algorithm can be modified to achieve high speed. For this the modifications suggested are:

1. Truncating error bits (μ_e): Modified SPIHT coding outputs only the most significant bits (MSB) of the DWT coefficients. Hence, the least significant bits (LSB) to be truncated are defined before encoding.
2. Absolute zero tree: The low-frequency subband of DWT contains most of the significant coefficients. The magnitudes of these coefficients decrease rapidly toward the lower levels of decomposition. Practically many trees become zero trees before the expected compression ratio. These zero trees occupy space in the LIS of SPIHT and unnecessarily increase the encoding time. A concept of absolute zero trees introduced to solve this issue. If the magnitudes of all descendants of a zero tree are smaller than 2^{μ_e}, it turns into an absolute zero tree and remains insignificant forever. So no need to store them in the LIS. Shorter LIS speed up the encoding process.
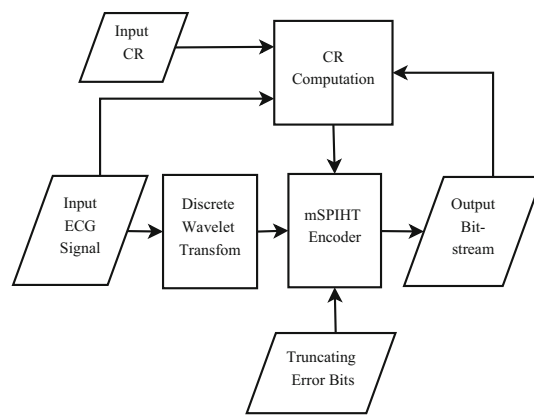

Fig. 2 Block diagram of mSPIHT encoder

3. For the decompression of signal, the original SPIHT decoder uses 3/2 times the threshold in sorting pass and adds or subtracts 1/2 times the threshold depending on the refinement bit in refinement pass. The proposed algorithm uses threshold value in the respective passes and achieves a reduction in computations.

The main feature of the proposed algorithm is the user can decide the required quality or CR of the reconstructed signal and also achieve high speed at low bit rate. During SPIHT coding, only the most significant bits in transform coefficients are outputted for later decoding. Hence, at low bit rate to increase speed lower bit planes can be truncated by forming zero trees. Figure 2 shows a block schematic of the proposed algorithm. With wavelet transform signal is decomposed up to five levels. The input CR and numbers of error bits to be truncated are two user-defined inputs. The inputs to the mSPIHT block are wavelet coefficients and computed CR from the CR computation block. The truncated error bits are given to mSPIHT encoder. The CR computation block compares the size of the encoded bitstream with the original signal and computes the CR. Depending on the comparison between computed and input CR, the algorithm terminates to give the exact bitstream required.

4 Coding algorithm

The MIT-BIH arrhythmia database is having the baseline drift of 1024, so the value 1024 is subtracted from every sample. Using orthogonal $db4$ and biorthogonal $bior4.4$ wavelets, signals are decomposed up to the fifth level. The mSPIHT algorithm steps are:
Step1: Initialization: Choose initial threshold by using Eq. (1). Define a number of truncating error bits. Find the absolute zero tree of the coefficients having a value less than 2^{μ_e}. Set LIP equal to the roots of the tree, i.e., highest detail

coefficient. LIS contains all the nodes, i.e., same nodes as LIP; the only condition is the node must have descendants.

Step 2: Sorting in LIP: For each entry in LIP, if it is significant output one followed by its sign bit and move the entry from LIP to LSP. In case the entry is insignificant, output 0.

Step 3: Sorting in LIS: In the case of Type-A entry, if any entry in LIS is significant output 1. Then check its offspring as a LIP coefficient. If $L(C_i)$ is not empty, the entry made is as Type-B else removes it from LIS. If any entry in LIS is insignificant, output 0. For Type-B entries, if it is significant output 1 and its offspring becomes a Type-A entry. If the entry is insignificant, output 0.

Step 4: Refinement pass: Using the concept of bit plane coding, SPIHT encodes MSB plane first and progresses toward LSB plane as per the defined bit rate. The refinement pass, output the refinement bits of the significant coefficients from the previous entries in LSP by bitwise ANDing with the current threshold.

Step 5: Update and threshold: Threshold is updated using Eq. (3).

$$T_{\text{new}} = \frac{T_{\text{old}}}{2} \tag{3}$$

Repeat steps 2–5 until the achievement of desired compression ratio or bit rate.

5 Example for mSPHIT bitstream generation process

The concept of output bitstream generation in mSPIHT coding is explained with a simple example in Fig. 3. Here, the wavelet coefficients with three-level decomposition are shown with the parent–child relationship. The number in top left corner of each cell represents the index of wavelet coefficient. The truncating error bits considered are two. Accordingly, coefficients with values less than four are becoming zero and form the absolute zero tree. Table 1 explains the coding process for mSPIHT output bitstream generation.

6 Experimentation and results

The proposed progressive coder 1-D mSPIHT is implemented on a personal computer with the hardware configuration: Intel i3 @ 2.1 GHz, 4GB RAM having operating system Windows 7 and software MATLAB R2013. For experimentation, the standard MIT-BIH arrhythmia database [24] is used which contains 48 records with a variety of rhythms and 30-min recording. The sampling frequency of these records is 360 Hz and 11-bit resolution. The signals are decomposed

Fig. 3 Illustrative example
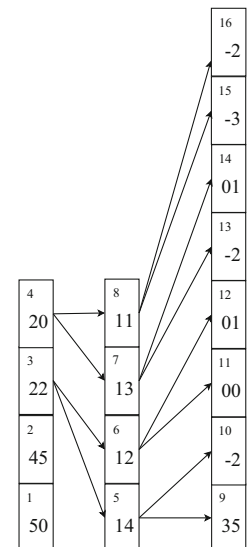
with orthogonal $db4$ and biorthogonal $bior4.4$ wavelet filters up to 5 levels with signal block size 2048. The performance of the proposed algorithms is compared with the state of art 1-D SPIHT algorithm. The studies of progressive coders involve rate-distortion evaluation, and complexity analysis based on encoding and decoding time.

To verify the rate-distortion performance parameters used are bits per samples or compression ratio(CR) and percentage root mean difference (PRD) and they are defined as,

$$CR = \frac{\text{actual size of original signal}}{\text{size of compressed signal}} \tag{4}$$

$$PRD = \sqrt{\frac{\sum_{n=0}^{N-1} \left(x(n) - \hat{x}(n) \right)^2}{\sum_{n=0}^{N-1} x^2(n)}} \times 100 \tag{5}$$

where $x(n)$ and $\hat{x}(n)$ are original and reconstructed signals, respectively. N represents the number of samples. The PRD is the popular parameter used as quality measures in the literature ECG compression. Compression ratio provides the ability of the algorithm to achieve a reduction in the file size, while reconstruction ability signifies faithfulness of the reconstructed ECG to that of the original.

6.1 The rate-distortion performance

The rate-distortion results with $db4$ and $bior4.4$ wavelets for record 100, 117, 210 and average of 48 records of MIT-BIH database are presented in Table 2. The proposed methods show almost same rate-distortion performance. Table 2 reveals that performance of mSPIHT is same as SPIHT at lower bit rates, but degrades as bit rate increases. The effect of truncating error bits for different bit rates in mSPIHT is presented in Fig. 4. At 0.5 bps the PRD remains con-
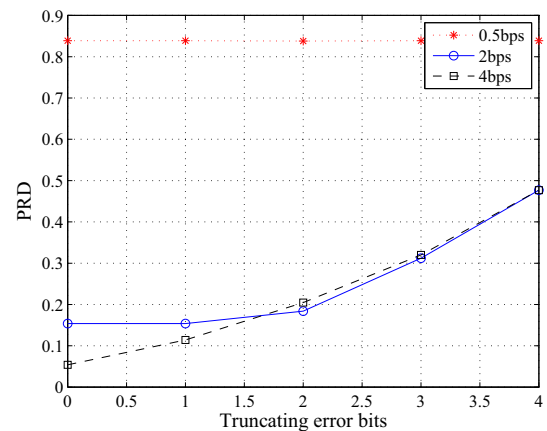
Table 1 mSPIHT coding process

Step	Coefficient/set	Output bitstream
$T = 32$		
LIP	C_1	11
	C_2	10
	C_3	0
	C_4	0
LIS (Type-A)	$D(C_3)$	1
	C_5	0
	C_6	0
	$D(C_4)$	0
LIS (Type-B)	$L(C_3)$	1
LIS (Type-A)	$D(C_5)$	1
	C_9	10
	C_10	0
	$D(C_6)$	0
LIS (Type-B)	$L(C_5)$	0
$T = 16$		
LIP	C_3	0
	C_4	0
LIS (Type-A)	$D(C_3)$	0
	$D(C_4)$	0
Refinement		1
		0
$T = 8$		
LIP	C_3	11
	C_4	10
LIS (Type-A)	$D(C_3)$	1
	C_5	1
	C_6	1
	$D(C_4)$	1
	C_7	1
	C_8	1
LIS (Type-B)	$L(C_3)$	0
	$L(C_4)$	0
Refinement		0
		1

Table 2 Rate-distortion performance for SPIHT and mSPIHT with orthogonal and biorthogonal wavelets at various bit rates

| bps (CR)	Record number	PRD (%)			
		SPIHT		mSPIHT	
		bior4.4	db4	bior4.4	db4
0.5	100	0.74	0.74	0.73	0.74
(22:1)	117	0.74	0.90	0.74	0.90
	210	0.83	0.65	0.83	0.65
Average		1.34	1.53	1.34	1.53
1	100	0.23	0.26	0.23	0.26
(11:1)	117	0.33	0.34	0.33	0.34
	210	0.62	0.64	0.62	0.64
Average		0.56	0.60	0.56	0.60
1.5	100	0.30	0.18	0.30	0.18
(7.33:1)	117	0.20	0.21	0.20	0.21
	210	0.59	0.62	0.59	0.62
Average		0.49	0.51	0.49	0.51
2	100	0.20	0.13	0.30	0.16
(5.5:1)	117	0.15	0.15	0.20	0.18
	210	0.59	0.60	0.60	0.64
Average		0.42	0.43	0.44	0.44
2.5	100	0.10	0.11	0.17	0.18
(4.4:1)	117	0.13	0.13	0.19	0.20
	210	0.58	0.60	0.60	0.66
Average		0.41	0.42	0.44	0.46

Fig. 4 Effect of truncating error bits in mSPIHT algorithm

stant for truncating error bits in the range of 0–4. At the bit rate, 2 bps PRD is constant up to two truncating error bits and increases after that. For higher bit rate 4 bps the effect of truncating error bits is prominent. It shows almost linear increment in the PRD values. The rate-distortion performance with bits per samples and PRD in Fig. 5 explains the usefulness for selecting desired quality, and corresponding bit rate. It also compares the performance of $db4$ and $bior4.4$ wavelets. It shows that up to 2 bps (lower bit rate), the performance of SPIHT and mSPIHT is same. The quality of reconstructed signal degrades for higher bit rate with

mSPIHT. From Table 2 and Fig. 4, it is clear that the biorthogonal wavelet performance is superior to orthogonal wavelet. It may be due to linear phase filter used in biorthogonal wavelets. As per application, the optimum length can be decided from the graph in Fig. 5. For achieving improved quality signal, the bit rate should be more or vice versa.

As biorthogonal wavelets are linear phase filters, the performance of $bior4.4$ is superior to $db4$. The quality of normal
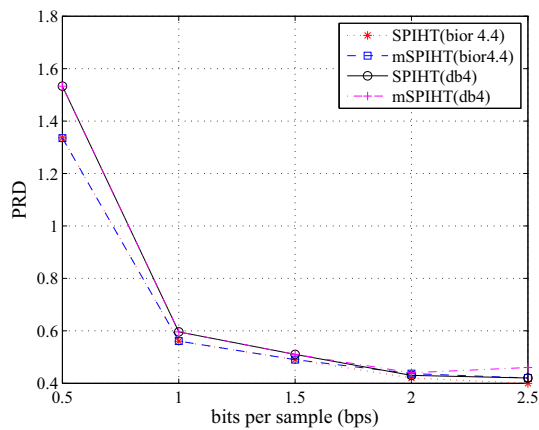
Fig. 5 Bits per sample vs. PRD

and abnormal reconstructed signals is expressed in terms of PRD as in Table 1. Also, from the reconstructed signal, it is possible to depict the visual quality of the signals. The original and reconstructed signals of normal sinus rhythm record 117 for mSPIHT with *bior*4.4 wavelet are presented in Fig. 6. The visual effect of all the reconstructed signals using this algorithm is quite comparable. At bit rate 0.5 bps the signal is reconstructed with PRD of 0.73 % for the compression ratio of 22:1, while for 2.5 bps PRD is 0.193 % with *bior*4.4 wavelet.

The suitability of the algorithm is tested for abnormal ECG signals also. The record 210 is an abnormal signal having ventricular ectopy. Figure 7 presents promising results for the abnormal signals too. At bit rate 0.5 bps the signal is reconstructed with PRD of 0.833 % for the compression ratio of 22:1, while for 2.5 bps PRD is 0.6 % with *bior*4.4 wavelet.

From Fig. 5, it is clear that one can achieve required quality reconstruction as per channel availability or application. It is the beauty of progressive coding method; the user can decide the desired CR-PRD pair. By providing desired CR, the bit stream length suitable for the application can be easily obtained.

6.2 Computational complexity analysis

Usually, computational complexity is described in terms of the encoding and decoding time requirement for the processor. These timings for the SPIHT and mSPIHT at various bit rates (0.5–2.5 bps) are as in Table 3. These timings are shown for sample records 100, 117 and 210 with sample block length 2048, as a representative of the normal and abnormal category. Also, readings in Table 3 show average encoding and decoding time values for 48 records from MIT-BIH arrhythmia database. The average readings show that encoding of mSPIHT is 5–15% faster than SPIHT as bit rate changes from 2.5 to 0.5 bps. At lower bit rate this improvement is more as the number of comparisons is less. So this method
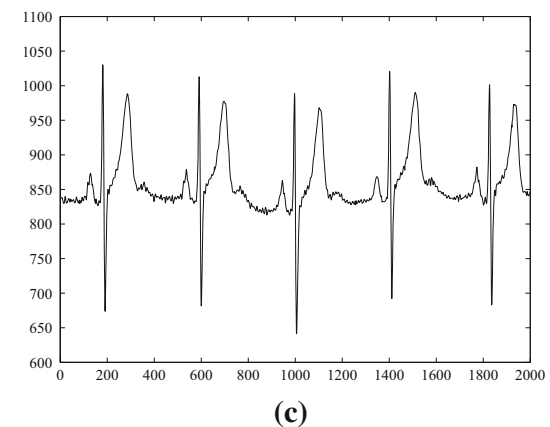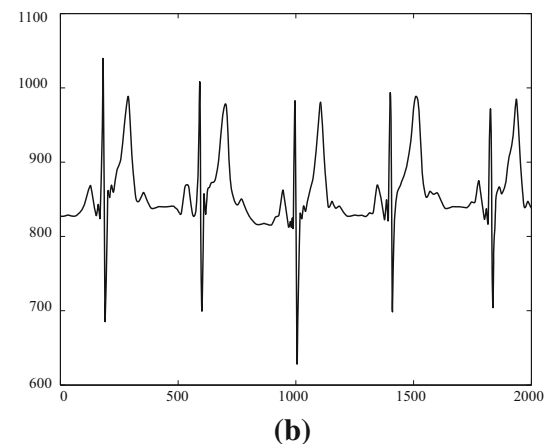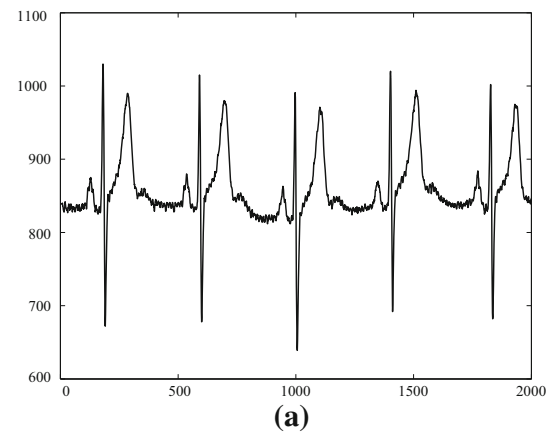
Fig. 6 Original and reconstructed signals at bit rate 0.5 bps and 2.5 bps for record 117 with *bior*4.4 wavelet, **a** original signal 117, **b** reconstructed signal with 0.5 bps, CR = 22:1 and PRD = 0.73, **c** reconstructed signal with 2.5 bps, CR = 4.4:1 and PRD = 0.13

is a solution for high speed at lower bit rate. Due to modification suggested in Sect. 3, the decoding time required is also less in the mSPIHT method. Almost for all bit rates, the total time required is less compared to SPIHT. Furthermore, the visual effect of all the reconstructed signals using these algorithms is quite comparable. The encoding, decoding time required for SPIHT and mSPIHT algorithms with *db*4 and
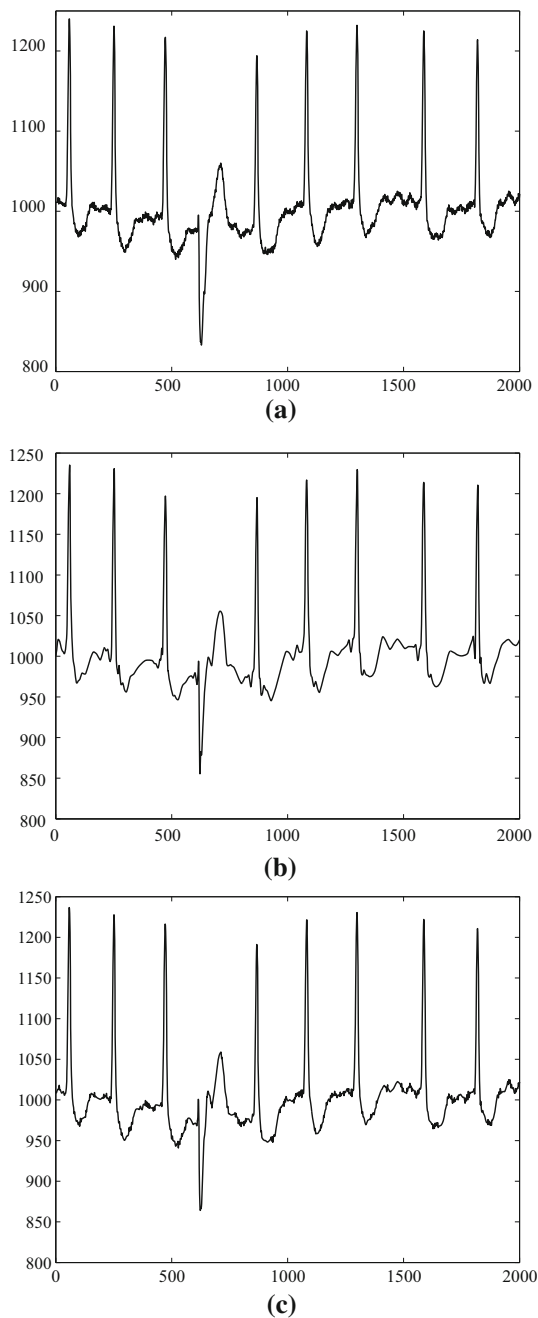
Fig. 7 Original and reconstructed signals at bit rate 0.5 and 2.5 bps for record 210 with *bior*4.4 wavelet, **a** original signal 210, **b** reconstructed signal with 0.5 bps, CR = 22:1 and PRD = 0.83, **c** reconstructed signal with 2.5 bps, CR = 22:1 and PRD = 0.58

*bior*4.4 wavelets is almost same. Hence, these timings for only *bior*4.4 wavelet are mentioned in Table 3.

6.3 Comparison of proposed algorithm with standard codecs

The comparison of proposed mSPIHT algorithm with related coders in the literature based on CR and PRD for record 117

Table 3 Computational complexity analysis for SPIHT and mSPIHT at various bit rates

bps (CR)	Record number	SPIHT		mSPIHT	
		Encode time(s)	Decode time(s)	Encode time(s)	Decode time(s)
0.5	100	0.06	0.04	0.06	0.03
(22:1)	117	0.06	0.03	0.05	0.02
	210	0.07	0.03	0.06	0.03
Average		0.06	0.03	0.05	0.02
1	100	0.10	0.03	0.10	0.03
(11:1)	117	0.10	0.03	0.10	0.03
	210	0.10	0.03	0.09	0.03
Average		0.10	0.04	0.09	0.03
1.5	100	0.14	0.10	0.14	0.04
(7.33:1)	117	0.14	0.05	0.13	0.04
	210	0.14	0.05	0.13	0.04
Average		0.14	0.07	0.13	0.05
2	100	0.16	0.05	0.14	0.07
(5.5:1)	117	0.19	0.07	0.15	0.08
	210	0.16	0.05	0.15	0.09
Average		0.17	0.16	0.16	0.05
2.5	100	0.20	0.06	0.14	0.70
(4.4:1)	117	0.20	0.09	0.15	0.07
	210	0.20	0.06	0.15	0.06
Average		0.20	0.09	0.16	0.05

Table 4 Comparison of CR and PRD for record 117

Methods	CR	PRD (%)
EZW [25]	16	1.1
SPIHT [23]	8	2.5
Hilton [14]	8	2.6
2D-SPIHT [21]	8	1.45
Proposed mSPIHT	8	0.23
Proposed mSPIHT	22	0.738

is presented in Table 4. It shows that the proposed algorithm provides superior reconstruction signal quality for desired compression ratio.

7 Conclusion

This paper proposed 1-D SPIHT and its modified version mSPIHT for ECG signal compression. mSPIHT has revealed the best combination of compression ratio and signal quality at the time of reconstruction. Considering the morphology of ECG signal and available bit budget, it is possible to achieve low bandwidth and low bit rate signal transmission with improvement in speed with the proposed mSPIHT algorithm. Also, the decoder is made faster by reducing the multiplica-

tion operations in sorting and refinement pass. High speed, high efficiency and straightforwardness in implementation can make the algorithm better contender for applications like telemedicine, wearable devices and ECG storage.

References

1. Cetin, A.E., Koymen, H., Aydin, M.C.: Multichannel ECG data compression by multirate signal processing and transform domain coding techniques. IEEE Trans. Biomed. Eng. **40**(5), 495–499 (1993)
2. Jalaleddine, S.M.S., Hutchens, C.G., Strattan, R.D., Coberly, W.A.: ECG data compression techniques-a unified approach. IEEE Trans. Biomed. Eng. **37**(4), 329–343 (1990)
3. Tompkins, W.J.: Biomedical Digital Signal Processing. Springer-Verlag, Berlin, Heidelberg (2000)
4. Sabarimalai, M., Dandapat, S.: Wavelet-based electrocardiogram signal compression methods and their performances: a prospective review, Biomed. Signal Process. Control **14**(1), 73–107 (2014)
5. Abo-Zahhad, M., Al-Ajlouni, A.F., Ahmed, S.M., Schilling, R.J.: A new algorithm for the compression of ECG signals based on mother wavelet parameterization and best-threshold levels selection. Digit. Signal Process. A Rev. J. **23**(3), 1002–1011 (2013). doi:10.1007/s11760-016-0944-z
6. Abdolhossein, Fathi., Fatemeh, Faraji-kheirabadi: ECG compression method based on adaptive quantization of main wavelet packet subbands. Signal Image Video Process. 1–8 (2016)
7. Rajankar, S., Talbar, S.: An optimized transform for ECG signal compression. ACEEE Int. J. on Signal and Image Process. **01**(03), 1–4 (2010)
8. Olmos, S., Laguna, P.: Multi-lead ECG data compression with orthogonal expansions: KLT and wavelet packets. In: Computers in cardiology, vol. 26 (Cat. No.99CH37004), pp. 539–542, (1999)
9. Bendifallah, A., Benzid, R., Boulemden, M.: Improved ECG compression method using discrete cosine transform. Electron. Lett. **47**(2), 87 (2011)
10. Khorrami, H., Moavenian, M.: A comparative study of DWT, CWT and DCT transformations in ECG arrhythmias classification. Expert Syst. Appl. **37**(8), 5751–5757 (2010)
11. Cetin, A. E., Tewfik, A. H., YardimciTewfi, Y.: Coding of ecg signals by wavelet transform extrema, in time-frequency and time-scale analysis, 1994. In: Proceedings of the IEEE-SP international symposium, pp. 544–547, (1994)
12. Blanco-Velasco, M., Cruz-Roldn, F., Bravo-Santos, A., Martnez-Muoz, D.: A low computational complexity algorithm for ECG signal compression. Med. Eng. Phys. **26**(7), 553–568 (2004)
13. Ahmed, S.M., Al-shrouf, A., Abo-zahhad, M.: ECG data compression using optimal non-orthogonal wavelet transform. Med. Eng. Phys. **22**(1), 39–46 (2000)
14. Hilton, M.L.: Wavelet and wavelet packet compression of electrocardiograms. IEEE Trans. Biomed. Eng. **44**(5), 394–402 (1997)
15. Mallat, S.: A wavelet tour of signal processing, Climate Change 2013—The Physical Science Basis, Intergovernmental Panel on Climate Change, Ed. Cambridge: Cambridge University Press, pp. 1–30, (2013)
16. Wang, Z., Zhu, P., Chen, Y.: A 2-D ECG compression algorithm based on modified SPIHT. In Proc. 5th Int. workshop on wearable and implantable body sensor networks, BSN2008, in conjunction with the 5th Int. summer school and symp. on medical devices and biosensors, ISSS-MDBS 2008, pp. 305–309, (2008)
17. Shapiro, J.M.: Embedded image coding using zerotrees of wavelet coefficients. IEEE Trans. Signal Process. **41**(12), 3445–3462 (1993)
18. Said, A., Pearlman, W.A.: A new, fast, and efficient image codec based on set partitioning in hierarchical trees. IEEE Trans. Circuits Syst. Video Technol. **6**(3), 243–250 (1996)
19. Pooyan, M., Taheri, A., Moazami-goudarzi, M., Saboori, I., Introduction, A.: Wavelet compression of ECG signals using SPIHT algorithm. World Acad. Sci. Eng. Technol. **2**(3), 212–215 (2005)
20. Tai, S.C., Sun, C.C., Yan, W.C.: A 2-D ECG compression method based on wavelet transform and modified SPIHT. IEEE Trans. Biomed. Eng. **52**(6), 999–1008 (2005)
21. Sahraeian, S. M. E., Fatemizadeh, E.: Wavelet-based 2-D ECG data compression method using SPIHT and VQ coding, EUROCON 2007—Int. Conf. Comput. as a Tool, pp. 133–137, (2007)
22. Islam, K.R., Abedin, M.A., Akter, M., Deb, R.: High speed ECG image compression using modified SPIHT. Int. J. Comput. Electr. Eng. **3**(3), 1–5 (2011)
23. Lu, Z., Kim, D.Y., Pearlman, W.A.: Wavelet compression of ECG signals by the set partitioning in hierarchical trees algorithm. IEEE Trans. Biomed. Eng. **47**(7), 849–856 (2000)
24. Moody, G.B.B., Mark, R.G.G.: The impact of the MIT-BIH arrhythmia database. IEEE Eng. Med. Biol. Mag. **20**(3), 45–50 (2001)
25. Ktata, S., Mahjoubi, H.: A zerotree coding for compression of ECG signal using EZW and SPIHT. In: IECON 2012—38th annual conference on IEEE industrial electronics society, pp. 1458–1464, (2012)