ISSN 2229-5518

Efficient Algorithm for ECG Coding

Ms. Manjari Sharma, Dr. A. K. Wadhwani

Abstract— Electrocardiogram (ECG) data compression algorithm is needed to reduce the amount of data to be transmitted, stored and analyzed, without losing the clinical information content. This work investigates a set of ECG data compression schemes in frequency domain to compare their performances in compressing ECG signals. These schemes are based on transform methods such as discrete cosine transform (DCT), fast fourier transform (FFT), discrete sine transform (DST), and their improvements. An improvement of a discrete cosine transform (DCT)-based method for electrocardiogram (ECG) compression is also presented as DCT-II. A comparative study of performance of different transforms is made in terms of Compression Ratio (CR) and Percent root mean square difference (PRD).The appropriate use of a block based DCT associated to a uniform scalar dead zone quantiser and arithmetic coding show very good results, confirming that the proposed strategy exhibits competitive performances compared with the most popular compressors used for ECG compression. Each specific transform is applied to a pre-selected data segment from the MIT-BIH database and then compression is performed.

Index Terms—Compression, Compression ratio, Cosine transform, ECG, Fourier transform, Frequency domain techniques, PRD, Time domain techniques.

---

1 INTRODUCTION

CODING is useful because it helps reduce the consumption of expensive resources, such as hard disk space or transmission bandwidth. On the downside, compressed data must be decompressed to be used and this extra processing may be detrimental to some applications. Basically, a data coding algorithm seeks to minimize the number of code bits stored by reducing the redundancy present in the original signal. The design of data compression schemes therefore involves trade-offs among various factors including the degree of compression, the amount of distortion introduced (if using a lossy compression scheme) and the computational resources required to compress and uncompress the data. To deal with the huge amount of electrocardiogram (ECG) data for analysis, storage and transmission; an efficient ECG compression technique is needed to reduce the amount of data as much as possible while pre-serving the clinical significant signal for cardiac diagnosis, for analysis of ECG signal for various parameters such as heart rate, QRS-width, etc. Then the various parameters and the compressed signal can be transmitted with less channel capacity. Compression connotes the process of starting with a source of data in digital form (usually either a data stream or a stored file) and creating a representation that uses fewer bits than the original [1]. An effective data compression scheme for ECG signal is required in many practical applications such as ECG data storage, ambulatory recording systems and ECG data transmission over telephone line or digital telecommunication network for telemedicine. The main goal of any compression technique is to achieve maximum data volume reduction while preserving the significant features [2] and also detecting and eliminating redundancies in a given data set. Data compression methods can be classified into two categories: 1) lossless and 2) lossy coding methods. Lossy compression is useful where a certain amount of error is acceptable for increased compression performance. Loss less or information preserving compression is used in the storage of medical or legal records. In lossless data compression, the signal samples are considered to be realizations of a random variable or a random process and the entropy of the source signal determines the lowest compression ratio that can be achieved. In lossless coding the original signal can be perfectly reconstructed. For typical biomedical signals lossless (reversible) compression methods can only achieve Compression Ratios (CR) in the order of 2 to 1. On the other hand lossy (irreversible) techniques may produce CR results in the order of 10 to 1. In lossy methods, there is some kind of quantization of the input data which leads to higher CR results at the expense of reversibility. But this may be acceptable as long as no clinically significant degradation is introduced to the encoded signal. The CR levels of 2 to 1 are too low for most practical applications. Therefore, lossy coding methods which introduce small reconstruction errors are preferred in practice. In this paper we review the lossy biomedical data compression methods.

Biomedical signals can be compressed in time domain, frequency domain, or time-frequency domain. ECG data compression algorithms have been mainly classified into three major categories [3]: 1) Direct time-domain techniques, e.g., turning point (TP), amplitude-zone-time epoch coding (AZTEC) [4], coordinate reduction time encoding system (CORTES) and Fan algorithm. 2) Transformational approaches [3], e.g., discrete cosines transformation (DCT), fast fourier transform (FFT), discrete

- *Manjari Sharma is currently pursuing masters degree program in Measurement and Control in Electrical Engineering Department, MITS, Gwalior, India, PH-08103780811. E-mail: manjari.sharma1@gmail.com*
- *Dr. A. K. Wadhwani is currently Professor in Electrical Engineering Department, MITS, Gwalior, India, PH-09425308846. E-mail: wadhwani_arun@rediffmail.com*

sine transform (DST), wavelet transform (WT) etc. 3) Parameter extraction techniques, e.g., Prediction and Vector Quantization (VQ) methods [2]. The time domain techniques which are based on direct methods were the earlier approaches to biomedical signal compression. Transform Coding (TC) is the most important frequency-domain digital waveform compression method.

When we compare these methods we find that direct data compression is a time domain compression algorithm which directly analyses samples where inter-beat and, intra-beat correlation is exploited. These algorithms suffer from sensitiveness to sampling rate, quantization levels and high frequency interference. It fails to achieve high data rate along with preservation of clinical information [5]. In Transform based technique [6] compressions are accomplished by applying an invertible orthogonal transform to the signal. Due to its de-correlation and energy compaction properties the transform based methods achieve better compression ratios [7]. In transform coding, knowledge of the application is used to choose information to discard, thereby lowering its bandwidth. The remaining information can then be compressed via a variety of methods. When the output is decoded, the result may not be identical to the original input, but is expected to be close enough for the purpose of the application. In parameter extraction methods a set of model parameters/features are extracted from the original signal (model based) which involves methods like Linear term prediction (LTP) and analysis by synthesis.

In this paper different transforms like DCT, FFT, DST and are studied for ECG signal compression. A threshold based algorithm is proposed to achieve better compression as DCT-II.

2 Necessity For ECG Compression

Compression of electrocardiography (ECG) is necessary for efficient storage and transmission of the digitized ECG signals. A typical ECG monitoring device generates a large amount of data in the continuous long-term (24-48 hours) ambulatory monitoring tasks. For good diagnostic quality, up to 12 different streams of data may be obtained from various sensors placed on the patient's body. The sampling rates of ECG signals are from 125Hz to 500Hz, and each data sample may be digitized into 8 to 12 bits binary number. Even with one sensor at the lowest sampling rate of 125 Hz and 8-bit encoding, it generates data at a rate of 7.5KB per minute and 450KB per hour. For a sampling rate of 500Hz and 12-bit encoding recording, it generates data at a rate of 540KB per minute and 30MB per hour. The data rate from 12 different sensors totally will generate 12 times amount of data and it is enormously big. Besides, recording over a period of time as long as 24 hours maybe needed for a patient with irregular heart rhythms . The monitor device such as Holter must have a memory capacity of about 400-800 MB for a 12-lead recording, but such a big memory cost may render a solid-state commercial Holter device impossible. Thus, efficient ECG data compression to dramatically reduce the data storage capacity is a necessary solution. On the other hand, it makes possible to transmit ECG data over a telephone line from one cardiac doctor to another cardiac doctor to get opinions.

3 Evaluation Criteria

Before studying the various algorithms it is important to understand the criteria on the basis of which results are compared. The evaluation of performance for testing ECG compression algorithms includes three components: compression efficiency, reconstruction error and computational complexity. The compression efficiency is given by compression ratio (CR). The compression ratio and the reconstruction error are usually dependent on each other. The computational complexity component is part of the practical implementation consideration [8], [9]. In present work we have tested the data on the basis of Compression ratio CR and Percent root mean square difference PRD.

3.1 Compression Ratio

This is one of the most important parameters in data compression algorithms which specifies the amount of compression. A large value of this ratio shows success of any algorithm. Compression Ratio (CR) is defined as the ratio between the rates of the compressed signal (in terms of bits per second) to the rate of the original signal. A data compression algorithm must also represent the data with acceptable fidelity while achieving high CR, given by (1).

$$CR = \frac{Number\ of\ samples\ before\ compression}{Number\ of\ samples\ after\ compression} \tag{1}$$

3.2 Distortion Measurement

One of the most difficult problems in ECG compression applications and reconstruction is defining the error criterion. The purpose of the compression system is to remove redundancy and irrelevant information. Consequently the error criterion has to be defined so that it will measure the ability of the reconstructed signal to preserve the relevant information. Since ECG signals generally are compressed with lossy compression algorithms, we have to have a way of quantifying the difference between the original and the reconstructed signal, often called distortion. Different objective error measures namely; root mean square error (RMSE), percentage root mean difference (PRD), signal to noise ratio (SNR) are used for calculation of reconstruction error. The most prominently used distortion measure is the Percent Root mean square Difference (PRD) [10] that is given by (2)

$$PRD = \sqrt{\frac{\sum_{n=1}^{L_b}[x(n)-x\prime(n)]^2}{\sum_{n=1}^{L_b}[x(n)]^2}} \quad (2)$$

where x(n) is the original signal, x'(n) is the reconstructed signal and L_b is the length of the block or sequence over which PRD is calculated. PRD provides a numerical measure of the residual root mean square (rms) error.

4 The Frequency Domain Techniques

In transform coding, knowledge of the application is used to choose information to discard, thereby lowering its bandwidth. The remaining information can then be compressed via a variety of methods. When the output is decoded, the result may not be identical to the original input, but is expected to be close enough for the purpose of the application. Transform Coding (TC) is the most important frequency-domain digital waveform compression method. The key is to divide the signal into frequency components and judiciously allocate bits in the frequency domain. In most TC methods, the input signal is first divided into blocks of data and each block is linearly transformed into the frequency domain. Transformation methods involve processing of the input signal by a linear orthogonal transformation and encoding of output using an appropriate error criterion. For signal reconstruction an inverse transformation is carried out and the signal is recovered with some error. Thus transformation techniques involve preprocessing the input signal by means of linear orthogonal transformation and properly encoding the transformed output (expansion coefficients) and reducing the amount of data needed to adequately represent the original signal. Various orthogonal transformations include Karhunen-Loeve Transform (KLT), Discrete Cosine Transform (DCT), Fast Fourier Transform (FFT), Discrete Sine Transform (DST), and WAVELET transforms etc.

In this work we have compared the performance of four different frequency domain transformation methods for ECG compression and then their performance is evaluated. The various compression techniques have been discussed below:

4.1 Discrete Cosine Transform (DCT)

The Discrete Cosine Transform (DCT) was developed to approximate Karhunen-Loeve Transform (KLT) when there is high correlation among the input samples, which is the case in many digital waveforms including speech, music, and biomedical signals.

The DCT v= $[v_0\, v_1 \ldots\ldots\ldots\ldots v_{N-1}]^T$ of the vector x is defined as follows:

$$v_0 = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x_n \quad (3)$$

$$v_k = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} x_n \cos\frac{(2n+1)k\pi}{2N} \quad , k=1, 2,\ldots\ldots, (N-1) \quad (4)$$

where v_k is the *kth* DCT coefficient. The inverse discrete cosine transform (IDCT) of v is given by (5).

$$x_n = \frac{1}{\sqrt{N}} v_0 + \sqrt{\frac{2}{N}} \sum_{k=1}^{N-1} v_k \cos\frac{(2n+1)k\pi}{2N} \quad , n=0, 1, 2,\ldots\ldots (N-1) \quad (5)$$

There exist fast algorithms, Order (*N* log *N*), to compute the DCT .Thus, DCT can be implemented in a computationally efficient manner. Two recent image and video coding standards, JPEG and MPEG, use DCT as the main building block.

A discrete cosine transform (DCT) expresses a sequence of finitely many data points in terms of a sum of cosine functions oscillating at different frequencies [11]. DCTs are important to numerous applications in science and engineering, from lossy compression of audio (e.g. MP3) and images (e.g. JPEG) (where small high-frequency components can be discarded), to spectral methods for the numerical solution of partial differential equations. The use of cosine rather than sine functions is critical in these applications. For compression, it turns out that cosine functions are much more efficient whereas for differential equations the cosines express a particular choice of boundary conditions [12]. In particular, a DCT is a Fourier-related transform similar to the discrete Fourier transform (DFT), but using only real numbers. DCTs are equivalent to DFTs of roughly twice the length, operating on real data with even symmetry (since the Fourier transform of a real and even function is real and even), where in some variants the input and/or output data are shifted by half a sample.

Discrete Cosine Transform is a basis for many signal and image compression algorithms due to its high decorrelation and energy compaction property [7]. A discrete Cosine Transform of *N* sample is defined as in (6):

$$F(u) = \sqrt{\frac{2}{N}} C(u) \sum_{x=0}^{N-1} f(x) \cos\left[\frac{\pi(2x+1)u}{2N}\right] \quad (6)$$

u=0,1,……..,N-1,

where $C(u) = \frac{1}{\sqrt{2}}$ for $u=0$

=1 otherwise.

The function *f(x)* represents the value of *xth* samples of input signals [7]. *F(u)* represents a DCT coefficients. The inverse DCT is defined in similar fashion as in (7):

$$f(x) = \sqrt{\frac{2}{N}} \sum_{u=0}^{N-1} C(u) F(u) \cos\left[\frac{\pi(2x+1)u}{2N}\right] \quad (7)$$

x=0, 1,… N-1.

4.2 Fast Fourier Transform (FFT)

A fast Fourier transform (FFT) [13] is an efficient algorithm to compute the discrete fourier transform (DFT) and it's inverse [14]. There are many distinct FFT algorithms involving a wide range of mathematics, from simple complex-number arithmetic to group theory and number theory. A DFT decomposes a sequence of values into components of different frequencies but computing it directly from the definition is often too slow to be practical. An FFT is a way to compute the same result more quickly. Computing a DFT of *N* points in the naive way, using the definition, takes O(N^2) arithmetical operations [15], while an FFT can compute the same result in only O(*N* log *N*) operations. The difference in speed can be substantial, especially for long data sets where *N* may be in the thousands or millions—in practice, the computation time can be reduced by several orders of magnitude in such cases, and the improvement is roughly proportional to *N* / log (*N*). This huge improvement made many DFT-based algorithms practical; FFTs are of great importance to a wide variety of applications, from digital signal processing and solving partial differential equations to algorithms for quick multiplication of large integers. The most well known FFT algorithms depend upon the factorization of *N*, but there are FFTs with O (*N* log *N*) complexity for all *N*, even for prime *N*. Many FFT algorithms only depend on the fact that is an *Nth* primitive root of unity, and thus can be applied to analogous transforms over any finite field, such as number-theoretic transforms.

Fast Fourier Transform is a fundamental transform in digital signal processing with applications in frequency analysis, signal processing etc [7]. The periodicity and symmetry properties of DFT are useful for compression. The u^{th} FFT coefficient of length *N* sequence {*f(x)*} is defined as in (8):

$$F(u) = \sum_{x=0}^{N-1} f(x) e^{-j2\pi ux/N} \quad (8)$$

u=0,1,….N-1.

And its inverse transform is calculated from (9):

$$f(x) = \frac{1}{N}\sum_{u=0}^{N-1} f(x) e^{j2\pi ux/N} \quad (9)$$

x=0,1,….N-1

4.3 Discrete Sine Transform (DST)

Discrete sine transform (DST) [16] is a Fourier-related transform similar to the discrete Fourier transform (DFT), but using a purely real matrix. It is equivalent to the imaginary parts of a DFT of roughly twice the length, operating on real data with odd symmetry (since the Fourier transform of a real and odd function is imaginary and odd), where in some variants the input and/or output data are shifted by half a sample. Like any Fourier-related transform, discrete sine transforms (DSTs) express a function or a signal in terms of a sum of sinusoids with different frequencies and amplitudes. Like the discrete Fourier transform (DFT), a DST operates on a function at a finite number of discrete data points. The obvious distinction between a DST and a DFT is that the former uses only sine functions, while the latter uses both cosines and sines (in the form of complex exponentials). However, this visible difference is merely a consequence of a deeper distinction: a DST implies different boundary conditions than the DFT or other related transforms [17].

Formally, the discrete sine transform is a linear, invertible function *F*: R^N -> R^N (where R denotes the set of real numbers), or equivalently an *N* × *N* square matrix. There are several variants of the DST with slightly modified definitions. The *N* real numbers $x_0,\ldots, x_{N-1}$ are transformed into the *N* real numbers $X_0, \ldots, X_{N-1}$ according to (10):

$$X_k = \sum_{n=0}^{N-1} x_n sin\left[\frac{\pi}{N+1}(n+1)(k+1)\right] \quad (10)$$

k=0,……N-1

4.4 Discret Cosine Transform-II (DCT-II)

The most common variant of discrete cosine transform is the type-II DCT [18]. The DCT-II is typically defined as a real, orthogonal (unitary), linear transformation by the formula in (11):

$$C_k^{II} = \sqrt{\frac{2-\delta_{k,0}}{N}}\sum_{n=0}^{N-1} x_n\, cos\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)k\right], \quad (11)$$

for *N* inputs x_n and *N* outputs C_k^{II}, where $\delta_{k,0}$ is the Kronecker delta (= 1 for k = 0 and = 0 otherwise).

DCT-II can be viewed as special case of the discrete fourier transform (DFT) with real inputs of certain symmetry [19]. This viewpoint is fruitful because it means that any FFT algorithm for the DFT leads immediately to a corresponding fast algorithm for the DCT-II simply by discarding the redundant operations.

The discrete Fourier transform of size *N* is defined by (12):

$$X_k = \sum_{n=0}^{N-1} x_n \quad (12)$$

where $\omega_N = e^{-2\pi i/N}$ is an *Nth* primitive root of unity. In order to relate this to the DCT-II, it is convenient to choose a different normalization for the latter transform [19] as in (13):

$$C_k = 2\sum_{n=0}^{N-1} x_n\, cos\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)k\right] \quad (13)$$

This normalization is not unitary, but it is more directly related to the DFT and therefore more convenient for the development of algorithms. Of course, any fast algorithm for *Ck* trivially yields a fast algorithm for C_k^{II} although the exact count of required multiplications depends on the normalization. In order to derive C_k from the DFT formula, we can use the identity $2cos(\pi l/N) = \omega_{4N}^{2l} + \omega_{4N}^{4N-2l}$ to write (14):

$$C_k = 2\sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)k\right]$$
$$= \sum_{n=0}^{N-1} x_n\, \omega_{4N}^{(2n+1)k} + \sum_{n=0}^{N-1} x_n\, \omega_{4N}^{(4N-2n-1)k}$$
$$= \sum_{n=0}^{4N-1} \widehat{x_n}\, \omega_{4N}^{nk} \quad (14)$$

where $\widehat{x_n}$ is a real-even sequence of length $4N$, defined as follows for $0 \leq n < N$:

$$\widehat{x_{2n}} = x_{4N-\widehat{(2n+1)}} = x_n \quad (15)$$

Thus, the DCT-II of size N is precisely a DFT of size $4N$, of real-even inputs, where the even-indexed inputs are zero.

5 The Coding Algorithms

The various compression techniques DCT, FFT, DST algorithms are compared with PRD and Compression ratio CR and best suitable is considered. A threshold based algorithm is proposed to achieve better compression as DCT-II. The algorithms are performed on MIT-BIH database shown in Fig. 1.

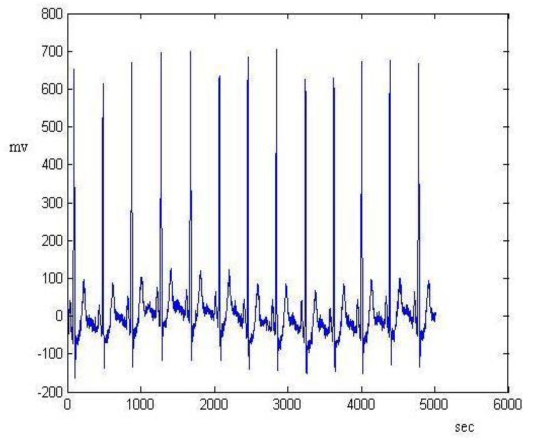

Fig. 1.The ECG signal before coding

5.1 The DCT Coding Algorithm

1. Separate the ECG components into three components *x, y, z*.
2. Find the frequency and time between two samples.
3. Find the DCT of ECG signal and check for DCT coefficients (before compression) =0, increment the counter A if it is between +0.22 to -0.22 and assign to Index=0.
4. Check for DCT coefficients (after compression) =0, increment the Counter B.
5. Calculate inverse DCT and plot decompression, error.
6. Calculate the compression ratio CR and PRD.

The plot for compressed ECG by DCT is shown in Fig. 2.

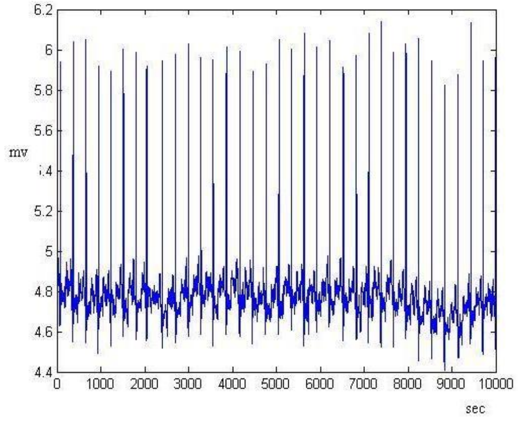

Fig 2. The ECG signal after DCT coding

The error signal after DCT coding is shown in Fig. 3.

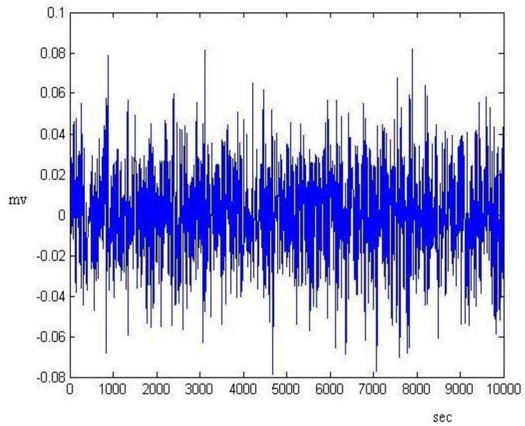

Fig.3. The Error signal after DCT coding

5.2 The FFT Coding Algorithm

1. Separate the ECG components into three components *x, y, z*.
2. Find the frequency and time between two samples.
3. Find the FFT of ECG signal and check for FFT coefficients (before compression) =0, increment the counter A if it is between +25 to-25 and assign to Index=0.
4. Check for FFT coefficients (after compression) =0, increment the Counter B.
5. Calculate inverse FFT and plot decompression, error.
6. Calculate the compression ratio CR and PRD.

ISSN 2229-5518

The plot for compressed ECG by FFT is shown in Fig. 4.

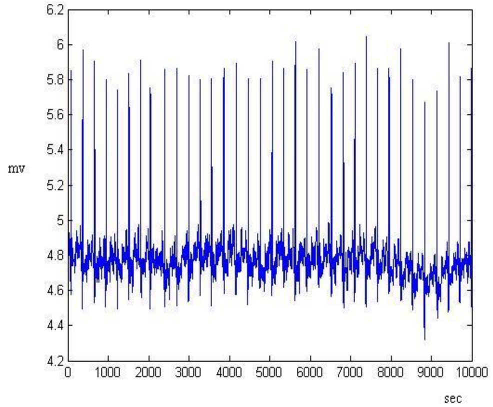

Fig.4. The ECG signal after FFT coding

The error signal after FFT coding is shown in Fig. 5.

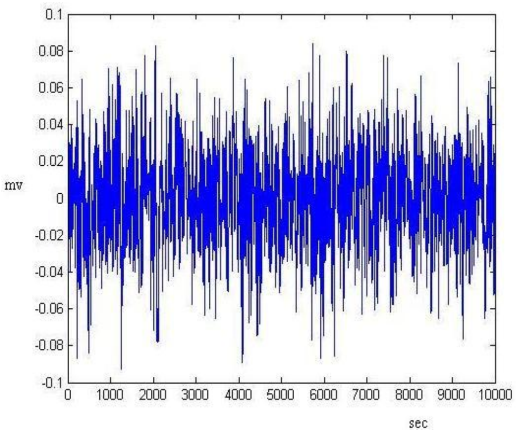

Fig.5. The Error signal after FFT coding

5.3 The DST Coding Algorithm

1. Separate the ECG components into three components *x*, *y*, *z*.
2. Find the frequency and time between two samples.
3. Find the DST of ECG signal and check for DST coefficients (before compression) =0, increment the counter A if it is between +15 to-15 and assign to Index=0.
4. Check for DST coefficients (after compression) =0, increment the Counter B.
5. Calculate inverse DST and plot decompression, error.
6. Calculate the compression ratio CR and PRD.

The plot for compressed ECG by DST is shown in Fig. 6.

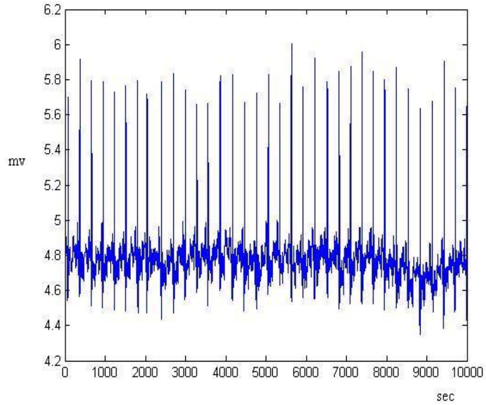

Fig.6. The ECG signal after DST coding

The error signal after DST is shown in Fig. 7.

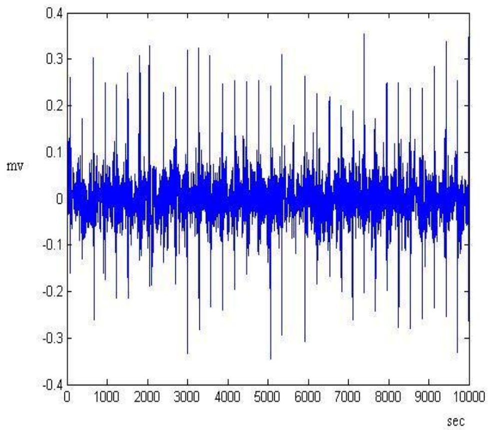

Fig.7. The Error signal after DST coding

5.4 The DCT-II Coding Algorithm

1. Partition of data sequence *x* in N_b consecutive blocks b_i, i =0, 1,…….., N_b-1, each one with L_b samples.
2. DCT computation for each block.
3. Quantization of the DCT coefficients.
4. Lossless encoding of the quantized DCT coefficients.

Increasing the block size increases the CR and the DCT computing time. Various results show that increasing the block size above a certain point results in a very modest CR gain, while the processing time increases. The type II DCT is commonly used for data compression due to its greater capacity to concentrate the signal energy in few transform coefficients.

The algorithm is explained in detail as:

1. Let $b_i[n]$, n=0,1,…,L_b-1,represent the L_b values in block b_i.

2. The one-dimensional DCT-II of this block generates a transformed block B_i constituted by a sequence of L_b coefficients $B_i[m]$, m=0,1,…..L_b-1,given by (16):

$$B_i[m] = \sqrt{\left(\frac{2}{Lb}\right)}\, c_m \sum_{n=0}^{L_b-1} bi\,[n] cos\left[\frac{(2n+1)m\pi}{2L_b}\right] \quad (16)$$

$$m = 0,1, \ldots L_b - 1$$

where c_m=1 for $1 \le m \le L_b$-1 and $c_o=(1/2)^{(1/2)}$.

The DCT can be seen as a one-to-one mapping for N point vectors between the time and the frequency domains. The coefficient $B_i[0]$,which is directly related to the average value of the time-domain block is called DC coefficient and the remaining coefficients of a block are called AC coefficients.

Given B_i, b_i can be recovered by applying the inverse DCT-II:

$$b_i[n] = \sqrt{\left(\frac{2}{Lb}\right)} \sum_{m=0}^{L_b-1} c_m Bi\,[m] cos\left[\frac{(2n+1)m\pi}{2L_b}\right] \quad (17)$$

n=0,1,….L_b-1.

3. To quantize B_i we use quantization vector q. Each element $q[n]$, n=0,1,….L_b-1,of q is a positive integer in a specified interval and represents the quantization step size for the coefficient $B_i[n]$.The elements $\widehat{B_i}[n]$of the quantized DCT block $\widehat{B_i}$obtained by the following operation:

$$\widehat{B_i}[n] = B_i[n]//q[n] \text{ , n=0,1,….}L_b\text{-1} \quad (18)$$

i=0,1,…..N_b-1

where // represents division followed by rounding to the nearest integer.

4. The lossless encoding of the quantized DCT coefficients involves run-length encoding, because the quantization normally generates many null values followed by an entropy encoder.

The plot for compressed ECG by DCT-II is shown in Fig. 8.

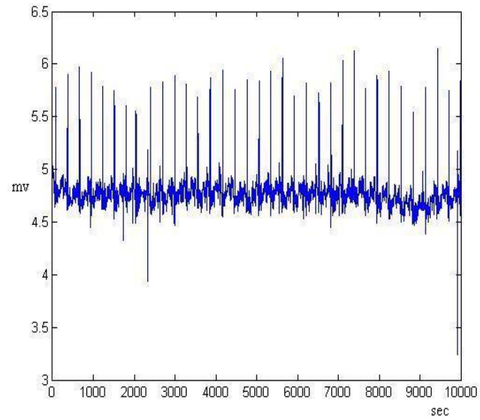

Fig. 8. The ECG signal after DCT-II coding

The error signal after DCT-II coding is shown in Fig. 9.

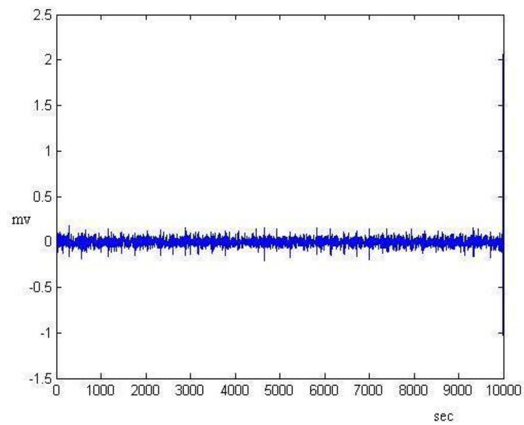

Fig.9. The Error signal after DCT-II coding

6 Results And Discussion

Since signal recording conditions and noise levels vary from study to study, a thorough comparison of the coding methods is very difficult to make. But, frequency-domain coding methods produce higher coding results than time-domain coding methods. We used data in the MIT-BIH database to test the performance of the four coding techniques. The ECG data is sampled at 360Hz and the resolution of each sample is 11 bits/samples that total bit rate is 3960bps. The amount of compression is measured by CR and the distortion between the original and reconstructed signal is measured by PRD. The comparison table shown in Table 1, details the resultant compression techniques. This gives the choice to select the best suitable compression method. A data compression algorithm must represent the data with acceptable fidelity while achieving high CR. As the PRD indicates reconstruction fidelity; the increase in its value is actually undesirable. Although DCT-II provides maximum CR, but distortion is more. So a compromise is made between CR and PRD.

TABLE 1
COMPARISON OF COMPRESSION TECHNIQUES

Method	Compression Ratio	PRD
DCT	91.6800	0.8392
FFT	89.5723	1.0237
DST	70.4073	1.1967
DCT-II	94.28	1.5729

Considering that the number of electrocardiogram records annually numbers in the millions and the use of sending electrocardiogram records over telephone lines for remote analysis is increasing, the need for effective electrocardiogram compression techniques is great. Many existing compression algorithms have shown some success in electrocardiogram compression; however, algorithms that produce better compression ratios and less loss of data in the reconstructed data are needed. This paper has provided an overview of several compression techniques and has formulated new algorithms that should improve compression ratios and lessen error in the reconstructed data. This has allowed comparing various compression schemes and analyzing reconstructed electrocardiogram records through a graphic interface, without detailed knowledge of the mathematics behind the compression algorithm.

6 CONCLUSIONS

Transform based techniques because of their high compression ability have gained popularity. In this paper the preprocessed signal is transformed to get the decorrelated coefficients. The thresholding or quantization of transformed coefficients gives the actual compression, which is lossy one. But it has good performance and low computational cost. Among the four techniques presented, DST provides lowest CR and distortion is also high. FFT improves CR and lowers PRD. So FFT is better choice than DST. Next is DCT which gives higher CR upto 91.68 with PRD as 0.8392. But DCT-II provides an improvement in terms of CR of 94.28 but PRD increases up to 1.5729. Thus an improvement of a discrete cosine transform (DCT)-based method for electrocardiogram (ECG) compression is presented as DCT-II. The appropriate use of a block based DCT-II associated to a uniform scalar dead zone quantiser and arithmetic coding show very good results, confirming that the proposed strategy exhibits competitive performances compared with the most popular compressors used for ECG compression.

ACKNOWLEDGMENT

The authors would like to thank UGC, New Delhi for providing financial support to carry out this research work.

REFERENCES

[1] Pranob K. Charles and Rajendra Prasad K. (2011): A Contemporary Approach For ECG Signal Compression Using Wavelet Transforms. Signal and Image Processing: An International Journal (SIPIJ). Vol. 2, No. 1, 178-183.

[2] Yucel Kocyigit, Mehmet Korurek and Bekir Karlik (1999): ECG Data Compression by Artificial Neural Networks. In *ELEC0'99 International Conference On Electrical And Electronics Enginering.* E01.44 / D-10, 338-339.

[3] S. Jalaleddine, C. Hutchens, R. Stratan, and W. A. Coberly (1990): ECG data compression techniques-*A* unified approach. *IEEE Trans. Biomed. Eng.*, 37, 329-343.

[4] J. R. Cox, F. M. Nolle, H. A. Fozzard and G. C. Oliver (1968), AZTEC: A preprocessing scheme for real time ECG rhythm analysis. *IEEE Tran. Biomed. Eng.,* vol-BME-15, 128-129.

[5] M. Sabarimalai Manikandan and S. Danpat (2006): Wavelet Threshold based ECG Compression using USZZQ and Huffman Coding of DSM. In *Science Direct Biomedical Signal Processing and Control*. 261-270.

[6] B. R. S. Reddy and I. S. N. Murthy (1986): ECG data compression using Fourier descriptors, IEEE Trans. Bio-med. Eng., BME-33, 428-433.

[7] Mrs. S. O. Rajankar and Dr. S. N. Talbar (2010): An Optimized Transform for ECG Signal Compression. In *Proc. Of Int .Conf. on Advances in Computer Science,* 94-96.

[8] Shang-Gang Miaou, Heng-Lin Yen, Chih-Lung Lin (2002): Wavelet based ECG compression using Dynamic vector Quantization with Tree Code vectors in single codebook. In IEEE Transaction on Biomedical Engineering, vol. 49, no. 7, pp. 671-680.

[9] R.shanta selva Kumari, V Sadasivam (2007): A novel algorithm for wavelet based ECG signal coding. *Science Direct Computers and Electrical Engineering*, 33, pp. 186-194.

[10] J. Abenstein and W. Tompkins (1982): A new data-reduction algorithm for real time ECG analysis. IEEE Tran. On Biomed. Engg., 29(BME-1):4, 3-8.

[11] N. Ahmed, T. Natarajan and K. R. Rao (1974): Discrete Cosine Transform. *IEEE Trans. Trans. On Computers.* C-23, 90-93.

[12] K. R. Rao and P. Yip (1990): Discrete cosine transform – algorithms, advantages, applications, San Diego: *Academic Press.*

[13] M. Clausen and U. Baum (1993): Fast Fourier Transforms. *BI-Wiss.-Verl.*

[14] L. Auslander, E. Feig and S. Winograd (1984): Abelian Semi-simple Algebras and Algorithms for the Discrete Fourier Transform. In *Advances in Applied Mathematics*.5, 31-55.

[15] Tinku Acharya and Ajoy K. Roy. Image Processing Principles and Applications. *John Wiley.*

[16] S. Chan and K. Ho (1990): Direct Methods for computing discrete sinu-

ISSN 2229-5518

soidal transforms. *IEEE Proceedings,* 137, 433-442.

[17] G. Steidl and M. Tasche (1991): A Polynomial approach to Fast algorithms for Discrete Fourier –cosine and Fourier-sine Transforms. In Mathematics in Computation, 56 (193), 281-296.

[18] E. Feig and S. Winograd (1992): Fast Algorithms for Discrete Cosine Trnsforms. *IEEE Tran. On Signal Processing.* vol-40(9), pp 2174-2193.

[19] Xuancheng Shao and Steven G. Johnson (May 10, 2007): Type-II/III DCT/DST algorithms with reduced number of arithmetic operations. *Preprint submitted to Elsevier.*