Computational Mechanisms for Exploiting Temporal Redundancies Supporting Multichannel EEG Compression

12

M. S. Sudhakar and Geevarghese Titus

Abstract

Multichannel electroencephalogram (MCEEG) recordings generally result in humongous volume of data that places constraint on space and time. Online transmission of such data demands schemes rendering significant performance with lesser computations. To compact such data, numerous compression algorithms have been introduced in the literature. Heretofore single channel algorithms when extended to multichannel applications do not accomplish remarkable results. If achieved it generally results in higher computational cost. Much of this chapter deals with the development of computationally simple algorithms that aim to reduce the computational aspects without comprising on the compression and decompression performance. Also, the amicability of this implementation supporting efficient storage with low bandwidth is performed. The objective of this chapter is to introduce some of the basic supporting concepts for exploiting data representation redundancies that aid compression. Accordingly, simple and novel compression schemes with its simulation comparison are presented in this chapter. The potency of the direct domain compression model is assessed in terms of compression ratio and reconstruction error between the original and reconstructed dataset. A significant compression with substantially low Percent Root-mean-square Difference (PRD) is accomplished by the novel compression schemes, thereby upholding diagnostic information of EEG for telemedicine applications with higher reconstruction accuracy and reduced computational load.

M. S. Sudhakar (✉)
SENSE, Vellore Institute of Technology, Vellore, India
e-mail: sudhakar.ms@vit.ac.in

G. Titus
Amal Jyothi College of Engineering, Kanjirapally, Kerala, India
e-mail: geevarghesetitus@amaljyothi.ac.in

S. Paul (ed.), *Application of Biomedical Engineering in Neuroscience*,
https://doi.org/10.1007/978-981-13-7142-4_12

Keywords
Compression ratio · Decoding · Encoding · Multichannel EEG · Percent root-mean-square difference

12.1 Introduction

EEG signal is a complex signal with both stochastic and deterministic properties. The redundancies present in the EEG or any biomedical signal can be removed without loss of signal quality and remain the basic principle supporting compression. In this context there are three forms of compression algorithms, namely lossless, lossy, and near lossless. For most critical clinical analysis of EEG, a lossless compression is the best bet, but the degree of compression is too low such that there is no significant reduction in storage space. Furthermore, most of such compression algorithms have complex architecture. The next best choice is to construct a near lossless compression system having higher compression with significant space saving. One critical factor to consider in such algorithms is the quality of the decompressed signal. Different quality parameters need to be defined like PSNR, PRD, and MSE, that will quantify the level of distortion in the decompressed signal. In other words, the features in the EEG should be retained such that any classification system can work properly irrespective of whether the process is automated or manual.

Another perspective of signal compression is by removing unwanted signals like noises and different forms of artifact from the EEG signal and then storing the clean signal. Different parameters like PSNR can be employed to quantify the quality of the denoised signal. Dimensionality reduction techniques fall under another class of method that can reduce the size of the data that needs to be stored. The subsequent section includes discussions on various time domain strategies and transform domain and compressive sensing methods for compressing the biomedical signal. Most of the methods are supported by coding schemes like Arithmetic, Huffman, Lempel-Ziv, and Lempel-Ziv-Welch that are generally classified as lossless coders, and other coders like SPIHT are EBCOT are lossy coders. These coders exploit the redundancies in the data thereby contributing to data compression.

EEG compression is an active research area for more than a decade. Various compression algorithms have been developed that are broadly classified into different groups based on their operational mode to achieve either lossless, lossy or near lossless compression. These are direct and transform mode based compression schemes. In the direct mode the compression is achieved by exploiting redundancies in the temporal or acquired domain itself. Whereas in transform mode, the compression is achieved by transforming the signal using various transforms and exploiting the redundancies in the transformed domain. These generic compression modes assist in data compression and specifically MCEEG compression. To further enhance the prevailing compression methods, they can be combined together resulting in next-generation compression algorithms.

The neural network-based predictor system of [22] defined the context of the EEG signal, based on which the corresponding coefficients are entropy coded to achieve compression. The MCEEG compression scheme of [21] packed EEG data in 2D form and later processed them using 2D Integer Lifting Transform. The produced coefficients were SPIHT coded along with the residue. Bazn-Prieto et al. [3] employed Cosine Modulated Filter Banks to perform domain transformation. The transformed coefficients were later quantized and entropy coded. A MCEEG compression system described in [21] represents MCEEG signal using tensor representation on which 2D wavelet transform is performed followed by uniform quantization and arithmetic coding of residues. This method tries to maintain an upper bound on the distortion level. Dauwels et al. [7] extended the earlier version using various tensor decomposition methods like SVD and PARAFAC along with residual coding to limit the error introduced in the system.

The compression scheme of [20] initially performed PCA on the EEG data, followed by Wavelet Packet Transform (WPT). The resulting high and low frequency coefficients were coded using wavelet-based and fractal coding techniques, respectively. These coded coefficients were finally entropy coded. Finally, genetic algorithm-based optimization technique was employed for enhancing compression performance. Lin et al. [18] combined dimensionality reduction methods such as the PCA with ICA to compress the MCEEG data represented as a tensor, and the resulting coefficients were then SPIHT coded.

The 1.5D compression algorithm of [24] employed 1D discrete wavelet transform on the 2D MCEEG data followed by No List Set Partitioning in Hierarchical Trees (NLSPIHT) coding. Another MCEEG compression algorithm by [4] employed Fast Discrete Cosine Transform (FDCT) followed by lossy coding. A near lossless MCEEG compressor by [11] used DCT and artificial neural network-based coder. Another compression scheme from [10] performed DPCM on the EEG channels that are then clustered using k-nearest neighbor (kNN). The resulting clusters were entropy coded using arithmetic coders.

To achieve compression, the above methods rely on computationally intensive progressive operations, limiting its use in real-time hardware critical applications. Some algorithms like [4] claim to be computationally fast but with compromise in either compression or distortion levels. So, prospective hardware-friendly compression scheme without significant signal deterioration needs to explored, such that the signal can be used in diagnosis or related applications.

12.1.1 MCEEG Compression Methods

In this chapter the concept of introducing and exploiting the data representation redundancies in the data for achieving compression is discussed. The proposed Logarithmic Spatial Pseudo CODEC (LSPC) methodology negates the necessity for domain transformation, for exploring redundancies, hence, leading to faster computations. The methodology was tested on different database using various performance parameters and discussed in subsequent sections. The proposed near lossless compression model illustrated in Fig. 12.1 is a two-stage process where the MCEEG signal is normalized using Translation Logarithmic (TL) block which is subsequently coded in the Integer Fraction Coder (IFC).

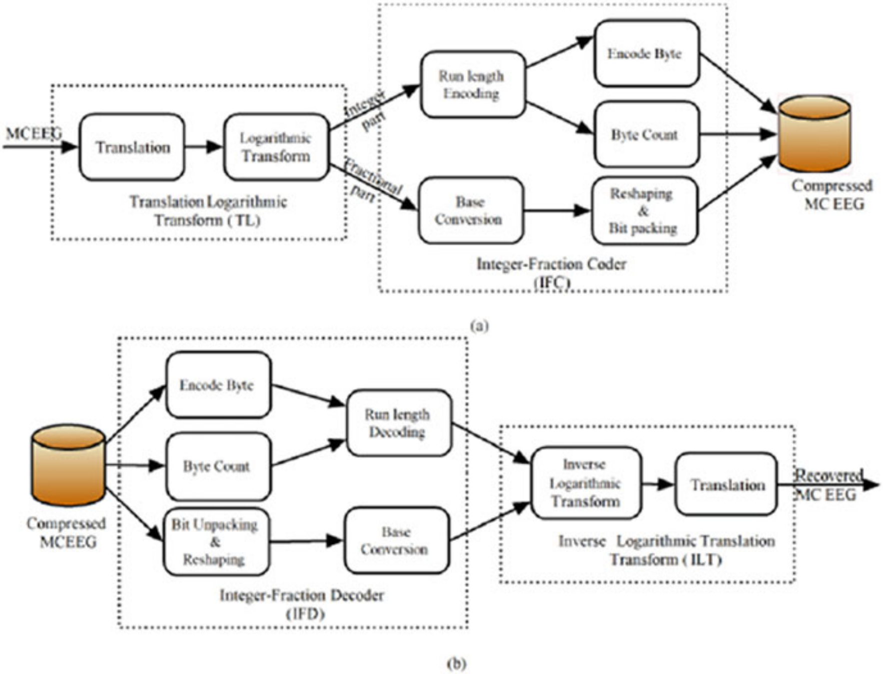

Fig. 12.1 Proposed LPSC system. (**a**) Encoder section. (**b**) Decoder section

12.1.2 Encoding Process

The LSPC encoding process illustrated in Fig. 12.1 a commences with the organization of MCEEG data in 2D form as represented in Eq. 12.1, where M and N correspond to the number of channels and samples per channel, respectively. Each of these sets is processed and stored as frames as illustrated in Fig. 12.2.

$$x = \begin{bmatrix} X_{11} & X_{12} & \ldots & X_{1N} \\ X_{21} & X_{22} & \ldots & X_{2N} \\ . & . & . & \\ . & . & . & \\ X_{M1} & X_{M2} & \ldots & X_{MN} \end{bmatrix} \tag{12.1}$$

To exploit the contribution of all samples and suit them for subsequent processing, the data is pre-processed by the TL block by translation and logarithm transformation. To ascertain the resulting normalized samples to be real and positive,

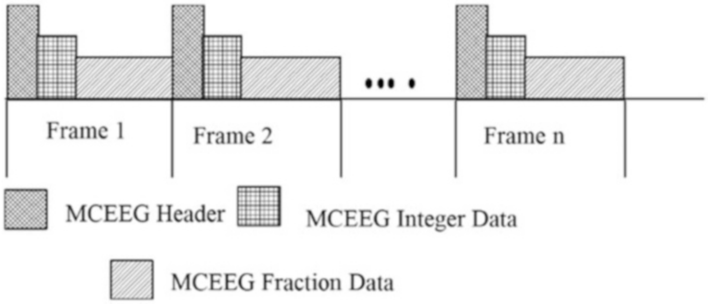

Fig. 12.2 Framing process in MCEEG compression

the MCEEG samples are translated by a factor f, which depends on the gain of the employed MCEEG recorders, and the operation is defined in Eq. 12.2.

$$g(x) = T(x+f) \tag{12.2}$$

Natural logarithm is employed in the LSPC for normalization; while other bases do not contribute to any significant improvement in compression and signal recovery, hence are not discussed here. The normalization process is defined by Eq. 12.3.

$$h(x) = \ln g(x) \tag{12.3}$$

The normalized data $h(x)$ is split into integer $I_r(x)$ and fractional $I_f(x)$ parts using Eqs. 12.4 and 12.5, respectively.

$$I_r(x) = h(x) \tag{12.4}$$

$$I_f(x) = h(x) - I_r(x) \tag{12.5}$$

$I_r(x)$ and $I_f(x)$ are encoded separately using the IFC. Later, $I_r(x)$ can be encoded using any spatial coding schemes that exploit the redundancies introduced by the TL transform. In the current architecture, Run Length Encoding (RLE) of [2] is employed because of its simplicity in implementation. The encoded data is represented as $(D_1, C_1)(D_2, C_2) \ldots (D_i, C_i) \ldots (D_n, C_n)$, where D_i and C_i are the ith distinct integer and their occurrence (count), respectively. As arithmetic and Huffman coders introduce complexity in the system with only a marginal improvement in compression, they are not considered suitable in the proposed architecture.

Subsequently, $I_f(x)$ is encoded to an equivalent representation with error deviation depending on the bit depth d of the base converter. First, the converter converts $I_f(x)$ to its equivalent binary stream b using Eq. 12.6, which is the generalized relation for converting fractions to or from any base.

$$b = a_0 m^{-1} + a_1 m^{-2} + a_2 m^{-3} + \cdots + a_{k+1} m^{-k} \tag{12.6}$$

where m is the base, k is the resolution, and a takes values 0, 1, . . ., $m - 1$. For example, for converting the fraction to binary, m is 2, and a takes value of either 0 or 1.

Data representations for faster encoding and decoding emphasized by [17] include methods like Variable Byte, Byte-Oriented Encoding, Binary Packing, Binary Packing with Variable-Length Blocks, and Patched Coding. Such representations are also useful in size reduction and faster access. A similar packing strategy of the binary stream has been employed in this architecture.

For data representation in the proposed algorithm, the binary stream b from base converter is reshaped and packed into groups of eight bits to form an integer equivalent representation of the fractional data called Pseudo Integers (PI). This representation supplemented with spatial coding technique helps the algorithm to achieve compression.

12.1.3 Decoding Process

The decoder of the SPC system illustrated in Fig. 12.1b takes RLE encoded data and PI as input. This is first processed by the Integer Fractional Decoder (IFD) wherein the RLE data is unpacked by interpolating the coded integer D_i by the parameter C_i to obtain $I_f(x')$. Subsequently, the PI are binarized and reshaped according to the bit depth value used for encoding, which is retrieved from MCEEG header. The reshaped data is converted to fractional base using Eq. 12.6 to obtain $I_f(x')$. Both processes are performed in parallel, thus reducing the decoding time. $I_f(x')$ is added with $I_r(x')$ resulting in the reconstructed normalized signal $I(x')$; this operation is represented in Eq. 12.7.

$$I(x') = I_r(x') + I_f(x') \tag{12.7}$$

Inverse transformation of $I(x')$ is processed by the Inverse Logarithmic Translation (ILT) block where an exponential operation defined by Eq. 12.8 is performed.

$$h(x') = \exp I(x') \tag{12.8}$$

Subsequently, the reconstructed MCEEG signal x' is obtained by translating $h(x')$, given by Eq. 12.9, where factor f is the same as that used in the encoder.

$$x' = T(h(x') - f) \tag{12.9}$$

Finally, the overall encoding and decoding performance of the LSPC on MCEEG signals were observed using the publicly available EEG datasets namely Berlin BCI Competitions [23], Swartz Center for Computational Neuroscience (SCCN) [5], PhysioNet [6], UCI MLR UCI machine learning repository [19], DREAMS sleep spindles database [15], the Bern-Barcelona EEG database [16], European Epilepsy Database [8], Child Mind Institute – Multimodal Resource for Studying Information Processing in the Developing Brain (MIPDB) Database [9], DREAMER Database [1], Stanford Research Data, and Australian Electroencephalography Database (AED) [12].

All channels of the EEG data sets were taken for simulation, though increasing or reducing the number of channels or samples per channel did not have any significant impact on the degree of compression. To evaluate the efficacy of the proposed scheme, few data sets have been chosen, labeled as data set 1 to 8, from each of these standard databases illustrated in Table 12.1.

The database is made up of MCEEG recording of different subjects performing various motor or imagery tasks, subjected to various constraints and stimuli. The performance metrics used in this study are CR, LAE, PAE, MAE, PRD, and PSNR [3] and are computed using Eqs. 12.10, 12.11, 12.12, 12.13, 12.14, and 12.15.

$$CR = \frac{Bits_{\text{orig}}}{Bits_{\text{comp}}} \tag{12.10}$$

Where, *Compression Ratio (CR)* represents the ratio of the number of bits of the uncompressed or original signal to the number of bits of the compressed signal. The terms $Bits_{\text{orig}}$ and $Bits_{\text{comp}}$ correspond to the number of bits of the uncompressed and compressed signal, respectively. Apart from CR, the impact of distortion in the decompressed EEG has a profound effect. Different quality indicators are used that are able to quantify this distortion introduced by the compression system. As a single indicator alone does not effectively quantify the distortions introduced, more than one of such measures and their relations needs to be investigated.

Local Absolute Error (LAE) is the absolute difference between the actual and the reconstructed value and is given in Eq. 12.11.

Table 12.1 MCEEG datasets employed for performance analysis

Dataset	No. of channels	Sampling frequency (in Hz)	Dataset source	Resolution (bits)
1	22	250	BCIC IV Set 2a	16
2	31	500	SCCN	16
3	64	256	UCI MLR	16
4	28	1000	BCIC II Set 4a	16
5	28	100	BCIC II Set 4b	16
6	64	240	BCIC III Set 2	16
7	64	1000	BCIC IV Set 1	16
8	64	80	Physionet	12

BCIC: Brain Computer Interface Competitions
SCCN: Swartz Center for Computational Neuroscience
UCI MLR: UCI Machine Learning Repository

$$LAE = abs\left(X_{\mathrm{ori}}(i) - X_{\mathrm{rec}}(i)\right) \tag{12.11}$$

The terms $X_{\mathrm{ori}}(i)$ and $X_{\mathrm{rec}}(i)$ correspond to the actual data and the decompressed data, respectively.

Peak Absolute Error (PAE) provides an indication of the maximum error occurring in the reconstructed signal or it is the maximum of the absolute difference between the actual and reconstructed value, given in Eq. 12.12.

$$PAE = max\left(abs(X_{\mathrm{ori}}(i) - X_{\mathrm{rec}}(i))\right) \tag{12.12}$$

MeanAbsoluteError(*MAE*) is the mean of the maximum absolute error difference between the actual and reconstructed signal and is given by Eq. 12.13

$$MAE = \frac{max\left(abs(X_{\mathrm{ori}}(i) - X_{\mathrm{rec}}(i))\right)}{N} \tag{12.13}$$

with N representing the data length

Percentage Root mean square Difference (PRD) given in Eq. 12.14 provides information on the amount of error in the decompressed signal.

$$PRD = \sqrt{\frac{\sum \left[X_{\mathrm{ori}}(i) - X_{\mathrm{rec}}(i)\right]^2}{\sum X_{\mathrm{ori}}(i)^2}} \tag{12.14}$$

Root Mean Square Error (RMSE) is the measure of the differences between actual and the decompressed signal or in other words it is the square root of the average of squared errors. It is sometimes referred to as *RootMean Deviation*(*RMSD*). As the effect of the error is proportional to the value of squared error, it can be observed as large variations in RMSD and can be computed using Eq. 12.15.

$$RMSE = \sqrt{\frac{\sum \left[X_{\mathrm{ori}}(i) - X_{\mathrm{rec}}(i)\right]^2}{n}} \tag{12.15}$$

n – Total number of samples

Finally, *Peak Signal to Noise Ratio (PSNR)* is mathematically the ratio between the maximum signal power to the noise power of the recovered signal and is represented in Eq. 12.16.

$$PSNR = 20 * \log \frac{\max\left(X_{\mathrm{ori}}\right)}{RMSE} \tag{12.16}$$

Based on the above metrics, further analysis is presented below. At the onset, the impact of logarithmic normalization is witnessed in Fig. 12.3. From Fig. 12.3b it can be clearly observed that the data is within the range of 3 and 4, with only few values lower than four.

An illustration of the original, reconstructed, and error signal at bit depths of 3, 4, and 5 is shown in Fig. 12.4. The bit depth of five can be considered as an optimal

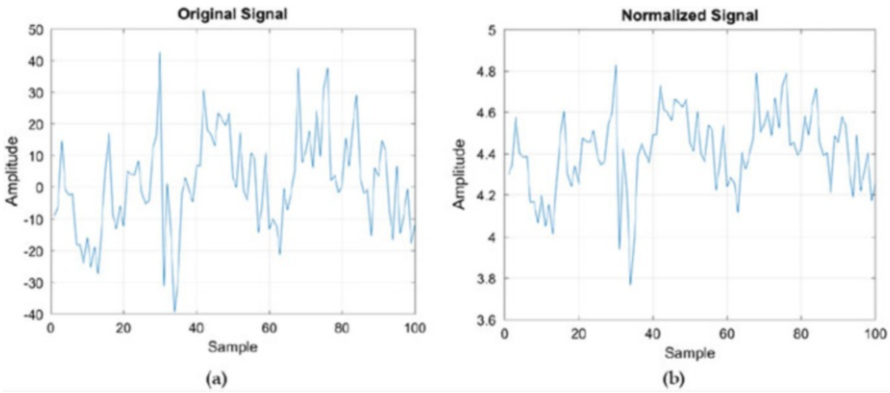

Fig. 12.3 Illustration of logarithmic normalization

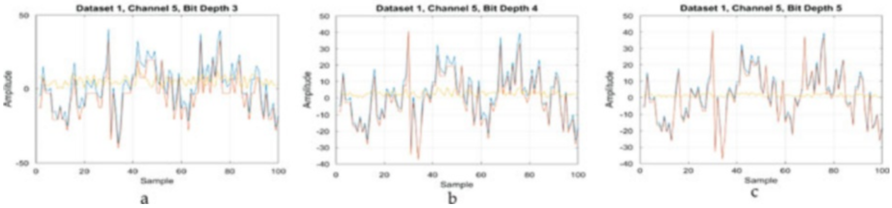

Fig. 12.4 Superimposed original, reconstructed, and residual error signals at a bit depth of 3, 4, and 5

choice as it gave almost similar and relatively good CR and distortion measures for all the data sets, and this finding is validated in subsequent discussions. The coded MCEEG is stored as frames as depicted in Fig. 12.2, having three fields MCEEG Header, Integer Data, and Fraction Data.

Figure 12.5 illustrates the encoding of four samples of 2D MCEEG signals represented by Distinct Integer, Integer Occurrence, and Pseudo Integers. The normalization process is fully reversible. The reconstructed signal quality was visually validated and quantified numerically using the LAE distortion parameter. The maximum error in the reconstructed signal was in the order of 10^{-7} to 10^{-9}.

12.1.4 Inferences from LSPC

The performance of the proposed algorithm for different data sets is illustrated in Figs. 12.6, 12.7, and 12.8. The bit depth *d* of base converter contributes largely to the compression as well as distortion in the recovered signal, as illustrated in Fig. 12.6a–d.

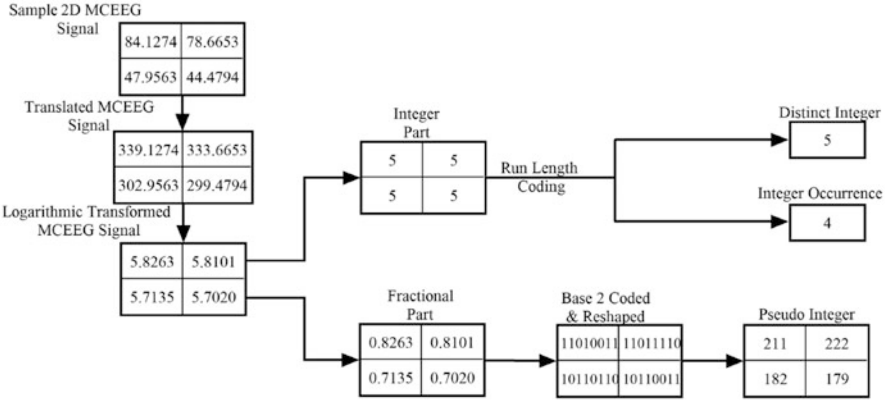

Fig. 12.5 Sample MCEEG encoding process

The lower value of bit depth results in higher compression, with larger distortion in the recovered or decompressed signal and vice versa. Hence, choice of optimal value of bit depth depends on the acceptable amount of distortion in the reconstructed signal. This value was found out with visual scoring along with numerical values quantified by the distortion metrics PRD, RMSE, PAE, MAE, and PSNR as elaborated in the subsequent paragraphs.

It was observed that at higher bit depths distortion is negligible and CR is comparatively less. For example, from Figs. 12.6a, 12.7a, d, CR at a bit depth of eight is nearly two, with PRD value very close to zero, and PSNR above 27 dB. The performance of the algorithm at different sampling frequencies (100 Hz, 1000 Hz) is almost the same as exemplified by the distortion indicators corresponding to data sets 4 and 5 in the performance plots. Another observation is that the resolution of the original data does play a significant role in determining the achievable CR. It can be concluded that higher CR can be achieved at higher resolutions of the ADCs and vice versa. This is a significant observation as currently available ADCs are operating at higher resolutions than those employed in this study.

Distortion performance indicators vary among the data sets at the same bit depth. PAE value indicates the largest difference between the original and reconstructed signal. MAE, on the other hand, indicates the mean error in the reconstructed signal. RMSE value, apart from being an error measure also serves as a performance indicator of the outliers. There was a fourfold increase in the RMSE value and a twofold increase in the PAE and MAE values for a successive reduction in bit depth.

For a bit depth of five, the acceptable average CR of 3.16 occurs at average MAE value of 3.88. Moreover, from Fig. 12.8b, at the average value of MAE, the maximum variation of PAE ranges from 10 to 15.

This provides an upper limit to the sample error at the current CR. Furthermore, average PSNR of 23.07 dB and PRD of 1.39 are observed at the specified CR. Hence, based on the visual scoring and critical inference from Fig. 12.6, 12.7, and 12.8, it can be concluded that the optimal bit depth across different data sets can

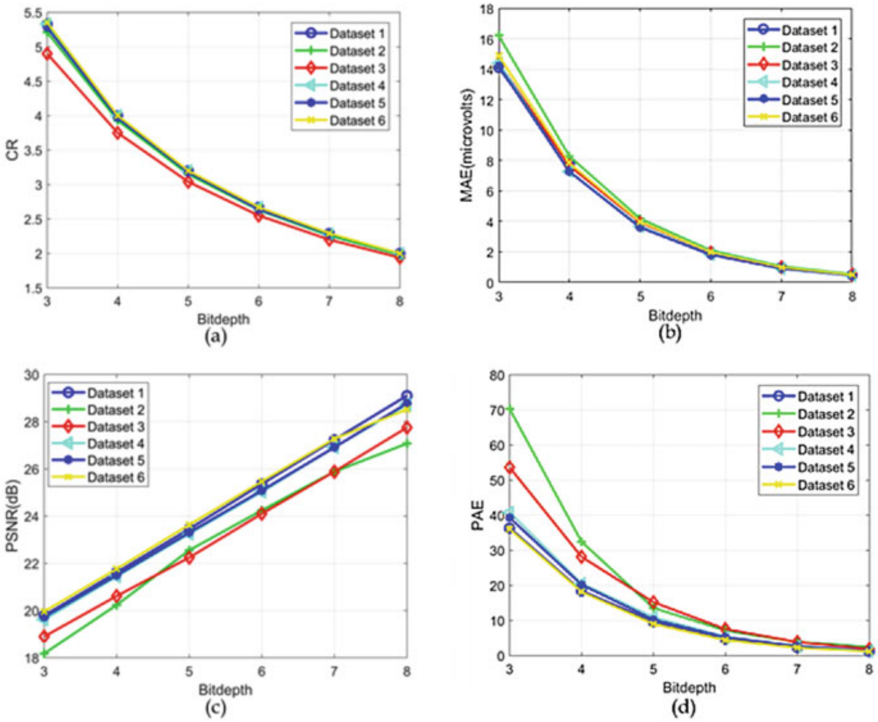

Fig. 12.6 Performance of LSPC on different MCEEG datasets for varying bit depths

be taken as five, though in some data sets lower bit depths can also be considered having similar distortion parameters.

Table 12.2 illustrates the encoding and decoding time of the proposed algorithm for different number of channels and samples per channel. Computations were performed on a computer having an Intel Core2Duo processor operating at 1.8 GHz with 2 GB RAM.

From Table 12.2, the average encoding and decoding times per sample can be computed, which are 0.3 and 0.04 ms respectively. This is a clear indicator that the algorithm is computationally simple and fast when compared to other progressive computation algorithms. Use of public and common data sets is required for relative performance analysis of novel compression algorithms. In this work, data sets 4, 7, and 8 were used for performance comparison as they were the ones used in recent compression algorithms discussed in [21, 4, 10]. The comparison is illustrated in Table 12.3, taking PRD and PSNR as reference quality indicators.

The comparative results signify that the performance of the proposed method in terms of compression and distortion is comparable with recent work but with a much lower time complexity. The main highlight of the proposed algorithm is its simplicity in compressing and decompressing the MCEEG data. Birvinskas et al. [4] claims to be computationally light, but it is not superior to the proposed algorithm, as

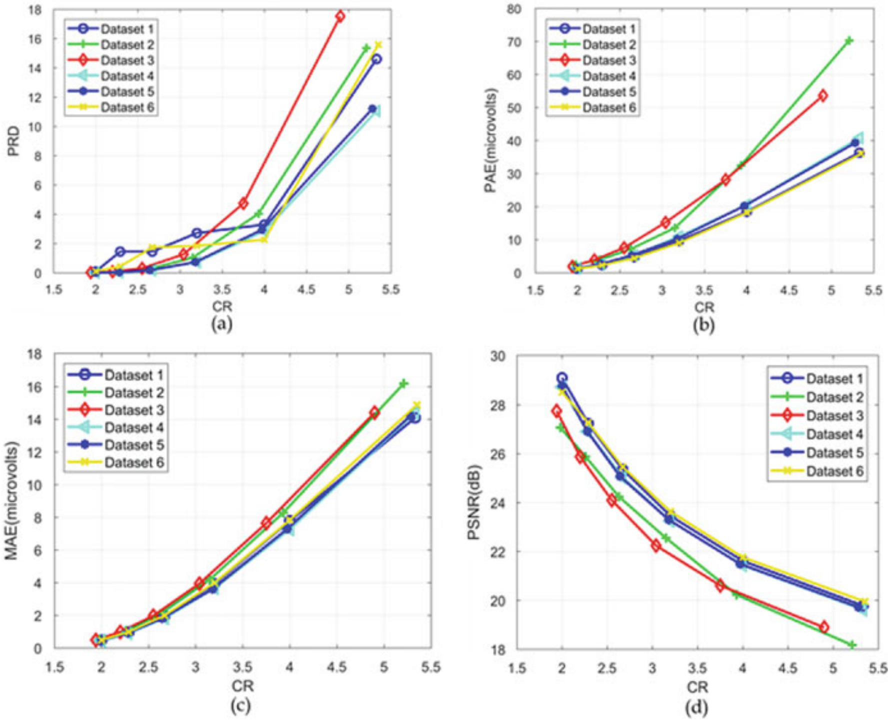

Fig. 12.7 LSPCs performance on different MCEEG datasets for varying CR

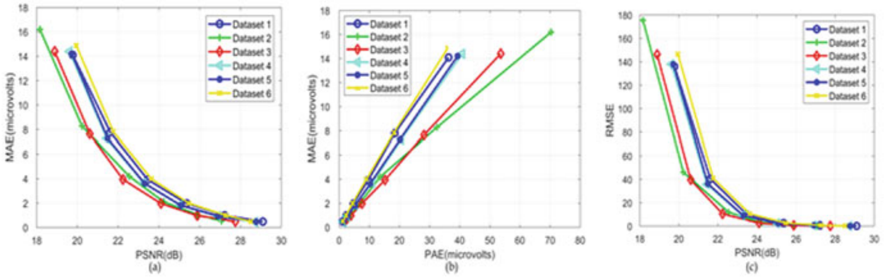

Fig. 12.8 LSPC analyzed using quantitative and qualitative metrics on diverse datasets

observed from the table, in terms of compression and distortion parameters of the reconstructed signal. Furthermore, the proposed algorithm outperforms [10] in terms of CR, with minimal distortion in the decompressed signal.

Extension of the aforesaid LSPC is further done for introducing and exploiting the data representation redundancies in the data for enhancing compression. The LSPC system based on logarithm-based normalization produces random outliers in the decompressed signal, due to the inherent nature of the logarithmic operation.

Table 12.2 Computation time of the proposed scheme, for different sample sizes

No. of channels	Number of samples per channel			
	800		400	
	Encoding time (s)	Decoding time (s)	Encoding time (s)	Decoding time (s)
10	3.38	0.23	2.06	0.1
20	6.03	0.58	3.37	0.23
30	8.58	1.09	4.74	0.39
40	11.26	1.76	6.02	0.58
50	13.76	2.56	7.28	0.82
60	16.48	3.52	8.67	1.1
61	16.87	3.63	8.64	1.1

Table 12.3 Relative performance analysis of LSPC

Data set	Algorithm	CR	PRD	PSNR	Optimal criteria
4	Fast DCT [4]	4	11.09	****	BinDCT
	LSPC	4	2.86	21.44	Bit depth of 4
7	Wavelet image and volumetric coding [21]	2.56	1.72	37.95	Wavelet-s/s/t
	Clustering method [10]	2.67	****	****	DPCM-kNN
	LSPC	3.20	4.43	23.63	Bit depth of 5
8	Wavelet image and volumetric coding [21]	6.63	9.21	28.92	Wavelet-s/s/t
	LSPC	3.89	2.43	20.32	Bit depth of 4

****The metric has not been evaluated in the algorithm

Furthermore, an additional translation process is required in the LSPC to handle negative samples, which requires the estimation of the maximum value, to guarantee that the signals after translation lie above zero. To overcome the aforesaid issues Min-Max normalization is used to replace the logarithm normalization. The scheme was tested on different database using various performance parameters and is discussed in subsequent sections.

12.2 MMSPC: Encoding Process

The modified scheme illustrated in Fig. 12.9 is able to achieve lossy to near lossless compression by a two-stage process. The signal is initially normalized and subsequently coded using the Integer Fraction Coder (IFC). In the encoder of the Min-Max Spatial Pseudo CODEC (MMSPC) system [14] illustrated in Fig. 12.9a, the raw MCEEG data is arranged in a 2D structure as shown in Eq. 12.1, where M and N correspond to the number of channels and samples per channel, respectively.

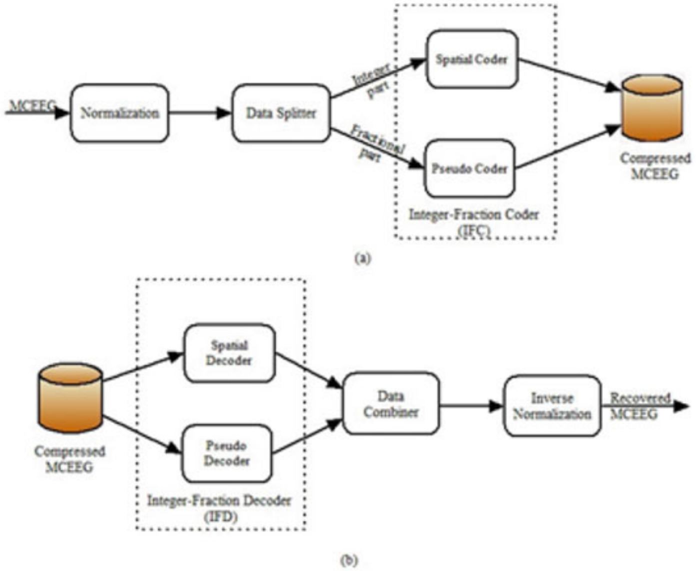

Fig. 12.9 Proposed MMSPC system. (**a**) Encoder section. (**b**) Decoder section

To increase the computational stability and memory requirement and maintain the contextual content, large sequence of MCEEG data are split into sets of maximum 1000 samples per channel, and each of these is processed and stored as frames as illustrated in Fig. 12.2. The next step is to normalize the MCEEG data using Feature Scaling (FS) that restricts the values between two arbitrary points a and b, and the normalized value is represented as $h(x)$ in Eq. 12.17

$$h(x) = a + \frac{(X - X_{\min}) * (b - a)}{X_{\max} - X_{\min}} \tag{12.17}$$

where X is the sample data, and $X_{\min}$ and $X_{\max}$ are the minimum and maximum of each channel. Size of $X_{\min}$ and $X_{\max}$ will be $M \times 1$. a and b take on values 0 and 1 in the proposed method. The normalized data $h(x)$ is split into integer $I_r(x)$ and fractional $I_f(x)$ parts using Eqs. 12.4 and 12.5 which are coded separately by the IFC.

Subsequently, $I_r(x)$ can be encoded using any spatial coding schemes that exploit the redundancies introduced by FS normalization. In the proposed system, Run Length Encoding (RLE) is employed because of its simplicity in implementation. The encoded data S presented in Eq. 12.18 is a sequence of codes representing a byte and the number of times it occurs consecutively.

$$S = (D_1, C_1)(D_2, C_2)\ldots(D_i, C_i)\ldots(D_n, C_n) \tag{12.18}$$

where D_i and C_i are the ith distinct integer and its occurrence, respectively. Alternatively, arithmetic coders could have been employed instead of RLE, but the coding efficiency was only marginally better at the expense of a slightly higher computational complexity.

Next a two-stage process encodes $I_f(x)$ and represents it by a novel PI representation. Initially a lossy process binarizes $I_f(x)$ based on the required resolution k. A base converter converts $I_f(x)$ to its equivalent binary stream b using Eq. 12.6.

Subsequently, in the second stage is a lossless process, based on the concept of data representation [17] for faster encoding and decoding is performed on the binary stream. The binary stream b from base converter is reshaped and packed into groups of eight bits to an equivalent PI representation. Outputs from the spatial coder and pseudo coder constitute the compressed MCEEG, and the resulting data is stored as frames as depicted in Fig. 12.2.

12.2.1 Decoding Process

The strength of the algorithm is its simple decoding architecture. In the decoder depicted in Fig. 12.9b, the coded MCEEG comprising the header, integer, and fractional data undergoes two processes, namely Spatial and Pseudo decoding, and the unit is collectively grouped as the Integer Fractional Decoder (IFD). In the Spatial Coder, the spatial coded data S is unpacked to obtain the integer part $I_r(x')$, based on the attributes in Eq. 12.18, where the coded integer D_i is repeated several times defined by C_i. Simultaneously, in the Pseudo Coder, the pseudo integers are binarized as normal unsigned integers. The binarized data is then reshaped with the same bit depth value used in the encoder, the value of which is retrieved from the corresponding header. This is followed by a fractional base conversion using Eq. 12.6 to obtain $I_f(x')$. The recovered integer and fractional parts are combined to reconstruct the normalized signal $I(x')$ presented in Eq. 12.7

Subsequently, inverse normalization of $I(x')$ defined by Eq. 12.19 results in reconstructing the MCEEG signal x'.

$$x' = X_{\min} + \frac{(I(x') - a) \times (X_{\max} - X_{\min})}{b - a} \tag{12.19}$$

where $X_{\min}$, $X_{\max}$, a, and b are values from the encoder.

Performance metrics on datasets that were discussed earlier for LSPC analysis are extended here for investigating the effectiveness of MMSPC for MCEEG compression. The effect of Min-Max normalization is illustrated in Fig. 12.10. Accordingly, the following observations were made:

It can be clearly observed that the data is now within the range of 0 and 1, depicting data representation redundancies. In Feature Scaling, normalization arbitrary values of a and b can be taken. It was observed that for all the variations of a, b other than $a = 0$ and $b = 1$ (Min-Max Normalization), the signal compression and reconstruction quality reduces. The normalization process is fully reversible, and the reconstructed signal follows the actual signal with visual validation. For numerical validation, Local Absolute Error (LAE) was computed, which is the deviation of

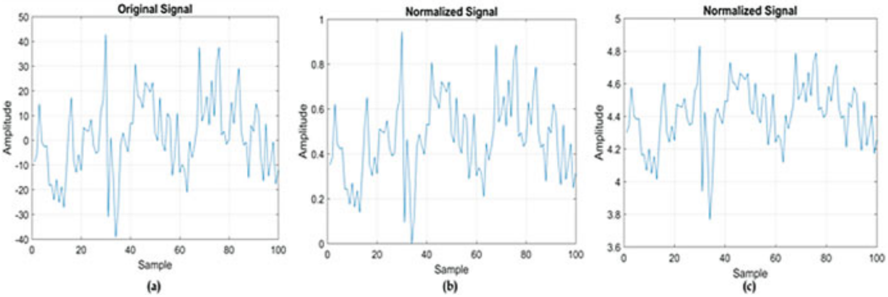

Fig. 12.10 Illustration of different normalization techniques. (**a**) Original, (**b**) Min-Max normalized, (**c**) logarithmic normalized

the decompressed sample from the original and was found to be in the order of 10^{-8} to 10^{-9}.

The performance of the compression algorithm at near optimal bit depths on the different MCEEG datasets has been numerically quantified in Table 12.4.

From the table for better signal quality a bit depth of four is better than its lower ranges, but results in lower CR, whereas lower bit depth of three provides a better CR with marginal increase in signal distortion in some of the datasets. An illustration of the original, decompressed and residual error signals at bit depths of 3, 4, and 5 is illustrated in Fig. 12.11.

It can be observed from the plots that at bit depths greater than 4 the error in the decompressed signal is significantly less. It can be observed that the algorithm is invariant to the data representation format and the performance metrics are correlated to the bit depth only. The choice of optimal bit depth is based on trade-off between acceptable amount of distortion in the decompressed signal and CR. This value can be found out with visual inspection along with numerical values of the distortion metrics PRD, RMSE, PAE, MAE, and PSNR in Fig. 12.12.

Furthermore, Fig. 12.12a–d, justifies that the performance metrics of the decompressed signal is highly correlated to the bit depth *d*. PRD provides an indication on the amount of error in the decompressed signal. For different data sets, at a bit depth of 3, the value varies from a minimum of 0.62 to a maximum of 12.01, while at a higher bit depth of 4, it varies from 0.15 to 2.98, indicating a higher signal quality. PAE which indicates the maximum absolute difference between the actual and decompressed sample varies from 11.73 to 61.02 at a bit depth of 3 and from 6.12 to 21.52 at bit depth of 4. MAE which provides a measure of closeness of the reconstructed sample varies from 3.15 to 9.93 and from 1.58 to 4.91 at bit depths 3 and 4, respectively. PSNR which is the ratio between the maximum signal power and the noise power of the recovered signal varies from 18.55 to 23.18 dB at bit depth of 3 and from 20.39 to 24.66 dB at bit depth of 4.

The distortion introduced in the decompressed signal depends on the bit depth or resolution of the base converter. At lower bit depths, distortions tend to increase but are complimented with larger signal compression. It can be observed from

Table 12.4 Performance evaluation of the proposed scheme at different bit depth

Data set	Min-Max normalization					
	Bit depth = 4			Bit depth = 3		
	CR	PRD	PSNR(dB)	CR	PRD	PSNR(dB)
1	3.38	0.23	24.06	5.03	12.01	22.21
2	6.03	0.58	20.39	5.12	7.06	18.55
3	8.58	1.09	21.32	5.12	4.21	19.48
4	11.26	1.76	24.63	4.66	0.78	22.79
5	13.76	2.56	24.66	2.60	0.62	23.18
6	16.48	3.52	24.22	5.10	7.31	22.33

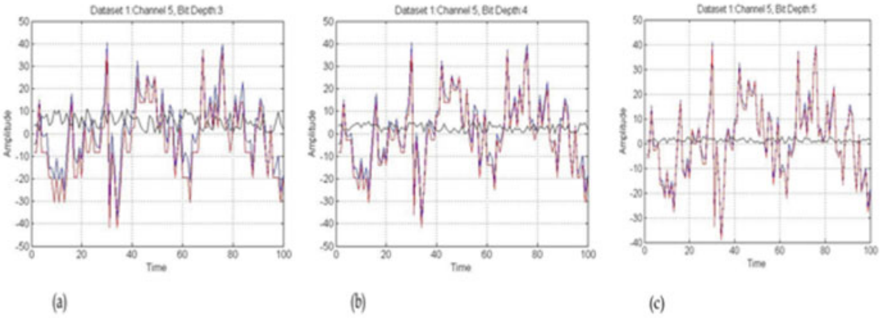

Fig. 12.11 Superimposed original, decompressed, and residual error signals at bit depth of three, four and five. (**a**) Signal compression at bitdepth of 3. (**b**) Signal compression at bitdepth of 4. (**c**) Signal compression at bitdepth of 5

Fig. 12.13a–d that the distortion parameters vary greatly among different data sets for varying CRs. Performance of the distortion parameters other than CR is almost the same for data sets 4 and 5, which are similar MCEEG signals with different sampling rates (1000 Hz and 100 Hz). The performance in terms of CR for data set 5 is comparatively lower than data set 4 as there were only 50 samples per channel, with the header information $X_{\min}$ and $X_{\max}$ having $M \times 1$ samples occupying 4% of the space. The header information for data set 4 with 500 samples per channel takes only 0.4% of the space, leading to higher CR. For MCEEG signals having longer duration, the overhead due to this header becomes insignificant.

The distortion introduced in the reconstructed signal depends on the bit depth or resolution of the base converter. At lower bit depths, distortions tend to increase, but are complimented with larger signal compression. It can be observed from Fig. 12.13a–d, that the distortion parameters vary greatly among different data sets for varying CRs. Figure 12.12a–d justify our initial theory that the performance metrics of the reconstructed signal are highly correlated to the bit depth *d*.

The relation between PSNR and PAE with MAE is depicted in Fig. 12.14a–b which can be used to decide on the choice of bit depth. From Fig. 12.14b the MAE can vary between 2 and 4 for an acceptable PAE of 10, which corresponds to the

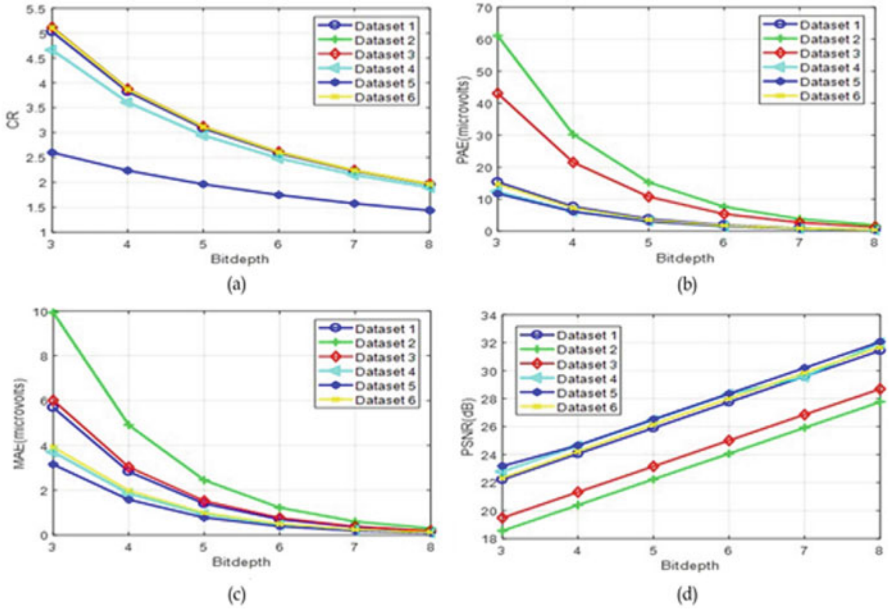

Fig. 12.12 Performance illustration of proposed scheme of MCEEG data sets for varying bit depth. (**a**) Bit depth versus CR. (**b**) Bit depth versus PAE. (**c**) Bit depth versus MAE. (**d**) Bit depth versus PSNR

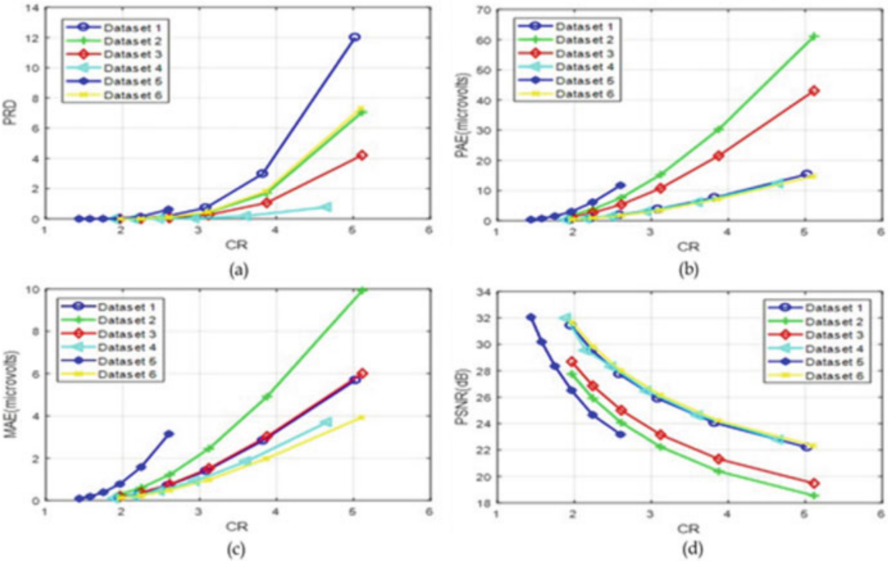

Fig. 12.13 Performance illustration of proposed scheme of MCEEG data sets for varying CR. (**a**) CR versus PRD. (**b**) CR versus PAE. (**c**) CR versus MAE. (**d**) CR versus PSNR

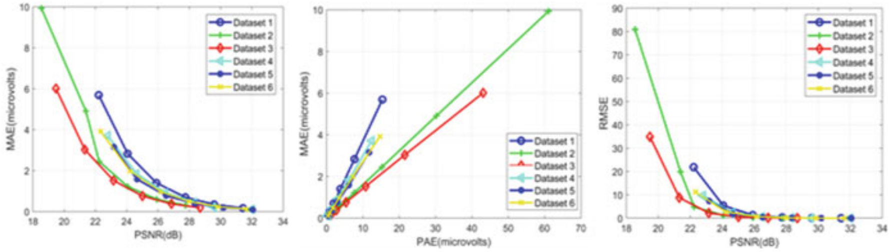

Fig. 12.14 Quantitative and qualitative analysis of MMSPC on different datasets

maximum error amplitude. On correlating this range of MAE with bit depth from Fig. 12.13c, an optimal bit depth can be set at 4, where the PAE is within 10 and MAE is within 4 in most of the data sets.

A relative comparison between LSPC and the proposed MMSPC system is illustrated in Figs. 12.15 and 12.16. It can be observed from the figures that MMSPC is better than LSPC in all grounds with reference to both CR and the degree of distortion introduced, with similar computational complexity of $O(zN)$.

Table 12.5 illustrates the time required for the encoding and decoding operations of MMSPC based on the variations in number of channels and samples per channel. Similar to LSPC, MMSPC computations were also performed on Intel Core2Duo system with two cores operating at 1.8 GHz, with 2 GB RAM. The average encoding and decoding time per sample was found to be 0.3 and 0.04ms, respectively, thereby strengthening the fact that the proposed algorithm is computationally fast.

Relative performance comparison of the proposed system with recent MCEEG compression algorithms [7, 4, 11, 13] is illustrated in Table 12.6 which uses CR, PRD, and computational complexity as measures.

CR of the proposed algorithm as observed from the table is greater when compared with other algorithms. The distortion indicator PRD is the lowest for the proposed algorithm suggesting that the proposed algorithm is less lossy than the other methods. Another highlight of the proposed algorithm is that it is computationally much lighter than all other algorithms, with a complexity of $O(zN)$, except for [4, 13] which claim to be computationally simple with comparable complexity as indicated in Table 12.6. But in both cases signal quality of the decompressed signal is more deteriorated with larger values of PRD as illustrated in the Table 12.6. One critical observation is that MMSPC performance is directly related to the bit resolution of the ADC, i.e., larger the bit resolution, the larger will be the CR with better decompressed signal quality and vice versa. This factor led to a lower performance of the algorithm for data set 8 which was recorded using a 12-bit ADC.

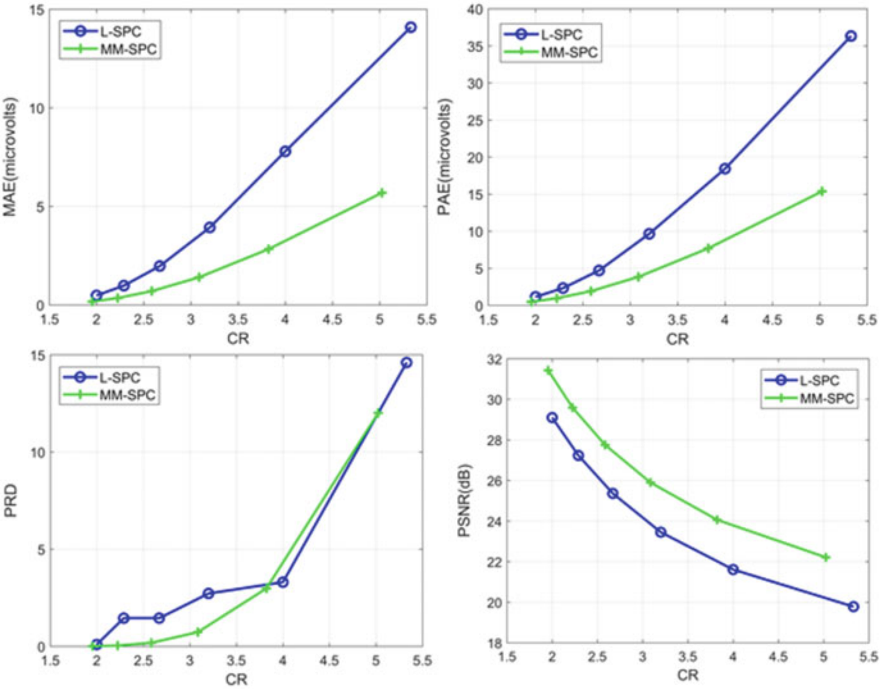

Fig. 12.15 Relative performance comparison of LSPC and MMSPC system under different CR

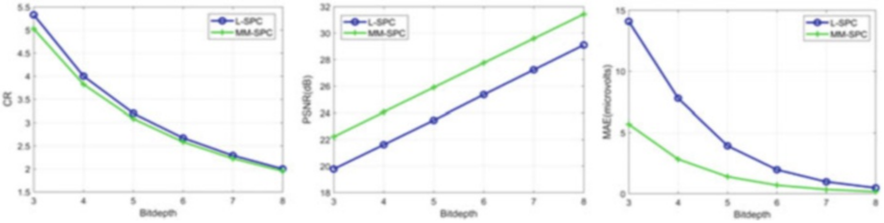

Fig. 12.16 Relative performance comparison of LSPC and MMSPC system for different bit depth

12.2.2 Complexity Analysis

In this section, an approximate estimation of performance of the MMSPC & LSPC is analyzed in terms of memory requirement and time complexity.

12.2.2.1 Memory Requirement Analysis

Lemma 1 *Let* $S = s_1,\ s_2 \ldots,\ s_n$ *be sample space of length n, with each sample represented by b bits. Then the total space B will be* $n \times \log \frac{2^b}{n}$

Table 12.5 Computation time of MMSPC for different sample sizes

No. of channels	Number of samples per channel			
	800		400	
	Encoding time (s)	Decoding time (s)	Encoding time (s)	Decoding time (s)
10	3.16	0.21	1.97	0.1
20	5.98	0.54	3.25	0.21
30	8.03	1.02	4.54	0.32
40	11.07	1.46	6.00	0.55
50	13.55	2.41	7.17	0.78
60	16.28	3.33	8.26	1.09
61	16.11	3.25	8.61	1.05

Table 12.6 Relative performance comparison of the compression algorithms

Authors & year	Dataset	CR	PRD	Complexity
Dauwels et al. [7]	7& 8	4.59	5.46	$O(3mN2)$
Birvinskas et al. [4]	4	4	11.09	$O(cN)$
Hejrati et al. [11]	7	2.67	****	$O(TKN)$
Titus and Sudhakar [13]	1 to 6	3.61	8.73	$O(zN)$
MMSPC	1 to 7	6.78	5.33	$O(zN)$

**** The metric has not been evaluated in the algorithm

Proof Consider n distinct b bit resolution samples. By the rules of information theory, the number of bits to represent the data is given by

$$B = \log \binom{2^b}{n} \tag{12.20}$$

This can be approximated using the property in combination to get the size of n elements or samples

$$B = n \times \log \frac{2^b}{n} \tag{12.21}$$

Spatial Coder returns two values namely distinct integer and its occurrence count of I_r defined in Eq. 12.3. The memory requirement of the coder is denoted by B_s

Lemma 2 *For given n samples with b bits per sample, there exists c set of distinct integers with size l bits per sample, such that* $1 \leq c \leq n$, $l < b$ *then from Lemma* 1 B_s *can be generalized by Eq.*12.20 ,

$$B_s \leq 2 \times c \times \log \frac{2^l}{c} \tag{12.22}$$

Pseudo Coder returns the integer representation of I_f defined in Eq. 12.4. Here the data is binarized using d bits defined by bit depth, and the number of samples n is taken such that product of $d \times n$ is divisible l the size of the integer. The memory requirement of the coder denoted by B_p

Lemma 3 *For a given n samples with b bits per sample, there exists n samples represented by d bits per sample such that* $d<b$ *then from Lemma 1* B_p *can be generalized by Eq.12.21 ,*

$$B_p = k \times \log \frac{2^l}{k} \tag{12.23}$$

where $k = \frac{d \times n}{l}$

Lemma 4 *Given* B_s *and* B_p, *the space requirements of spatial and pseudo coders, respectively, the total space requirement for the proposed system* B_t *will be the sum of* B_s *and* B_p.

Proof Given B_s and B_p from Lemmas 2 and 3 B_t, the total space complexity of the proposed algorithm is defined by Eq. 12.22.

$$B_t = 2 \times c \times \log \frac{2^l}{c} + k \times \log \frac{2^l}{k} \tag{12.24}$$

Lemma 5 *Let B,* B_t *be the space requirements of the original and compressed signal, then* $B_t << B$.

Proof Given B, B_t the space requirements from Lemmas 1 and 4 subject to the conditions $c \leq n$, $d<b$ and $l<b$, then

$$n \times \log \frac{2^b}{n} >> 2 \times c \times \log \frac{2^l}{c} + \frac{d \times n}{l} \times \log \frac{2^l}{\frac{d \times n}{l}} \tag{12.25}$$

Time Complexity

Majority of EEG compression algorithms available in the literature commonly employ PCA, Fast ICA, and compressed sensing to get an equivalent structure of the actual data. To the best of our knowledge the lowest computational complexity that can be attained by any MCEEG compression algorithm is above $O(N^2)$, where N is the number of samples.

Lemma 6 *For a given sample space S, the complexity of the proposed system is O* (zN)

Proof Given N the number of samples per channel, M the number of channels, and d the bit depth. The complexity of the system is $O(NMd)$. As the number of channels M and precision or bit depth d requirement are fixed, their contribution to the complexity is minimal. Thus, the algorithm has a linear dependence with the number of samples taken over a period with a complexity of $O(zN)$, where $z = d \times M$ and $z << N$.

12.3 Conclusion

The significance of this study is that it provides a computationally simple model for compressing MCEEG signals. In the proposed LSPC system, the signal is normalized using the TL transform and the resulting coefficients are coded using IFC, ensuing in data compression. The proposed scheme achieved an average CR of 3.16 with a computational complexity of only $O(zN)$ and average encoding and decoding times per sample of 0.3 and 0.04 ms, respectively. This is reasonably better than the other methods which rely on computationally intensive operations, to achieve similar CR. Also, an extension of LSPC in the form of MMSPC is proposed. Upon FS normalization, the normalized coefficients are coded in the IFC, resulting in data compression. The proposed scheme was able to achieve an average CR of 3.54 for a decent average PSNR of 23.38 dB with a computational complexity of O (zN) only. The average encoding and decoding time for the scheme per sample is 0.3 and 0.04 ms. The signal compression achieved is at par or reasonably better than the other compression schemes employing computationally intensive operation. The compression range is highly related to the recording resolution; more the resolution, the better will be the compression, while maintaining better signal quality. Moreover, the distortion level indicators such as PRD, MAE, PAE, and PSNR showed promising results to substantiate that the algorithm is suitable for MCEEG compression.

References

1. Andrzejak RG *et al* (2012) Nonrandomness, nonlinear dependence, and nonstationarity of electroencephalographic recordings from epilepsy patients. Phys Rev E 86:1–17
2. Antoniol G, Tonella P (1997) EEG data compression techniques. IEEE Trans Biomed Eng 44 (2):105–114
3. Bazn-Prieto C *et al* (2012) Retained energy-based coding for EEG signals. Med Eng Phys 34 (7):892–899
4. Birvinskas D *et al* (2015) Fast DCT algorithms for EEG data compression in embedded systems. Comput Sci Inform Syst 12:49–62

5. Blankertz B *et al* (2002) Classifying single trial EEG: towards brain computer interfacing. In: Dietterich TG, Becker S, Ghahramani Z (eds) Advances in neural information processing systems, vol 14. MIT Press, Cambridge, pp 157–164
6. Blankertz B *et al* (2006) The BCI competition III: validating alternative approaches to actual BCI problems. IEEE Trans Neural Syst Rehabil Eng 14(2):153–159
7. Dauwels J *et al* (2013) Near lossless multichannel EEG compression based on matrix and tensor decompositions. IEEE J Biomed Health Inform 17:708–714
8. Delorme A, Makeig S (2004) EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. J Neurosci Methods 134(1):9–21
9. Dheeru D, Karra Taniskidou E (2017) UCI machine learning repository
10. Hejrati B *et al* (2017a) Efficient lossless multichannel EEG compression based on channel clustering. Biomed Signal Process Control 31:295–300
11. Hejrati B *et al* (2017b) A new near-lossless EEG compression method using ANNbased reconstruction technique. Comput Biol Med 87:87–94
12. Hunter M *et al* (2005) The Australian EEG database. Clinical EEG Neurosci 36(2):76–81
13. Titus G, Sudhakar MS (2017) A simple and efficient algorithm operating with linear time for MCEEG data compression. Australas Phys Eng Sci Med 40(3):759–768
14. Titus G, Sudhakar MS (2018) A simple but efficient EEG data compression algorithm for neuromorphic applications. IETE J Res 1:1–12
15. Goldberger AL *et al* (2000) Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. Circulation 101(23):215– 220
16. Katsigiannis S, Ramzan N (2018) DREAMER: a database for emotion recognition through EEG and ECG signals from wireless low-cost off-the-shelf devices. IEEE J Biomed Health Inform 22(1):98–107
17. Lemire D, Boytsov L (2012) Decoding billions of integers per second through vectorization. ArXiv e-prints 1209.2137, 1–30
18. Lin L *et al* (2015) Multichannel EEG compression based on ICA and SPIHT. Biomed Signal Process Control 20:45–51
19. Naeem M *et al* (2006) Seperability of four-class motor imagery data using independent components analysis. J Neural Eng 3:208–216
20. Saeedi J *et al* (2014) Hybrid fractal-wavelet method for multi-channel EEG signal compression. CSSP 33(8):2583–2604
21. Srinivasan K *et al* (2013) Multichannel EEG compression: Wavelet-based image and volumetric coding approach. IEEE J Biomed Health Inform 17:113–120
22. Sriraam N (2009) Context based near lossless compression of EEG signals using neural network predictors. Int J Electron Commun (AE) 63:311–320
23. Tangermann M *et al* (2012) Review of the BCI competition IV. Front Neurosci 6:1–31
24. Xu G *et al* (2015) A 1.5-D multi-channel EEG compression algorithm based on NLSPIHT. IEEE Signal Process Lett 22(8):1118–1122