High-fidelity compression of electroneurographic signals from motor cortex

S. Zhang[a,b], S. Li[c], S. Lic[c], S. Lid[b,*]

[a]X University,China
[b]Y University,USA
[c]z University,China

不要科普性质

加一个范围，而非具体数字

Abstract

In the invasive Brain Machine Interface, multi-electrode array is implanted onto cerebral cortex to get the highest quality electroneurographic signal. Such signal recorded at 30 kHz will put a significant load on data storage and transmitting, making compression requisite to decrease the data volume. In this paper, a high-fidelity compression algorithm is proposed combined with the characteristics of motor cortex signals. Experiments on mammalian motor cortex shows compression ratio at 18% of original size without obvious compensation for reconstruct performance. The signal-to-noise (SNR) reaches at 36dB and detected spikes reserved more than ~~90%~~. Compared with existed works on ~~one-dimension~~ lossy signal compression, our method provides perfect fidelity for electroneurographic signal with acceptable compression ratio. ~~Accordingly, low cost storage devices can be efficiently used to store longtime neural signal after compression.~~

Keywords: Electroneurographic signal; motor cortex; High-fidelity Compression; Discrete Cosine Transformation; Amplitude filter; Dual-Phase Encoding

1. INTRODUCTION

1.1. BMI and Electroneurographic Signal

Biological brain is one of the most complex systems ever to be studied. It has a huge sensory system, transmitting mechanism and action executor which alter the state of mind or body continuously. Recent development on neural recording techniques has made it possible to collect neuron activities from brain, leading to the development of Brain-Machine Interfaces (BMI) system. A BMI is an interface connecting neurons to current circle, accomplishing information interaction and control. This is inspired by the requirements of substituting the BMI for impaired nervous systems [10, 11] and it has been demonstrated that BMI can transform motor intentions into actions by decoding the cerebral neuronal signals [4, 5, 6, 7, 8, 9].

BMI systems can be classified to invasive and non-invasive according to the neural activity recording mode. The non-invasive methods include functional magnetic resonance imaging and scalp-placed electrodes sampling to record thousands of cortical neurons activities, such as scalp electroencephalogram (EEG). Although this type of signal is easily accessible, the signal precision decreases due to the distance of skull. As a consequence, signal generated by individual neurons cannot be separated to analyze. On the contrary, the invasive methods use surgically implanted electrodes, recording extracellular neurons signals with finest detail [12, 13], which is referred to action potential, or spikes in individual neurons. The spike is a transient electrical membrane voltage change represents as an abruptly rising and falling phase in potential, which is initiated by a cells membrane of voltage-gated ion channels [29]. When excited, neurons create ion currents through their membranes, causing the cell to depolarize and trigger a spike. Such spikes are significant for characterizing the relationship between neuronal response and stimulus because spikes occur in excitable cells when given stimulus such as motor actions. The advantage of invasive BMI is that, in contrast with other methods used for investigating neuronal activity, the extra-cellular field techniques used in invasive BMI achieve well understood biophysics measurements [1]. Moreover, the invasive BMI provides the highest quality electroneurographic signal that can be used to decode neuron activity individually. Therefore, many BMI systems sample electroneurographic signals by implanting microelectrode onto cerebral cortex despite irritation might be caused by the invasive methods.

1.2. Motivation

This paper focuses on the motor cortex neuronal signals recordings. As an important part of cerebral cortex, motor cortex is in charge of planning, controlling and executing voluntary movement of body. Neurons in this region send neuron impulses to muscles to activate and coordinate movements. Unfortunately, injury or lesions to motor cortex may lead to motor diseases such as muscular atrophy and paralysis, disabling those patients from normal movements. To assist them with the development of biophysics, motor neuroprosthetics is researched. In invasive BMI system, motor neuroprosthetics aims at restoring movements for those people with paralysis or help them by motor nerve controlling devices. To research the neural mechanisms of human cognition, mammalian such as monkey has been used for the study for over 70 years [27].

*The author to whom the correspondence should be addressed to; Email: lisi@edu.

In the researches of motor cortex function, spike recorded by invasive BMI serves as an important data source, reflecting activity of excitatory neurons. However, a spike usually lasts for less than one thousandth of a second in the brain cells of animals, therefore high-resolution data acquiring device is required. The most common way to get the extracellular neuronal activity is by multi-electrode array (MEA). In this technique, the high-resolution electroneurographic signal is measured by penetrating MEA into tissues to get signals from hundreds of interested neurons. Instructional signals such as the location and velocity of the movement are then obtained by the implanted arrays. The mammalian neuronal signal of motor cortex is usually recorded by MEA with 128 channels. A sampling rate at 20-30 kHz is need to well-preserve the detail of spikes. Consequently, with 16-bit A/D resolution and a maximum sampling rate of 30 kHz, data stream of such raw format in overall 128 channels is recorded at 7.68MB/s. In other words, an hour of such data produces 28.8GB. This not only brings significant financial cost for data storage, but also challenges data transferring from several sources. Therefore, compression is desirable for neuronal recordings.

2. RELATED WORK

Although BMI systems are well established, compression for cerebral electroneurographic signal is not deeply investigated. Some relevant works such as the compression on Electromyography (EMG) and Electroencephalography (EEG) take signal characteristics into consideration for effective compress. However, the invasive electroneurographic signal is quite different.

Take EEG as example, it is commonly sampled from 250Hz to 2000Hz, because more detailed individual neuron activity cannot be captured as we have mentioned, therefore, high sampling rate is not required. Main scalp EEG signals reflecting the activity of millions of neurons under 100 Hz [26]. Therefore, only a small fraction at low frequency band is to be saved for further processing. Based on this idea, some compression methods are proposed. Giuliano et al uses Huffman coding, linear predictive coding and adaptive linear prediction [19] to encode each sequence signal. To achieve an efficient compression, basis pursuit is used to get a sparse representation in the work of Monica et al [24]. Afterwards, the correlation between channels has been explored for further compression. Agarwal et al used a long-tem prediction for multi-channel EEG compression. A compressed sensing framework is also used given sparse constrained representation [25].

However, compression techniques of EEG cannot be used to compress electroneurographic signal directly. Because lossy EEG compression algorithms lost high frequency band signal, but the high frequency signal in electroneurographic signal includes components significant in exploring how the brain encodes information, i.e., depicting the relationship between the stimulus and neuronal responses.

As a typical one-dimension sequential signal, audio also has large redundant information to be compressed. Lossless audio compression methods like FLAC and TTA use linear prediction to estimate audio spectrum, but the compression ratio is higher than 50% due to the complexity of original waveform. On the other hand, lossy audio compression is supposed to save the components more audible to human hearing. Such audio compression methods typically compress to 5%-20% of original data stream by reducing perceptual redundant data. Nevertheless, typical audio compression may loss exactly the information of neuronal activity, so audio compression algorithms havent taken the peculiarities of electroneurographic signal into consideration.

Existing compression algorithms for multi-channel electroneurographic signal is implemented from two threads. One is to compress signal of each channel individually using intra-channel properties; the other is to decrease the redundancy among channels using inter-channel correlation. From the 1ŝt perspective, Weber et al. [28] compress somatosensory cortex (S1) neuronal responses of rat by a wavelet based coder, achieving the compression ratio up to 1:20. However, this compensates for the loss of 25% of the spikes, which is not desirable for future analysis. For the same recorded data of rats S1 response, Chen et als result achieved Signal to Noise Ratio (SNR) at about 25db with compression ratio larger than 4:1 [15] by adaptively quantified, in which both compression ratio and signal quality is not guaranteed perfectly. To improve their work in terms of the 2n̂d point of view, Chen et al. [16] take advantage of correlation between channels, achieving 20:1 compression ratio with SNR at 25db by a video compression method. However, all of the above works have a loss at detail signal, making the high quality original signal acquired in vain. To better process signal in further research, preserving the original electroneurographic signal seems to be of more importance.

In this paper, we developed a comprehensive framework that compress electroneurographic signal of motor cortex with high fidelity. Incorporating with the characteristics of motor neuronal recording, we consider to compression using intra-channel properties. A number of instances have been examined on our proposed framework, achieving an average SNR at 36db and compression ratio of 18%. The fidelity of spikes is also kept higher than 92%, making reconstruction performance guaranteed. For signals with inter-channel correlation, this algorithm could be utilized for further compression improvements. The framework is organized as follows. Section 3 analyzes the peculiarities of neural signal; Section 4 gives the overview of our proposed compression method; Section 5 depicts the core Dual-Phase Encoding algorithm; Section 6 presents the experimental results to show the effectiveness of the proposed method; Section 7 devotes to the conclusion and future work.

3. CHARACTERISTICS OF ELECTRONEUROGRAPHIC SIGNALS FROM MOTOR CORTEX

Data recorded from MEA is a multi-channel signal, composed of multiple one-dimension signals, each is a temporal sequence. In previous analysis of such recorded data, electroneurographic signal of a channel is split into different frequency bands depending on the signal sources and different properties

of the brain tissue. Lower frequency (cut off at 100Hz for example) could be the local field potential (LFP) while higher frequency components in general action potential events. The LFP is largely originated from pre-synaptic activity, composed of more sustained currents reflecting the averaged dendrictic activity [2, 18, 20]. The other important part of electrophysiological signal is spike. Unlike LFP, spikes locate at medium to high frequency (depends on the intensity of stimulus), representing activities transmitted by individual neuron.

To compress effectively while maintaining the quality of signal, the properties of recorded multi-channel electroneurographic signal are analyzed in temporal and spectral domain. Our dataset is composed of signals recorded from motor cortex neurons of 2 male monkeys by a 96-electrode array sampling at 30 kHz, see Section 7.1 for detail. According to the statistics on the dataset, three characteristics are summarized from intra-channel property to inter-channel correlation.

1. Power centralizes on low frequency in single channel

To investigate the characteristic of the recorded temporal signal in spectral domain, discrete cosine transformation (DCT) has been adopted. DCT reconstructs the original signal by a sum of cosine functions, corresponding at different frequencies. As a variation of Fourier Transformation, DCT is preferable because it derives a set of real number coefficients, called DCT coefficients. Moreover, it also has fast solution with optimal signal reconstruction ability.

The transformed DCT coefficients vector of the i-th channel is denoted by $x_i = [x_i^1, x_i^2, ..., x_i^N]$ Let x_i^j be the j-th DCT component of x_i. The energy proportion of the low frequencies part is calculated on the whole dataset:

$$P = \frac{\sum_i \|x_i^1, x_i^2, ..., x_i^{T_{p0}}\|_2}{\sum_i \|x_i\|_2} \quad (1)$$

where the denominator is the total amount of energy over all channels and the numerator is the energy summation of the first T_{p0} DCT coefficients, i.e., energy of the low frequency part with the cutoff frequency at T_{p0}. For each sampled signal, P is calculated according to Eq.1. The average value of P on the whole dataset with T_{p0} is shown in fig.1. It is clearly illustrated that few number of DCT components occupy the dominant energy. In other words, considerable power is centralized on low frequency domain, that is, the LFP.

2. Remarkable peaks locate at high frequency

Last property has demonstrated the predominant energy of LFP. Besides, we have also find out some characteristics in the spectrum of the electroneurographic signal at medium and high frequency. Take the first channel as example, calculate the average spectrum of this channel over a period of time (e.g 10 mins), clear peaks can be found at the medium and high frequency, as fig.2 shows a conceptual illustration of a truncated part at high frequency of the first channel. There is a peak at 7325Hz, which corresponds to a frequent neuronal firing pattern. Actually, experiment shows that some peak frequency locations are shared by different channels while some are not. This can be comprehended from the sampling mechanism of Multielectrode array, by which the electroneurographic signal of a channels is composed by signal generated from 3 to 5 neurons with different firing patterns.

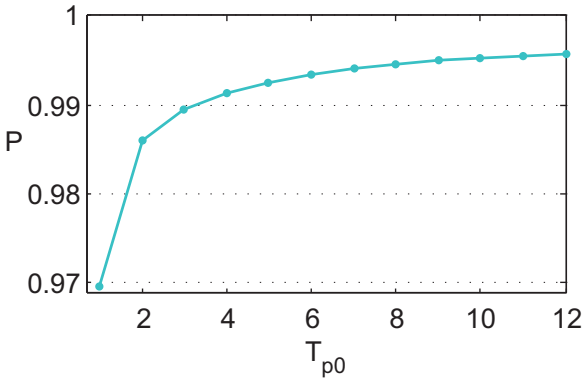

Figure 1: Statistical Energy ratio of the first 12 dimension DCT coefficients. The horizontal axis T_{p0} denotes the number of components to be taken into consideration. The vertical axis is the energy proportion P of the first T_{p0} components.

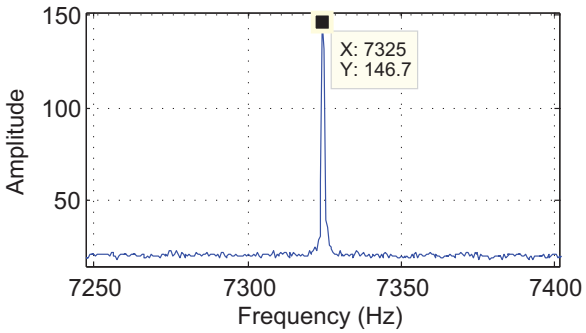

Figure 2: DCT coefficients amplitude distribution of Electrophysiological Signal in High frequency

3. Inter-channel Correlation is not steady

In Chen et al s work [16], the somatosensory cortex (S1) neuronal responses of Wistar rat is studied. Taking the signal correlation among channels into consideration, the compression algorithm achieves a good performance. Accordingly, we use the electroneurographic signal of mammalian motor cortex to explore whether same correlation exists among our recorded channels.

Partition a raw signal R into 10 equal-length temporal successive segments, let the average DCT coefficients be $\boldsymbol{F} = \{\boldsymbol{F^1}, \boldsymbol{F^2}, \ldots, \boldsymbol{F^{10}}\}$ in spectral domain, $F^i \in R^{N_c \times S_b}$ represents the average of DCT coefficients amplitude of the ith segment, where N_c is the number of channels and S_b is the block size to be transformed by DCT each time. For each vector $\boldsymbol{F^i}$, calculate the correlation coefficient between every two channels to get the correlation matrix $\boldsymbol{C} \in R^{N_c \times N_c}$. Fig.3(a),(b) shows schematically the correlation matrix of two sequential segments in a sample of 96 channels. It can be seen that the correlation varies dramatically. Use coefficient of Variation (CV) to formulate the relative dispersion extent of correlation coefficient:

$$CV = \frac{\sigma}{\mu} \quad (2)$$

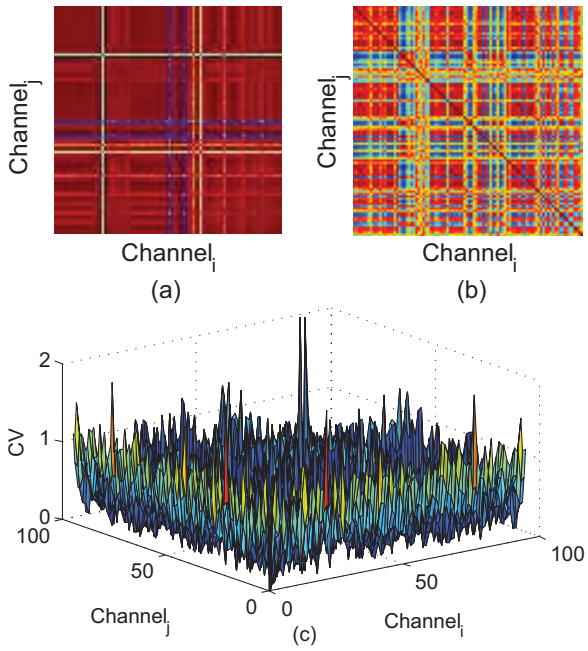

Figure 3: (a),(b) is the inter-channel correlation matrix of 2 segments in spectral domain. In the two figures, value at position (i,j) denotes the correlation coefficient between the ith channel and jth channel. Darkness of the point means the relativity of corresponding channels. (c) is the average CV matrix over all segments in the dataset. Height on position (i,j) denotes the CV of correlation between channel i and channel j.

where σ is the standard deviation and μ represents the mean of the variable correlation coefficient. A signal is considered to be stable when σ is negligible to μ, reaching at a small CV. For each segment $\boldsymbol{F^i}$, its coefficient of variation matrix $\boldsymbol{CV} \in R^{N_c \times N_c}$ measures the fluctuation of each element in correlation matrix $\boldsymbol{C}$. The average of $\boldsymbol{CV}$ over all segments in the entire dataset is shown in fig.3(c). The average of such $N_c \times N_c$ CV values is 0.68. That is to say, the correlation coefficient varies seriously over time, so the correlation between channels is not steady in our obtained motor cortex neuronal responses, leading to difficulties in reducing redundancy among channels.

4. THE PROPOSED COMPRESSION METHOD: AN OVERVIEW

In this paper, we propose a high-fidelity compression framework for electroneurographic signal incorporating with its above mentioned characteristics. This framework involves first a frequency domain preprocess, then applying a systematic method for efficient compressing and encoding. The diagram in fig.4shows the key steps of this framework, contained in two data stream based modules namely Preprocess and Dual-Phase Encoding. To reconstruct the signal, decoder of our proposed method can be derived by directly reverse the processing steps.

The processing strategies for electroneurographic signal from motor cortex are enlightened by its properties. Firstly, as considerable power centralized on low-frequency, it is insufficient to merely use a power-based criterion to measure the performance of reconstruction quality (Section 6.2 specifies the compression criterion). Besides, the second property has pointed out that some peaks at high frequency may correspond to specific spike firing patterns, making it unavailable to use low-pass filter for compressing recorded signal. The reason is that signals out of cut-off frequency will be largely attenuated in this way. To address this problem, we use an Amplitude Filter on spectrum, dichotomize the given signal according to amplitude instead. The partitioned signal with high amplitude consists of salient LFP and action potential, while the residual is contained in the low amplitude components. In the Dual-Phase Encoding module, the two parts are compressed individually by different strategies. Finally, it has been demonstrated in the third property that correlation coefficients fluctuate with time in the motor neuronal recordings. As a result, we only use intra-channel correlation for compression. The encoding of entire signal is composed by compression on all channels.

For one channel, the long signal obtained is firstly grouped into small portable blocks of S_b discrete points in time domain, where S_b denotes the size of a block. Then the two modules implement the following operations for each block:

1) Preprocess:. Input the grouped signal generated from motor cortex, the temporal sequence is presented in spectral domain by S_b DCT coefficients, each corresponds to a spatial frequency. These coefficients are to be dichotomized by an Amplitude filter.

Let $\boldsymbol{f} \in R^{N_c \times S_b}$ be the DCT coefficients of all channels within the time of a block S_b, each row of $\boldsymbol{f}$ presents the spatial frequency of a channel. Based on an amplitude filter mentioned before, the components comprising the signals spectrum are to be separated into two parts, namely High-Amplitude-Component (HAC) and Low-Amplitude-Component (LAC).

$$\begin{aligned} &L_c = find(|f(c,:)| \le T_{LH}),\ L_c \in \mathbb{R}^{S_l},\ S_l < S_b, \\ &H_c = [h_1, h_2, \ldots, h_{s_b}],\ h_i = \begin{cases} f(c,i),\ if\ |f(c,i)| > T_{LH} \\ 0,\ else \end{cases} \\ &c = 1, 2, \ldots, N_c \end{aligned} \tag{3}$$

where T_{LH} is the amplitude threshold, H_c and L_c denotes High-Amplitude-Component and Low-Amplitude-Component of the c^{th} channel of $\boldsymbol{f}$ respectively. Note that L_c is composed only by those DCT coefficients whose amplitude less than T_{LH}, while HAC save the coefficients out of threshold by zeros instead. Therefore H_c has the same number of elements as in $\boldsymbol{f}$, whereas the size of L_c recorded as S_l is less than S_b. This data-handling method is designed for conveniently processing in the subsequent encoding methods.

2) Dual-Phase Compression:. For neural signal analysis, it has been demonstrated in rate coding that a neurons response is completely characterized by its mean firing rate [31], which is the number of action potential per unit time. This concept was reformed by [3] in 2007, in which claims that combining LFP and spike activities will enhance the neural decoding accuracy. As a consequence, high amplitude component is more vital for representing the neuronal responses due to this part

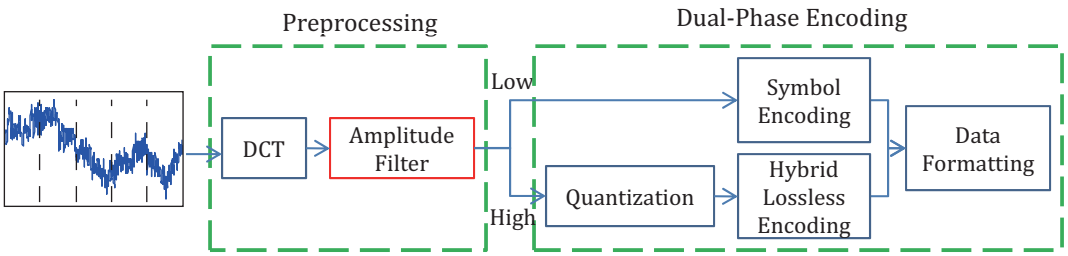

Figure 4: Flow diagram of the overall compression algorithm

contains LFP and spike activities. On the contrary, the low amplitude component is paid on less attention, but we also keep it to retain global signal information and for further investigation. Details of this leverage will be specified in Section 5.

From the different functions played by L_c and H_c, the input dichotomous signal is compressed in dual-phase as illustrated in fig.4. In the first phase, the part with low amplitude is compressed by Symbol Encoding. This is an element-wise encoding method, recording one-bit symbol in the place of original DCT coefficient in L_c. In the second phase, the high amplitude component is quantified and encoded by a hybrid encoding method blending Huffman Encoding with Zero-Length-Encoding. In the end, data is formatted to store. While the intuition of the Dual-Phase compression seems simple at first glance, a reasonable blending strategy in the hybrid encoding method and properly parameterize is requisite for providing an efficient compression architecture.

5. DUAL PHASE ENCODING

The Dual-Phase Encoding module is proposed to retain the global and local electroneurographic signal while signal compressing. It is defined as two independent single-phase operations on the preprocessed signal. The first phase use Symbol Encoding to compress low amplitude component. In this phase, the 16-bit original value is replaced with one-bit symbol, which is implemented by average smoothing. The second phase compresses high amplitude component in the following two steps. First the high amplitude component is quantized to a relative small range, then the quantized data is further compressed in a lossless fashion by Hybrid Lossless Encoding.

5.1. Symbol Encoding for Low-Amplitude Component

At the input of Dual-Phase Module, signal is separated into two parts by an amplitude filter. To get a high compression ratio, low amplitude values are discarded normally. However, in this strategy, sizeable coefficients to be lost in are likely to introduce significant errors that impact the signal fidelity and future analysis. To address this problem, we propose an efficient representation for this part, which is inspired by a processing step of neural decoding.

In the research of neural decoding, one of the difficulties in predicting motivation from neurons responses is noisy recording: Given identical stimulation, same pattern of neuronal activity never appears twice. To get accurate information of neuronal responses, some sort of averaging must be performed on the electrophysiological signal [30]. Therefore, average value is considered to be the surrogate of original coefficient, keeping signal average characteristics while compressing.

In Symbol Encoding, this substitution is implemented in frequency domain. Low-Amplitude DCT coefficients are taken place by the average of the low amplitude values at corresponding position. Notice that the positive and negative coefficients may counteract to zero, thus the average is computed based on amplitude instead of value directly. For efficient compression, the average amplitude of Low Amplitude Component is precalculated for all channels. So each element of L_c is encoded by one-bit symbol representing its sign, as calculated in symbol (l_i):

$$symbol(l_i) = \begin{cases} 0, & -T_{LH} < l_i \leq 0 \\ 1, & 0 < l_i < T_{LH} \end{cases} \tag{4}$$

where 0 and 1 is the coefficient symbol, 0 for negative and 1 for positive. According to Eq.4, a vector composed of S_l low amplitude component symbols is obtained in Symbol Encoding, which is used to be recovering signal when decompressing.

In the decompression steps, we use sgn function to transform the symbols into sign value (± 1), the signal is then recovered by multiplying it to the average amplitude at corresponding frequency. Let $\boldsymbol{F}$ be all the grouped blocks within a given recording, that is, a set of $\boldsymbol{f}$ at different time slices. $\boldsymbol{M} \in R^{N_c \times S_b}$ is the mean of low amplitude coefficients at different entries in $\boldsymbol{F}$.

$$L_c(v)' = sgn(symbol(L_c(v)) - 0.5)\dot{M}(c, v'), i = 1, 2, \ldots, S_l \tag{5}$$

where *sgn* is the function returning 1 for positive and -1 for negative value, v is the original position of $L_c(v)$ in $\boldsymbol{f}$. M is calculated by

$$\begin{aligned} A_F(c, v) &= \{x \mid |x| < T_{LH}, x \in F(c, v)\} \\ M_F(c, v) &= \frac{1}{\|A_F(c,v)\|} \sum A_F(c, v) \end{aligned} \tag{6}$$

where $A_F(c, v)$ is the set of DCT coefficients at (c,v) in F whose amplitude is less than TLH, $\|\cdot\|$ is the size of a set. As a necessity in recovering the low amplitude component, M_F should be calculated and prestored for each recording to be compressed.

The last problem when decompression is how to get the original position v of $L_c(v)$ in Eq.5. Intuitively, we can record the positions of LAC, but it is space consuming. This problem is solved by taking advantage of the assigned zeros in HAC

as Eq.3 shows. In decompression, we can locate the low amplitude component as the zero entries positions in HAC. There is no need to worry about the number inconsistency of corresponding positions, because it is guaranteed that all the nonzero entries in HAC maintain non-zero in subsequent steps. So each element in LAS can be correctly located.

5.2. Quantization for High-Amplitude Component

Since high amplitude component include LFP and salient spikes, this vital part is to be fine preserved than the low amplitude component. In order to compress without loss of quality, this component is first quantized to a small range, then a hybrid encoding method is taken for further compression. The first process step referred to as quantization is presented in this section. After output from the Preprocess module, each of the S_b coefficients of high amplitude component is uniformly quantized in conjunction with a Quantization Table. This is fundamentally a lossy process, but redundancy is discarded by representing signal with precision no larger than necessary to preserve valuable information. The quantization is defined as division original DCT coefficients by corresponding quantizer, followed by rounding to the nearest integer. Those quantizers for different channels compose the Quantization Table (QT). Let $QT_F \in R^{N_c \times S_b}$ denotes the quantization table for set $\boldsymbol{F}$, the v-th entry of quantized signal $\boldsymbol{H}_c^Q$ can be derived from Eq.7

$$H_c^Q = round(H_c./QT_F(c,:)) \tag{7}$$

where ./ is a matlab operator representing vector division at each corresponding position; round(X) is the operation that rounds the elements of X to the nearest integers. In the decompression, dequantization functions inversely for HAC by multiplying QT by the quantized result.

$$H_c' = QT_F(c,:) \times H_c^Q \tag{8}$$

When aims at compressing the electroneurographic signal as much as possible without decreasing its feasibility, a proper way to design quantization table should be required. For high amplitude component to be quantized, its scale is determined by the individual signal as well as the threshold $T_L H$. However, such signal recorded from different individuals could obtain divergent results [32]. Whats more, different neural units may have different spectral distribution. Therefore quantization table varies on different channels for different samples. In our method, the average amplitude of low amplitude coefficients are assigned to QT, i.e., setting QT equals M as defined in Eq.6.

This designing strategy for QT is proposed from several angles. First, diversity of individual recordings has been taken into consideration by yielding unique QT for each sample. Second, unshared quantizer for each channel takes account of the discrepant spectrum distribution among channels. Third, with the use of round function, the reverted value differs by no more than half of QT on corresponding position. Moreover, QT calculated by Eq.6 makes all the quantizer less than T_{LH}. As a consequence, the quantized values are no less than one, sufficing the non-zero condition mentioned at the end of section5.1. Finally, as M has been already defined and saved in Symbol Encoding module, no additional space is required if reuse M as quantization table.

5.3. Hybrid Lossless Encoding for Quantized Data

As all the elements in high amplitude component are integer after quantization, the discrete distributed signal can be compressed losslessly by encoding, which is implemented by blending Huffman Encoding with Zero-Length-Encoding.

As an optimal symbol-by-symbol coding (symbols are unrelated), Huffman Encoding yields the optimal variable-length code, which can be efficiently used in our quantized data. However, notice the amplitude distribution illustrated in fig.2, many DCT coefficients at high frequency have small amplitude, which are likely to be included by LAC. After partitioned by the amplitude filter with Eq.3, a lot of coefficients at high frequency turn to be zero in HAC. Therefore, more efficient encoding can be achieved by replacing separately recorded zeros with recording the number of continuous zeros at high frequency, which is referred to as Zero-Length-Encoding. Consequently, all the nonzero DCT coefficients are presented by Huffman Encoding; zeros before and after B are signified by Huffman Encoding and Zero-Length-Encoding respectively. To exploit the two encoding methods effectively, the boundary B between them should be well-designed, which depends on the distribution of zeros.

To elaborate the hybrid lossless encoding, we first give a brief introduction of Huffman Encoding and Zero-Length-Encoding. Then we show how to set the boundary between the two methods.

5.3.1. Huffman Coding

Entropy Encoding is a lossless compression technique, typically creates a unique prefix-free code to each symbol of a set. As the most common method of Entropy Encoding, Huffman Encoding [22] was used in our lossless encoding method, aiming at establishing an optimal tree that minimizes the weighted sum of heights [21] (i.e. the total length of code). To transform original value into binary sequences, Huffman coding derives variable-length code based on the estimated occurrence frequency of each symbol to be encoded.

Note that we dont calculate the occurrence of all the DCT coefficients. For the non-zero entries, we only calculate those within [-Z,Z], where Z is the coefficient statistical range. This is because the coefficients of HAC follow the Gaussian distribution approximately according to our experiment, in which high amplitudes rarely appear. Calculating all coefficients will largely decrease the efficiency of Huffman encoding. Therefore, coefficients with amplitude larger than Z are saved by their value and position independently. In addition, not all the zero coefficients are taken into consideration either. Let B be the boundary between Huffman Encoding and Zero-Length-Encoding. As only the zeros before B use Huffman Encoding, we only calculate on the occurrence of zeros before B.

As a result, the occurrence number of zero coefficients before B and those nonzero entries within [-Z, Z] over channels of all samples is averaged, as shown in fig.5. In this case, B

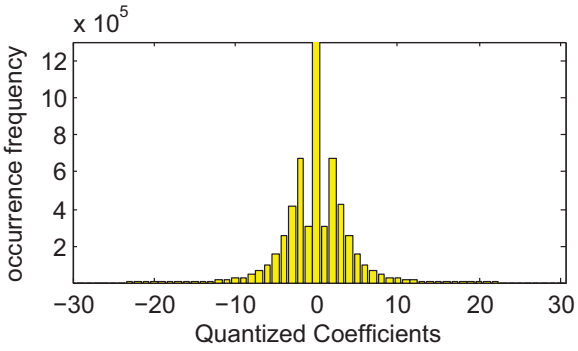

Figure 5: Quantified Coefficient Frequency distribution, horizontal axis denotes the coefficients value ranging from -31 to 31. The vertical axis is the occurrence frequency of each value in x.

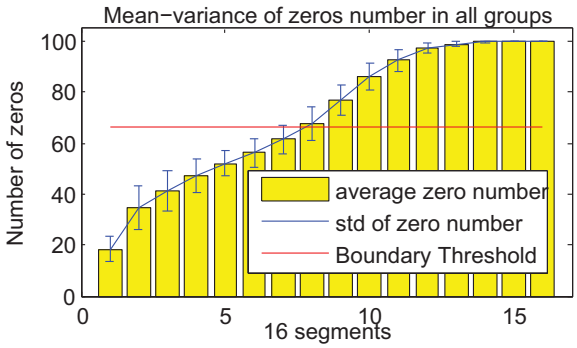

Figure 6: Average Zero distribution of 16 segments. Divide the 1600 elements into 16 segments, each has 100 components. Yellow bar and blue line denote average and standard deviation of zeros in each segment respectively. The red line is the threshold $T_{HZ} * (S_b/N_s)$ to separate the two encoding method. It can be derived that crossing point lies on the 7th segment.

is determined by the delimitation strategy according to section 5.3.3. After the calculation, Huffman Encoding is derived from the statistical result. Use ***HCT*** to represent the Huffman Code Table, i.e. the variable-length code table for encoding the given DCT coefficients.

5.3.2. Zero-Length-Encoding

The Huffman Encoding has fulfilled encoding zeros before the boundary B. In the high frequency domain after the B^{th} DCT coefficient, use Zero-Length-Encoding to record the repetition length of continuous zero coefficients, while Huffman encoding is still used to encode the nonzero ones.

Let k_z be the number of zeros before a nonzero coefficient, use octal number notation to encode k_z in the following way. Use 3 digits to represent [0,7] at different orders, each order has a basis of an exponential of 8. The number of orders is not constant, increasing with k_z if necessary. Let g(k) be the number of orders to represent zero number, we have

$$g(k_z) = \begin{cases} 1, \ k_z \in [0, 7] \\ 2, \ k_z \in [8, 7 \times 8 + 7] \\ 3, \ k_z \in [64, 63 \times 8 + 7] \\ \dots \end{cases}$$

which is formulated to

$$g(k_z) = \begin{cases} 1, k_z = 0 \\ \lceil \log_8(k_z + 1) \rceil, else \end{cases} \tag{9}$$

As the Huffman encoding of zero is no longer used to present zero individually in this method, it can be used to connect adjacent orders of zeros. For example, the format below shows $k_z = (8A + B) \times 8 + C$, HCT***HCT***(0) is the Huffman Encoding of zero, and A, B, C denotes 3^{rd}, 2^{nd}, 1^{st} order respectively.

$$\underbrace{\_\ \_\ \_}_{A} HCT(0) \underbrace{\_\ \_\ \_}_{B} HCT(0) \underbrace{\_\ \_\ \_}_{C}$$

In the decoding step, a 3-bit binary code is read firstly, then detect whether a sequence following is the same as ***HCT***(0). If the condition is met, read 3 bits again and repeat until the condition is dissatisfied. Finally, the sequence to represent zero is decoded. Due to the Huffman Encoding is prefix-free code, it can be guaranteed the unambiguous of zero representation.

Zero-Length-Encoding benefits compression ratio by coding the number of continuous zeros instead of recording individually. It is, therefore, expected that the continuous zeros in this part is sufficient that recording in this way is more space saving. Thus it is indispensable to determine the boundary between the two methods.

5.3.3. Delimitation of Quantized Data

To make the best use of the above mentioned two encoding methods, how to delimitate between them appears especially important. As the two methods are motivated by the special zero distribution of quantized data, the partition strategy is also to be get from it.

A rough estimation of the boundary between the two encoding method is calculated according to the tendency of zero coefficients distribution. Divide every S_b-point discrete temporal sequence into N_s equidistant shorter segments, then calculate the number of zeros of each segment. Fig.6 illustrates the average zero distribution of a 1600 DCT coefficients vector split into 16 segments, each has 100 elements (Sb=1600, Ns=16). It indicates that the number of zeros increases with frequency. A threshold of zero occurrence frequency can therefore be determined for dichotomizing the two encoding methods according to the average zero distribution $\Omega_0 \in R^{N_s}$, i.e. the yellow bars in this figure.

Let T_{HZ} be the threshold of zero occurrence probability at the boundary between Entropy Encoding and Zero-Length-Encoding, then the number of zeros in the segment is denoted by $T_{HZ} \cdot \frac{S_b}{N_s}$, where $\frac{S_b}{N_s}$ is the number of elements in each segment. Then the boundary can then be derived from Eq.10, coefficients before B use Huffman encoding and those after B use Repetition Count Encoding.

$$B = CrossSegment\left(\Omega_0, T_{HZ} \cdot \frac{S_b}{N_s}\right) \cdot \frac{S_b}{N_s} \tag{10}$$

In this formulation, CrossSegment function returns the index of segment of the intersection of 0 and the threshold of zero

occurrence frequency. Fig.6 illustrates the zero distribution of a channel divided into 16 segments, with the red line indicates the threshold, the point meets the threshold denotes cross of intersection is the 7th segment. In this case, Sb=1600, Ns=16, define THR =0.75 and we have B=700.

In a more precise way, we consider the accurate position of intersection in practice. Then the boundary is calculated according to Eq. (8). Use x to denote the index of crossing segment, instead of multiplying the number of elements of a segment directly, the calculation of B in (8) has taken account of the proportion of the threshold to the corresponding segment.

$$\begin{aligned} x &= CrossSegment\left(\Omega_0, T_{HZ} \cdot \tfrac{S_b}{N_s}\right) \\ B &= \left((x-1) + \tfrac{T_{HZ} \cdot \frac{S_b}{N_s} - \Omega_0(x-1)}{\Omega_0(x) - \Omega_0(x-1)}\right) \cdot \tfrac{S_b}{N_s} \end{aligned} \tag{11}$$

The last problem is the deterministic of T_{HZ}. As T_{HZ} denotes the percentage of zeros at B, it can be represented by the average number of zeros before a nonzero coefficient at the boundary:

$$T_{HZ} = \frac{k_z(B)}{k_z(B)+1} \tag{12}$$

where 1 represents the nonzero coefficient and $k_z(B)$ denotes the average number of zeros before a nonzero efficient at the boundary.

This delimitation achieves additional lossless compression based on the statistical characteristic of DCT coefficients. However, there is a deadlock in the calculation of B and T_{HZ} in Eq.11and 12. Eq.11 needs the value of T_{HZ} which is associated with B in Eq.12. An iterative-checking algorithm is proposed to address this problem, which need to formulate the code length derived by the two encoding methods separately to delimitate rationally.

Let ***HCT*** be the Huffman Code Table derived before, ***HCT***(x) means the Huffman Encoding of x , $l_0 \in Z_+$ denotes the length of ***HCT***(0), k_z be the average number of contiguous zeros before a nonzero coefficient. The code length of quantized coefficients by Huffman encoding only is l_1:

$$l_1 = \sum_{i=1}^{I} [HCT(x_i)] + l_0 \cdot k_z I,\ x_i \in H_c^Q, x_i \neq 0 \tag{13}$$

where I is the size of the set including all the nonzero coefficients in H_c^Q. In this equation, the first term summates all the nonzero elements x_i, while $l_0 \cdot k_z I$ denotes the length of the zeros because each nonzero coefficient has $l_0 \cdot k_z$ digits before it in average.

Likewise, if all zero entries are denoted by Zero-Length-Encoding, the length of code is l_2:

$$l_2 = \sum_{i=1}^{I} [HCT(x_i) + (3+l_0)g(k_z) - l_0],\ x_i \in H_c^Q, x_i \neq 0 \tag{14}$$

According to the Zero-Length-Encoding format shown in section 5.3.2, $(3 + l_0)$ is the number of bits required for each additional order. Use $g(k_z)$ to be the number of orders for k_z as defined in Eq.(9), then $(3+l_0)g(k_z) - l_0$ is the number of bits denoting the average number of zero coefficient before a nonzero one. Note that k_z is approximated to be the nearest integer for convenient calculation.

To compare the two encoding length, we use Eq.(13) minus Eq.(14) and take the part in the bracket as $f(k_z)$,

$$l_1 - l_2 = \left[l_0 \cdot k_z - (3+l_0)g(k_z) + l_0\right] I = f(k_z) \cdot I \tag{15}$$

As I is a constant, we only consider function f of variable k_z. Put Eq.(9) into Eq.(15), we have

$$f(k_z) = \begin{cases} -3, & k_z = 0 \\ l_0 \cdot k_z - (3+l_0)\lceil \log_8(k_z+1) \rceil + l_0, & else \end{cases} \tag{16}$$

For discrete integer k_z, difference between adjacent items is calculated:

$$\begin{aligned} f(k_z) - f(k_z - 1) &= \begin{cases} f(0) = -3; \\ f(k_z) = f(0) + l_0 k_z - (3+l_0)\lfloor \log_8 k_z \rfloor \end{cases} \\ &where\ k_z \in Z_+, l_0 \in Z_+ \end{aligned} \tag{17}$$

For this interleaved arithmetic progression, it can be easily derived from Eq.(17) that $f(k_z)$ can only be negative at the very beginning of k_z for l_0 no less than 1. With the increase of frequency, $f(k_z)$ has a tendency of raise, making $f(k_z)$ has only one intersection point with zero, where $\lfloor \log_8 k_z \rfloor$ is equal to zero. In this intersection, we can derive Eq.(18) from Eq.(17)

$$k_z = \frac{3}{l_0} \tag{18}$$

Consequently, the threshold between the two methods can be determined according to Eq.(12) and (18). But there still have problems in the parameter choosing step. As we have mentioned, there is a deadlock among the boundary B, threshold T_{HZ} and Huffman Code Table. The problem lies that, T_{HZ} to be calculated by k_z (Eq.(12)) depends on l_0 (Eq.(18)), which is determined by the Huffman Code Table. However, the ***HCT*** calculate the zeros before boundary B, which results from T_HZ. To break this deadlock, we initialize k_z by an assumption $l_0 = 1$ in Boundary Descent algorithm.

Algorithm 1 demonstrates the flow of this algorithm. Let ε be the flag of iteration, $l_0(HCT)$ is the length of zeros Huffman Encoding in ***HCT***. Starting with the assumption that l_0(***HCT***)is 1, the algorithm initializes k_z to be 3 (line 1). While the iteration flag is true, calculate the zero distribution from all samples (line 3-7). Then the method threshold T_HZ is calculated (line 8) to derive boundary B (line 9-10) via Eq.(11,12). Taking B and quantified signal H_c^Q as input, Huffman Code Table ***HCT*** is derived (line 11). Then our assumption at the beginning is checked that whether parameters calculated satisfy the condition in Eq.(18). If the derived l_0 do not consistent with the assumption, substitute it for $\frac{3}{l_0}$ and do next iteration, until the boundary is meet at l_0(***HCT***)$\cdot k_z = 3$ (line 12-16). Notice that

Algorithm 1: BOUNDARY DESCENT ALGORITHM

Input: Quantized HAC to be compressed H_c^Q
Output: ***HCT***,B

1 $k_z \leftarrow 3, \varepsilon \leftarrow 1$
2 **while** $(\varepsilon = 1)$ **do**
3 **for** $c \leftarrow 1 to N_c$ **do**
4 **for** $i \leftarrow 1 to S_b$ **do**
5 $\Omega_0(i) \leftarrow \sum_{c=1}^{C} 1_{\{H_c^Q[i]=0\}}$
6 **end**
7 **end**
8 $T_{HZ} \leftarrow \frac{k_z}{k_z+1}$
9 $x = CrossSegment\left(\Omega_0, T_{HZ} \cdot \frac{S_b}{N_s}\right)$
10 $B = \left((x-1) + \frac{T_{HZ} \cdot \frac{S_b}{N_s} - \Omega_0(x-1)}{\Omega_0(x) - \Omega_0(x-1)}\right) \cdot \frac{S_b}{N_s}$
11 $HCT \leftarrow HuffmanEncoding(H_c^Q, B)$
12 **if** $(l_0(HCT) = \frac{3}{k_z})$ **then**
13 $\varepsilon \leftarrow 0$
14 **end**
15 **else if** $(l_0(HCT) \leq 3)$ **then**
16 $k_z \leftarrow \frac{3}{l_0(HCT)}$
17 **else**
18 $B \leftarrow 0,\ \varepsilon \leftarrow 0$
19 **end**
20 **end**
21 **return HCT**,B

l_0 should be limited within 3, or it is unnecessary to use Huffman Encoding (line 17-19). This algorithm returns ***HCT*** and boundary B eventually (line 21).

The key point of this algorithm is iteratively checking our assumption to suffice eq. (16). With the initialization of l0=1 and kz =3, if the acquired HCT not satisfies $l_0(\boldsymbol{HCT})\cdot k_z = 3$ in an iteration, i.e., l0(***HCT***) is larger than expected, then kz will be decreased in line 12, making THR smaller (line 6) and dwindling B, as illustrated by fig.6. This makes l0 no shorter than the current one, just in accordance with our target. This loop will eventually converges as at $l_0(\boldsymbol{HCT})\cdot k_z = 3$. Thus the boundary declines iteratively and achieves the optimal according to Boundary Descent Algorithm.

5.4. Data Format for Encoding

The previous sections discussed the key processing steps of the DCT-based and lossless encoding algorithm. In this section, the data format of compressed signal is expounded. As mentioned in the preprocess step, signal recorded in all channels is first grouped into blocks at time domain. Each block contains N_c channels, which is recorded in the order illustrated in fig.(7).

According to the three binary encoding methods, code of each channel includes three parts, representing the result of Huffman Encoding, Zero-Length-Encoding and Symbol Encoding respectively. For the High Amplitude Component after quantized, assume the Huffman Code Table shows ***HCT***(0) = 1, ***HCT*** (3) = 010, ***HCT***(6) = 110010. It is understandable in Huffman Encoding where the ***HCT*** is directly used to represent values. In Zero-Length-Encoding, $(3+l_0)g(k_z)-l_0$ bits are used to record the number of zeros k_z before each nonzero coefficient (in this case, $l_0 = 1$). Take the same recorded binary digits 010 1 110 as example, in Huffman Encoding, the decode result is three values: 3,0,-6 while in Zero-Length-Encoding it denotes 22 (2*8+6) zeros and a value 3. After encode the high amplitude component, the encoding of recover the Low Amplitude Component is recorded at last, representing sign of each value. This is implemented by Symbol Encoding, as illustrated in fig. 7. In the encoding process, encoding results derived from the three methods are connected to get a binary stream. However, it is necessary to separate them for decoding by different methods when decompressing. Two margins should therefore be saved at the begining, one is between Huffman Encoding and Zero-Length-Encoding, the other is the coding length after Zero-Length-Encoding. Besides, the Huffman Code Table and quantization table should also be recorded.

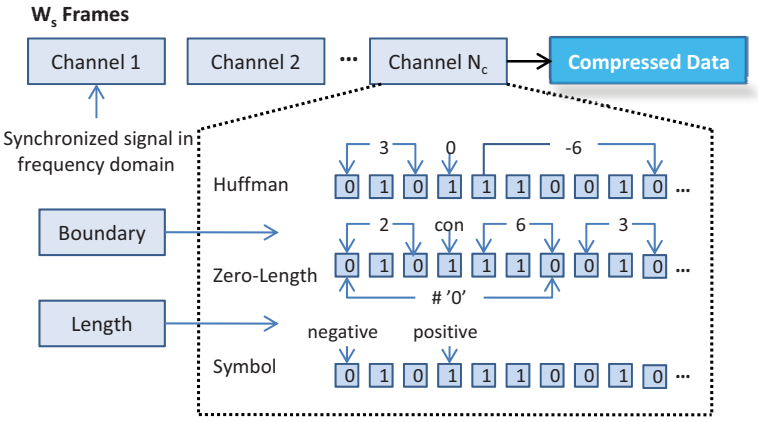

Figure 7: Schematic the compressed data format

6. EXPERIMENTAL RESULTS

This section presents results of our compression methods applied to a corpus of neural signals examinations randomly extracted from 2 monkeys of ZJU in the last four years. Compression performance compared with other signal compression techniques are presented as well.

6.1. Dataset

Dataset of our experiments comes from two male rhesus monkeys (Macaca mulatta) at Zhejiang University Qiushi Academy BMI system [14]. In order to train the monkey, a training system was established. In this system, each monkey was trained to perform a four-direction centered-out task by turning a joystick (by its hands) to move a cursor and hit the target according to the prompt (up, down, left, right) on a visual display. After acquiring this task, the monkey was implanted with a multi-electrode array in the primary motor cortex (M1) of the cerebral hemisphere contralateral to track the neural signal as

hand performing. Each experiment takes for approximately 60 min.

The present work is performed on a multi-channel acquisition device Cerebus 128TM (Blackrock Microsystem, Salt Lake City, UT, USA) to record 106 neurons signal simultaneously. The signal is sampled from 96-electrode array (1.0 mm electrode length; 7.0 cm wire bundle length), i.e, 96 channels, with 16-bit accuracy at a sampling rate of 30 kHz. The experiment result to data stream with 5.76MB/s, leading to 1.73GB signal recorded in 5 minutes. To verify our compression algorithm, we randomly selected 12 records with each has a length of approach to 300s.

6.2. Criterions Used

To ensure the available exploration on the electroneurographic signal, the compression algorithm is supposed to reduce information while retaining all relative information in the reconstructed signal [24]. From this perspective, a compression criterion is required to measure the compression performance. In communication theory, Signal to Noise Ratio (SNR) is used to judge the fidelity of compressed data by comparing original signal with the reconstruction error. Let S_o and S_r be the original signal and recovered signal in a channel respectively, SNR is defined by the power of So divided by the power of background noise:

总分结构。我们用了两个criteria，分别是……他们用什么表示

$$S/N = \frac{\|S_o\|_2^2}{\|S_o - S_r\|_2^2} \tag{19}$$

As a power-based criterion, SNR can well reflect the proportion of error energy, but it is insufficient. Look back to the first characteristic of electroneurographic signal declared in Section 3, the signals power is centralized in the low frequency. This makes low frequency occupies a much more important position in SNR, while the high frequency suffers from an inequality. To reconcile this discrepancy, we focus on spike, the primary concerns of the high frequency in the second criterion.

To examine the retention of spikes, the Spike Ratio is taken into consideration, representing the ratio of spikes reserved after reconstruction. In our validation, the well-known amplitude threshold technique [33] is used for spike detection, where the threshold (Thr) is set by

$$Thr = \alpha \cdot \sigma_n,\ \sigma_n = median\left(\frac{|x|}{0.6745}\right) \tag{20}$$

where α is a constant factor, σ_n is an estimate of the standard deviation of the background noise. A point is regarded as the beginning of a spike with amplitude higher than Thr. Notice that this is not only an issue of counting, but counting the spikes correctly matched. In this process, spikes are detected on both original and the decompressed one, the percentage of matched spikes is defined as Spike Ratio.

（SR）

To help understanding and perceiving the reconstruction performance, fig. gives an example of a short segment truncated from two compressed results with different SNR and Reserved Spike Ratio, where black line represent the original signal, blue and green are the reconstructed ones, detected spikes are labeled by short line segments at the bottom. It can be seen that the spike detected at x=779 in original and right signal is not detected in the first left one. This is because the signal has some loss at this point, making it miss the spike threshold in Eq.(20). Notice that higher SNR usually accompanies with higher Spike ratio, but it is not always go in this way. Because the SNR more reflects the signal power at low frequency as mentioned before.

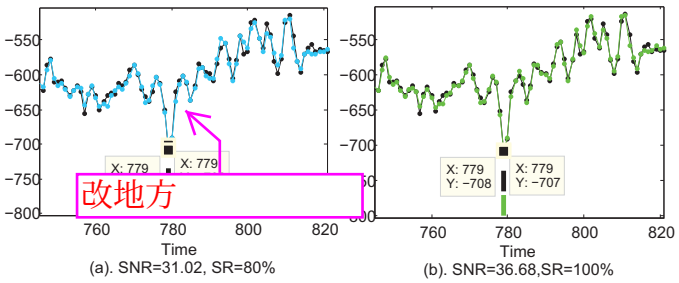

Figure 8: A truncated segment of two reconstruction results for a same original signal (black). The two reconstructions truncations come from the result of (a). SNR = 31.02, Spike Ratio = 80% of the blue line, (b). SNR = 36.68, Spike Ratio = 100% of the green line, respectively. The line segments at the bottom shows the detected spike positions.

In addition to measure the signal fidelity by SNR and the spike retention, compression ratio (CR) also related to the compression effect, in a data reduction perspective. The compression ratio is defined as the size ratio of compressed file to the original one.

To summarize, our proposed method use three criteria to evaluate a compression method, they are SNR, Spike ratio and Compression ratio.

6.3. Parameter setting

~~For neural signal compression, one does not have the luxury of varying the parameters for good performance. Hence, some fixed parameters or parameter choosing strategies that work across a wide range of samples is sought~~. The determination of Huffman Code Table and boundary for lossless encoding algorithm has been mentioned before. In this section, we investigate the choosing of three parameters: T_{LH}, the threshold of Amplitude Filter; ω, the scale of Quantization Table and S_b, the size of a block grouped in preprocessing.

In the basis of our proposed compression method, information losses from two operations fundamentally: the mean substitution strategy in Symbol Encoding for low amplitude component, and quantization for high amplitude component. Both of them compress in frequency domain, but Symbol Encoding losses at most the corresponding value in Quantization Table, while the quantization step losses less than half of those. The choice of compression method depends on the value belongs to low or high amplitude component, where the boundary between them is the threshold T_{LH}. As T_{LH} increases, more DCT coefficients are processed by Symbol Encoding, bringing more loss while decreasing compresson ratio. Therefore, the deterministic of T_{LH} can be viewed as a tradeoff between reconstruction performance and compression ratio. In our simulation, T_{LH} is tested on entire dataset for an optimal compressing result.

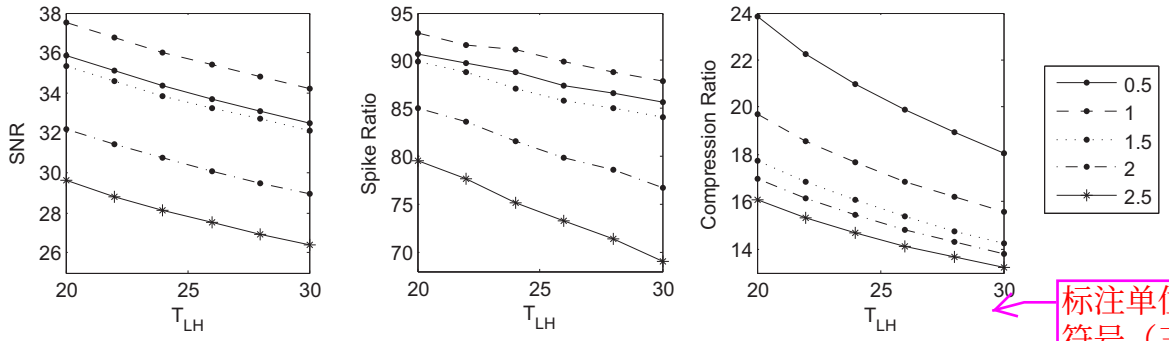

Figure 9: SNR, Spike Ratio and Compression Ratio for different T_{LH} and QT scenarios. The horizontal axis of each subplot is the thre Table at different scales are represented by factor ω = 0.5(ł), 1 (—), 1.5 (), 2(-.-.-) and 2.5(-*-*-).

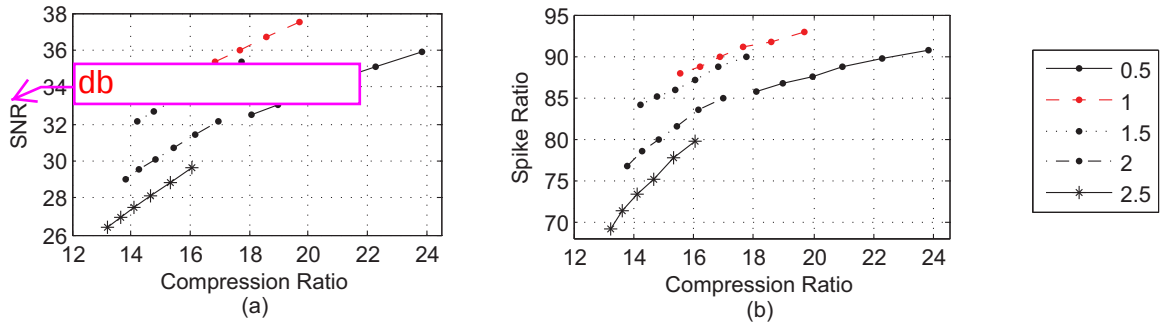

Figure 10: Compression performance comparison of different QT scales. The horizontal axis represents the compression ratio and the vertical axis represents (a). SNR, (b). Spike Ratio. Each curve corresponds to the performance taking different threshold under different QT scales.

Another important parameter is quantization table (QT). In order to save less parameter for decompression, QT is shared with Symbol Encoding in our proposed method, which is designed to be the average magnitude of Low Amplitude Component. Therefore, QT increases with T_{LH} and coarsen the compression result. However, QT is usually to be composed of small value under a small T_{LH}. Therefore, wed like to know whether the compression result will be improved by scaling QT, that is, fine-tuned QT by multiplying different factors from 0.5 to 2.5 with equal difference of 0.5. Due to the association between T_{LH} and QT, they are measured simultaneously as two variables.

Results of our simulation under different parameters settings are shown in fig. 9. Each evaluation is obtained by the average result on the whole dataset by a systematic variation of T_{LH} and Quantization Table multiplier ω. With the increase of T_{LH}, it is clearly that both SNR and Spike Ratio went down under the same ω, because higher threshold will amplify QT. At the cost of reconstruction performance, compression ratio reduces from saving less information. Performance of different ω is also shown in this figure. Accordingly, ω=1 is always optimal in the view of SNR and Spike ratio, and its compression ratio ranks second. It is easy to understand the compression ratio ranks in the order of ω, for larger ω compresses more information. For SNR and Spike Ratio, it can be comprehend in this way. As a critical factor in Quantization, smaller QT will refine the quantized DCT coefficients undoubtedly. However, the QT is defined as the average amplitude under T_{LH} in corresponding position, which would be the best substitution in Symbol Encoding if the variance of LAC is not very large.

Fig. 9 above gives us an intuitive concept of how ω affect the fidelity and compression ratio respectively, but it is still hard to select ω because the multiple criteria. Therefore, signal fidelity is compared under fixed compression ratio as shown in fig. 10. This is a transpose of fig.9, with compression ratio as independent variable, (a) SNR and (b) Spike Ratio as dependent variables. For same compression ratio (at horizontal axis), we see that with ω equal to 1 consistently outperforms others. Therefore, for unsupervised compression, QT with multiplier of one offers a reasonable compromise between compression ratio and reconstruction performance, becoming a good choice for this parameter.

The choice of T_{LH} depends on our compression requirement on compression ratio. For a desired SNR higher than 30db and Spike ratio no less than 90%, T_{LH} = 24 is selected, resulting to average compression ratio at 17.75%, SNR at 36.24dB and Spike ratio at 90%.

The last undetermined parameter is the size of a block divided in the preprocessing. All the above experiments are taken on a fixed grouped block size of Sb=1600, but it is still to be explored whether compression result can be improved by changing it. Our experiment is shown in fig. 11, with S_b tested between 1500 to 28500, T_{LH} = 24 and ω=1.

This figure illustrates that with the increasing of S_b, the signal fidelity firstly increases and then decreases. This phenomenon can be interpreted in the following way. First, larger S_b brings more refined DCT coefficients, therefore, decreasing the error in decompression at IDCT (inverse DCT). However,

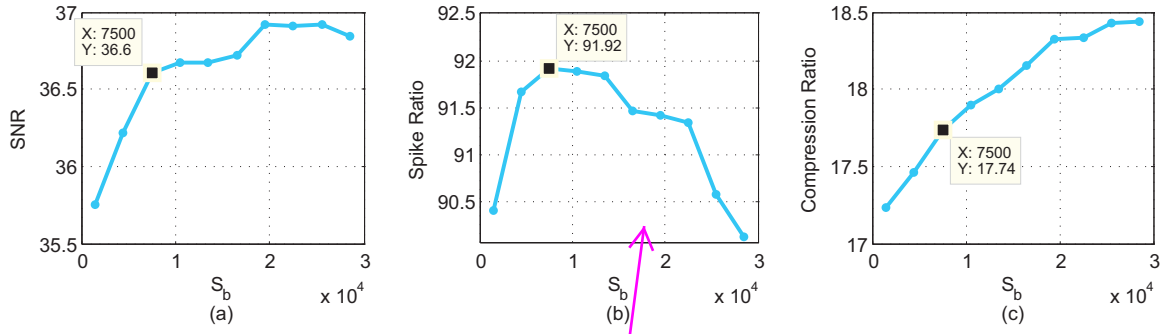

Figure 11: Compression result of different block size Sb. The block.

smaller amplitudes are assigned to more elaborated DCT coefficients, making more coefficients involved into the low amplitude component, which means more loss according to the previous analysis. Therefore, the increase first and then descend fidelity reflect the negotiation between the two factors on reconstruct error. According to the given experiment result, an optimal size of block is determined at $S_b = 7500$, achieving average compression ratio at 17.7%, SNR at 36.6dB and Spike ratio at 91.9% on the entire dataset.

6.4. *Effect of Symbol Encoding*

To compress the DCT coefficients with low amplitude effectively, Symbol encoding method has been proposed according to section 5.1. This section gives experimental results to prove the feasibility of it. A contrast experiment without symbol encoding has been tested on the whole corpus. Last subsection we have chosen the optimal parameter. Given these settings as prior, this section calculated the improvement brought by symbol encoding. The comparison result is shown in Table 1.

Symbol Encoding	SNR(db)	SR	CR
with	31.4	83.7%	13.1%
without	36.6	91.9%	17.7%

Table 1: SNR, Spike Ratio and Compression Ratio With and Without Symbol Encoding (Low-Amplitude)

It is clearly that taken the Low-Amplitude part into consideration has to save more information, i.e., leads to higher compression ratio, but the reconstruct performance has been greatly improved in with LA encoded. This is achieved by the inherent coherence of coefficients in the high frequency part. Therefore, symbol encoding is adopted as an effective way to preserve high frequency signal.

6.5. *Comparison with other approaches*

For the lack of other neural signal compression standards, the state-of-art audio compression algorithms are considered to be contrast with our compression algorithm as another one-dimension signal family. Some traditional data compression methods are also compared to ours. By investigating the compression result of both lossless and lossy audio compression algorithm, we find that our compression enables a balance of compression ratio and reconstruction performance to be achieved.

6.5.1. *Lossless compression*

Lossless audio compression and generic data compression techniques are explored in this section. Lossless compression methods produce exact duplicate of the original file and information will be reduced by a more compact coding format. For audio compression, Codecs like FLAC use linear prediction to estimate the spectrum of the signal, achieving a compression ratio at 50%-60% for general waveforms [23]. However, unlike audio signal, neural signal is more complicate and difficult to be predicted. Consequently, such compression methods applied to neural signal cannot attain lower compression ratio. Likewise, data file compression format such as Zip, 7-Zip and RAR also cannot achieve a relative high compression ratio. Table 2. shows the compression ratio of different lossless compression techniques. The best compression method for the given neural data is APE (Monkeys Audio), attaining lowest compression ratio at 56.88%. Note that, although neural signal cannot be compressed effectively, file can be exactly reconstructed from the compressed one, i.e., SNR is infinite.

6.5.2. *Lossy Compression*

Different from lossless compression, lossy audio compression takes advantage of human acoustic perception that only sensitive to specific frequency band and amplitude, loss some trivial signal that not contribute to the discrimination of audio, only quantify and encode the perceptible parts. As a state-of-art audio encoding algorithm, Advanced Audio Coding (AAC) is used on neural signal and compared to our coding method. AAC is part of MPEG-2 standard and provide better signal quality than MP3 with 30% reduce of file size. Fig. 12 shows the Compression ratio C reconstruction performance comparison between the two methods.

In audio compression, the compressed signal quality improves with increasing bitrate. Simulation in this case uses high variable bitrate ranging from 300kbps to 600kbps, intending to achieve good reconstruction performance. However, the result is not ideal for electroneurographic signal. Fig. 16 illustrates

	Lossless Compression Techniques					
Compression Ratio	Audio Codec			Data File Compression		
	Lossless WMA	FLAC	APE	Zip	7-Zip	RAR
	70.89%	70.50%	58.72%	70.04%	59.58%	60.91%

Table 2: Lossless Compression Techniques Compression Result

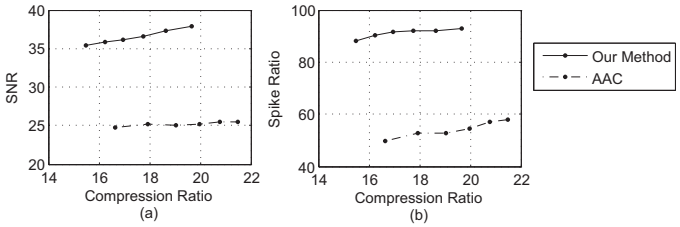

Figure 12: Compression Result comparison between AAC encoding and our compression method.

that our method is higher than AAC in both SNR and Spike ratio, under the same compression ratio. The obvious drawback comes from that audio compression method doesnt care about what is significant to neural signal processing. Therefore, lots of signal at high frequency is loss for the useless for audio discrimination, making AAC achieves an extremely low reserved spike ratio.

6.6. Computational cost

Previous sections have demonstrated the availability of our method for balancing compression ratio and reconstruction error. At last, the compression efficiency is taken into consideration. The following results come from the statistical experimental result implemented on MATLAB. The initialization process to get Quantization table, Huffman Code Table and boundary in Lossless encoding needs 2.86Mb/s, the compression consumes 0.13Mb/s and decompression speed is 0.14Mb/s. The algorithm compression process would be speed up by other program languages and hardware implementation, which would be explored in the future.

7. CONCLUSION

In this paper, a lossy compression algorithm for electroneurographic signal is given. It is based on the dual-phase encoding after spectrum analysis. The temporal signal transformed into frequency domain is passed through an amplitude filter, then compress the low and high amplitude component separately in the dual-phase encoding. To compress the low amplitude component, we use a Symbol Encoding method, recording only one symbol for each value; to the high amplitude component, it is quantified and encoded by Hybrid Lossless Encoding.

Comparisons between the proposed method and other compression techniques have been performed. It has been found that our method outperforms others in reconstruction performance (SNR and spike ratio) with fixed compression ratio on the neural signal obtained from Multiple Electrode Array, achieving at a compression ratio of 17.7% with SNR values at 36.6dB and reserved 91.9% of original spikes. The result is outstanding compared to other biomedical signal compression methods, which attach a SNR at 15-26dB and compress to 1%-20% of original file [15, 16, 25]. Our proposed method is validated on the motor cortex electroneurographic signal, but it can also be applied to electroneurographic signals besides recordings from motor cortex. With the correlation between channels in other related signals, compressed file can be further reduced without significant decreasing signal quality.

Moreover, the average frequency for quantization table, Huffman Code Table and other parameters can be predefined adaptively according to the signal to be processed. The prototype was successfully tested in the sampled Rhesuss motor cortex neural signal obtained at Zhejiang University Qiushi Academy for Advanced Studies. Compression result is desirable in compression ratio as well as reconstruction performance. The obvious drawback, however, is that it is time consuming in the process of compression and decompression. But the process speed can be improved by rewriting the algorithm from MATLAB to C in the further work.

Acknowledgements

The work is partially supported by NSFC

Reference

[1] György Buzsáki, Costas A. Anastassiou and Christof Koch The origin of extracellular fields and currents-EEG, ECoG, LFP and spikes Nature Reviews Neuroscience 13, June 2012, 407-420

[2] Hansjörg Scherberger, Murray R. Jarvis, Richard A. Andersen Cortical Local Field Potential Encodes Movement Intentions in the Posterior Parietal Cortex, Neuron, Volume 46, Issue 2, 21 April 2005,347-354

[3] Kraskov A, Quiroga RQ, Reddy L, Fried I, Koch C. Local field potential and spikes in the human medial temporal lobe are selective to image category Journal of Cognitive Neuroscience archive, Volume 19 Issue 3, March 2007, 479-492

[4] Gilja V, Chestek C A, Diester I, et al. Challenges and opportunities for next-generation intracortically based neural prostheses. IEEE Trans Biomed Eng, 2011, 58: 1891-1899

[5] Vaadia E, Birbaumer N. Grand challenges of brain computer inter-faces in the years to come. Front Neurosci, 2009, 3: 151-154

[6] Hatsopoulos N G, Donoghue J P. The science of neural interface systems. Annu Rev Neurosci, 2009, 32: 249-266

[7] Schwartz A B, Cui X T, Weber D J, et al. Brain-controlled interfaces: Movement restoration with neural prosthetics. Neuron, 2006, 52:205-220

[8] Lebedev M A, Nicolelis M A. Brain-machine interfaces: Past, present and future. Trends Neurosci, 2006, 29: 536-546

[9] Nicolelis M A. Actions from thoughts. Nature, 2001, 409: 403-407

[10] Donoghue J P. Bridging the brain to the world: A perspective on neural interface systems. Neuron, Nov. 2008, 60: 511-521

[11] Chapin J K, Moxon K A, Markowitz R S, et al. Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex. Nat Neurosci, 1999, 2: 664-670

[12] P. J. Rousche and R. A. Normann, B Chronic recording capability of the Utah intracortical electrode array in cat sensory cortex. J Neurosci Methods. 1998 Jul 1;82(1):1-15.

[13] D. R. Kipke, R. J. Vetter, J. C. Williams, and J. F. Hetke, B Silicon-substrate intracortical microelectrode arrays for long-term recording of neuronal spike activity in cerebral cortex. IEEE Trans Neural Syst Rehabil Eng. 2003 Jun;11(2):151-5.

[14] Zhang Q S, Zhang S M, Hao Y Y, et al., Development of an invasive brain- machine interface with a monkey model. Chinese Science Bulletin, 57(16):2036-2045, 2012

[15] Chen Han Chung, Yu-Chieh Kao, Liang-Gee Chen, Fu-Shan Jaw Intelligent Content-Aware Model-Free Low Power Evoked Neural Signal Compression. Advances in Multimedia Information Processing - PCM 2008, Lecture Notes in Computer Science Volume 5353, 2008, pp 898-901

[16] Chen Han Chung, Liang-Gee Chen et al, Multichannel Evoked Neural Signal Compression Using Advanced Video Compression Algorithm, Proceddings of the 4th International IEEE EMBS conference on Neural Engineering, 2009

[17] Goh A, Craciun S, Rao S et al. Wireless transmission of neuronal recordings using a portable real-time discrimination/compression algorithm. Conf Proc IEEE Eng Med Biol Soc. 2008

[18] Nikos K. Logothetis, "The Underpinnings of the BOLD Functional Magnetic Resonance Imaging Signal", J. Neurosci. 2003 May 15;23(10):3963-71.

[19] G Antoniol and P Tonella EEG data compression techniques. IEEE Trans. Biomed. Eng., vol. 44, no.2, pp. 105-114, February 1997

[20] U. Mitzdorf,"Current source-density method and application in cat cerebral cortex: investigation of evoked potentials and EEG phenomena",Physiol Rev January 1, 1985 65:(1) 37-100

[21] W. Hunt, B. Maher, K. Coons, D. Burger, K.S. McKinley, Optimal Huffman tree-height reduction for instruction level parallelism, unpublished manuscript provided by authors, 2006

[22] Huffman, D.A. A method for the construction of minimum redundancy codes. Proceedings of IRE , Sept. 1952, vol. 40, Iss:9. 1098-1101.

[23] Coalson, Josh. "FLAC Comparison". Retrieved 6 March 2013.

[24] Monica Fira and Liviu Goras, Biomedical Signal Compression based on Basis Pursuit, vol. 14, pp. 53-64, Jan 2010

[25] S. Aviyente, Compressed sensing framework for EEG Compression, IEEE workshop on statistical Sig. Proc., pp. 181-184, August 2007

[26] A Belitski et al. Low-Frequency Local Field Potentials and Spikes in Primary Visual Cortex Convey Independent Visual Information, J Neurosci. 2008 May 28; 28(22):5696-709.

[27] Jacobsen C.F. Functions of the frontal association area in primates. Arch NeurPsych. 1935;33(3):558-569.

[28] Birgitta Weber, Thomas Malina, Kerstin M. L. Menne, Volker Metzler, Andre Folkers, Ulrich G. Hofmann Handling large files of multisite microelectrode recordings for the European VSAMUEL consortium, Neurocomputing 01/2001; 38-40:1725-1734.

[29] Murata Y, Iwasaki H, Sasaki M, Inaba K, Okamura Y. "Phosphoinositide phosphatase activity coupled to an intrinsic voltage sensor.", Nature. 2005 Jun 30; 435(7046):1239-43.

[30] Bruno B. Averbeck, Peter E. Latham, and Alexandre Pouget Neural correlations, population coding and computation, Nature Reviews Neuroscience 7, May 2006, 358–366

[31] Anne Hsu, Alexander Borst and Frédéric E Theunissen Quantifying variability in neural responses and its application for the validation of model predictions, Network, 2004 May;15(2):91-109.

[32] Prentice JS, Homann J, Simmons KD, Tkačik G, Balasubramanian V, et al. (2011) Fast, Scalable, Bayesian Spike Identification for Multi-Electrode Arrays. PLoS ONE 6(7): e19884.

[33] Quiroga RQ, Nadasdy Z, Ben-Shaul Y. "Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering.",Neural Computation. Aug. 2004; 16(8): 1661 - 1687.