

Impact of Lossy Compression on Neural Response Characteristics extracted from High-Density Intra-cortical Implant Data

Michael A. Shetliffe, Awais M. Kamboh, Andrew Mason and Karim G. Oweiss

Abstract—In this paper we examine the impact of lossy wavelet compression on the information contained within high-density microelectrode array neural recordings. We have previously reported on the ability of our hardware architecture to perform under the constraints imposed by implantable hardware, as well as on its performance from a compression and signal distortion standpoint. Here we extend that work by examining the amount of information that is lost from the recorded data as a result of the finite precision integer arithmetic and thresholding operations inherent in our system. One method commonly used for the classification and sorting of recorded extracellular action potentials is principal component analysis. This technique is used to statistically obtain the most significant attributes of the spikes, thereby allowing for more accurate classification. We use the separability of the resultant clusters as a measure of the information content within the data, and present the results of simulations demonstrating the impact of various hardware design parameters on this separability.

I. INTRODUCTION

The recording of large ensembles of neurons in the brain using implanted high density microelectrode arrays is becoming increasingly popular for neuroprosthetic devices and brain machine interfaces (BMIs). Such recordings provide a greater insight into underlying neural function and connectivity, however they do so at the cost of a vastly increased volume of data requiring processing and transmission to the extra-cranial space. A typical state-of-the-art array can have as many as 1000 electrodes integrated into a single device, however a current generation device may only be capable of transmitting a small fraction of those channels in real time due to telemetry bandwidth restrictions. Therefore, any on-chip processing that can be done to reduce the amount of data requiring transmission will reduce transmission latencies and improve real time performance.

Previous work in our lab [1] has demonstrated the applicability and overall effectiveness of using wavelet transform techniques to reduce the required bandwidth for transmission of data with minimal distortion of the output signal reconstructed from thresholded (denoised) wavelet coefficients. Furthermore, we have also shown that this performance can be achieved in real-time with a custom VLSI architecture under the severe constraints of implantable hardware [2]. Additionally, it has been demonstrated that not only does the wavelet transformation provide this necessary compression,

This work was supported in part by the National Institutes of Health under Grant NIH-NINDS-NS047516.

M. Shetliffe, A. Kamboh, and A. Mason are with the ECE Dept at Michigan State University, East Lansing, MI 48824 USA (shetliff@egr.msu.edu, kambohaw@msu.edu, mason@egr.msu.edu).

K. Oweiss is with the ECE Dept and Neuroscience Program at Michigan State University, East Lansing, MI 48824 USA (koweiss@msu.edu).

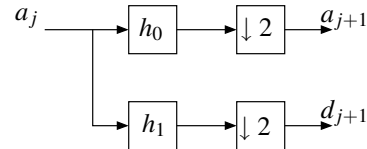


Fig. 1. Mallat's filter bank algorithm for computation of wavelet transform coefficients via recursive convolution of approximation coefficients from the previous level with high-pass, h_1 , and low-pass, h_0 , filters followed by decimation in time.

but that the compressed representation itself captures unique features of recorded extracellular spike waveforms, allowing for improved classification of their source [3].

As a next step, we are therefore interested in analyzing to what extent these underlying spike waveform features (which we loosely refer to as the *information* content of the spike train) are retained when the wavelet processing is performed using the hardware architecture we have already developed. In the work presented here, we compare classification performance using Principal Component Analysis (PCA) of spike waveforms before and after processing with this architecture.

Section II of this paper follows with some brief theoretical background, and we then describe our simulations and results in Sections III and IV respectively.

II. THEORY

A. The Lifting Wavelet Transform

The primary motivation for wavelet processing of the recorded multielectrode neural data is its excellent ability to denoise and compress the data as originally demonstrated in [4]. Essentially, the signal can be expressed in relatively few large coefficients, while the noise becomes expressed in many small coefficients that can be removed by thresholding.

The most common architecture for computation of the wavelet transform coefficients is that shown in Fig. 1 in which the approximation coefficients from level j are convolved with the decomposition filters h_0 and h_1 and then decimated in time to obtain the approximation and detail coefficients at level $j+1$ [5].

The lifting wavelet transform [6] allows for more efficient computation of these same wavelet coefficients through a scheme that splits the incoming coefficients from level i into even and odd samples, and then predicts the odd samples from the even samples resulting in the d_{i+1} coefficients as the prediction error, and updates the even samples with the d_{i+1} coefficients to obtain the a_{i+1} coefficients.

The basic lifting structure is shown in Fig 2 where the filters h_0 and h_1 have been factored into N lifting steps

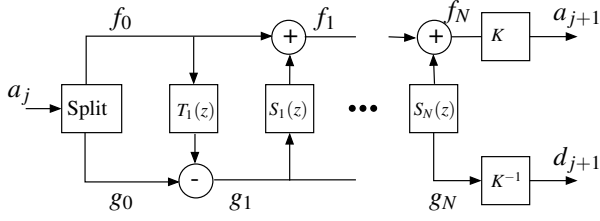


Fig. 2. Block diagram of the lifting scheme for computing a single level of wavelet decomposition. The filters T and S are obtained from factorization of the f and g filters

TABLE I
SYMMLET 4 LIFTING COEFFICIENTS

Coef.	Value	Scaled	Coef.	Value	Scaled
C_1	0.39114	6	C_5	0.16203	3
C_2	-0.12439	-2	C_6	0.43128	7
C_3	-0.33924	-5	C_7	0.14598	2
C_4	-1.41951	-23	C_8	-1.04925	-17

represented by the update filters $T_N(z)$ and $S_N(z)$. For a lifting factorization of the symmlet 4 wavelet, which have been shown in previous work to yield the highest compression performance for neural signals [1], the filtering steps in the scheme can be written as [7]

$$\begin{aligned}
\text{step 1: } g_1[i] &= g_0[i] + C_1 f_0[i] \\
\text{step 2: } f_1[i] &= f_0[i] + C_2 g_1[i] + C_3 g_1[i+1] \\
\text{step 3: } g_2[i] &= g_1[i] + C_4 f_1[i] + C_5 f_1[i-1] \\
\text{step 4: } a[i] &= f_1[i] + C_6 g_2[i] + C_7 g_2[i-1] \\
\text{step 5: } d[i] &= g_2[i] + C_8 a[i-1]
\end{aligned} \quad (1)$$

The important point for the scope of this work, is that within our hardware implementation, we must quantize the incoming data (and the resultant wavelet coefficients), as well as the coefficients used in the lifting filters, C_1 through C_8 . For reference, Table I shows the symmlet 4 filter coefficients and their associated 6 bit (5 bit plus sign) integer values. It is the choice of the level of quantization used for these quantities, and the associated impact on the information content of the data, that we investigate herein. We shall denote the number of bits used for the time-domain data and wavelet coefficients by B_d , and the number of bits used for representation of the filter coefficients by B_c .

B. Principal Component Analysis

The objective of Principal Component Analysis (PCA) is to find those features within a set of data that best capture the variation in the data [8], thereby making the separation of different classes contained within the data more straight forward. In the present context, the data is a set of n spike realizations, each of length d samples, and contained within the matrix $S \in \mathbb{R}^{n \times d}$. By decomposing the $d \times d$ temporal covariance matrix, Σ , of S^T via the singular value

decomposition, we obtain

$$\Sigma = U D U^T = \sum_{i=1}^d \lambda_i \mathbf{u}_i \mathbf{u}_i^T \quad (2)$$

where λ_i denotes the i^{th} eigenvalue corresponding to the i^{th} diagonal entry in D , and the columns of U are the eigenvectors of Σ . The vectors \mathbf{u}_i represent the principal components of the original data S which are the directions of maximum variance within the data.

By projecting the original data onto these principal components through the linear transformation

$$\hat{S} = U^T S \quad (3)$$

we obtain a transformed dataset in which the major characteristics of the data — or *information* contained within the data — has been isolated.

In practice, the matrix S would be constructed through the isolation of individual spikes observed in multielectrode recordings for the purpose of assigning the spikes to particular neurons (i.e. spike sorting). For this study, however, we are essentially performing a supervised analysis in that we start with spike waveforms of a known class, and investigate whether there is any degradation in our ability to separate the classes contained within \hat{S} as a result of our hardware processing.

C. Performance Measure

As mentioned above, in this work we are looking at the information contained within a set of spikes as being related to our ability to classify the spikes as belonging to a particular neuron — that is, the *separability* of the clusters obtained from the PCA step in (3).

For a set of clusters, $\{C_i | i = 1, 2, \dots, p\}$, we define a performance metric, P , as

$$P(\{C\}) = \frac{\text{Average inter-cluster distance}}{\text{Average intra-cluster distance}} = \frac{d_{\text{inter}}}{d_{\text{intra}}} \quad (4)$$

The average inter-cluster distance is given by

$$d_{\text{inter}} = \frac{\sum_{i=1}^p \sum_{x \notin C_i} \left(\sum_{j=1}^{|C_i|} \|x - s_{ij}\| \right)}{\sum_{i=1}^p \sum_{j \neq i} |C_j|} \quad (5)$$

for p clusters (spike classes) with $|C_i|$ equal to the number of elements in cluster C_i , s_{ij} representing the j^{th} element of cluster i , and $\|\cdot\|$ representing the euclidean distance (L2 norm) between two d -dimensional cluster elements. This quantity provides a factor related to the overall *separation* between clusters.

The average intra-cluster distance is given by

$$d_{\text{intra}} = \frac{\sum_{i=1}^p \left(\sum_{j=1}^{|C_i|} \sum_{k=1}^{|C_i|} \|s_{ij} - s_{ik}\| \right)}{\sum_{i=1}^p |C_i| (|C_i| - 1)} \quad (6)$$

and provides a factor related to the overall *spread* within individual clusters. It can therefore be intuitively seen that this measure gives a quantitative indication of the *separability* of a set of clusters, since a greater overall spread within individual clusters will *reduce* separability, while a greater overall separation of clusters will *enhance* separability.

III. METHODOLOGY

We obtained 5 spike templates by averaging multiple occurrences of spikes from distinct neural sources that had been previously identified and isolated manually from data recorded from the dorsal cochlear nucleus of an adult guinea pig. These templates served as the mean vectors, μ_i , of $p = 5$ spike classes. These templates are shown in the right column of Fig. 3(a). In the left column we show the 128 realizations of each template that were used in this analysis. Each realization was drawn from a multivariate normal distribution using its corresponding template as a mean vector, μ_i , and a covariance matrix, σ_i , that was created from random elements, but corresponding in form to covariance matrices that have been observed in sets of spikes isolated from real neural recordings.

We developed a software simulation of the hardware wavelet processing that incorporated the effects of quantization of data and filter coefficients, and then compared the cluster performance (4) obtained from PCA of the original data, with that which had been decomposed with various amounts of quantization, and then reconstructed perfectly. Fig. 3(b) shows the spike waveforms projected onto their first 2 principal components, and Fig. 3(c) shows the same but for the coefficients across the nodes of the wavelet decomposition tree. The separability of the classes can be seen to vary in different nodes, illustrating the power of using the DWT not only for compression, but also for aiding in spike classification. It is a quantitative assessment of how this separability is affected by our hardware processing that we are investigating here.

IV. RESULTS

Simulations were run for a range of values of the parameters B_d and B_c . Fig. 4 shows the variation in performance, P , with quantization in the data and wavelet coefficients for various values of B_c . We observe increasing performance up to around 10 bits of data quantization, after which there appears to be minimal gain in performance. The dark dashed line shows the performance obtained with the original data (i.e. what would be obtained with perfect decomposition).

Fig. 5 shows the variation in performance with quantization in the filter coefficients. Again, the dark dashed line shows “perfect” performance. Here we observe that beyond around an 8 bit representation, there is no significant improvement in performance.

An interesting and unexpected result is the performance obtained across all results using 5 bit representations of the filter coefficients. (It should be noted here that performance with less than 5 bits was tested, but was found to be lower than all results presented here in all cases.) This appears

to indicate that the resultant filter that is obtained with 5 bit filter coefficients is somehow *better* at capturing the spike features (information) than the symmlet 4 filter with floating point representation (Table I) and therefore appears to warrant further investigation in another study.

Similarly, we show the cluster separability performance obtained across wavelet decomposition nodes (i.e. clustering of the actual wavelet coefficients in a given node) versus B_d and B_c in Figs. 6(a) and 6(b) respectively. In these results, Node i corresponds to the detail coefficients, d , (from Fig. 2) at level $i/2$ for even i , and to the approximation coefficients, a at level $(i+1)/2$ for odd i . We observe similar general characteristics in these curves to those obtained for the reconstructed data with respect to both B_d and B_c . These results also show clearly that in some nodes (e.g. nodes 6, 8, and 10), the cluster performance in the wavelet domain, even with some level of quantization, exceeds that of direct clustering of the time-domain signals. This serves as further validation of our overall approach and verifies that performance does not suffer in the hardware implementation.

V. CONCLUSION

We have presented extensions to our previous work that looked at the distortion produced with integer lifting wavelet processing, by examining the information loss within a PCA framework. The results show the clear benefit of working with a sufficiently high number of bits for the incoming spike train data and associated wavelet transform coefficients, as well as that using anything higher than our current planned implementation of 10 bits does not yield a significantly higher information output. With respect to the filter coefficient quantization, we find very little increase in information output beyond 8 bits. The combination of these results with those obtained from previous and ongoing studies discussed above will assist us in finalizing some of the key design decisions for the implantable hardware.

REFERENCES

- [1] K. G. Oweiss, “A systems approach for data compression and latency reduction in cortically controlled brain machine interfaces,” *IEEE Trans on Biomedical Engineering*, vol. 53, no. 7, pp. 1364–1377, July 2006.
- [2] K. G. Oweiss, A. Mason, Y. Suhail, A. M. Kamboh, and K. E. Thomson, “A scalable wavelet transform VLSI architecture for real-time signal processing in high-density intra-cortical implants,” *IEEE Transactions on Circuits and Systems*, vol. 54, pp. 1266–1278, June 2007.
- [3] K. G. Oweiss and D. J. Anderson, “Tracking signal subspace invariance for blind separation and classification of nonorthogonal sources in correlated noise,” *EURASIP Journal on Advances in Signal Processing*, vol. 2007, pp. Article ID 37485, 20 pages, 2007.
- [4] D. L. Donoho, “De-noising by soft-thresholding,” *Information Theory, IEEE Transactions on*, vol. 41, no. 3, pp. 613–627, 1995.
- [5] S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, 1999.
- [6] W. Sweldens, “The lifting scheme: A new philosophy in biorthogonal wavelet constructions,” in *Wavelet Applications in Signal and Image Processing III*, A. F. Laine and M. Unser, Eds. 1995, pp. 68–79, Proc. SPIE 2569.
- [7] K. Oweiss, Y. Suhail, K. Thomson, J. Li, and A. Mason, “Augmenting real-time DSP in implantable high-density neuroprosthetic devices,” in *3rd IEEE/EMBS Special Topic Conference on Microtechnology in Medicine and Biology*, 2005, pp. 108–111.
- [8] P. N. Tan, M. Steinbach, and V. Kumar, *Introduction to data mining*, Pearson Addison Wesley, Boston, 2006.

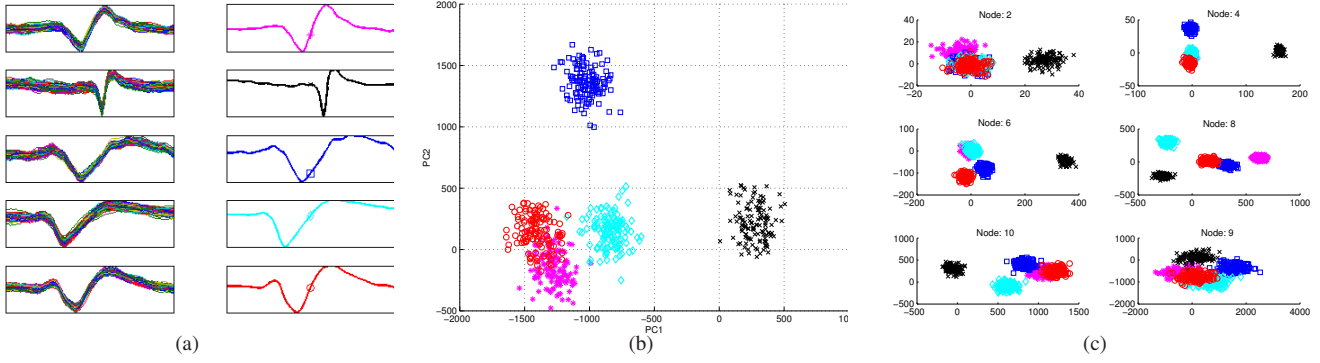


Fig. 3. (a) Spike realizations from 5 classes (left), with the mean spike waveform, μ_i , for each class, i , shown on the left. (b) First two dimensions of the principal components obtained directly from the original time-domain signals. (c) First two dimensions of the principal components obtained from the wavelet coefficients at various levels of decomposition showing visible changes in separability across classes in different nodes.

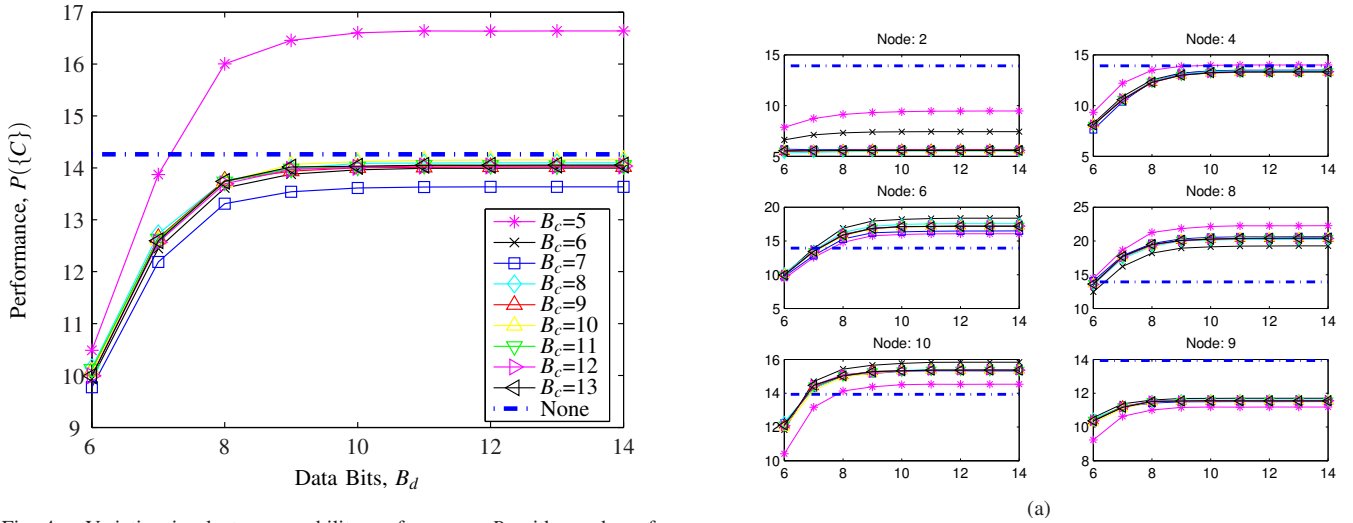


Fig. 4. Variation in cluster separability performance, P , with number of bits used for data quantization, B_d , for selected values of filter coefficient quantization, B_c . The thick dashed line represents the performance obtained without quantization.

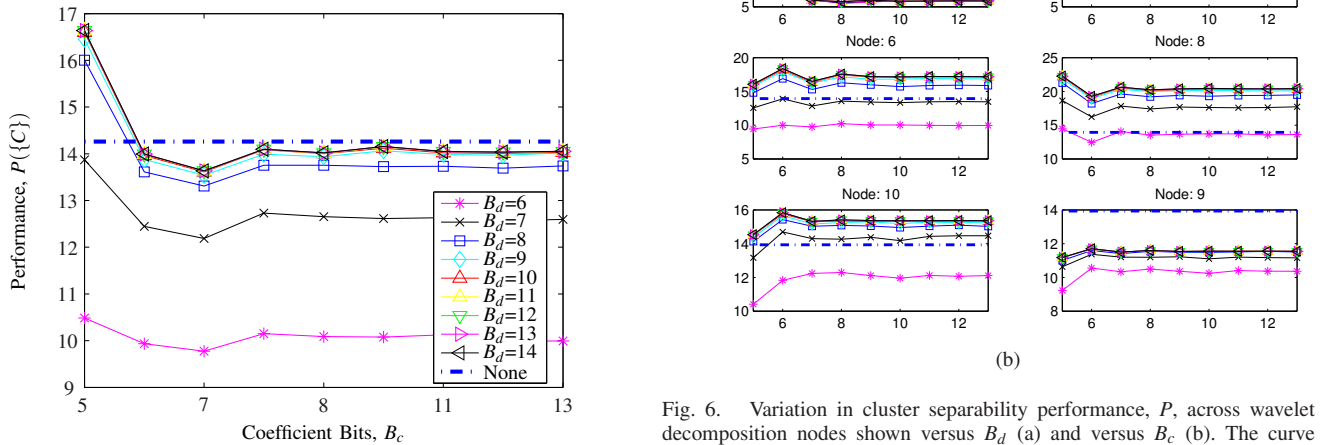


Fig. 5. Variation in cluster separability performance, P , with number of bits used for filter coefficient quantization, B_c , for selected values of data quantization, B_d . The thick dashed line represents the performance obtained without quantization.

Fig. 6. Variation in cluster separability performance, P , across wavelet decomposition nodes shown versus B_d (a) and versus B_c (b). The curve labels match those of Fig. 4 and 5. The dark dashed line shows the cluster separability of the time-domain signals without quantization, and we observe improved performance in certain wavelet nodes. Node i corresponds to the detail coefficients, d , at level $i/2$ for even i , and to the approximation coefficients, a at level $(i+1)/2$ for odd i .