Efficient Data Compression in Wireless Sensor Networks for Civil Infrastructure Health Monitoring

Shengpu Liu Liang Cheng
Department of Computer Science and Engineering
Lehigh University, 19 Memorial Drive West, Bethlehem, PA 18015
{shl204, cheng}@cse.lehigh.edu

Abstract- **In this paper, we present an efficient sensor data compression process for civil infrastructure health monitoring applications. It integrates lifting scheme wavelet transform (LSWT) and distributed source coding (DSC), which can reduce the raw data size by 1:27 to 1:80 while having a minor effect on the modal parameters identified from the sensor data. We have compared our algorithms with other data compression algorithms for structural health monitoring. Results show that our algorithms can achieve 80% ~ 100% higher compression ratios with the same signal-restoration quality.**

I. INTRODUCTION

Civil infrastructure systems are critical to the nation's economic growth and public safety. Sensor networking technologies show great potential in monitoring fields since they enable dense in situ sensing and simplify deployment of instrumentation. Therefore, wireless sensor networks (WSNs) are considered vital to the functioning of structural health monitoring systems. A crucial practical issue arising from the full-scale implementation of WSNs is the sensor data compression. Large volumes of sensor data generated will make the data transmission between sensor nodes and a remote data acquisition center a very challenging task, especially given the limited power and bandwidth of currently available wireless sensors. Data compression facilitates power conservation of WSNs since the energy of a sensor node is consumed primarily by wireless communications [1].

Data compression has been used to reduce the redundancy of the raw data. In a dense sensor network, the redundancy exists in both data collected at individual sensor node which is called local redundancy and data obtained from correlated sensor nodes called distributed redundancy. Classical data compression techniques [7], which involve transform, quantization and encoding, can reduce the local redundancy. Due to the nature of distributed sensor deployment in WSNs, an appealing technology for reducing distributed redundancy is DSC [3]. DSC refers to the compression of multiple correlated sensor outputs without communication between sensor nodes: sensor data is encoded locally according to a predefined correlation and decoded at the remote sink based on the side information.

For the implementation of DSC in WSNs, one of the requirements is that the correlation is well known by each sensor node and sink. Most of existing DSC algorithms take the correlation in time or space domain that constraints DSC's implementation only for smooth (low frequency) signal processing. For high frequency signal, the correlation is hard to be found duo to the high frequency noise pollution. In the structural health monitoring applications, the collected raw vibration sample data consists of both high frequency component, which is induced by high sample frequency (50Hz) and noise, and low frequency component, which is our primary interest in because it contains the key information of structure health menace [5].

In this paper, we describe a constructive algorithmic framework that supports DSC for high- and low-frequency signal compression in WSNs. To separate the low frequency component from the high frequency component, and strength the correlation among distributed sensor data, LSWT is used to preprocess the original data for signal decomposition and noise deduction. LSWT is better than traditional transforms and suitable for WSNs because 1) it is more efficient than FFT or DCT, 2) the transformed data supports time domain analysis, and 3) it supports multi-scale analysis and integer to integer mapping. After the LSWT, scalar quantization is used to reduce the individual redundancy. And last, the low frequency component of the sensor data can be processed by DSC to reduce the distributed redundancy. To the best of our knowledge, it is the first time that the LSWT and DSC are employed for vibration data compression, which achieves a higher compression ratio than the classical compression technique [5] while obtain the same data quality.

II. RELATED WORK

The problem of distributed data compression and data aggregation in sensor networks have led to new research challenges in networking, information theory and algorithm. In [2], Slepian and Wolf have theoretically shown that separate encoding (with increased complexity at the joint decoder) is as efficient as joint encoding for lossless compression. Similar results were obtained by Wyner and Ziv with regard to lossy coding of joint Gaussian sources. Currently, DSC is an active research area – more than 30 years after Slepian and Wolf laid the theoretical foundation. S.S. Pradhan et al. [3] provide a constructive practical framework based on algebraic trellis codes dubbed as Distributed Source Coding Using Syndromes (DISCUS). They address the problem of compressing correlated distributed sources. They also discuss the rate loss from the DISCUS which is separated into source coding loss and channel coding loss.

Although DSC has been implemented successfully in some sensor network scenarios [3][8], most of them are based on time or space domain correlation. These algorithms

work well only for low frequency signal. In our application, the sample frequency is 50Hz, making it difficult to decide the correlation of the sensor data as traditional DSC compression algorithms duo to noise pollution. In this paper, we apply DSC in frequency domain, and the proposed algorithm is suitable for both high- and low- frequency sensor data. In our work, the original data is decomposed into the low frequency component and the high frequency component by LSWT. Scalar quantization is then utilized to treat each input symbol separately in producing the output to reduce the noise and strengthen the correlation. This algorithm can achieve much higher compression ratio that another vibration data compression algorithm [5] with favorable signal-restoration quality.

There has been some implementations of LSWT in the structure health monitoring system based on WSNs [4][5]. Both of them utilized the LSWT to compress vibration data. Although their methods successfully reduced the high frequency information and achieved a good compression performance, e.g. a compression ratio of 1:14 in [5], they only compressed the individual data in every single sensor node instead of reducing the redundancy of distributed source data. In our algorithms, we not only compress the data generated by the individual source, but also consider the correlation of data from neighbor nodes. Experiment results indicate that our proposed algorithms can achieve a higher compression ratio while attain the same signal to noise ratio as theirs because DSC is a lossless algorithm.

III. Distributed Source Coding and Lifting Scheme Wavelet Transform

A. Distributed Source Coding

When DSC is implemented in WSNs, the correlated sensor nodes send their encoded data to the base station (sink) for joint decoding. Assume $\{X_i\}$ and $\{Y_i\}$ are the sequences of sample values collected in the sensor node A and B respectively (see Figure 1). In the individual source coding, the entropy $H(X)$ and $H(Y)$ must be sent respectively to the base station. However, according to Shannon's theory, if they are correlated discrete random variables of independent and identical distribution (i.i.d), only joint entropy $H(X, Y)$ is needed for lossless compression if they are encoded together. Slepian-Wolf extends Shannon's theorem further to that even if X and Y must be separately encoded, a rate $H(X, Y)$ can also be achieved if decoding of X and Y is done jointly. In WSNs, that means every individual node can compress its data and reduce the distributed redundancy only based on its own information. DSC is very suitable for WSNs because it excludes the data exchange, which would be very expensive for the extremely limited power and bandwidth, among the correlated neighbor nodes in WSNs. Slepian-Wolf source coding is lossless. While in practice, Slepian-Wolf coding is often combined with quantization to provide an approach to address lossy DSC problems.

B. An Example of Slepian-Wolf Coding [3]

For binary sample value X_i, $Y_i \in \{000, 001, \cdots, 111\}$, each of them needs to be encoded by 3 bits/sample. However

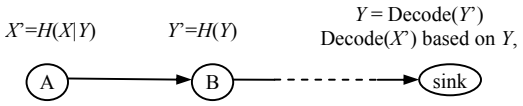


Figure 1. Basic structure of distributed source

if the correlation is known that the hamming distance between X_i and Y_i is $d_H \leq 1$, the value space can be divided into four cosets: $Z_{00} = \{000, 111\}$, $Z_{01} = \{001, 110\}$, $Z_{10} = \{010, 101\}$ and $Z_{11} = \{100, 011\}$. In every coset, the hamming distance between any two values is larger than or equal to 3. Y is encoded into $H(Y) = 3$ bits/sample and this original data is sent to the base station. Additionally, X is encoded as the index of the cosets $H(X|Y) = 2$ bits/sample and this compressed data is also sent to the base station. In the base station, Y is first decoded, and then X can be decoded depending on the side information Y. For a given Y, there are only four possible choices which belong to four separate cosets under the condition $d_H \leq 1$. For example, when $Y = 000$, $X \in \{000, 001, 010, 100\}$. Assume encoded value $X' = 01$ is the index of the coset, the only answer $X = 001$ can be decided because the hamming distance between $Y = 000$ and 110 (the other value in the coset) is 2 which is larger than $max(d_H)$. Thus the Slepian-Wolf limit of $H(X, Y) = H(Y) + H(X|Y) = 3 + 2 = 5$ bits is indeed achieved in this example with lossless decoding.

From the above example, we can generalize the Slepian-Wolf coding to the case when X and Y are equiprobable 2^n bit binary sources. Here $n \geq 3$ is a positive integer. The correlation model between X and Y is again characterized by $d_{H(X, Y)} \leq 2^k - 1$. Let $m = k + 1$, then $H(X) = H(Y) = n$ bits per sample, $H(X|Y) = m$ bits per sample, and $H(X,Y) = n + m$ bits per pair of samples for joint encoding.

For the implementation of DSC in WSNs, the choice of parameters n (the source codebook size) and k (the correlation) is an interesting topic. It could be automatically adjusted based on the history information. In our research, we decide these values by statistic analysis.

C. Wavelet and Lifting Scheme Wavelet Transform

Because our interested information lies in the low frequency component in the structure monitoring application, it is important to decompose it from the high frequency component. Wavelet Transform (WT) can analyze the signals in a frequency domain and decompose signals into the low frequency and high frequency components. DSC works well for the low frequency component, and the high frequency component is quantized to a small value later. WT outperforms traditional frequency transforms, i.e. FFT and DCT, because it does not need to know the global time domain information and can detect both the low frequency and the high frequency information automatically. Another important advantage of WT is that it can analyze the signal in multi-scales – the low frequency component can be decomposed again (figure 2), which provides us the potential to achieve a tradeoff between the data quality and compression ratio. The computation complexity of Mallat WT is $O(n)$ which is also called *fast wavelet transform* comparing the complexity $O(n\log n)$ of FFT and DCT. All of these characteristics indicate that WT is suitable for data compression and DSC naturally in WSNs.

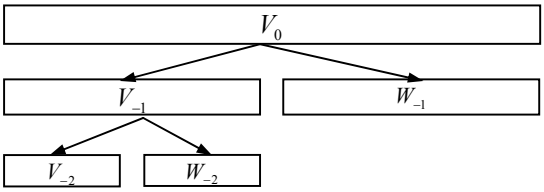


Figure 2: The wavelet decomposition tree with scale level *n*=2

LSWT is the second generation WT which is extended from Mallat algorithm. LSWT replaces the *translating* and *dilating* operations of conventional WT with *splitting*, *prediction* (dual lifting) and *updating* (primal lifting) operations. Compared to the first generation WT, LSWT has three main advantages when implemented in sensor networks [6]. First, it is faster than the Mallat algorithm although the computation complexity is still $O(n)$. Second, unlike the first generation WT, its inverse transform is easy to find and implement. Last, LSWT provides integer to integer mapping which is favorable in WSNs because the sensed data is a 10 bit integer.

In our simulation, we test two kinds of popular wavelets: CDF(1,1) (Haar wavelet) and Daubechies D4 wavelet. In the version of Daubechies D4 transform, LSWT consists of splitting, two updates, prediction, and normalization. Splitting refers to splitting the original data set λ_{j+1} into the even part λ_j and the odd part γ_j. The first update is to use the odd part to update the even part. After that, the even part is used to predict the odd part, followed by the update process again. The last step is normalization. The sequence of the steps is listed as equations (1) ~ (4):

Update 1: $$\lambda_j = \lambda_j + \sqrt{3}\gamma_j \quad (1)$$

Predict: $$\gamma_j = \gamma_j - \frac{\sqrt{3}}{4}\lambda_j + \frac{\sqrt{3}-2}{4}\lambda_{j-1} \quad (2)$$

Update 2: $$\lambda_j = \lambda_j - \gamma_{j+1} \quad (3)$$

Normalize: $$\lambda_j = \frac{\sqrt{3}-1}{\sqrt{2}}\lambda_j \quad \text{and} \quad \gamma_j = \frac{\sqrt{3}+1}{\sqrt{2}}\gamma_j \quad (4)$$

Assume that the length of data set λ_{j+1} is $2n$. To handle the edge problem, λ_{-1} and γ_n is replaced by λ_{n-1} and γ_0 respectively.

IV. System Design

A. Simulation System Structure

In our simulation, we use the same sample data as used in [5], which obtained by a civil engineering research group which utilizes the Micaz motes with 128K Program Flash Memory and 10 bit Analog to Digital Converter developed by UC Berkeley, and collect the data from a five layer civil infrastructure model [5]. The distance between each layer is 15cm. The first layer is attached to a motherboard which is driven by a vibration exciter. All the upper layers oscillate along with the lower layers. One sensor node is put in each layer and acceleration data is collected at a sample frequency 50Hz. Related research [4] has evaluated the accuracy of the ADXL202E onboard accelerometer for structure health monitoring and its modifications have been proposed. All the data collected are saved in the RAM. After collecting 4096 samples, the data will be compressed in the sensor node and sent to the base station which is connected to a PC. The data compression process is described in details in the next section.

B. Compression Process

To take advantage of DSC, the node in the first layer of the structure sends the original (self compressed) data as side information to the base station, and each other node sends the DSC compressed data.

The compress process is illustrated in Figure 3.

- The first step is LSWT. LSWT can be repeated by iteration on the λ_j, creating a multi-level or multi-resolution decomposition.
- The second step is quantization. We choose scalar quantization in our application to reduce the individual redundancy because it has been widely studied and successfully implemented in data compression combined with WT.

  After the quantization, most of the high frequency data value is set to zero. We use modified unary coding to encode the high frequency data set. That is: only the nonzero data set $\{x_i\}$ are encoded. If $x_i > 0$, we encode it with $2 \times x_i$ bits 1 and 10 bits relative position information. If $x_i < 0$, we encode it with $2x_i$-1 bits and 10 bits relative position information. For the low frequency component, we encode them as DSC described in the following steps.
- The third step is to map the coefficient to the source codebook. The codebook area is from 0 to 2^n-1. n is decided by the vibration character and statistic analysis. After the mapping, we can encode every data into n bits per sample which represents $H(Y)$. We rename the encoded data as base data. If the data is collected in the first layer sensor node, the compress process will be ended here and the base data will be sent to the base station, otherwise we continue to the next step.
- The fourth step is DSC. The data set is partitioned into different cosets as the channel codebook, and the original data is replaced with the channel codeword which is the coset index and represents $H(X|Y)$. We rename this kind of data as fully compressed data.

After the base station receives all the collected data, it will decompress all the data step by step. The decompress process includes:

- First, the base station decompresses base data received from first layer sensor node. Because this data represents $H(Y)$, we can decompress it without any side information.
- Then, we decompress the second layer data based on the decompressed first layer data as side information.
- After that, we can utilize the decompressed data of the second layer node to estimate and decompress the data from the third layer. This process is repeated till all the data is decompressed.

The proposed compress algorithm is a lossy one. The distortion includes quantization error conduced by scalar quantization and estimation error conduced by the channel coding. To achieve tolerable distortion ratios, we can adjust

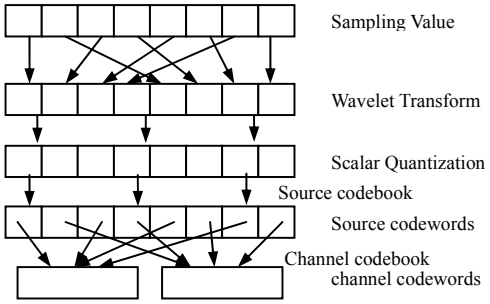


Figure 3: The compression process in sensor nodes

the quantization parameter and correlation parameter *k*.

C. Data Format and System Topology Extension

The encoded data structure in the application in our system is illustrated in Figure 4:

Bits (1) (8) (16)

type	base_ID	len	payload

Figure4: The compressed data format

Type field is to depart the base data (0) from fully compressed data (1). If *type* field is 0, *base_ID* field is its own node ID. Otherwise *base_ID* is the ID of its correlated node whose data is used as side information. *Len* field is the length of payload data.

Considering the implementation of our algorithms to large scenarios of WSNs, we should also consider the topology management for DSC. In [8], four DSC encoding schemes are provided, with the discussion of compression rate and loss factor separately. The cluster head, which sends the original data, could be selected dynamically to balance the power consumption. The topology controlled data compression is also our future research.

V. EXPERIMENTS AND SIMULATIONS

The overall goal of our experiments is to measure the compression ratio, restored data performance (distortion ratio) and computation complexity of our algorithms and compare them with other peer algorithms in various structure vibration scenarios. The system structure of our experiment is depicted in the previous section. The vibration exciter generates the vibration and drives the motherboard which connects the five layers structure. The basic methodology we use to measure the properties of our proposed compression algorithms is to change the vibration frequency of the exciter. However, because the highest sample frequency of the accelerometer (ADXL202E) in the sensor board is 60Hz, we can't detect higher frequency information. To justify the proposed WT-DSC based algorithms, White Gauss Noises of different degree are added to the collected sample value. We compare the results of our algorithms with those of other existing compression algorithms [5]. We call the proposed algorithms Haar-DSC (Haar wavelet based DSC) and Daub-DSC (Daubechies4 wavelet based DSC). We also rename the algorithms in [5] as Haar-MUC and Daub-MUC.

Our experiments also compare the compression properties based on different wavelets and DSC parameters. The results are analyzed for each experiment. In order to enable a direct, fair comparison between different algorithms, we implement each selected algorithm on 20 sets of raw data sampled from 20 scenarios with disparate vibration frequencies. Because these algorithms are challenged in the same identical condition, we can compare their performance directly.

In our experiments, we only measure the performance and results of the compression algorithms, rather than simulate the wireless sensor network topology. We choose C++ to implement our experimental code.

A. Compression Ratio

Compression ratio is one of most important criterions for a compression algorithm. Figure 5 highlights the relative compression ratio of the three compression algorithms as the noise degree increases. From the results, we can see that for the collected original signal without high frequency noise, Haar-DSC can achieve a higher compression ratio than that of Daub-DSC. However, as the noise degree increases, the compression ratio of Daub-DSC increases rapidly while that of Haar-DSC decreases. Haar WT performs an average and difference on each pair of neighbor values. For the original signal without noise, the high frequency component that stands for the difference consists of mostly zeros, and the low frequency component that stands for the average is smoother than that of Daubechies wavelet transformed signal, which picks up some neighbor nodes for high pass and low pass filters, because there is an overlap between iterations in the transform step, and the overlap makes the transformed data not as smooth as that of Haar WT. The smoother the signal, the higher the correlation, allowing Haar-DSC to achieve a better compression ratio than Daub-DSC. However, as the noise power increase, the high frequency component in the Haar-DSC conserved more high frequency information which can not be filtered and it makes Modified Unary Coding inefficient.

DSC based compression algorithms, which reduce both the local and distributed redundancy, always outperform MUC based algorithms. Haar-DSC achieves almost the same compression ratio with the Haar-MUC when the noise ratio is larger than 0.5dBW. The reason is that the high frequency component contains large values under this condition, and most of the compressed data bits come from this part which is encoded by MUC in both algorithms.

B. Compression Performance

The proposed compression algorithms in this paper are lossy algorithms, and the information is lost for quantization and estimation error. We evaluate our compression performance using three means: Peak Signal to Noise Ratio (PSNR), time domain analysis and low frequency domain analysis. PSNR of a reconstructed signal x_i^* compared to the original signal x_i is defined as:

$$\mathrm{PSNR} = 20\log_{10}\left(\frac{x_{peak}}{RMSE}\right)dB,$$

where $x_{peak} = \max_i |x_i|$ and the Root Mean Square Error:

$$RMSE = \sqrt{\sum_{i=0}^{n}(x_i - x_i^*)^2 \Big/ n}$$

where n is the length of the sample data set.

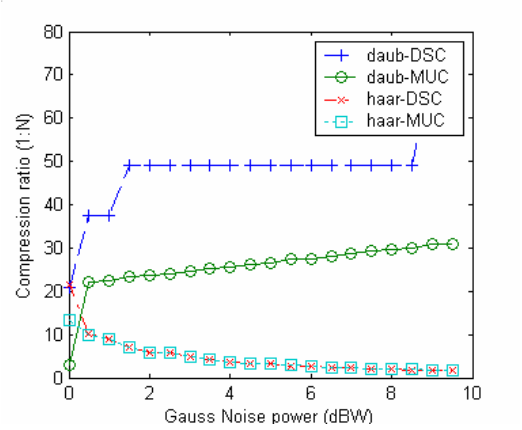


Figure 5: Compression ratio vs. noise degree

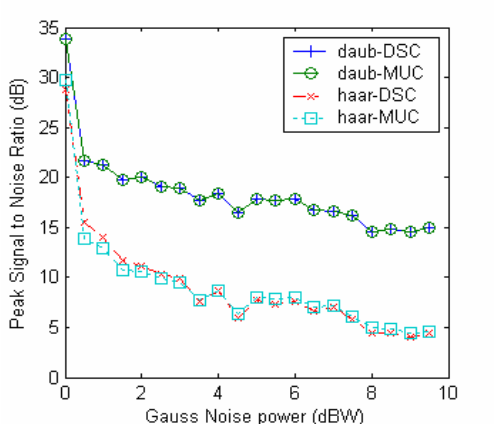


Figure 6: The peak signal to noise ratio vs. noise degree

PSNR is related to the properties of (bi)orthogonal wavelets [6]: neglecting the wavelet coefficients with the smallest magnitudes is a good compression approach if one wants to keep a high PSNR. Figure 6 shows that our proposed DSC based compression algorithms can always get the same compression quality as the MUC based algorithms. Because MUC is a lossless entropy coding algorithm, it also justifies the estimation error is neglectable in our proposed algorithms.

To better illustrate the performance of LSWT based algorithms and compare them with other algorithms, the original and reconstructed signals in time and frequency domains are shown in Figures 7 and 8. Both the Haar-DSC algorithm and Daub-DSC algorithm can achieve favorable performance in the low noise situation. However, when the noise degree increases slightly, the effect of Haar-DSC is weakened quickly, while Daub-DSC can still achieve fairish effect for the favorable characters of Daubechies wavelet. In the structure health monitoring applications, low frequency information of the sample signal is the important part. As the noise degree increase, the Haar-DSC algorithm can't restore the original low frequency signal, while Daub-DSC algorithm can still restore the original signal when the noise power increases to 5dBW. The results validate the feasibility of the proposed algorithms in the structure health monitoring applications. It is suitable for vibration and other high frequency correlated data compression.

C. Computation Complexity

Computation complexity is another important criterion when evaluating the compression algorithm, especially in the WSNs with limited resources. The compression process consists of three steps in our algorithms: LSWT, scalar quantization and DSC or MUC. The computation complexity can be expressed as:

$$C(n) = C_{\mathrm{LSWT}} + C_{\mathrm{quan}} + p \times C_{\mathrm{DSC}} + (1\text{-}p) \times C_{\mathrm{MUC}}. \quad p=1/2^n$$

where n is the scale level of LSWT.

As analyzed in section III, the complexity of LSWT (C_{LSWT}) is $O(n)$. However, the complexity of Haar LSWT is less than Daubechies D4 LSWT because Haar LSWT only has two steps and counting two filter coefficients, while Daubechies D4 has four steps and counting four filter coefficients.

The computation of scalar quantization matrix is nontrivial. However, we found that the sampled value from the same layer observed the same curve model in every experiment. To improve the efficiency of the algorithms, we only calculate the quantization matrix one time, and use the same matrix in all the later quantization process, so that we reduced the quantization complexity (C_{quan}) to $O(n)$. Results show it does not affect the compression performance under this condition.

The computation complexity of both DSC (C_{DSC}) and MUC (C_{MUC}) is $O(n)$, while DSC is still faster than MUC because the coding complexity of DSC for every symbol is 1 compared to the complexity $2\times|x_i|$ of MUC for symbol x_i. Overall, the total computational complexity of our algorithms is still $O(n)$.

From the above analysis, we can get the total complexity:

$$C(n) \in O(n)$$

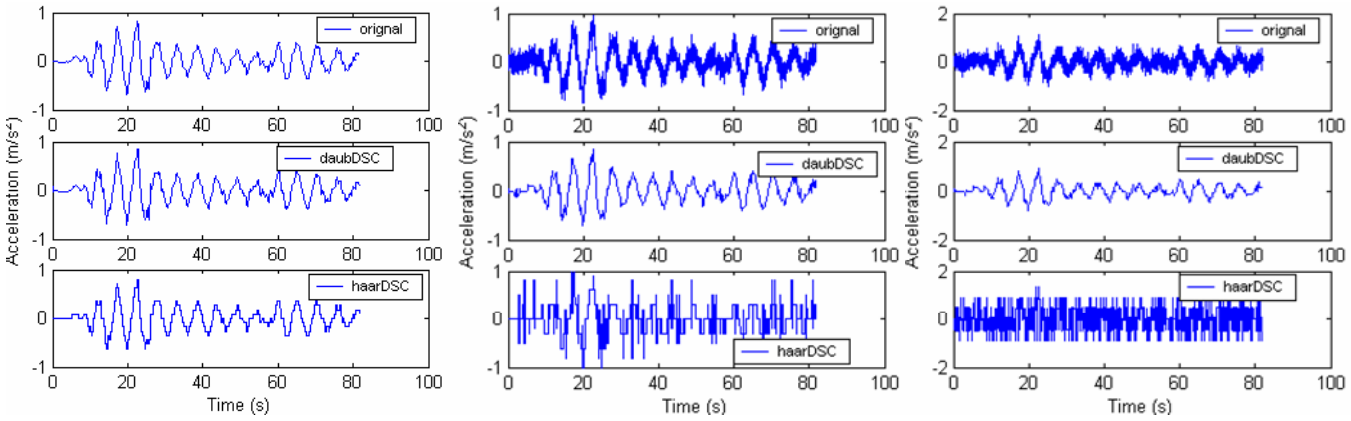
The running time of all the algorithms is listed in table 1. All the data is measured in Micaz platform associated with an ADXL202E onboard accelerometer.

As the experiments and analysis result demonstrated previously, we can see that the proposed LSWT and DSC based algorithms outperform their peer algorithms on compression ratio with the same data quality and similar computation overhead. Comparing Haar-Wavelet and Daubechies-Wavelet, Haar-Wavelet is more suitable for low-noise conditions because it is simpler and can also achieve good performance. In high-noise conditions, Daubechies-Wavelet, which outperforms Haar-Wavelet significantly, is the better choise.

VI. CONCLUSIONS

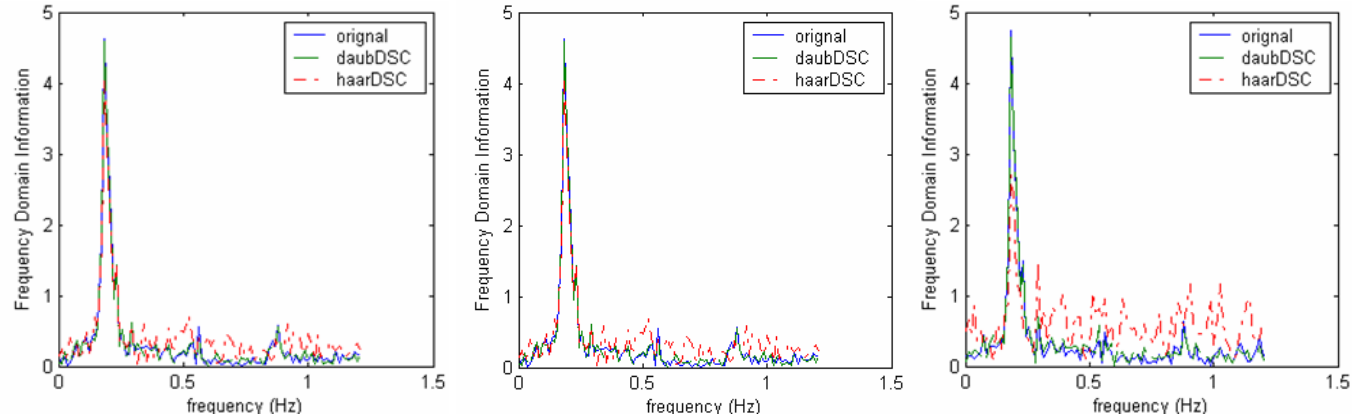
The research area of in-network processing in WSNs has been receiving more and more attention in recent years. Civil infrastructure health monitoring is an important application for WSNs. In this research, we propose new compression algorithms for high frequency sensor data based on LSWT and DSC. The proposed algorithms can achieve 1:27 to 1:80 compression ratios without weakening the data quality.

We first theoretically validate the correctness of our proposed algorithms, and analyze the appropriate choice of



(a) (b) (c)

Figure 7: The comparison of original and restored signal (a) noise: 0dBW, (b) noise: 0.5dBW, (c) noise: 5dBW



(a) (b) (c)

Figure 8: The frequency domain analysis (a) noise: 0dBW, (b) noise: 0.5dBW, (c) noise: 5dBW

different wavelet functions in special scenarios. We then implement the proposed algorithms in the experiment system. Results are analyzed based on compression ratio, data quality, and computation complexity. After the comparison with other peer algorithms, we validate that the LSWT and DSC based compression algorithms are favorable for high frequency data compression in WSNs.

Table 1: Computation time (s) in the Micaz sensor mote for 4096 sample value compression

Algorithms	Wavelet transform	Quanti-zation	Source coding	Total time
Haar-DSC	0.0102	0.0070	0.0629	0.0801
Haar-MUC	0.0102	0.0071	0.0611	0.0783
Daub-DSC	0.0450	0.0070	0.1196	0.1717
Daub-MUC	0.0451	0.0070	0.0447	0.0968

ACKNOWLEDGEMENTS

The authors would like to thank Dr. Yuefeng Zhang for his helpful discussion and data supply. This research has been supported by National Science Foundation (NSF) and a grant from the Commonwealth of Pennsylvania, Department of Community and Economic Development, through the Pennsylvania Infrastructure Technology Alliance (PITA). Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the sponsors of this research.

REFERENCES

[1] L. Doherty, B.A. Warnake, B. Baser, and K.S.J. Pister, "Energy and performance considerations for smart dust," in *Int. J. Parallel and Dtstributed Sensor Networks,* 2001.

[2] D. Slepian and J.K. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inform. Theory,* Vol. IT-19, pp. 471-480, July 1973.

[3] S. S. Pradhan, J. Kusuma, and K. Ramchandran, "Distributed compression in a dense microsensor network," *IEEE Signal Processing Magazine*, Vol. 19, pp. 51–60, Mar. 2002.

[4] N. Xu, S. Rangwala, K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan and D. Estrin, "A Wireless Sensor Network for Structural Monitoring," *In Proceedings of the ACM Conference on Embedded Networked Sensor Systems,* November 2004

[5] Y. Zhang and L. Cheng, "Issues in applying wireless sensor networks to health monitoring of large scale civil infrastructure systems" *ASCE Structure Congress 2005*, New York City, New York, April 20-24, 2005

[6] G. Uytterhoeven, D. Roose, and A. Bultheel, Wavelet transforms using the Lifting Scheme, *ITA-Wavelets Report WP 1.1,* Department of Computer Science, Katholieke Universiteit Leuven, Belgium, November 1996.

[7] D. Salomon, Data Compression: The Complete Reference, 3rd Edition, Springer 2004.

[8] D. Marco and D. L. Neuhoff, "Reliability vs. Efficiency in Distributed Source Coding for Field-Gathering Sensor Networks", *IPSN'04*, Berkeley, CA, Apr. 22-23, 2003