



Using DWT Lifting Scheme for Lossless Data Compression in Wireless Body Sensor Networks

Joseph Azar, Rony Darazi, Carol Habib, Abdallah Makhoul, Jacques Demerjian

► To cite this version:

Joseph Azar, Rony Darazi, Carol Habib, Abdallah Makhoul, Jacques Demerjian. Using DWT Lifting Scheme for Lossless Data Compression in Wireless Body Sensor Networks. International Wireless Communications and Mobile Computing Conference, Jun 2018, Limassol, Cyprus. hal-02952695

HAL Id: hal-02952695

<https://hal.science/hal-02952695>

Submitted on 29 Sep 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Using DWT Lifting Scheme for Lossless Data Compression in Wireless Body Sensor Networks

Joseph Azar^a, Rony Darazi^b, Carol Habib^{a,b}, Abdallah Makhoul^a, Jacques Demerjian^c

^aFEMTO-ST Institute/CNRS, Univ. Bourgogne Franche-Comté, Belfort, France

^bTICKET Lab, Antonine University, Hadat-Baabda, Lebanon

^cLARIFA-EDST, Faculty of Sciences, Lebanese University, Fanar, Lebanon

Emails: josph_azar@outlook.com, rony.darazi@ua.edu.lb, carol.habib@univ-fcomte.fr, abdallah.makhoul@univ-fcomte.fr, jacques.demerjian@ul.edu.lb

Abstract—Recently, interest in Wireless Body Sensor Networks composed by low-power devices which are placed in, on or around the body has been increasing. Wireless Body Sensor Networks open up tremendous healthcare and wellness applications such as continuous monitoring of a patient's vital signs. One of the fundamental challenges in Wireless Body Sensor Networks is energy consumption due to wireless transmission of collected data. In this paper, we aim to extend the life-time of battery-powered biosensors by applying a data reduction technique that works efficiently under constrained processing, storage, and energy resource conditions. The presented technique is a lossless transform-based compression technique based on the Discrete Wavelet Transform using the lifting scheme extended with Lagrange polynomial interpolation. To evaluate our approach, we have run multiple series of simulation on real sensor data. The results show that our proposed method reduces the amount of data by up to 90% without losing any information.

Keywords—Data reduction, lossless data compression, wireless body sensor network, lifting scheme, wavelet.

I. INTRODUCTION

THE development of Wireless Body Sensor Networks (WBSNs) offers the ability to monitor a patient's physical and biochemical parameters continuously, under natural physiological status of the patient, and in any environment. This technology supports a number of innovative and interesting applications in the medical and healthcare areas such as surveilling patients with chronic disease, examining hospital and elderly patients, and distant patients monitoring where the patients' vital signs are checked constantly in order to control their health condition and provide treatment in case of emergencies [1].

The collection and analysis of vital signs' data can be achieved by deploying different types of biomedical sensors (e.g., body temperature, heartbeat, blood pressure, electrocardiogram (ECG), electroencephalogram (EEG), etc.) [2] [3]. A network of biosensors is to be placed on or implanted in the body of patients. The amassed information is sent to a coordinator (e.g., Body Control Unit), located on or near the body. This coordinator is responsible of the aggregation, the fusion and the forwarding of the collected data to the sink node which in turn sends the received data to the network [4].

Although the WBSN platforms aim to provide and support a number of innovative and interesting

applications [5] [6] [7], there are many technical challenges that lie ahead such as the need for better sensor design, context-awareness, secure data transfer, and longer battery lifetime. Energy consumption is one of the key considerations for WBSNs since it not only determines the size of the battery required but also the duration that a biosensor can be left in situ [1]. One of the main techniques to reduce the energy consumption of a biosensor node is by decreasing the energy consumption related to wireless transmission of the collected data since it is the likeliest cause of energy consumption [8] [9]. Data reduction can be considered as a direct way to reduce the energy consumption due to wireless transmission. Some of the techniques, such as aggregation, adaptive sampling, and data compression aim to reduce the amount of data that is to be delivered to the coordinator [10].

Several compression techniques have been proposed in literature to tackle the energy consumption problem. These techniques can be classified into three categories: sampling compression, data compression, and communication compression [8]. Sampling compression focuses on reducing the number of sensing operations while keeping the loss of information within an acceptable margin. Data compression converts an input data stream into another data stream that can be represented with a lesser number of bits. Communication compression focuses on reducing the number of packet transmissions and receptions. Unfortunately, conventional compression algorithms have a high complexity, thus in order to gain sufficient storage they're purposefully made for desktops and servers. As such, they cannot be directly applicable to WBSNs where the primary objective is to save energy.

Many resource-aware compression techniques have been used and developed for data reduction in the context of Wireless Sensor Networks (WSNs) and WBSNs.

In [9], the authors proposed the simple Delta encoding algorithm, known as Differential Pulse Code Modulation (DPCM) for accelerometer data compression in WBSN, and compared it to the Huffman encoding. The results showed that the Delta encoding outperformed the Huffman encoding in terms of data reduction, computational complexity, and energy savings. A method referred to as LiftingWise has been proposed in [11]. The LiftingWise method is a modified version of the original Discrete Wavelet Transform (DWT) Lifting Scheme (LS) algorithm in which it can be applied on

a set of data with arbitrary length while the original LS is applied on a signal S_n of length 2^n . This method uses the Haar wavelets to process the data disseminated from objects deployed in a monitoring environment. It was compared with two other simple compression techniques appropriate for usage in WSNs: The Offset compression and Marcelloni compression. The results had proved the effectiveness of this method in reducing the number of bits of the collected data while taking into consideration the limited resources of sensor nodes.

In this paper, we aim to reduce the energy consumption of biosensor nodes due to the size of the information of patients' vital signs that is being transmitted periodically. To do so, we propose to use a transform-based compression technique based on the Discrete Wavelet Transform (DWT) using the lifting scheme (LS) extended with Lagrange polynomial interpolation. The wavelet transform provides a time frequency representation of the original data set. The motivation behind using the DWT resides in the transformation of redundant samples in the spatial domain to decorrelated coefficients in the time frequency domain where the original samples are compacted and represented with fewer coefficients.

To evaluate our approach, we compare our proposed algorithm with the Discrete Cosine Transform, DWT LS based Haar transform used in [11], and the simple Delta encoding used in [9].

The following of the paper is organized as follows: Section II provides additional detail on energy consumption at node-level in the context of WBSN. Section III presents some background information about the DWT LS. Section IV details our proposed DWT lifting scheme algorithm. Section V presents the experimental results. Section VI discusses the obtained results, and Section VII concludes the paper.

II. ENERGY CONSUMPTION AT NODE-LEVEL

The four main constituents of a biosensor node that consume energy are the Microcontroller (MCU), Transceiver, Memory, and Sensor units. The MCU consumes energy for processing the data, executing code, and controlling tasks. The transceiver consumes energy for transmitting and receiving data. The energy consumption of the memory depends on the number of memory read and write, the number of stored bits, and the duration of storage. Also, the number of bits generated from the sensing unit contributes very little to the overall energy consumption of the biosensor node [12].

Proceeding from these facts, the compression algorithm applied to biosensor nodes must reduce the number of transmitted bits and increase the percentage of energy saving. However, processing and memory units cannot be neglected, and the trade-off between computational cost and the energy saved from compression ratio must be considered. The applied method must have a low computational time/memory complexity in a way that the energy saved from transmitting the compressed data must be greater than the energy consumed by performing additional computation and processing [13].

III. BACKGROUND ON THE DWT LIFTING SCHEME

Sweldens introduced the “lifting” procedure in [14] [15]. The LS is a transform that uses means and differences to compute the DWT coefficients. The advantages of the LS over conventional wavelet transform method are that it allows a faster implementation of the wavelet transform, requires half number of operations as compare to traditional convolution based DWT, and allows a fully in-place calculation [16]. The aforementioned advantages make the LS suitable for low power applications.

In the following, the DWT LS is presented step by step. Consider a set of data x_j with 2^n values, the LS performs n forward transform steps (n -level decomposition). Each forward transform consists of three operations: split, predict, and update as shown in Figure 1.

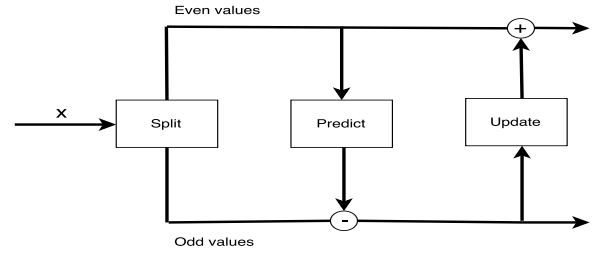


Fig. 1: Lifting scheme forward step

a) Split operation: This operation splits the data into even and odd sets. The even and odd sets each contain 2^{n-1} values.

Afterwards, the even set is transformed into the average set (approximation coefficients) denoted by s_{j+1} , and the odd set is transformed into the differences set (detail coefficients) denoted by d_{j+1} , each of length 2^{n-1} .

b) Predict operation: In this step, a prediction function is used to approximate the data set. A correlation exists between an odd element and its even neighbors, and the even elements can be used to predict the odd element. The difference between the actual odd value and the predicted value replaces the odd elements. The even elements are left unedited and become the input for the next operation in the transform. The prediction, which is the difference, is denoted by Equation 1:

$$d_{j+1} = x[2n+1]_j - P \times (x[2n]_j), \quad (1)$$

where P is the prediction function. The importance of the prediction operation is that it results in smaller values that can be represented in fewer bits. The more the data are correlated, the more the difference between an odd element and its prediction is small and some level of compression can be achieved.

c) Update operation: The update operation replaces the even entry by the average of the data being processed. The importance of this operation is that it results in a smoother input for the next decomposition level [17]. The update operation is defined by Equation 2:

$$s_{j+1} = x[2n]_j + U \times (d_{j+1}). \quad (2)$$

The LS reverse transform is similar to the forward transform and is based on the three operations: reverse update, reverse prediction, and merge. These three operations are defined in Equations 3, 4, and 5 respectively.

$$x[2n]_j = x[2n]_{j+1} - U \times (d_{j+1}). \quad (3)$$

$$x[2n+1]_j = x[2n+1]_{j+1} + P \times (x[2n]_j). \quad (4)$$

$$x_j = \text{Merge}(x[2n+1]_j, x[2n]_j). \quad (5)$$

The described operations (split, predict, and update) can be repeated n times on a set of data of length 2^n . Figure 2 illustrates 2-level wavelet decomposition of 128 PhotoPlethysmoGram (PPG) samples. Each decomposition level l_j breaks down the signal into two sets: approximation coefficients set (averages) s_j and detail coefficients set (differences) d_j . We can notice that the values of the detail coefficients are smaller than the ones of the original PPG samples, which means that they require fewer bits to be represented. The final result is a set s_n of one average (the mean value of all entries) and n sets of differences (d_1, d_2, \dots, d_n).

IV. LIFTING SCHEME BASED HAAR WAVELETS EXTENDED WITH POLYNOMIAL INTERPOLATION

Multiple prediction functions can be used to calculate the detail coefficients (differences). One of the simplest predict functions is the Haar prediction used in [11]. The Haar prediction operation predicts that the odd element will be equal to the even element. The odd element is then replaced by the difference between the predicted value (the even element) and the actual value of the odd element as defined in Equation 6. In the update step, each even element in the even set is replaced with the average of the even/odd pair as in Equation 7.

$$d_{j+1} = x[2n+1]_j - x[2n]_j. \quad (6)$$

$$s_{j+1} = x[2n]_j + d_{j+1}/2. \quad (7)$$

The Haar wavelet is well known for its simplicity. However, it performs poorly when the signal is not constant (jagged time series), and misses the changes between odd and even values. In this paper, we propose to extend the Haar wavelet with Lagrange polynomial interpolation to predict the odd values.

The Lagrange interpolating polynomial is the polynomial $P(x)$ of degree $\leq (n-1)$ that passes through the n points $(x_0, y_0 = f(x_0)), (x_1, y_1 = f(x_1)), \dots, (x_n, y_n = f(x_n))$, and is given by

$$P(x) = \sum_{j=0}^n F_j(x) y_j, \quad (8)$$

where

$$F_j(x) = \prod_{k=0, k \neq j}^n \frac{x - x_k}{x_j - x_k}. \quad (9)$$

The implementation of the polynomial interpolation had been done based on the model discussed in [17]. The forward transform step consists of the three steps: split, update, and interpolate as shown in Figure 3. Note that the update step was placed before the interpolate step, so the latter can calculate the prediction value from the smoothed values that result from the update step.

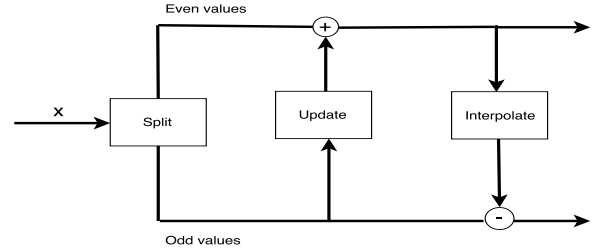


Fig. 3: Lifting scheme extended with polynomial interpolation forward step

In the update step, the even elements are updated using Equation 10.

$$s_{j+1} = (x[2n]_j + x[2n+1]_j)/2. \quad (10)$$

In the interpolate step, an odd element is predicted from the even points using n -point Lagrange polynomial interpolation, and replaced by the difference between the interpolated (predicted) value and the actual value of the odd element as defined in Equation 11.

$$d_{j+1} = x[2n+1]_j - P_{interpolated}. \quad (11)$$

In our implementation, 4-point polynomial interpolation has been used to predict the odd elements.

Consider four even points: e_0, e_1, e_2 , and e_3 . Disregarding their actual x-coordinates, which are timestamps in the case of WBSN data, we position these points at x-coordinates 0, 1, 2, 3. The first odd point in the odd set will be positioned between the first and second even points (e_0 and e_1), at 0.5. The last two points in the odd set are positioned at 2.5 and 3.5 respectively. The rest of the odd points are positioned between e_1 and e_2 , at x-coordinate 1.5. Figure 4 shows the logical positioning of the even and odd points.

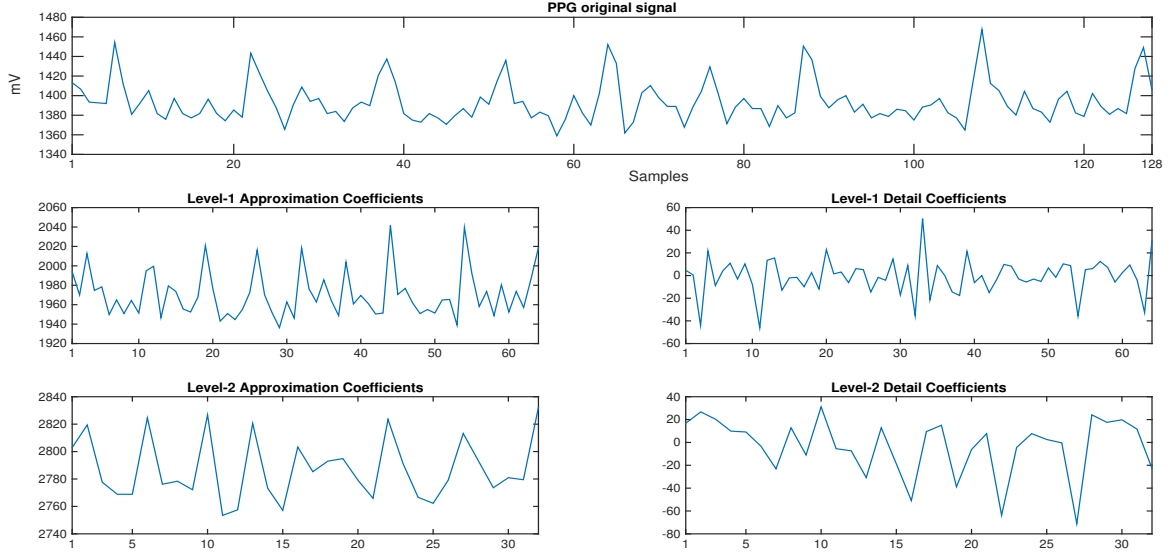


Fig. 2: PPG signal wavelet decomposition

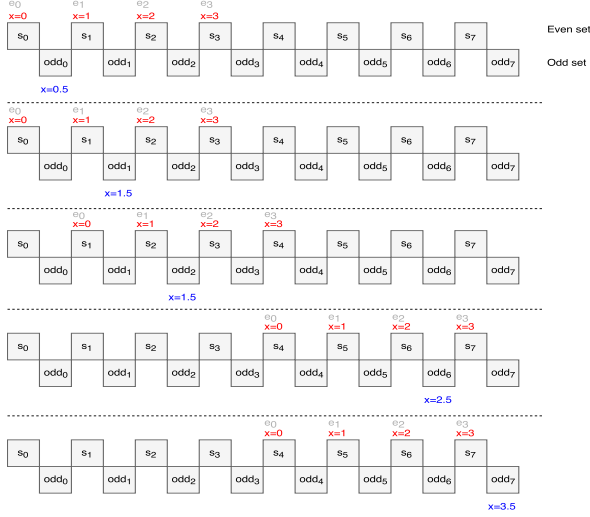


Fig. 4: Example of even and odd elements positioning in a data series

Based on Equations 8 and 9, $P_{\text{interpolated}}$ is calculated as:

$$P_{\text{interpolated}} = F_0(x)e_0 + F_1(x)e_1 + F_2(x)e_2 + F_3(x)e_3. \quad (12)$$

The coefficients F_0 , F_1 , F_2 , and F_3 are constant and can be calculated once for the x-coordinates 0.5, 1.5, 2.5, and 3.5 using Equation 9.

The advantage of the polynomial interpolation prediction over the Haar prediction is that it is more accurate and able to properly capture changes between odd and even values. Note that the more the prediction step is accurate, the more the detail

coefficients (differences) are close to zero and require fewer bits to be represented.

V. EXPERIMENTAL RESULTS AND ANALYSIS

A custom Java-based simulator has been used to test the proposed compression technique. Therefore, the following shows the results of applying the DWT LS based Haar transform used in [11] (LS Haar), the Delta encoding algorithm used in [9] (Delta), the DCT, and the proposed LS extended with polynomial interpolation (LS Poly) on different time series recorded using Shimmer3 GSR+ Unit [18] (Figure 5). The data are sampled at a fixed rate of 16 Hz over 10 minutes (5 minutes at rest, and 5 minutes walking).

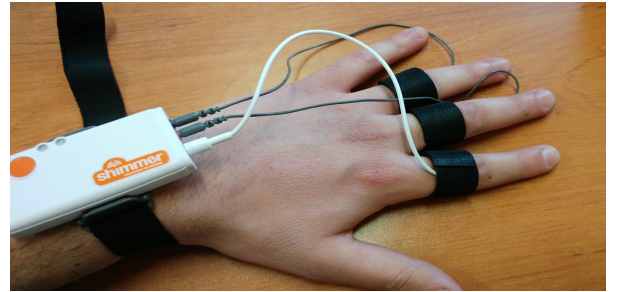


Fig. 5: Shimmer3 GSR+ Unit

Table I presents the compression ratios (CR) and the reduction percentages (Red %) obtained by the four compression algorithms over 50 periods (1 period = 8 seconds) on different sensor data: Pressure (kPa), Temperature (Degrees Celsius), PPG (mV), Accelerometer (m/s^2), and Gyroscope (deg/s). The results show that the

TABLE I: Performance comparison of LS Haar, Delta encoding, DCT, and LS Poly on different sensor data collected from Shimmer GSR+ Unit over 50 periods

	Uncompressed	LS Haar			Delta			DCT			LS Poly		
		Bits	CR	Red (%)	Bits	CR	Red (%)	Bits	CR	Red (%)	Bits	CR	Red (%)
Pressure	102400	8282	12.36	91.91	9074	11.28	91.13	7775	13.17	92.41	7274	14.08	92.9
Temperature	87479	7897	11.08	90.97	7299	11.99	91.66	8306	10.53	90.5	7285	12.01	91.67
PPG	127992	80636	1.59	37.11	80956	1.58	36.71	80070	1.6	37.5	77143	1.66	39.76
Accelerometer_x	76784	15934	4.82	79.25	18085	4.25	76.47	16493	4.66	78.54	12844	5.98	83.28
Accelerometer_y	56611	13278	4.26	76.53	15070	3.76	73.4	12534	4.52	77.88	9821	5.76	82.64
Accelerometer_z	76800	11619	6.61	84.87	12722	6.04	83.44	12458	6.16	83.77	9819	7.82	87.21
Gyroscope_x	52073	37376	1.39	28.06	39790	1.31	23.66	37750	1.38	27.54	32886	1.58	36.71
Gyroscope_y	64205	39847	1.61	37.89	41919	1.53	34.64	40518	1.58	36.71	35332	1.82	45.05
Gyroscope_z	41146	33610	1.22	18.03	36185	1.14	12.28	33824	1.22	18.03	28782	1.43	30.07

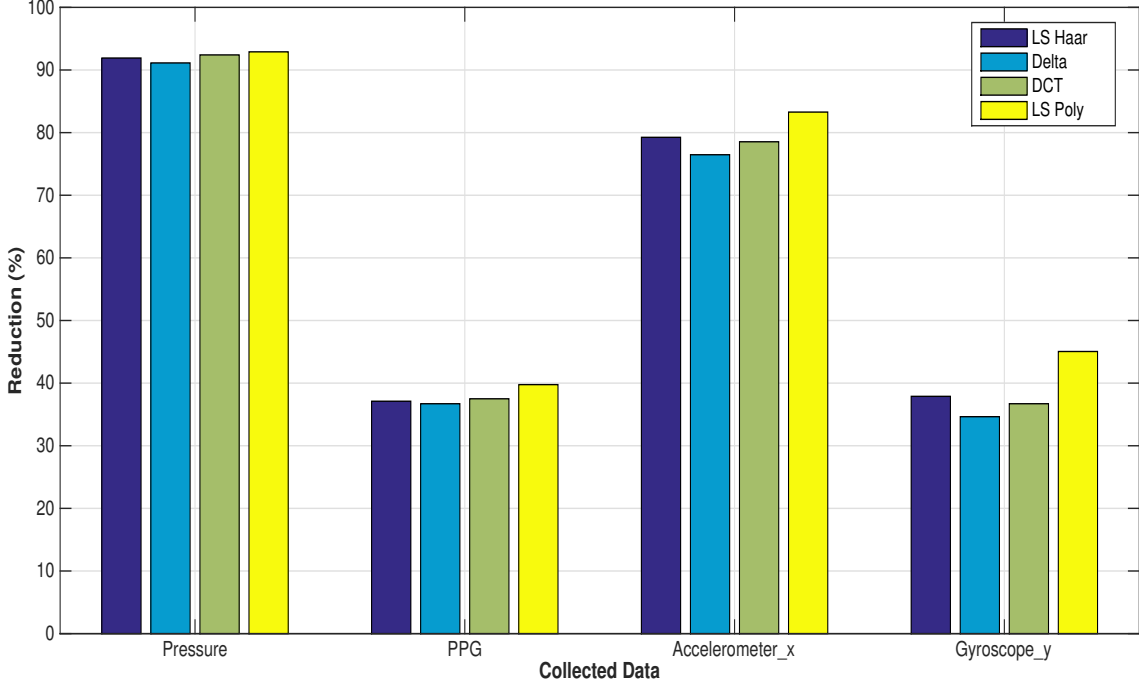


Fig. 6: Comparison of the reduction percentage achieved by the four compression algorithms on different sensor data over 50 periods

highest compression ratios and reduction percentages were achieved by the proposed LS Poly algorithm.

We can notice that the four compression algorithms reduced the pressure and temperature time series by more than 90%, since the data in these time series are constant and highly correlated. In the case of PPG and gyroscope time series, where the data are not correlated, the maximum reduction percentage achieved was 45% by the proposed LS Poly algorithm.

VI. DISCUSSION

In real world applications, different types of data are usually collected using a single sensor node. The statistical characteristics of the collected data may also vary. As shown in Figure 6, we can see that the compression algorithms achieve high reduction percentage on certain time series (pressure, accelerometer), and low reduction percentage on

other time series (PPG, gyroscope). Therefore, the implemented compression technique must have the ability to adapt and perform well across different types of data, since it is not feasible to implement a compression algorithm for each of the collected time series. Additionally, the implemented compression technique must have low processing and memory requirements. The time complexity of the Delta coding, LS, and DCT algorithms are $\Theta(n)$, $\Theta(n \log n)$, and $\Theta(n \log n)$ respectively. The four algorithms are lossless, and most of the performed operations are simple arithmetic operations, which make them suitable for limited resources devices.

In order to further reduce the energy consumption due to transmission, multiple compression techniques can be combined. However, the trade-off between computational cost and the energy saved from compression ratio must be considered.

VII. CONCLUSION

Given the fact that the majority of the total energy consumption in WBSN applications is consumed by the wireless transmission of the collected data, data compression can be considered an efficient way to increase the energy efficiency of WBSNs. Our main objective in this paper was to reduce the number of bits needed to represent the sensed data prior to transmission. The proposed technique is a lossless transform-based compression technique that does not require complex mathematical calculation and additional memory space. This technique is based on the DWT using the lifting scheme extended with polynomial interpolation. A series of simulations have been run on real data collected from Shimmer3 GSR+ Unit to prove the effectiveness of the proposed algorithm. We compared the proposed algorithm with the lifting scheme based Haar transform, the simple delta encoding, and the discrete cosine transform. The results show that the proposed algorithm achieved the higher compression ratio on different types of data.

For future work, we intend to combine the proposed lifting scheme algorithm with the compressive sensing to achieve a higher compression ratio on non-correlated data such as PPG and gyroscope time series.

ACKNOWLEDGMENT

This work is partially funded with support from the Labex ACTION program (contract ANR-11-LABX-01-01), the National Council for Scientific Research in Lebanon CNRS-L, the Hubert Curien CEDRE programme, and the Lebanese University Research Program (Number: 4/6132).

REFERENCES

- [1] G. Z. Yang, *Body Sensor Networks*. Springer, 2006.
- [2] J. I. Bangash, A. H. Abdullah, M. H. Anisi, and A. W. Khan, "A survey of routing protocols in wireless body sensor networks," *Sensors*, vol. 14, no. 1, pp. 1322–1357, 2014. [Online]. Available: <http://www.mdpi.com/1424-8220/14/1/1322>
- [3] N. Boudargham, J. B. Abdo, J. Demerjian, C. Guyeux, and A. Makhoul, "Investigating low level protocols for wireless body sensor networks," *CoRR*, vol. abs/1611.08401, 2016. [Online]. Available: <http://arxiv.org/abs/1611.08401>
- [4] N. Bradai, L. C. Fourati, and L. Kamoun, "Wban data scheduling and aggregation under wban/wlan healthcare network," *Ad Hoc Networks*, vol. 25, pp. 251 – 262, 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1570870514002339>
- [5] R. Negra, I. Jemili, and A. Belghith, "Wireless body area networks: Applications and technologies," *Procedia Computer Science*, vol. 83, pp. 1274 – 1281, 2016, the 7th International Conference on Ambient Systems, Networks and Technologies (ANT 2016) / The 6th International Conference on Sustainable Energy Information Technology (SEIT-2016) / Affiliated Workshops. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S187705091630299X>
- [6] T. Yilmaz, R. Foster, and Y. Hao, "Detecting vital signs with wearable wireless sensors," *Sensors*, vol. 10, no. 12, pp. 10 837–10 862, 2010. [Online]. Available: <http://www.mdpi.com/1424-8220/10/12/10837>
- [7] X. Lai, Q. Liu, X. Wei, W. Wang, G. Zhou, and G. Han, "A survey of body sensor networks," *Sensors*, vol. 13, no. 5, pp. 5406–5447, 2013. [Online]. Available: <http://www.mdpi.com/1424-8220/13/5/5406>
- [8] M. A. Razzaque, C. Bleakley, and S. Dobson, "Compression in wireless sensor networks: A survey and comparative evaluation," *ACM Trans. Sen. Netw.*, vol. 10, no. 1, pp. 5:1–5:44, Dec. 2013. [Online]. Available: <http://doi.acm.org/10.1145/2528948>
- [9] S. Arrabi and J. Lach, "Adaptive lossless compression in wireless body sensor networks," in *Proceedings of the Fourth International Conference on Body Area Networks*, ser. BodyNets '09, 2009.
- [10] T. Rault, A. Bouabdallah, and Y. Challal, "Energy efficiency in wireless sensor networks: A top-down survey," *Computer Networks*, vol. 67, pp. 104 – 122, 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1389128614001418>
- [11] E. Aboelela, "Liftingwise: A lifting-based efficient data processing technique in wireless sensor networks," *Sensors*, vol. 14, no. 8, pp. 14 567–14 585, 2014. [Online]. Available: <http://www.mdpi.com/1424-8220/14/8/14567>
- [12] N. Kamyabpour and D. B. Hoang, "Modeling overall energy consumption in wireless sensor networks," *CoRR*, vol. abs/1112.5800, 2011. [Online]. Available: <http://arxiv.org/abs/1112.5800>
- [13] T. Srisooksai, K. Keamarungsi, P. Lamsrichan, and K. Araki, "Practical data compression in wireless sensor networks: A survey," *Journal of Network and Computer Applications*, vol. 35, no. 1, pp. 37 – 59, 2012, collaborative Computing and Applications. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1084804511000555>
- [14] W. Sweldens, "The lifting scheme: A construction of second generation wavelets," *SIAM Journal on Mathematical Analysis*, vol. 29, no. 2, pp. 511–546, 1998. [Online]. Available: <https://doi.org/10.1137/S0036141095289051>
- [15] W. Sweldens, "The lifting scheme: A custom-design construction of biorthogonal wavelets," *Applied and Computational Harmonic Analysis*, vol. 3, no. 2, pp. 186 – 200, 1996. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S106352039600159>
- [16] A. Jensen and A. Cour-Harbo, *Ripples in Mathematics: The Discrete Wavelet Transform*. Springer, 2001.
- [17] I. Kaplan, "Wavelets and signal processing," http://www.bearcave.com/misl/misl_tech/wavelets/lifting/basiclift.html, accessed: 2018-3-1.
- [18] "Shimmer sensing," <http://www.shimmersensing.com/products/shimmer3-wireless-gsr-sensor>, accessed: 2018-3-2.