

Data-Smoothness based Preprocessing Strategy for Wavelet Data Processing in Wireless Sensor Networks

Yalin Nie¹, Haijun Wang², and Yujie Qin¹

¹Dept. of Computer and Information Engineering, Luoyang Institute of Sci. and Tech., Luoyang 471023, China

²School of Mathematics and Statistics, Henan University of Sci. and Tech., Luoyang 471023, China

Email: nieyalin111@163.com; hkdwhj@163.com; qyjswjtu@vip.sohu.com

Abstract—Wavelet based data compression in wireless sensor networks can reduce in-network data transmissions and gain better data approximation. To improve the performance of algorithms based on wavelet data compression, data-smoothness based preprocessing strategy for wavelet data processing is proposed. The strategy can adjust the order of data to be processed for better smoothness through sample mean and control the frequency of data order adjustment by a threshold, achieving better data reconstruction precision with acceptable network control overhead, and higher data compression degree under a given requirement of data reconstruction precision. Theoretical analysis and experiments prove the effectiveness of the strategy.

Index Terms—Wavelet, data smoothness, sensor network, data compression, data preprocessing

I. INTRODUCTION

In order to monitor environment sufficiently, a large number of sensor nodes are deployed, often resulting in amounts of redundant in-network raw data. A lot of redundant data transmission will greatly reduce the monitoring performance of wireless sensor networks. Therefore, it is necessary to process raw data to reduce the redundancy among data before transmitting them, decreasing the amount of data transmissions and prolonging network lifetime [1].

Compared with the Fourier analysis, wavelet can characterize a signal in time-domain and frequency-domain simultaneously and has multi-resolution analysis features. When a signal is processed by wavelet transform at different scales, its statistical features can still be maintained. Currently, Discrete Wavelet Transform (DWT) has been applied widely in many fields such as digital image processing, encoding theory, wireless sensor networks, *etc.* The amount of redundant information in raw sensory data is often large. DWT can mine spatial and temporal correlation among raw data to decrease the redundant information. Through DWT, raw sensory data can be transformed into a series of wavelet coefficients (approximation coefficients and detail coefficients) which can be efficiently compressed by a

suitable encoding algorithm. Abandoning parts of detail coefficients, approximation data can be reconstructed by performing inverse DWT on the rest of coefficients satisfying the given error bound [4, 5], which can be used to compress raw data further.

Cluster is a common architecture for wireless sensor networks to process in-network data [2, 3]. Generally, cluster members send sensory data to cluster heads and cluster heads are responsible for intra-cluster data processing to decrease in-network data transmissions. With DWT, cluster heads process the intra-cluster data which can be regarded as a discrete signal and report approximation coefficients and parts of detail coefficients to the sink. The sink reconstructs approximation data on the basis of the received coefficients through inverse DWT. The wavelet features [6] show that the smoother the discrete data to be processed is, the more concentrated the energy distribution of the transformed data is, which is conducive for compressing coefficients. Therefore, if the intra-cluster data to be processed by wavelet has good smoothness, the effect of wavelet compression will be fine.

Current researches on wavelet based data processing in wireless sensor networks are on the basis of some assumptions about data correlation or mine data correlation dynamically [6-14]. All of them do not pay attention to the smoothness of data to be processed. In order to improve the data compression performance of algorithms based on wavelet in wireless sensor networks, we propose a novel Data-Smoothness based Preprocessing Strategy (DSPS), which is simple but effective. It can gain smoother data to be processed and improve the performance of data compression and data reconstruction based on wavelet obviously. The strategy can be combined with any wavelet based data processing algorithm in wireless sensor networks, promoting the performance of original algorithms effectively. Theoretical analysis and experiments demonstrate that the strategy is effective.

The rest of the paper is organized as follows: Section II introduces some related works. Section III describes the Data-Smoothness based Preprocessing Strategy (DSPS). Section IV analyzes our strategy and does some experiments to prove its effectiveness. Section V presents our conclusions and future work finally.

Manuscript received May 1, 2014; revised October 17, 2014.
Corresponding author email: nieyalin111@163.com.
doi:10.12720/jcm.9.10.762-770

II. RELATED WORK

Wavelet has been used widely in various applications of WSNs. In the help of wavelet error tree, Zhang JM *et al.* proposed a data compression scheme based on 1D Haar wavelet with infinite norm error bound [5]. Further, they designed the MWCEB algorithm for multivariate monitoring sensors based on a base signal selection algorithm which is used to select signals, a linear regression scheme and the former proposed 1D Haar wavelet compression scheme. RACE [7] is a time series compression algorithm based on wavelet with rate adaptivity and error bound. It builds a gradient error tree, selects wavelet coefficients by error-based zeroing and adjusts its maximum normalized error to current network capacity. Acimovic J *et al.* proposed several distributed Haar-based data compression algorithms [8]. Under the algorithms, network is divided into groups and the data processing based on wavelet for each group is carried out in a distributed manner, with more energy-efficient communication. An energy-efficient data representation and routing scheme based on a distributed wavelet compression algorithm is proposed by Ciancio *et al.* [9]. It uses the lifting factorization of wavelet transform, exploits the natural data flow and aggregates data by computing partial wavelet coefficients which are refined as the data flows towards the central node. The algorithm also computes an optimal combination of data representation algorithm on a selected routing strategy at each node for each route, further reducing the overall cost. Zhou *et al.* designed a ring topology and proposed a wavelet based spatio-temporal data compression algorithm [10] which can support a broad scope of wavelets. Later, they designed another overlapping cluster topology. Combined it with the former ring topology, they proposed 2D and 3D wavelet-based data compression transmission algorithms [11] which are efficient in memory requirement and data compression. R. Wanger *et al.* [12] designed a new wavelet basis. The wavelet basis can form a tight frame and adapt to the structure of the network. Then they performs an irregular wavelet transform under the wavelet basis which can adapt to an arbitrary, multiscale network routing hierarchy. Because of the limited computing and memory resources in multimedia sensor networks, Rein S. *et al.* first conducted a fractional wavelet filter [13] and then proposed a fractional wavelet transform algorithm based on fixed-point arithmetic [14]. The algorithm can reduce the consumption on memory and computation greatly and degrade image quality only a little. Hu *et al.* designed a wavelet basis generating algorithm which is running at the sink. Based on the basis, sensor measurements are compressed and reconstructed by their wavelet transform-based distributed compressed sensing algorithm [15], with high performance on energy and reconstruction accuracy. For low power wavelet-based coder in visual sensor networks, Hadjou *et al.* compare the implementations of the classical convolutional based wavelets and

the relatively new lifting based wavelets, for choosing appropriate wavelets gaining tradeoff of energy and construction quality in image processing [16]. For multimedia sensor networks where nodes are deployed regularly in a 2D grid, Dutta *et al.* used red black wavelet lifting accompanied by difference detection technique for capturing spatial and temporal correlation respectively, and they proposed an energy-saving audio data compression technique and an energy-efficient routing scheme, which has good performance [17]. Hasan *et al.* studied the convolution based and the lifting based DWT implementation with the embedded hierarchical image compression structures using set partitioning in hierarchical trees (SPHIT) [18]. They found that the lifting based cdf 9/7 filter with five levels of decomposition produces excellent results in SPHIT image compression especially in low bit rates, with minimal performance degradation in memory reduction. For detecting data anomalies in WSNs, Takiannam *et al.* proposed an integrated data compression and anomaly detection algorithm [19]. In the help of only half of sensor measurements, the algorithm uses DWT to compress data first and then employs one-class support vector machine to detect anomaly, with good detection performance.

III. DATA-SMOOTHNESS BASED PREPROCESSING STRATEGY FOR WAVELET DATA PROCESSING

The general process of DSPS is as follows: Each cluster member calculates a sample mean and a sample standard deviation based on K sensory data as its approximate environment data characteristics. The change of the sample mean is used to measure the change of the environment approximately. A cluster member will update and report its sample mean to its cluster head if it discovers that the sample mean changes drastically. Each cluster head builds and holds a node order index (NOI) about its cluster members based on their sample means, generates an intra-cluster data vector by sorting intra-cluster data according to the NOI , takes wavelet transform on the data vector and finally sends the approximation coefficients and some detail coefficients to the sink. The sink reconstructs sensory data by taking inverse wavelet transform on its received coefficients for each cluster with the help of the corresponding NOI .

The key of our strategy is that the sequence of data to be processed is adjusted according to the NOI to improve the data smoothness and the update opportunity of the sensory data sample mean and the NOI is determined heuristically to decrease extra energy cost while better data smoothness is maintained.

A. Relevant Symbols and Indicators

- v_i : The i -th node;
- CH_i : The i -th cluster head;
- CM_i : The i -th cluster member;
- $s_i(j)$: The j -th sensory data of v_i ;

$\tilde{\mu}_i$: The sample mean of sensory data on v_i ;

$\tilde{\sigma}_i$: The sample standard deviation of sensory data on

v_i ;

P_i/Q_i : The Node Order Index (NOI) of CH_i ;

K : The number of sample data for calculating the sample mean and sample standard deviation;

$Threshold_P$: The threshold which measures the maximum position change, within $[0,1]$.

$[cA, cD]=DWT(s)$: DWT is a function that carries out a wavelet transform on data s , returning approximation coefficients and detail coefficients stored in cA and cD respectively.

$Data_C=F(cA, cD)$: F is a function that carries out zeroing and encoding on approximation and detail coefficients stored in cA and cD respectively, returning compressed coefficients which is stored in $Data_C$.

CR (Compression Rate): Suppose $s=(s_1, s_2, \dots, s_n)$ is raw data. Transform it by wavelet and set some detail coefficients 0. The number of non-zero coefficients is n_C . Assuming that a raw data s_i needs $Data_1$ bytes to be represented after encoding and a coefficient needs $Data_2$ bytes. Then the CR is

$$CR = \frac{n_C \times Data_2}{n \times Data_1}$$

MSE (Mean Squared Error): Suppose original data $s=(s_1, s_2, \dots, s_n)$. Transform it based on wavelet and set some detail coefficients 0. Gain reconstructed data $r=(r_1, r_2, \dots, r_n)$ by taking inverse wavelet transform on the non-zero wavelet coefficients. Then, the MSE of r against s is

$$MSE = \frac{1}{n} \sum_{i=1}^n (s_i - r_i)^2$$

E_s : The energy of data $s=(s_1, s_2, \dots, s_n)$ is $E_s = \sum_{i=1}^n s_i^2$

AEC (Average Energy Consumption): Suppose that a network has n nodes and the energy has been consumed by v_i is e_i . AEC is defined as follows:

$$AEC = \sum_{i=1}^n e_i / n$$

B. Strategy for Cluster Member

The data processing strategy for cluster members is shown in Table I. Each cluster member, say v_i , has two parameters: a sample mean $\tilde{\mu}_i$ and a sample standard deviation $\tilde{\sigma}_i$. $\tilde{\mu}_i$ is initialized by the first sample data, and $\tilde{\sigma}_i$ is initialized by the initial standard deviation $\sigma_{initial}$ which is dependent on experience. Afterwards, once v_i collects K data, it calculates the new sample mean $\tilde{\mu}_i^{new}$. Based on $\tilde{\mu}_i^{new}$, v_i calculates its sample Mean Varying degree (MV) according to formula (1), for measuring the changes of the environment approximately.

$$MV = \frac{|\tilde{\mu}_i^{new} - \tilde{\mu}_i|}{\tilde{\sigma}_i} \quad (1)$$

TABLE I: DATA PROCESSING OF CLUSTER MEMBER

1.	v_i collects its data $s_i(1)$;
2.	$\tilde{\mu}_i = s_i(1)$;
3.	$\tilde{\sigma}_i = \sigma_{initial}$; % $\sigma_{initial}$ is obtained from experience
4.	v_i sends $s_i(1)$ to its cluster head;
5.	$N_S = 2$; % N_S is the number of sensed data
6.	While <i>True</i>
7.	v_i collects and sends its data $s_i(N_S)$ to its cluster heads;
8.	If $\text{mod}(N_S, K) = 0$
9.	$\tilde{\mu}_i^{new} = \frac{1}{K} \sum_{j=N_S-K+1}^{N_S} s_i(j)$;
10.	$MV = \frac{ \tilde{\mu}_i^{new} - \tilde{\mu}_i }{\tilde{\sigma}_i}$;
11.	If $MV \geq 1$
12.	$\tilde{\mu}_i = \tilde{\mu}_i^{new}$;
13.	$\tilde{\sigma}_i = \left(\frac{1}{K} \sum_{j=N_S-K+1}^{N_S} (s_i(j) - \tilde{\mu}_i)^2 \right)^{1/2}$;
14.	v_i sends $\tilde{\mu}_i$ to its cluster head;
15.	$N_S = N_S + 1$;

If the sample mean changes much, *i.e.* $MV \geq 1$, v_i thinks its sensory data distribution has changed significantly and its $\tilde{\mu}_i$ and $\tilde{\sigma}_i$ should be updated. After updating the $\tilde{\mu}_i$ and $\tilde{\sigma}_i$, v_i informs its cluster head the new data feature. If $MV < 1$, v_i thinks there is only few changes occurring in its monitoring environment and the distribution of its sensory data remains unchanged. It is not necessary for v_i to calculate a new data feature. All of those are shown in Table I Lines 8-14. The value of the parameter K can be different for each node and it can also be adjusted according to the actual monitoring environment. When the distribution of the environment data changes frequently, K should be decreased properly, making the intra-cluster data to be processed have good smoothness for better wavelet data processing but increasing extra communication cost. When the environment changes slowly, the cluster member should increase its K to reduce energy consumption on the updates of sample mean and NOI .

C. Strategy for Cluster Head

Suppose a cluster, say the i -th cluster and its head CH_i , has m members $Mem=\{CM_j | j=1, 2, \dots, m\}$. In order to generate data to be processed with good smoothness, CH_i preserves two parameters: a mean list $\tilde{\mu}_i = \{\tilde{\mu}_i^{CM_1}, \tilde{\mu}_i^{CM_2}, \dots, \tilde{\mu}_i^{CM_m}, \tilde{\mu}_i^{CH_i}\}$ and an NOI $P_i = \{P_i(j) \in Members \cup \{CH_i\} | j=1, \dots, m+1\}$. The data processing strategy for cluster heads is detailed in Table II. Each cluster member sends one sensed data at a time. After getting the first batch of data ($S_i(1)$) sent by

cluster members, the cluster head sorts them in descending order, initializes $\tilde{\mu}_i$ and records the corresponding node order index in P_i (shown in Table II Lines 1-2). Once receiving an update information about sample mean of a cluster member, CH_i updates $\tilde{\mu}_i$ and descends it, getting a new node order index stored in Q_i . In order to determine that whether the order of nodes, which is used to adjust the order of intra-cluster data to be processed to gain good smoothness, needs to be updated, CH_i compares P_i with Q_i and calculates the degree of node Order Varying (OV) according to formula (2).

$$OV_i = \frac{\max_{\substack{1 \leq j_1, j_2 \leq m+1 \\ P_i(j_1) = Q_i(j_2)}} \{|j_1 - j_2|\}}{m+1} \quad (2)$$

TABLE II: DATA PROCESSING OF CLUSTER HEAD

% CH_i is a cluster head and its cluster members are $Mem = \{CM_j j=1, 2, \dots, m\}$.
1. Collect data from its cluster members, gaining $S_i(1) = \{s_i^{CM_1}(1), s_i^{CM_2}(1), \dots, s_i^{CM_m}(1), s_i^{CH_i}(1)\}$, and initialize $\tilde{\mu}_i = \{\tilde{\mu}_i^{CM_1}, \tilde{\mu}_i^{CM_2}, \dots, \tilde{\mu}_i^{CM_m}, \tilde{\mu}_i^{CH_i}\}$ with $S_i(1)$;
2. Sort $S_i(1)$ in descending order and gain the sorted data $S_s_i(1)$ and the NOI P_i :
$S_s_i(1) = \{s_i^{P_i(1)}(1), s_i^{P_i(2)}(1), \dots, s_i^{P_i(m+1)}(1)\}$,
$P_i = \{P_i(j) P_i(j) \in Mem \cup \{CH_i\}, j \in Z, 1 \leq j \leq m+1, \tilde{\mu}_i^{P_i(j+1)} \leq \tilde{\mu}_i^{P_i(j)}\}$;
3. $[cA_i(1), cD_i(1)] = DWT(S_s_i(1))$;
4. $Data_C_i = F(cA_i(1), cD_i(1))$;
5. Send $Data_C_i$ to the Sink;
6. $N_S = 2$;
7. While True
8. Collect data from its cluster members:
$S_i(N_S) = \{s_i^{CM_1}(N_S), s_i^{CM_2}(N_S), \dots, s_i^{CM_m}(N_S), s_i^{CH_i}(N_S)\}$
9. If receive $\tilde{\mu}^{CM_j}$ from CM_j ($j=1, 2, \dots, m$) OR $\tilde{\mu}^{CH_i}$ is updated
10. Update $\tilde{\mu}_i = \{\tilde{\mu}_i^{CM_1}, \tilde{\mu}_i^{CM_2}, \dots, \tilde{\mu}_i^{CM_m}, \tilde{\mu}_i^{CH_i}\}$
11. Sort $\tilde{\mu}_i$ in descending order and gain the NOI Q_i
$Q_i = \{Q_i(j) Q_i(j) \in Mem \cup \{CH_i\}, j \in Z, 1 \leq j \leq m+1, \tilde{\mu}_i^{Q_i(j+1)} \leq \tilde{\mu}_i^{Q_i(j)}\}$;
12. $OV_i = \frac{\max_{\substack{1 \leq j_1, j_2 \leq m+1 \\ P_i(j_1) = Q_i(j_2)}} \{ j_1 - j_2 \}}{m+1}$;
13. If $OV_i \geq Threshold_P$
14. $P_i = Q_i$;
15. Send P_i to the Sink;
16. Sort $S_i(N_S)$ according to P_i and gain the sorted data:
$S_s_i(N_S) = \{s_i^{P_i(1)}(N_S), s_i^{P_i(2)}(N_S), \dots, s_i^{P_i(m+1)}(N_S)\}$
17. $[cA_i(N_S), cD_i(N_S)] = DWT(S_s_i(N_S))$;
18. $Data_C_i = F(cA_i(N_S), cD_i(N_S))$;
19. Send $Data_C_i$ to the Sink;
20. $N_S = N_S + 1$;

If $OV_i \geq Threshold_P$, it is demonstrates that a new node order is necessary to smooth intra-cluster data to be

processed better under the new approximated data features of intra-cluster nodes. Therefore, CH_i updates P_i and informs the sink the new data processing order. Otherwise, CH_i does not think that a new node order can improve the smoothness of the intra-cluster data to be processed or much. Under the new node order, the performance of data compression can't be improved or can only be promoted a little, but increasing the energy cost on the update of NOI a lot. The process is shown in Table II Lines 9-15. After an intra-cluster data collection, CH_i sorts the data to be processed according to its NOI P_i and performs data compression based on some wavelets, as shown in Table II Lines 16-19.

Sort intra-cluster data to be processed, say o , according to P_i , a discrete signal, say s , is obtained. Compared with o , the change between any two adjacent discrete data of s is often more gradual. According to wavelet theory, the smoother the signal to be processed is, the more concentrated the energy distribution of the transformed signal is. Therefore, the wavelet compression performed on s is beneficial for improving the precision of reconstructed data at the sink and increasing the degree of data compression to decrease the in-network data transmission.

IV. THEORETICAL ANALYSIS AND EXPERIMENTS

The purpose of DSPS is to be applied in improving the performance (reconstruction precision and compression rate) of the algorithms based on wavelet data compression in WSNs, so we compare our strategy only with the algorithm which processes intra-cluster data directly by some wavelets without data smoothness preprocessing, say common wavelet based algorithm (CWA), in both theoretical analysis and experiments.

A. Theoretical Analysis

Suppose that $s = (s_1, s_2, \dots, s_n)$ is raw data. Take wavelet transform on s and get the approximation and detail coefficients $cA = (cA_1, cA_2, \dots, cA_k)$ and $cD = (cD_1, cD_2, \dots, cD_l)$. The corresponding low and high frequency energy are $E_{cA} = \sum_{i=1}^k cA_i^2$ and $E_{cD} = \sum_{i=1}^l cD_i^2$ respectively. Wavelet theory shows that there exists complementary relationship between E_{cA} and E_{cD} . For compressing signal, some unimportant coefficients are set 0 and the signal is reconstructed based on the approximation coefficients and parts of detail coefficients. In order to reduce the discrete signal reconstruction error, the lost signal energy caused by zeroing parts of detail coefficients must be decreased, that is the lost high frequency energy must be decreased. Furthermore, wavelet has good time-frequency characteristics. Reconstructing data mainly based on the low frequency energy which are around the high frequency energy (*i.e.* large detail coefficients) would cause large reconstruction

error inevitably. So here we analyze the strategies in the light of energy.

Though the wavelet used for our strategy can be any wavelet, we only analyze DSPS and CWA which are based on Haar wavelet for simplicity. Due to the theoretical complexity of other wavelets, we do not analyze the corresponding strategies but compare them through experiments.

For the sake of simplifying the problem analysis process, we only take the extreme case that $Threshold_P=0$ and $K=1$ for DPSP to explain. Correspondingly, the main process of DPSP (adjusting the order of intra-cluster data to be processed according to $Threshold_P$ and the sample mean of cluster members) is simplified as descending intra-cluster data to be processed.

(1) Under the same data compression rate, when $K=1$, the energy of reconstructed data gained by DSPS is better than CWA.

Suppose a cluster has 2^n nodes. The cluster head collects its members' sensory data and forms a data vector $s = (s_0, s_1, \dots, s_{2^n-1})$. Sort s in descending order and gain $s_{-}s = (s_{-}s_0, s_{-}s_1, \dots, s_{-}s_{2^n-1})$. Suppose the cluster head adopts 1-level Haar wavelet, the detail coefficients obtained by CWA and DSPS are

$$cD = (cD_0, cD_1, \dots, cD_{2^{n-1}-1})$$

$$cD_{-}s = (cD_{-}s_0, cD_{-}s_1, \dots, cD_{-}s_{2^{n-1}-1})$$

respectively, where:

$$cD_i = \frac{s_{2i+1} - s_{2i}}{\sqrt{2}}, \quad cD_{-}s_i = \frac{s_{-}s_{2i+1} - s_{-}s_{2i}}{\sqrt{2}}$$

For $|s_{-}s_{2i} - s_{-}s_{2i+1}| \leq |s_{2i} - s_{2i+1}|$, $|cD_{-}s_i| \leq |cD_i|$.

Under the same data compression rate, the number of detail coefficients which are not set 0 by CWA is the same with DSPS, say m . So, there are $m' = 2^{n-1} - m$ detail coefficients being set 0 correspondingly.

Denote the subscript sets of the best m' detail coefficients to be set 0 by CWA and DSPS as $zero$ and $zeroS$ respectively, we have

$$zero = \arg \min_{J = \{z_i | z_i \in Z \wedge z_i \in [0, 2^{n-1}-1], i=1, \dots, m'\}} \sum_{j \in J} cD_j^2,$$

$$zeroS = \arg \min_{J = \{z_i | z_i \in Z \wedge z_i \in [0, 2^{n-1}-1], i=1, \dots, m'\}} \sum_{j \in J} cD_{-}s_j^2.$$

For

$$\sum_{j \in zeroS} cD_{-}s_j^2 = \min_{J = \{z_i | z_i \in Z \wedge z_i \in [0, 2^{n-1}-1], i=1, \dots, m'\}} \sum_{j \in J} cD_{-}s_j^2$$

$$\leq \sum_{j \in zero} cD_{-}s_j^2 \leq \sum_{j \in zero} cD_j^2,$$

$$E_{zeroS} = \sum_{j \in zeroS} cD_{-}s_j^2, \quad E_{zero} = \sum_{j \in zero} cD_j^2,$$

there is $E_{zeroS} \leq E_{zero}$. Because

$$E_{total} = \sum_{i=0}^{2^n-1} s_i^2 = \sum_{i=0}^{2^n-1} s_{-}s_i^2,$$

$$E_{DSPS} = E_{total} - E_{zeroS} \text{ and } E_{CWA} = E_{total} - E_{zero},$$

we have $E_{DSPS} \geq E_{CWA}$.

Therefore, when the cluster head performs 1-level Haar wavelet compression, $E_{DSPS} \geq E_{CWA}$.

For multi-level Haar wavelet compression, similar to the above discussion, we can have $E_{DSPS} \geq E_{CWA}$.

In conclusion, under the same data compression rate, the energy of reconstructed data gained by DSPS is better than CWA.

(2) Under the same requirement on reconstructed data energy, when $K=1$, the data compression performance of DSPS is better than CWA.

Suppose that the energy difference between the original data and the reconstructed data should not be larger than $Threshold_E$. Then, the lost high frequency energy caused by zeroing some detail coefficients should not be larger than $Threshold_E$. For the same raw data, suppose $cD = (cD_0, cD_1, \dots, cD_h)$ and $cD_{-}s = (cD_{-}s_0, cD_{-}s_1, \dots, cD_{-}s_h)$ are the detail coefficients gained by CWA and DSPS under the same multi-level wavelet transform respectively. When $K=1$, it is easy to know that $|cD_{-}s_i| \leq |cD_i|$, $0 \leq i \leq h$. Denote the maximum numbers of detail coefficients which can be set 0 by CWA and DSPS as m and mS respectively, while the corresponding subscript sets of those detail coefficients gained by CWA and DSPS are denoted as M and MS .

$$M = \arg \max_{P \subseteq \{0,1,\dots,h\}} \{ |P| \mid \sum_{j \in P} |cD_j|^2 \leq Threshold_E \},$$

$$MS = \arg \max_{P \subseteq \{0,1,\dots,h\}} \{ |P| \mid \sum_{j \in P} |cD_{-}s_j|^2 \leq Threshold_E \},$$

$$m = |M|, \quad mS = |MS|.$$

For $\sum_{j \in M} |cD_j|^2 \leq Threshold_E$ and $\forall j \in \{0,1,\dots,h\}$,

$|cD_{-}s_j| \leq |cD_j|$, then $\sum_{j \in MS} |cD_{-}s_j|^2 \leq Threshold_E$. So, we have $m \leq mS$.

From the above, DSPS can lead to more detail coefficients able to be set 0 compared with CWA under the same requirement on reconstructed data energy. So the data compression performance of DSPS is better than CWA.

(3) Suppose the environment does not change drastically. Given a data compression rate, the energy consumption of DSPS is larger than that of CWA, but not much. With a fixed requirement of reconstruction precision, DSPS can save more energy compared with CWA.

Here free space energy consumption model [10] is used to calculate the energy consumed by node. The energy spent on a node transmitting an l -bit message over

a distance d is $E_{Tx}(l, d) = lE_{elec} + l\epsilon_{fs}d^2$ and the energy spent on a node receiving an l -bit message is $E_{Rx}(l) = lE_{elec}$, where $E_{elec} = 50\text{nJ/bit}$, $\epsilon_{fs} = 100\text{pJ/bit/m}^2$.

Suppose n is the average number of clusters in network, k is the average number of nodes in a cluster, d_1 is the average distance from a cluster head to the sink, d_2 is the average distance between a cluster member and its cluster head, $Data_1$ is the data volume of an encoded raw data, $Data_2$ is the data volume of an encoded node ID, and CR_{DPS} and CR_{CWA} are the average data compression rates gained by DPS and CWA respectively. For DPS, intra-cluster data should be sorted according to the *NOI* before being processed, and each cluster head should update its *NOI* dynamically in line with the environment. Here we suppose each cluster head needs to update its *NOI* every T data transmissions.

For CWA, the energy spent on T data transmissions is

$$E_{CWA}^T = n \times T \times (E_{Tx}(k \times Data_1 \times CR_{CWA}, d_1) + E_{Rx}((k-1) \times Data_1) + (k-1) \times E_{Tx}(Data_1, d_2))$$

For DPS, the energy spent on T data transmissions is

$$E_{DPS}^T = n \times T \times (E_{Tx}(k \times Data_1 \times CR_{DPS}, d_1) + E_{Rx}((k-1) \times Data_1) + (k-1) \times E_{Tx}(Data_1, d_2)) + E_{Tx}(k \times Data_2, d_1) + E_{Rx}((k-1) \times Data_1) + (k-1) \times E_{Tx}(Data_1, d_2))$$

The difference between E_{CWA}^T and E_{DPS}^T is

$$E_{CWA}^T - E_{DPS}^T = n \times k \times E_{Tx}(T \times Data_1 \times (CR_{CWA} - CR_{DPS}) - Data_2, d_1) - n \times (k-1) \times E_{Tx}(Data_1, d_2) - n \times E_{Rx}((k-1) \times Data_1)$$

1) Same data compression rate

For $CR_{DPS} = CR_{CWA}$,

$$E_{CWA}^T - E_{DPS}^T = -n \times k \times E_{Tx}(Data_2, d_1) - n \times (k-1) \times E_{Tx}(Data_1, d_2) - n \times E_{Rx}((k-1) \times Data_1) = -n \times (E_{elec} \times (2(k-1) \times Data_1 + kData_2) + \epsilon_{fs} \times (k \times Data_2 \times d_1^2 + (k-1) \times Data_1 \times d_2^2))$$

From the above expression, the energy consumption of DPS is larger than CWA under the same compression rate, but not much. That is because DPS has two kinds of extra energy cost caused by sample mean update at cluster members and *NOI* update at cluster heads. And compared with energy spent on transmitting data, the two kinds of extra energy cost are small. When the environment does not change drastically, the update of *NOIs* and sample means is not frequent. So, the total energy consumption of DPS is larger than CWA, but not much.

2) Same reconstruction precision requirement

From (2), it is known that $CR_{DPS} \leq CR_{CWA}$.

If we want that $E_{CWA}^T - E_{DPS}^T > 0$, then

$$k \times E_{Tx}(T \times Data_1 \times (CR_{CWA} - CR_{DPS}) - Data_2, d_1) > (k-1) \times E_{Tx}(Data_1, d_2) + E_{Rx}((k-1) \times Data_1)$$

must be required to hold. *i.e.* T must be required to satisfy the following inequality:

$$T > \frac{1}{CR_{CWA} - CR_{DPS}} \left(\frac{Data_2}{Data_1} + \frac{(k-1)(2E_{elec} + \epsilon_{fs}d_2^2)}{k(E_{elec} + \epsilon_{fs}d_1^2)} \right).$$

$$\text{When } d_1^2 - d_2^2 > 500, \text{ then } \frac{2E_{elec} + \epsilon_{fs} \times d_2^2}{E_{elec} + \epsilon_{fs} \times d_1^2} < 1.$$

Generally, $d_1^2 - d_2^2 > 500$ holds for almost all wireless sensor networks. So, when

$$T > \frac{1}{CR_{CWA} - CR_{DPS}} \times \frac{Data_2 + Data_1}{Data_1},$$

then $E_{CWA}^T - E_{DPS}^T > 0$ holds certainly.

For example, if $Data_2/Data_1 = 0.5$, $CR_{CWA} = 0.7$ and $CR_{DPS} = 0.5$, then DPS can save more energy as long as $T > 7.5$. When the environment changes slowly, T is large under general cases, and so DPS is more energy-efficient compared with CWA.

TABLE III: PRIMARY PARAMETERS

Parameters	Value
Network Size	120m×60m
Number of Nodes	512
Communication Radius of Node	Adjustable, ≤80m
Single Raw Data Size	64bits
Single Coefficient Data Size	64bits
Node ID Size	16bits
Message Head Size	160bits
Number of Clusters	8
Number of Cluster Members	64
Wavelet Type	{db1, db2, coif1, bior2.2, bior4.4}
Level of Wave Transform	5
Number of Data Collection	2000
Sample Data Distribution	U(a-b, a+b), $a \in [20, 80]$, $b \in [5, 15]$
Compression Rate (CR)	0.2-0.9
Mean Square Error (MSE)	20-100
Threshold _P	0.5
K	10

B. Experiments

Within a 120m×60m area, 512 nodes are deployed to form a network. The network is divided into 8 clusters, and each cluster has 64 cluster members. Intra-cluster data processing based on wavelets is performed by cluster heads. In experiments, we compare the performance of

DSPS with CWA under different wavelets: 1) the average CR under the same MSE, 2) the average MSE under the same CR and 3) the AEC. The primary experimental parameters are shown in Table III.

1) Performance of data compression

To evaluate the performance of data compression of DSPS and CWA with different wavelets, MSE is designated as the requirement of reconstruction error. We increase it from 20 to 100 and calculate average data compression rate, with results shown in Fig. 1. Among the four wavelets db1, db2, coif1, bior2.2 and bior4.4, db1 wavelet performs best regardless of DSPS or CWA. And it is obvious that the data compression rate gained by DSPS is lower than CWA under different wavelets, *i.e.* DSPS is better for data compression. From Fig. 1, we find that the best data compression is gained by db1, and the compression performance is degraded by db2, coif1, bior2.2 and bior4.4 in sequence on the whole.

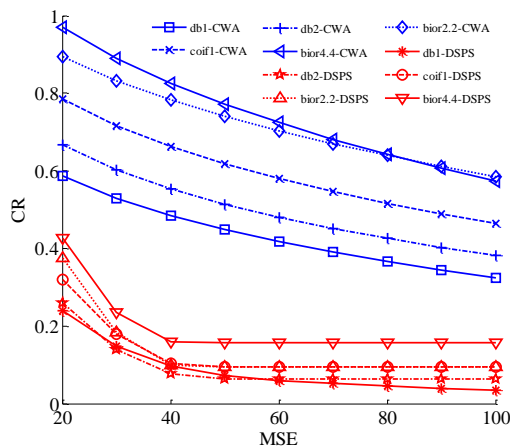


Fig. 1. Comparison of data compression rate

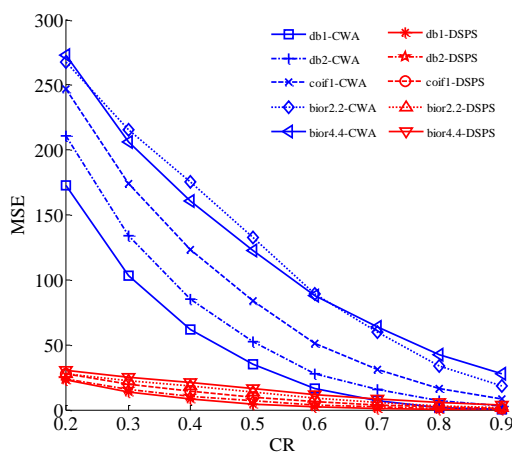


Fig. 2. Comparison of data reconstruction precision

2) Performance of data reconstruction

In order to compare the data reconstruction performance of DSPS with CWA, we do experiments with fixed data compression rates which range from 0.2 to 0.9 and collect corresponding average MSE on the whole network for the strategies with different wavelets,

shown in Fig. 2. When the compression rate is 0.2, the MSE gained by DSPS is much less than the MSE gained by CWA. As compression rate increases, the gap of MSE between DSPS and CWA decreases, but the MSE of DSPS is smaller than that of CWA under a same wavelet type. Compared with CWA, the data reconstruction performance of DSPS is better. The difference of average MSE gained by DSPS with wavelets db1, db2, coif1, bior2.2 and bior4.4 is not large compared with CWA. And either DSPS or CWA, db1 leads to the best data reconstruction and bior4.4 the worst.

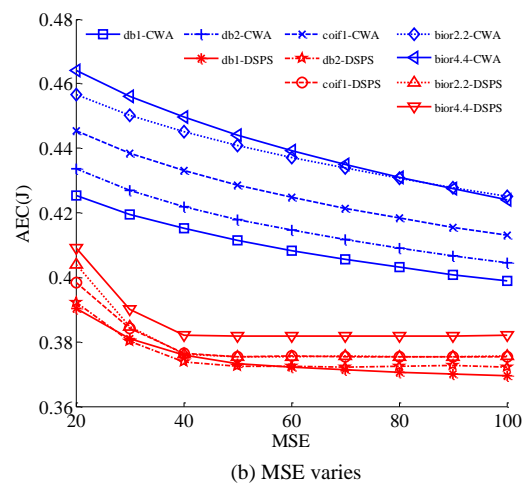
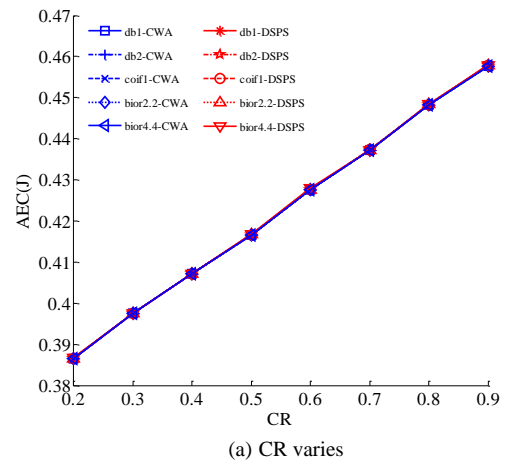


Fig. 3. Comparison of average energy consumption per node

3) Energy consumption

Under the same data compression rate, DSPS and CWA have the same energy consumption on coefficient data transmission. But DSPS has two extra energy costs compared with CWA: 1) energy cost on sample mean update and 2) energy cost on NOI update. In this case, DSPS consumes more energy than CWA. As compression rate varies, the AECs of the two strategies with different wavelets are shown in Fig. 3(a). From Fig. 3(a), we find that the energy consumption increase for DSPS against CWA is not large, but DSPS can gain more precise data at the sink which is shown in Fig. 2.

Under a fixed requirement for MSE of reconstruction data, DSPS can gain smaller data compression rate

compared with CWA, *i.e.* the data compression performance of DSPS is better than that of CWA, which is shown in Fig. 1. So, although DSPS has to spend two extra energy costs, its total energy consumption should be usually smaller than CWA. This energy consumption is proved by our experiments whose results are shown in Fig. 3 (b): the AEC of DSPS is lower than that of CWA. From Fig. 3, we find that db1 brings the best energy performance and bior4.4 the worst roughly.

V. CONCLUSIONS

The node deployment of wireless sensor networks is often dense, causing the raw data sensed by nodes in network have greater relevance. Data compression based on wavelet can remove redundant information among the raw in-network data, which contributes it to a feasible data processing scheme for wireless sensor networks. For the sake of improving the performance of data compression algorithms based on wavelets, we propose a Data-Smoothness based Preprocessing Strategy (DSPS) for wavelet data processing in wireless sensor networks. On one hand, DSPS can promote the data reconstruction precision under a given data compression rate. On the other hand, it can improve the data compression performance under a fixed data reconstruction precision, decreasing the in-network data transmissions greatly and prolonging network lifetime. Theoretical analysis and experiments show that DSPS can improve the performance of wavelet based data processing algorithms in wireless sensor networks.

K is an important parameter of DSPS. In this paper, we determined K approximately according to some experiences. How to find the best K dynamically according to the real network situation to optimize the performance of DSPS is one of our future works.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China No. 61373174, the National Funds of China for Young Scientists No. 11205080, and Youth Fund of Luoyang Institute of Science & Technology No. 2008QZ11.

REFERENCES

- [1] H. Abusaimah, M. Shkoukani, and F. Alshrouf, "Balancing the network clusters for the lifetime enhancement in dense wireless sensor networks," *Arabian Journal for Science and Engineering*, vol. 39, no. 5, pp. 3771-3779, May 2014.
- [2] A. Sinha and D. K. Lobiyal, "Probabilistic data aggregation in information-based clustered sensor network," *Wireless Personal Communications*, vol. 77, no. 2, pp. 1287-1310, Jul. 2014.
- [3] H. Lin and H. Uester, "Exact and heuristic algorithms for data-gathering cluster-based wireless sensor network design problem," *IEEE-ACM Transactions on Networking*, vol. 22, no. 3, pp. 903-916, Jun. 2014.
- [4] G. Minos and K. Amit, "Wavelet synopses for general error metrics," *ACM Transactions on Database Systems*, vol. 30, no. 4, pp. 888-928, Dec. 2005.
- [5] J. M. Zhang, Y. P. Lin, S. W. Zhou, and J. C. H. Ouyang, "Haar wavelet data compression algorithm with error bound for wireless sensor networks," *Journal of Software*, vol. 21, no. 6, pp. 1364-1377, Jun. 2010.
- [6] L. Z. H. Cheng, H. X. Wang, and Y. Luo, *Theory and Application of Wavelet*, 1st ed. Beijing: Science Press, 2004, ch. 3-4, pp. 55-131.
- [7] H. M. Chen, J. Li, and P. Mohapatra, "RACE: Time series compression with rate adaptivity and error bound for sensor networks," in *Proc. 2004 IEEE Int. Conf. Mobile Ad Hoc Sensor Syst.*, Fort Lauderdale, 2004, pp. 124-133.
- [8] J. Acimovic, R. Cristescu, and B. Lozano, "Efficient distributed multiresolution processing for data gathering in sensor networks," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Philadelphia, 2005, pp. IV837-IV840.
- [9] A. Ciancio, S. Pattem, A. Ortega A, and B. Krishnamachari, "Energy-efficient data representation and routing for wireless sensor networks based on a distributed wavelet compression algorithm," in *Proc. 5th Int. Conf. on Information Processing in Sensor Networks*, New York, 2006, pp. 309-316.
- [10] S. W. Zhou, Y. P. Lin, J. M. Zhang, J. C. H. Ouyang, and X. G. Lu, "A wavelet data compression algorithm using ring topology for wireless sensor networks," *Journal of Software*, vol. 18, no. 3, pp. 669-680, Mar. 2007.
- [11] S. W. Zhou, Y. P. Lin, S. T. Ye, and Y. P. Hu, "A wavelet data compression algorithm with memory-efficiency for wireless sensor network," *Journal of Computer Research and Development*, vol. 46, no. 12, pp. 2085-2092, Dec. 2009.
- [12] R. Wagner, S. Sarvotham, and R. Baraniuk, "A Multiscale Data Representation for Distributed Sensor Networks," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Philadelphia, 2005, pp. 549-552.
- [13] S. Rein and M. Reisslein, "Performance evaluation of the fractional wavelet filter: A low-memory image wavelet transform for multimedia sensor networks," *Ad Hoc Network*, vol. 9, no. 4, pp. 482-496, Jun. 2011.
- [14] S. Rein and M. Reisslein, "Low-memory wavelet transforms for wireless sensor networks: A tutorial," *IEEE Communications surveys & Tutorials*, vol. 13, no. 2, pp. 291-307, May 2011.
- [15] H. F. Hu and Z. H. Yang, "An energy-efficient distributed compressed sensing architecture for wireless sensor networks based on a distributed wavelet compression algorithm," in *Proc. 2011 Int. Conf. Wirel. Commun. Signal Process.*, Nanjing, 2011, pp. 1-4.
- [16] B. Hadjou, A. Mammeri, and A. Khoumsi, "Determining suitable wavelet filters for Visual Sensor Networks," in *Proc. Saudi Int. Electron., Commun. Photonics Conf.*, Riyadh, 2011, pp. 1-5.
- [17] I. Dutta, R. Banerjee, and S. Das Bit, "Energy efficient audio compression scheme based on red black wavelet lifting for wireless multimedia sensor network," in *Proc. Int. Conf. Adv. Comput., Commun. Informatics*, Mysore, 2013, 1070-1075.
- [18] K. K. Hasan, U. K. Ngah, and M. F. M. Salleh, "The most proper wavelet filters in low-complexity and an embedded hierarchical image compression structures for wireless sensor network implementation requirements," in *Proc. IEEE Int. Conf. Control Syst., Comput. Eng.*, Penang, 2012, pp. 137-142.
- [19] S. Takiannangam and W. Usaha, "Discrete wavelet transform and one-class support vector machines for anomaly detection in wireless sensor networks," in *Proc. 19th Int. Symp. Intelligent Signal Process. Commun. Syst.: "Decade Intelligent Green Signal Process. Commun."*, Chiang Mai, 2011, pp. 1-6.



Yalin Nie was born in Yiyang, Hunan, China, August 1981. She received her B.S. in Computer Science and Technology in 2004 and M.S. in Computer application technology in 2007, both from School of Computer and Communication of Hunan University in Changsha, Hunan, China. Currently, she is working toward the Ph.D. in School of

Computer Science and Technology of Xidian University in Xi'an, Shaanxi, China. She is a lecturer in Department of Computer and Information Engineering of Luoyang Institute of Science and Technology in Luoyang, Henan, China. Her research interest includes routing protocols, mobile computation and data aggregation in wireless sensor networks.



Haijun Wang was born in Xinyang, Henan, China, October 1979. He received his B.S. in Computer Software in 2002 from School of Mathematics and Econometrics, and M.S. in Computer Software and Theory in 2007 from

School of Computer and Communication, both of Hunan University in Changsha, Hunan, China. Currently, he is working toward the Ph.D. in School of Electronic Engineering of Xidian University in Xi'an, Shaanxi, China. He is a lecturer in School of Mathematics & Statistics of Henan University of Science & Technology in Luoyang, Henan, China. His research interest includes machine learning, image processing and network optimization methods.



Yujie Qin was born in Huining, Gansu, China, September 1976. She received her B.S. in Safety Engineering from Harbin University of Science and Technology and her Ph.D. in Electrical Theory and New Technology from Southwest Jiaotong University. She is an associate professor in the Department of Computer and Information Engineering of

Luoyang Institute of Science and Technology in Luoyang, Henan, China. Her research interest includes network analysis and design, electromagnetic calculation and RFID.