2016 International Conference on Distributed Computing in Sensor Systems

Tree-Structured Linear Approximation for Data Compression over WSNs

Chu-Ming Wang
Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan
cmwang@ vc.cs.nthu.edu.tw

Chia-Cheng Yen
Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan
ccyen@ vc.cs.nthu.edu.tw

Wen-Yen Yang
Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan
wanyaneyang@yahoo.com.tw

Jia-Shung Wang
Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan
jswang@cs.nthu.edu.tw

***Abstract*—In wireless sensor networks (WSNs), how to reduce the power consumption thus lengthen the system life time is one of the key issues to sustain the services. According to the radio model, packet transmission depletes a much more substantial amount of the energy budget when compared to sensing and processing. Therefore, it is desirable to compress or filter the sensing data effectively in order to save the transmission power eventually. Recently, the model-based scheme is proved to be a promising solution, which usually approximate temporal data by a piecewise linear function. In this paper, a tree-structured linear approximation scheme is proposed to compress sensing data according to an optimal rate-distortion (R-D) relationship. The main design goals are two: (1) providing a bottom-up procedure to explore the best-fit piecewise partition for modeling globally; (2) considering the heterogeneity of sensors simultaneously using our proposed rate-distortion adjustment. That is, a distortion allocation procedure is designed to allocate the distortions to sensor nodes for aware of the heterogeneous properties. Thus the proposed spatio-temporal scheme is adaptable to heterogeneous sensors, various sampling rate, and outliers of data. A real-world dataset simulation is applied to demonstrate the effectiveness. For nearly all combinations with distortion requirements, the proposed method shows better performance than the earlier approaches in terms of data reduction.**

Keywords: WSNs; model-based scheme; linear approximation; compression; convex hull; rate-distortion; outliers removal

I. INTRODUCTION

Wireless sensor networks (WSNs) are basically data gathering systems that utilize a large number of small, battery-powered, and resource-limited sensing devices to collect heterogeneous signals simultaneously. These wirelessly linked disposable sensor nodes observe physics data in fixed time intervals and then communicate either among each other or directly to the base station. The sink (base station) will store up all the data it receives in order to be applied to designated fields of human activity including surveillance, healthcare, environmental and utility monitoring. Power consumption is a key issue to sustain the services effectively. Zhang, Meratnia, and Having [1] showed that sensors with low battery level cause more errors and outliers than those with a full battery. According to the radio model, packet transmission consumes a much more substantial amount of the energy budget compared to sensing and processing.

Many techniques have been developed to reduce the transmission power consumption while ensuring fidelity during gathering. Modeling or sampling is a better solution to deal with this problem, which uses a small set of features to represent the complete data signals.

There are three main common applicable categories of data collection models. The first one is simply prolonging the updating interval in periodic-based data collection, which will reduce the frequency of data being sent. However, longer updating periods might decrease the data fidelity. Second, event-based data collection, which means sensors will only report their status to the sink when predefined events happen. This is a more preferable method, but it also creates additional overhead for event monitoring and communication between sensors. The third, and perhaps the most promising solution [2], is the model-based data collection scheme, which transmits models to approximate data instead of sending the full set of original signals. In WSNs, sensor will continuously sense the data in short intervals, so the data stream in a WSN is highly correlated temporally. It is a good idea to use a small number of model parameters to approximate data, which would lower both the storage cost and the transmission cost. In the discussions in [2], the authors categorize model-based compression schemes into four types: Constant Model, Linear Model, non-Linear Model and Correlation Model.

For examples, Piecewise Constant Approximation (PCA) [3] and Adaptive Piecewise Constant Approximation (APCA) [4] are simple methods which model the data in windows with a maximum error constraint (tolerance) = (d_{max}-d_{min})/2, where d_{max} and d_{min} are the maximum value, minimum value within a window, respectively. Piecewise Linear Histogram (PWLH) [5] is an extension of APCA. The only difference between these two is a constant or a line (in PWLH) to fit the data. The Slide Filter (SF) [6] is another piecewise linear approximation method which is currently the best method in compression ratio [2]. SF maintains a set of possible lines initially, and when new data arrives, SF checks whether these lines can be used for approximation. A line will extend for as long as possible. And it will stop when the line set becomes empty due to unmatchable input. Chebyshev Approximation (CHEB) [7] is a nonlinear-model approach, which used the Chebyshev

2325-2944/16 $31.00 © 2016 IEEE
DOI 10.1109/DCOSS.2016.37

model to approximate the data in fixed-length windows. For the Grouping and Amplitude Scaling (GAMPS) approach [8], it groups similar sensors together, and chooses one as the base and only transmits the margin between the base and rest of the signals in the group to the sink.

Recently, Compressive Sensing (CS) has attracted great research attentions, which can acquire and compress data at the same time via randomness measurements. In addition, due to the simplicity of the acquisition at the sensor nodes, the CS scheme is a fit for distributed applications, such as distributed video coding [9] and WSNs [10] [11]. The application of the spatial-temporal CS scheme to WSNs [12] [13] provides promising possibility for data gathering with stochastic spatio-temporal models. Three steps in ST-HDACS [13] are: (1) in each time snapshot a randomly selected subset of sensors participate the data aggregation; (2) employs an adaptive hierarchical aggregation for data routing; (3) utilizes the Matrix Completion algorithm to recover all the data for the entire network over the whole data collection time.

Unlike the above scheme, our proposed method approximates temporal data by a piecewise linear function, consisting of connected line segments. Initially, a set of possible line segments of equal size is maintained as a complete binary tree for W consecutive data of current stream in each of sensor nodes. Linear regression is employed to draw the best-fit line first, and then the rate-distortion (R-D) pruning process is applied to trim the tree to several candidate forms of incomparable R-D pairs. Optimal distortion allocation will be assigned to each sensor. After that, each sensor node will locate the most fitting R-D pair and retrace the R-D pruning process to extract the most satisfactory tree. The tree further shrunken by merging possible pairs of segments iteratively to a minimum bit rate obeying the distortion. Finally, a refinement procedure with the help of outlier removal is implemented to further compress the data streams.

SF is an on-line method, whether the line should be further extended to or stopped is based on the incoming data. Our approach is an off-line method, which draw the approximation line globally in the window of W samples. This means that our method can use a global view to deal with the data.

Our contributions in this paper are two: (1) providing a bottom-up procedure to explore the best-fit linear approximation globally; (2) considering the heterogeneity of sensors simultaneously using our rate-distortion adjustment. Furthermore, with our outlier removal procedure, thus, the proposed scheme is adaptable to heterogeneous sensors, various sampling rate, and outliers of data.

The rest of this paper is organized as follows. In Section II related studies are introduced and discussed. A detailed description of the proposed method is delineated in Section III. In Section IV, experimental results of several life datasets are presented to demonstrate and compare the performance between our proposed method and the Slide Filter method. The conclusion is drawn in Section V.

II. BACKGROUND

Following are the introduction to the studies which are highly related to or being employed in our proposed method.

A. The Slide Filter

As mentioned before, the Slide Filter [6] is currently the best method in compression ratio. To compare our method with SF, this section will explain the SF model in detail.

Initially, temporal numerical data stream is represented as a 2-D data point (t_i, x_i) sequence, $i \in [1, n]$. Given the target distortion ε, SF then finds an approximate line, the difference between this line and each point should be lower than ε. SF names the first two points of a line, x_1 and x_2, and then uses these two points as references to draw two lines. One of the lines, lower bound L_1, passes through x_1+ε and x_2-ε, while the other line, upper bound U_1, passes through x_1-ε and x_2+ε. Every line that passes through the intersection of these two lines and between L_1 and U_1 will have difference lower than ε with either x_1 or x_2. So the lines sloping between L_1 and U_1 can be combined as a set of lines to be used in the model.

Now add a new point, x_3. Using the same rule as above to draw four lines: L_{13} , U_{13}, L_{23} and U_{23}. With a simple observation, a line with a slope higher than U_{13}, U_{23} or U_1 will fail to fit all three points. Similarly, all line with a slope lower than L_{13}, L_{23} or L_1 will fail to fit all three points. So we can set the new upper bound as $U = \min(U_{13}, U_{23}, U_1)$, and the new lower bound as $L = \max(L_{13}, L_{23}, L_1)$. Using the same process and iterative to reduce the upper bound and lower bound until the new point x_i enters the model with x_i+ε being smaller than the current lower bound or x_i-ε being higher than current upper bound. Then the line will stop at x_{i-1}. Figure 1 is a simple demonstrative example.

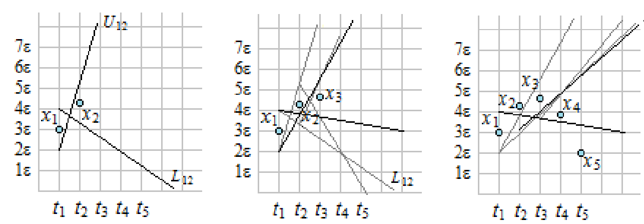

Figure 1: A demonstrative example of the Slide Filter.

B. Tree Structure Optimal Pruning

Tree-structured Optimal Pruning method begins with an initial large tree, and prunes back until the pruned sub-tree have the fewest number of leaves while maintaining a given level of fidelity. This method is first introduced by Breiman, Friedman, Olshen, and Stone [14] for classification applications. The method was later adopted by Chou, Lookabaugh, and Gray [15] to conduct a complete analysis in streaming data encoding.

In the tree structure, each leaf holds a part of the sequential data for compressing. Parent node combines all of the data of its offspring, thus a higher level node will contain fewer data.

A node also retains its distortion and bit rate after compression.

With these principles, a full tree with n leaves (L_1, L_2, …, L_n) divide up all data in the root to n disjoint parts. Leaves in this tree support a small amount of data so that a low distortion is preserved at the cost of a high bit rate. When the R-D pruning is applied, the number of leaves will decrease, and each remaining leaf will hold more data. This can reduce the bit rate, but at the expense of an increase in distortion.

C. Rate Distortion Optimal Allocation

Here we describe how to consider the heterogeneity of sensors simultaneously. We provide: (1) a model of characterizing the rate-distortion behavior of each sensor signal; and (2) a distortion allocation procedure of distributing the distortions to sensor nodes for aware of the heterogeneous properties.

When bit rate R is fixed and large enough, the distortion $D(R)$ is proportional to 2^{-2R} [16]. If R is fixed with a small number (i.e. R < 1 bit/pixel), $D(R)$ varies as $R^{1-2\gamma}$[16]. Thus we can arrive at the following relation:

$$D(R) = CR^{1-2\gamma}. \tag{2.1}$$

Here $\gamma > 1/2$ and $C > 0$ are parameters dependent on the fluctuation of signals. Based this rate-distortion relationship, we can reorganize the allocation of bit rates among sensors to reach a lower distortion result. For example, if a sensor with low C and γ, at the same rate allocation, its distortion will be lower than one with higher C and γ. So we can reallocate a lower bit rate to this sensor, and reserve additional bit rate for sensors with higher C and γ.

In our early works [17][18], we utilize this rate-distortion technique to solve the optimal rate-distortion allocation problem for WSNs. The problem is formulated as below:

$$\mathit{Minimize} \sum_{i=1}^{N} R_i(D_i) \quad \text{subject } \mathit{to} \sum_{i=1}^{N} D_i \le D. \tag{2.2}$$

In [18], we have the optimal solution for distortions D_i and bit rates R_i as below:

$$D_i = \zeta_i^{\eta_i} \cdot (x^*)^{-\eta_i} \quad \text{and } R_i = K_i \cdot (D_i^*)^{\frac{1}{1-2\gamma_i}}$$

where $K_i = C_i^{\frac{1}{2\gamma_i - 1}}$, $\quad \zeta_i = \frac{-K_i}{1-2\gamma_i}$, $\quad \eta_i = \frac{2\gamma_i - 1}{2\gamma_i}$

and x^* is a root of $\sum_{i=1}^{n} (\zeta_i^{\eta_i} \cdot x^{-\eta_i}) - D = 0.$ (2.3).

III. THE PROPOSED METHOD

A. Problem Formulation

The wireless sensor network consists of N sensors, s_1, s_2, …, s_n, and a sink node D. The network graph $G = (V, E)$ has a node set $V = \{s_1, s_2, …, s_n, D\}$, and edge set E consisting of edges between nodes that can communicate with each other directly (in one hop). Each sensor has an instantaneous reading X_i^t at time t, and we denote the sensor signal of node i within a window of size W as $(X_i^1, X_i^2, \dots, X_i^W)$. In the model-based approach, these W number of sensing data are usually approximated by piecewise linear approximation of line segments, L_1, L_2, …, L_p, ($1 \le p \le T$), each of which can be modeled by a linear regression with intercept a and slope b. This p number piecewise line segments can be recognized as a partition π of these W sensing data. That is, these W consecutive data is divided into a sequence of variable-length data segments. Thus, the t^{th} estimated value of X_i^t (say in the τ^{th} point of the q^{th} line segment) can be calculated as:

$$\hat{X}_i^{\tau} = a_q + \tau \cdot b_q \tag{3.1}$$

Usually, the Mean Square Error (MSE) is used to measure the estimation error $d(X_i^t, \hat{X}_i^t)$ between predicted value and its corresponding raw data. Thus, the distortion of node i for a partition π can be represented as:

$$D_i^{\pi} = \sum_{t=1}^{W} \left(X_i^t - \hat{X}_i^t\right)^2. \tag{3.2}$$

As mentioned above, the goal is to reduce the volume of transmission while maintaining data fidelity. So, the problem can be formulated as:

$$\underset{\pi \in \Pi}{\text{minimize}} \sum_{i=1}^{N} \alpha_i \cdot R_i^{\pi}$$

$$\mathit{subject} \text{ to} \sum_{i=1}^{N} D_i^{\pi} \le D_{target} \tag{3.3}$$

where α_i is the transmission cost of each sensor i to the sink D.

As discussed in Section II.C, our optimal distortion allocation procedure collects the model parameters c and γ from all sensor nodes. With these data, the sink node could calculate the optimal distortion allocation for the sensors. Thus, a simple tree is constructed for this purpose. We describe the flowchart of the proposed method for solving Problem (3.3) in the following subsection.

B. The Flowchart

In this paper, a tree-structured linear approximation with optimal rate-distortion (R-D) control method to deal with the sensor data signal is proposed. Similar to other approaches, our method approximates each sensor data signal by a piecewise linear function.

Initially, a set of possible line segments of equal size is maintained as a complete binary tree for W consecutive data collected in each of sensor nodes. Linear regression is employed to draw the best-fit line first, and then the rate-distortion (R-D) pruning process is applied to trim the tree to several candidate forms of incomparable R-D pairs. Then, an optimal distortion allocation procedure is employed to allocate the distortions to sensor nodes accordingly. With the assigned distortion value, each tree is further shrunken by iteratively merging possible pair of segments to a minimum rate (# of line segments) while obeying the distortion. Finally, a refinement procedure, with the assistance of outlier removal, is implemented to further compress the data streams. The flowchart of our method is depicted as the Figure 2 below.

C. Initialization

There are two parameters. The first one is window size W, a parameter that decides how many consecutive sensing data will be processed simultaneously. The second one is target distortion D_{target}, which is the total distortion limitation. In other words, in the whole process of compression, and the sum of all sensors distortion will not be greater than D_{target}. Even though the Optimal Distortion Allocation step will allocate the optimal target distortion automatically, the sum of these allocated distortions should be within the designated threshold.

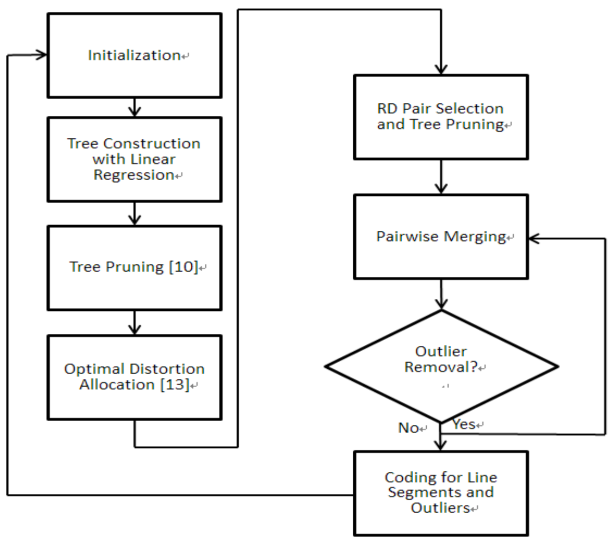

Figure 2: The flowchart of the proposed method.

D. Tree Construction with Linear Regression

A complete binary tree is constructed initially. The root hold the entire W (assumed to be power of two for simplicity) data, $\{d_1, d_2,\ldots, d_W\}$. Two children of the parent node hold left half and right half of data, respectively. An example of initial tree with $W = 512$ is depicted in Figure 3.

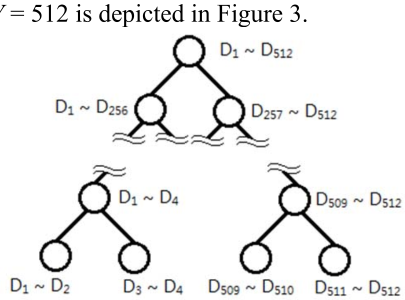

Figure 3: Example for complete binary tree with $W = 512$.

Each tree node uses linear regression to extract the slope and the intercept of possible line that can be used to approximate its own data set perfectly. With the slope and the intercept, the compression rate and distortion can be calculated. The lowest level of the tree is $\log W$. Each node in level-$\log W$ has exactly two sensing data, the slope and the intercept can be obtained by simple calculation. And these nodes have no distortion because the line perfectly represents the two sensing data. That is, the distortion D of this complete binary tree is 0; however its rate R will be the largest.

Consider the three cases in Figure 4. In case (b) the binary tree is fully formed. Since the window size W and tree-structure are known, it is easy to locate the start point and end point of the line without transmitting their location to the sink. In the following steps, this tree will be trimmed with fewer nodes eventually so as to find out a better (suboptimal) partition with corresponding line segments. In the extreme case (d), only one line segment left, its corresponding tree is singleton with $\{0\}$. Its distortion D will be the largest, while rate R is the smallest.

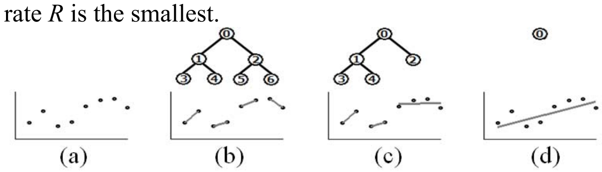

Figure 4: An example of tree with eight nodes. (a) the data, (b) the complete binary tree before pruned, (c) the tree trimmed at node 2 and (d) the tree with only root node, which corresponds to one single line.

E. Tree Pruning

In a graph with bit rate and distortion as the X-Y axes, every sub-tree of the full tree can be drawn as a point in the graph according to its distortion and bit rate. The most desirable sub-tree will be the one with lowest distortion and bit rate, hence this means that sub-trees that fall within the lower boundary of the convex hull has priority over ones that do not. In [15], their algorithm considers the slopes between each pair of vertices in this R-D plane, and then locates the vertices that results from the sub-trees after pruning. By locating the vertices in this manner, the curve in lower boundary of the convex hull is traced in a clockwise fashion.

For a pruned sub-tree S, let the bit rate denoted as $R(S)$ and the distortion be $D(S)$. All of the R-D pairs ($R(S)$, $D(S)$) can be depicted in the R-D plane, see Figure 5(a). As mentioned, the singleton tree $T(t_0)$ has the smallest R and the largest D while the whole tree T has the largest R and the smallest D. For each pruned sub-tree S of T, $R(S) \le R(T)$ and $D(S) \ge D(T)$. Thus, ($R(S)$, $D(S)$) is fundamentally monotonic affine. Therefore, in the R-D plane, ($R(T(t_0))$, $D(T(t_0))$) is the upper-left corner of the convex hull and ($R(T)$, $D(T)$) is the lower-right corner.

The goal of the Tree Pruning procedure is to locate the vertices clockwise around the lower boundary of the convex hull, $t_0 \le S_n \le \ldots \le S_2 \le S_1$. We start with T and prune it back to the root t_0.

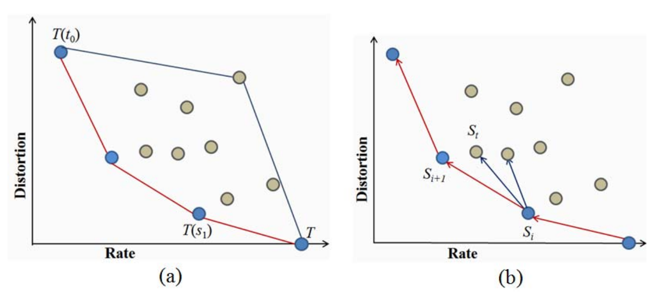

Figure 5: (a) The R-D plane and the R-D pairs corresponding to pruned trees. (b) An example to locate points on the convex hull.

Consider the case of pruning a tree from S_i to S_{i+1} in Figure 5(b). First, we prune off a single branch from some interior node $t\in S_i$ out of S_i to get the set of all pruned sub-tree of S_i. Each of the pruned cases has its own R-D pair depicted as a point on the R-D plane. For a case, say S_t, the slope of the line segment between S_t and S_i is $\Delta D(S_t)/\Delta R(S_t)$. Then S_{i+1} is chosen to be such a S_t with the minimum slope among all pruned sub-tree of S_i.

When all such nodes being fixed, we can model the R-D function with appropriate c and γ later. The worst-case time complexity to locate this lower boundary of the convex hull is $O(W\log_2 W)$ [15], where W is the number of data points in a processing window.

F. Optimal R-D Rate Allocation

Once the R-D pairs, (C, γ)s, corresponding to the all sensors received by the sink, based on Eq.(2.3), the Optimal Rate-Distortion Allocation (ORDA) procedure will calculate the optimal distortion for each of sensors under a given target distortion D_{target}..

G. R-D Pair Selection and Tree Pruning

Once the assigned distortion (by ORDA) returned to each sensor, it retraces the convex-hull path to locate the first R-D pairs lower than the given distortion. Consider Figure 6 as an example.

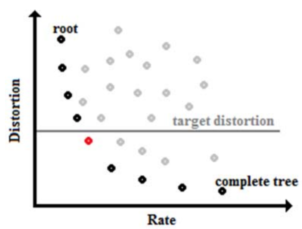

Figure 6: Locate the appropriate R-D pair in the convex path.

Each node in the above figure is an R-D pair corresponding to a sub-tree of the complete binary tree. The left-top one is the tree with only one node, the root, and the right-bottom one corresponding to the original complete binary tree. After executed the Tree Pruning procedure, the R-D pairs in the lower convex hull was fixed and marked, and the other nodes not in the lower convex hull are ignored in the following steps. After ORDA, each sensor gets its assigned (target) distortion, which is marked as a horizontal line in the above figure. And the appropriate R-D pair just lowers than the horizontal line is marked in red. Finally, the pruned tree corresponding to this R-D pair will be used to construct the adequate piecewise linear approximation.

H. Pairwise Merging

There exists some limitation in the Tree Pruning procedure, that is, the line segments are of size 2^l for some l, because of the binary tree structure. Consider Figure 7 as an example. In the tree pruning procedure, the data corresponding to leaf b is not similar to the data in leaf c, so they cannot be merged, thus two line segments are still required ultimately. Conversely, the data in leaf a and leaf b are quite similar, however, these two nodes cannot be merged because in the tree-structured model.

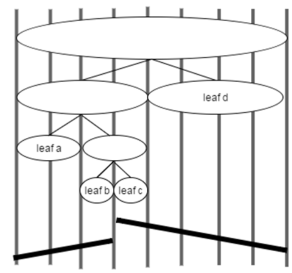

Figure 7: An illustration of pruned tree that can be merged further.

Here we give a pairwise merging procedure to solve this problem. For a pruned tree and the corresponding approximation line segments, we conduct a scan procedure as follows. Each pair of adjacent segments is considered being merged, linear regression is applied then to get the approximated solution for this merged data set and calculate the R-D pair after merging. If both the rate is smaller and the distortion is not greater than the assigned (target) distortion, this merging is seen as an acceptable choice. This scan procedure continues until no more feasible merging found.

I. Merging with Outlier Removal

Outlier (noise) is another issue [1]. Two data segments cannot be merged further, sometimes is influenced by the existing noises. Outliers are sensing data that are far away from the mean than considered acceptable; however, how to justify and remove outliers is a challenge. In this paper, we provide an Outlier Removal procedure to extract and remove these odd data points in the sense of the merging criterion. Actually, we do not ignore these points instead of sending them as extra data. The extraction is simply selecting the point with the extreme value per iteration.

Similar to the processing in the Pairwise Merging procedure, after one outlier is eliminated from the data set, two neighboring segments are assumed being merged, linear regression is applied to get the approximated solution for this merged data set. And then calculate the R-D pair after merging to check the value for merging (the same criterion used in Pairwise Merging). If it is feasible, this merging will proceed (using Pairwise Merging); otherwise, this step is terminated.

IV. Experimental Results

A. Data set abd Parameter Settings

We conducted series of simulations to demonstrate both the coding efficiency and computing feasibility of the proposed approach through the dataset in [2][19]. This dataset was extracted from a wide variety of sensors deployed for an environmental monitoring project. Several issues are considered here, such as the heterogeneity of sensors, the volatility of data values, the diversity of sampling rates, and the range of regions where the data were collected. In our simulation, 16 heterogeneous sensors of 3,072 data items are employed. The dataset and corresponding statistics are listed in Table 1.

Table 1: Typical sensors in data set [19].

	Sampling rate	min	max	mean	Standard deviation	Short-term fluctuation
moisture	300 sec.	-4.13	1340.00	221.22	5.15	medium
humidity	60 sec.	-3.19	129.93	56.14	14,24	medium
lysimeter	300 sec.	1642.25	2347.99	1935.86	47.82	medium
snow-height	600 sec.	-3062	2613	21.83	87.54	low
temperature	60 sec.	-58.4	275	0.85	6.81	medium
CO_2	16 sec.	350	2000	541.34	162.91	high
radiation	600 sec.	-1912	7997	680.37	224.562	medium
wind-direction	60 sec.	0.0	359.99	122.16	74.49	high

In [2], The Slide Filter (SF) is recognized as one of the most powerful methods with leading compression ratios. SF is included in the convenient framework [2].

All the experiments were run on a PC with Intel Core i5 processor @ 3.10 GHz and 3.42 GB of main memory.

As in [2], the size of each signal is of 32 bits, and the parameters of each line segment, slope and intercept are also of 32 bits. The overhead of coding a tree is 2 bits times the number of internal nodes. Each merging requires 1 bit for each leaf, and each outlier costs 32 bits for the value and 9 bits for the position if the window is of size 512.

The indices for comparison are: the compression ratio, the distortion and the execution time. The distortion is measured using the Normalized Root-Mean-Square Error (NRMSE) and the Bit Rate is defined below:

$$Bit\ Rate = \frac{\#Compress\ \text{Data Bit}}{\#Original\ \text{Data Bit}} \times 100\%$$

B. Comparing with the Slider Filter

Table 2 lists the performance comparison between our proposed method and the Slide Filter. Figure 8 gives a demonstrative example. Note that the target distortion of SF is the maximum error tolerance ε = $(d_{max}-d_{min})/2$, but it is NRMSE instead, in our solution. To tune with the same distortion (NRMSE) for comparison, we manually choose ε for SF, and modulate our target distortion for an agreement in both NRMSEs.

Table 2: The comparison of bit rate and NRMSE.

	Proposed Method		Slide Filter	
	Bit Rate	NRMSE	Bit Rate	NRMSE
moisture	3.5%	8.2%	12.5%	7.8%
humidity	0.6%	6.7%	2.0%	7.2%
lysimeter	0.9%	7.6%	4.4%	6.3%
snow-height	7.9%	2.8%	11.1%	4.1%
temperature	1.5%	7.6%	2.1%	8.4%
CO_2	5.6%	7.3%	17.1%	7.8%
radiation	12.1%	6.3%	17.4%	8.0%
wind-direction	14.3%	6.5%	26.3%	7.2%

In Table 2, it is easy to see that our proposed method outperforms SF in terms of compression rate with the similar distortion. The main reasons are:

- Less overhead: Our overhead includes: (i) the slope and the intercept for each node, (ii) the tree structure for each window *W*, which amounts to almost 2 bits times the number of leafs in the tree. The overhead of SF includes: the start and end points, which results in almost 1.5 times the overhead of our method. So, even though our method has to transmit more lines than SF, we are able to still achieve a better compression ratio.

- Global optimization: SF is an on-line method which dealing with data on the fly, whether the line should be further extended to or stopped is based on the characteristics of incoming data. Our method is an off-line method, which draw the approximation line globally in the window of *W* samples. This means that our method can use a global view to deal with data. When most of the data in the data segments are very close to the line, the method allows for some data far away from the approximation line to be processed. An example is shown in Figure 8.

- Outlier removal: When the new coming data is an outlier, SF approximation line extension will stop immediately and start a new line. Outliers also afflict our method, but because of the global view of data, it is possible to locate outliers and deal with them through the outlier removal procedure. An example is shown in Figure 9.

Table 3 lists the execution time of our proposed method and SF. Beyond our expectations, the execution time of our method is basically on par with SF. In some cases, our proposed method is slower than SF; however in other cases the result is reversed. The execution time of SF is largely based on the stability of dataset. SF will extend the line as longer as much as possible. When it comes a new data point, SF will draw two lines for each point in the line segments with the new point, which means longer line segments will consume much more time than line segments of shorter length.

Our proposed method is much stable than SF in execution time because the same amount of linear regression are used in different datasets. For a data segment of 16 sensors with 3,072 data each, our method requires 0.23~0.39 seconds to execute, while SF needs 0.08~0.63 seconds.

C. Comparison of our four procedures

To better understand how the proposed method works, the statistics of bit rate and distortion of four key procedures are listed in Table 4.

Unlike comparing with SF, we fix the target distortion so we are able to determine the effectiveness of each procedure in terms of distortion and bit rate.

It is interesting to note that distortions are reduced in some test cases (i.e. radiation, temperature, and CO_2) when we apply ORDA. The main goal of ORDA, as listed in Table 4, is for bit rate reduction. As expected, the NRMSE values are raised because we share the allocated distortion to multiple/all sensors which necessitate more distortion to reduce their compression rate. It is also apparent that the NRMSEs of these datasets are raised when ORDA is applied. So we can infer that this is because the NRMSE is a normalizing parameter which is affected by the difference between maximum data and minimum data. The unanticipated results of the NRMSE values may be caused by the sensors being allocated larger target distortions to also acquire greater differentials between their max and min data.

Table 3: Execution time (seconds) compared with SF.

	Proposed Method	Slide Filter
moisture	0.281	0.56
humidity	0.234	0.357
lysimeter	0.265	0.0829
snow-height	0.281	0.632
temperature	0.25	0.345
CO_2	0.328	0.104
radiation	0.296	0.151
wind-direction	0.391	0.0809

Table 4: Bit rate and distortion of four key procedures.

	Tree Pruning	ORDA	Pairwise Merging	Outlier Removal and Merging
	Bit Rate			
moisture	7.7%	6.6%	5.0%	3.5%
humidity	0.68%	0.60%	0.60%	0.61%
lysimeter	2.4%	1.7%	1.1%	0.9%
snow-height	12.3%	11.6%	9.4%	7.9%
temperature	2.6%	2.0%	1.6%	1.5%
CO_2	12.0%	10.7%	8.2%	5.5%
radiation	18.2%	16.2%	14.3%	12.1%
wind-direction	21.7%	20.9%	17.0%	14.2%
	NRMSE			
moisture	6.1%	6.5%	6.8%	8.2%
humidity	6.4%	6.7%	6.7%	6.7%
lysimeter	5.4%	5.5%	5.6%	7.7%
snow-height	2.1%	2.4%	3.4%	2.8%
temperature	6.5%	6.3%	6.4%	7.6%
CO_2	6.7%	6.7%	6.8%	6.9%
radiation	6.9%	6.4%	6.7%	6,4%
wind-direction	6.2%	6.2%	6.2%	6.5%

With the addition of the ORDA step, the reduction in compression ratio highly fluctuates in each case. The result is highly depends on the differences between sensors in the same test case. In this step, manually tuning the target distortion sensor by sensor does not affect the outcome.

The merge step and outlier removal step lower almost the same compression ratio in every test case, but the NRMSE increases in the outlier removal step are much drastically than it increases in the merging step.

Table 5 lists the execution time of the seven steps. The Outlier Removal procedure is included in the Merging step, because after Outlier Removal, the Merging step is processed again. The "LR" in this table represents the total time of all linear regression in our method. We can observe that the Initial step demands the most time because the step has to calculate most of the linear regression in the tree. Linear regression is not only utilized in the Initial step, but is also applied in the Merging step and the Outlier step, thus the execution time of the Outlier step affects the execution time of total linear regression.

Table 5: Execution time (seconds) of each step.

	Initial	Pruning	ORDA	Merging	Outlier	Data read	LR
moisture	0.094	0.031	0.031	0.047	0.031	0.078	0.079
humidity	0.062	0.047	0.016	0.031	0.016	0.078	0.078
lysimeter	0.142	0.048	0.04	0.015	0.015	0.015	0.062
snow-height	0.077	0.031	0.016	0.063	0.031	0.078	0.093
temperature	0.126	0.048	0.015	0.015	0.015	0.046	0.062
CO_2	0.124	0.032	0.031	0.032	0.032	0.031	0.093
radiation	0.111	0.047	0.015	0.077	0.062	0.03	0.172
wind-direction	0.128	0.031	0.046	0.124	0.078	0.046	0.142

D. *Comparing with different window sizes*

The different window sizes, 1024, 512, and 256 are also considered as shown in Table 6. Bit rate can be reduced as window size increased. The complete tree contains more data when window size become larger, otherwise less data is included in complete tree. Thus, the efficiency of compression drops as the window size is small and the bit rate grows as well.

Table 6: The comparison of different window sizes.

	W=1024	W=512	W=256
Rate			
moisture	2.62%	3.64%	4.97%
humidity	0.48%	0.61%	0.94%
lysimeter	0.81%	0.96%	1.34%
snow-height	7.86%	7.96%	8.22%
temperature	1.41%	1.54%	1.78%
CO_2	6.04%	5.96%	6.08%
radiation	12.13%	12.21%	12.29%
wind-direction	14.53%	14.38%	14.18%
NRMSE			
moisture	6.82%	6.48%	6.24%
humidity	7.66%	6.73%	5.69%
lysimeter	5.86%	5.63%	5.13%
snow-height	2.52%	2.46%	2.36%
temperature	7.27%	6.29%	5.48%
CO_2	6.75%	6.74%	6.75%
radiation	6.46%	6.44%	6.28%
wind-direction	6.26%	6.21%	6.06%

E. *Comparing with ST-HDACS*

Here, we roughly compare our proposed spatio-temporal method with ST-HDACS because it is the CS-based data collection scheme to reduce the transmission redundancy from both spatial domain and temporal domain [13]. In the simulation results of ST-HDACS, λ percent of sensor nodes are selected at each collection step. In simulation, the values of λ are set as 0.25, 0.50, and 0.75, respectively. For the Sea Surface Temperature case with λ = 0.25, the errors (Normalized Mean Absolute Error) are ranging from 2.3% to 6.8% (for different network sizes) and the corresponding energy savings is around 54.3%. That is, the (energy saving, NMAE) pair is (54.3%, 2.3-6.8%), or the (rate, NMAE) pair is roughly equal to (25%, 2.3-6.8%). And ours (rate, NRMSE) is (1.78%, 5.48%) as shown in the row temperature of Table 6.

In [11], the energy model consists of computation and communication costs, and the communication cost is proportional to the amount of data in transmission.

V. CONCLUSION

In this paper, a spatio-temporal tree-structured linear approximation scheme for serving heterogeneous sensors simultaneously is presented. The main contributions are two: (1) providing an efficient bottom-up procedure to explore the best-fit piecewise solutions instead of scanning approach as SF; (2) considering the heterogeneity of sensors simultaneously using the R-D distortion allocation so as to maximize the distortion usage.

A real-world dataset simulation is applied to demonstrate the effectiveness of the model. The results of proposed method are compared with SF and ST-HDACS. For nearly all test datasets, the proposed method shows better performance than SF and ST-HDACS in terms of data reduction under similar distortion condition. The comparison of execution time as given, our method is basically on a par with SF. The statistics of bit rate and distortion of our four key procedures are listed. Bit rate are all reduced when we apply optimal distortion allocation because of the consideration of the heterogeneity of sensors spatially. Some sensors which necessitate more distortion to reduce their compression rate will be assigned the suitable amount since the distortion is sharable.

REFERENCES

[1] Yang Zhang, Nirvana Meratnia, and Paul Having, "A Outlier Detection Techniques for Wireless Sensor Networks: A Survey," *IEEE Communication Surveys & Tutorials*, Vol. 12, No. 2, pp. 159-170, second quarter 2010.

[2] Nguyen Quoc Viet Hung, Hoyoung Jeung, and Karl Aberer, "An Evaluation of Model-Based Approaches to Sensor Data Compression," *IEEE Transaction on Knowledge and Data Engineering*, Vol. 25, No. 11, pp. 2334-3447, November 2013.

[3] I. Lazaridis, and S. Mehrotra, "Capturing Sensor-Generated Time Series with Quality Guarantees," *Proceedings of 19th IEEE ICDE*, pp. 429-440, March 2003.

[4] K. Chakrabarti, E. Keogh, S. Mehrotra and M. Pazzani, "Locally adaptive dimensionality reduction for indexing large time series databases," *ACM Transactions on Database Systems*, Vol. 27, No. 2, pp. 188-228, June 2002.

[5] C. Buragohain, N. Shrivastava, and S. Suri. "Space Efficient Streaming Algorithms for the Maximum Error Histogram," *Proceedings of* 23rd *IEEE ICDE*, pp. 1026-1035, April 2007.

[6] Hazem Elmeleegy, Ahmed K. Elmagarmid, Emmanuel Cecchet, Walid G. Aref, and Willy Zwaenepoel, "Online Piece-wise Linear Approximation of Numerical Streams with Precision Guarantees," *VLDB*, Vol. 2, No. 1, pp. 145-156, August 2009.

[7] Andrew Cheng, S. Hawkins, Christopher Monaco, Lillian Nguyen, and Gordon Seagrave, "Data compression using Chebyshev transform," Patent US 7,249,153 B2, August 2007.

[8] Sorabh Gandhi , Suman Nath , Subhash Suri , and Jie Liu, "GAMPS: compressing multi sensor data by grouping and amplitude scaling," *SIGMOD*, pp. 771-784, July 2009.

[9] T.T. Do, Yi Chen, D.T. Nguyen, N. Nguyen, L. Gan, and T.D. Tran, "Distributed Compressed Video Sensing," in *Proc. IEEE Conf. on Image Processing* (ICIP '09), pp. 1393-1396, 2009.

[10] Jin Wang, Shaojie Tang, Baocai Yin, and Xiang-Yang Li, "Data Gathering in Wireless Sensor Networks Through Intelligent Compressive Sensing," *INFOCOM, Proceedings IEEE*, Orlando, FL, March 25-30, 2012.

[11] X. Xu, R. Ansari, A. Khokhar, A. V. Vasilakos, "Hierarchical data aggregation using compressive sensing (HDACS) in WSNs." *ACM Transactions on Sensor Networks*, Vol. 11, No. 3, Article 45, February 2015.

[12] M. Roughan, Y. Zhang, W. Willinger, and L. Qiu, "Spatio-Temporal Compressive Sensing and Internet Traffic Matrices (Extended Version)," *IEEE Transactions on Networking*, Vol. 20, No. 3, pp. 662-676, June 2012.

[13] X. Xu, R. Ansari, A. Khokhar, "Spatio-Temporal Hierarchical Data Aggregation using Compressive Sensing," *IEEE International Conference on Distributed Computing in Sensor Systems* (DCOSS), Fortaleza, Brazil, June 10-12, 2015.

[14] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, "Classification and Regression Trees," CA: *Wadsworth*, 1984.

[15] Philip A. Chou, Tom Lookabaugh, and Robert M. Gray. "Optimal Pruning with Applications to Tree-Structured Source Coding and Modeling," *IEEE Transactions on Information Theory*, Vol. 35, No. 2, pp. 299-315, March 1989.

[16] S. Mallat, and F. Falzon, "Analysis of low bit rate image transform coding," *IEEE Transactions on Signal Processing ,* Vol. 46, p.p. 1027-1042, April 1998.

[17] Chun Lung Lin and Jia-Shung Wang, "Optimal Rate Allocation of Compressed Data Streams in Multihop Sensor Networks," 5th *IEEE/ACM International Conference on Distributed Computing in Sensor Systems* (DCOSS'09), Marina Del Rey, California, June 8 - 11, 2009.

[18] Chun-Lung Lin, Chen-Lung Chan, and Jia-Shung Wang, "Optimization of Rate Allocation with Distortion Guarantee in Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 22, No. 7, pp. 1230-1237, July 2011.

[19] http://www.swiss-experiment.ch.

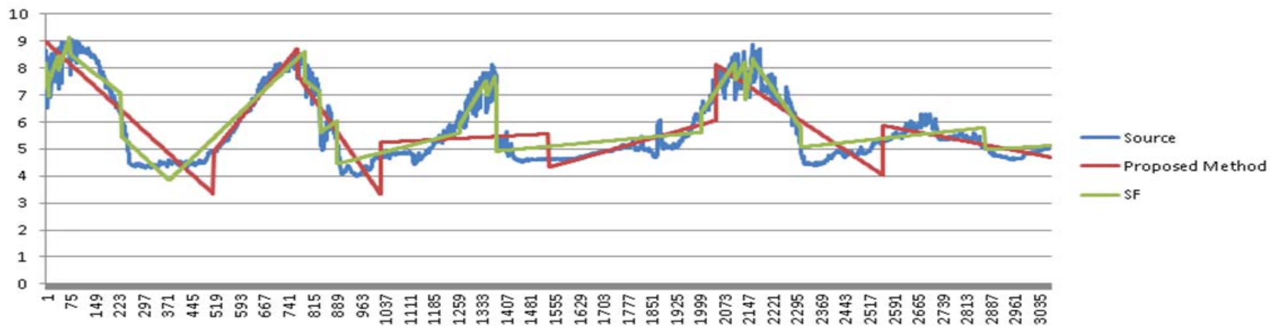

Figure 8: A demonstrative example.