RACE: Time Series Compression with Rate Adaptivity and Error Bound for Sensor Networks

Huamin Chen, Jian Li, and Prasant Mohapatra
Department of Computer Science
University of California at Davis
Davis, CA 95616.
Email: {chenhua, lijian, prasant}@cs.ucdavis.edu.

Abstract

Sensor networks usually have limited energy and transmission capacity. It is beneficial to reduce data volume for dissemination in a sensor network that monitors continuous physical processes in order to reduce energy consumption. Data compression schemes in use should be able to adapt to limited bandwidth while preserving high data quality. We propose a wavelet-based, error aware compression algorithm that is targeted to achieving these goals. First of all, it can adjust its maximum normalized error to current network capacity. Additionally, errors due to multiple passes of compression during multi-hop relaying are additive and thus can be estimated easily upon data reconstruction. Moreover, during data dissemination, error ranges can be narrowed through an opportunistic patching process when excess bit rate is available. Consequently, the performance is less subject to the volatility of physical processes. The algorithm has been evaluated in various aspects and demonstrated to be effective in rate adaptivity, error range narrowing, and preservation of statistical interpretation.

1 Introduction

A sensor network is formed of a number of networked sensors that are deployed to gather information collectively from their proximity and relay it to a data sink for further processing. Sensor networks are widely used in environmental exploration, military systems, and factory production monitoring. Sensor Networks usually have limited battery source and link bandwidth. A large body of research has been done to address some challenging issues in deploying large scale sensor networks, for example, route setup and maintenance [1, 2, 3], energy consumption and conservation [4, 5]. Closer to the context of our work, authors in [6, 7] proposed methods for data volume shrinkage, which, however, are for single variable cases and thus not directly applicable to multivariate monitoring sensor networks.

Based on characteristics of traf c sources, data dissemination in sensor networks can be divided into two major classes: continuous dissemination and event-triggered delivery. In the class of continuous dissemination, sensing devices monitor certain physical processes continuously and periodically relay the collected data to a sink in a hop-by-hop manner. For instance, environmental observations, and production process monitoring fall into this class. In the class of event-triggered delivery, sensors do not transmit data on a regular basis. Instead, the sink advertises its interest of certain types of events to the entire network, and individual sensors generate a report to the sink only upon the occurrence of an event of interest. Moving object detection and tracking in battle eld are examples of this category.

Our motivation for this work was based on the observations that sensor networks are usually energy and bandwidth constraint and that data rates in continuous time series may be large and varying over time. It is desirable that we can compress time series from volatile physical processes into streams with constant or limited bit rates in order to meet network capacity limitation. To the best of our knowledge, there has been no reported work on time series compression with rate adaptivity and the ability to preserve statistical interpretation of time series. We believe such a compression method will signi can tly enrich the research efforts in network capacity planning and estimation, data quality maintenance, network traf c scheduling, congestion alleviation, and energy conservation.

Our work attempts to address multiple issues, namely, data dissemination with capacity constraint, preservation of statistical interpretation, and energy conservation from an integrated view. In particular, we propose a Rate Adaptive Compression with Error bound (RACE) algorithm in the context of M^3 (multi-terminal, multi-hop, and multivariate) sensor networks. Major contributions of this work are as follows.

- We propose an error-based wavelet coef cients zero-

0-7803-8815-1/04/$20.00 ©2004 IEEE

ing algorithm. This algorithm has low computation complexity and its error bounds are additive when multiple error thresholds are imposed. It can generate ef - cient data encoding patterns and meet various rate constraints.

- We adopt an opportunistic patching process to smooth and minimize error ranges for the proposed compression algorithm.
- We evaluate the algorithm and demonstrate its capability of adapting to bit rate, minimizing and smoothing error range, and preserving statistical interpretation.

Although our work is focused on adaptive compression of time series in the class of continuous dissemination sensor networks, some of the principles are applicable to event-triggered sensor networks as well.

The rest of the paper is organized as follows. Section 2 outlines main challenges in sensor networks, some related efforts in existing literature, and motivation of our approaches. Section 3 gives some preliminaries which facilitate our discussion. Section 4 presents our RACE compression algorithm and analyzes its effectiveness in time series compression. Section 5 elaborates main procedures of our RACE algorithm. Performance evaluation of RACE algorithm is reported in Section 6. Concluding remarks and future work plan are presented in Section 7.

2 Related Efforts and Our Approaches

The following questions motivate our design philosophy. They are answered in the rest of this section.

- Why data compression is not replaceable by lowering sampling frequency/precision?
- Why constant or limited bit rate (CBR and LBR) is preferred over other forms (e.g. variable bit rate, VBR)?
- How can we bound error for lossy compression, especially multi-pass compression on multivariate time series in M^3 sensor networks?
- Why encoding is necessary during compression?

Intuitively, low sampling rate and sparse sensor deployment can shrink the data volume. However, such approach cannot capture the detailed movement of observed variables and thus only delivers information of low quality. Compression of high precision data is able to preserve more details and can trade off accuracy for bandwidth when the transmission capacity limit is encountered. A threshold triggered dissemination scheme is proposed in [7], which, however, does not use data compression to meet bandwidth constraints.

It has been known that wireless transceivers are major energy consumers in sensor networks [4]. So, transmitting raw data without compression will result in too much energy consumption. But only compression is not enough. Due to the volatility of physical processes, common compression algorithms with hard precision requirement may produce variable bit rate output stream, which is not desirable for transmission scheduling in sensor networks. Without a good transmission scheduling, sensor nodes have to listen more frequently for pass-by traf c in hope of not incurring too much delay. As pointed out in [5], this kind of *idle listening* is also a major source of energy consumption. Thus an algorithm that can generate constant bit rate or limited bit rate output streams is important for better transmission regulation and energy conservation. To this end, we believe that it is very desirable to have a rate adaptive compression algorithm in order to shrink data volume and reduce transmission energy consumption. An interesting related work is Embedded Zerotree Wavelet (EZW) proposed in [8] that interprets errors in forms of noise signal strength, which is different from those used in numerical error analysis and is thus not suitable for our purpose.

Because we want to compress time series from volatile physical processes into CBR or LBR output streams which can meet network capacity constraint, lossy compression becomes the only choice. There are two types of compression based on the retention of accuracy: lossy and lossless. Lossy compressed data cannot be reconstructed to the original accuracy as lossless compression does. This naturally leads to the concern of data quality. In the context of time series compression, data quality is quanti ed as their numerical accuracy. As discussed in [9], lossy compression based on wavelet decomposition needs special mechanisms to protect against unbounded errors. The wavelet coef cient zeroing techniques (explained in Section 3) in that work, however, cannot be directly translated to compression algorithms: there is lack of mechanisms of how these wavelet coef cients should be encoded and how we can estimate data error after zeroing some leaf tree(s). Hence we propose Gradient Error Tree structure and error-based zeroing technique in order to achieve error aware compression.

In a multivariate monitoring sensor networks, multiple time series are converged at a sink for statistical interpretations. The interpretations are either based on individual variables' features such as mean and variance, or on multiple variables' correlation and regression. While there are plenty of network research efforts on single variable dissemination [6, 7], their conclusions are not well applicable into the multivariate sensor networks. Since the compressibility of different time series may be quite different, some compression methods may treat some better than the others. For instance, the shrinkage based on some principal components [10] may incur different levels of delity among the multiple time series. This differentiation may dimin-

ish statistical signi can ce of different variables. In other words, differentiated treatment of multiple variables may not render meaningful statistical values. Variables with diverse patterns should be treated with adequate fairness in order to preserve their individual characteristics as well as the correlationship among them. Additionally, we need to deal with the problem of multi-pass compression on multiple time series in multi-hop sensor networks. With the error additivity property of our technique, it is easy to estimate data error after multi-pass compression in multi-hop relaying.

Now let us explain the importance of encoding schemes in adaptive data compression. For a compression to be effective in volume shrinkage, it is necessary to exploit data patterns and derive an ef cient encoding scheme. Otherwise compression outcome may not be satisfying. For instance, the two sequences with the identical membership but different patterns {ababab} and {aaabbb} have varying compression ratios using different encoding schemes. When compressed using run length encoding [11] based on a single character's occurrence frequency, the rst sequence is encoded as {ababab} and the second is encoded as {a3b3}, the second one is more ef ciently encoded than the rst one. But if the encoding counts the occurrence frequency of two characters, then the rst sequence is encoded as {(ab)3}, its compression ratio is improved. Therefore, it is pivotal to incorporate encoding into wavelet coef cient zeroing to implement an ef cient compression algorithm.

Finally, we would like to point out that it is dif cult, if not impossible, to provide *hard guarantee* on data quality in a sensor network with potential con icting between available network capacity and data volume of time series. So it is more practical to pursuit a *soft guarantee* on data quality, which is the guide for the design of our rate adaptive compression algorithm which attempts to meet bandwidth constraint and provide minimal error bound in a best effort manner.

3 Preliminaries

In this section we brie y review some basics in order to facilitate later discussion on our compression algorithm.

3.1 Error Norm

An L^2 error norm was chosen to be the accuracy objective for quality guarantee in [6]. The L^2 error norm is de ned as

$$L^2 \equiv \sqrt{\sum_i e_i^2}, \tag{1}$$

where e_i is the error of each element. This choice of de nition is demonstrated to be effective in trend prediction. We n d that L^2 error norm is weak at estimating error bound on operation results from functions such as mean, variance, correlation, etc, because it only captures the overall error range of the data set instead of per element errors. We thus choose to use L^∞ as our error norm, which is de ned as the maximum absolute error among all the data elements:

$$L^\infty \equiv max_i|e_i|. \tag{2}$$

It is observed that the values of the absolute errors from multiple physical processes are likely to be in different magnitude orders. When they are treated with the same precision objective, those in lower magnitudes might be diminished by others in higher magnitude orders. We are thus motivated to use a variant of the absolute error-*normalized error* as the error norm. The normalized error refers to the absolute error of the normalized data. More speci cally , a time series is rst normalized between its maximum and minimal values. An element of the time series d_i is normalized into $norm(d_i)$ using the following equation:

$$norm(d_i) = \frac{d_i - d_{min}}{d_{max} - d_{min}}, \tag{3}$$

where d_{min} and d_{max} are the maximum and minimal values that have been observed. Note that the normalized value range is thus between [0,1]. The normalized error between the actual data element d_i and its reconstructed value d_i^r is $|norm(d_i) - norm(d_i^r)|$.

The normalization is illustrated as follows. Ten temperature samples that are used in Section 6 are {24.33, 24.40, 24.31, 24.35, 24.41, 24.43, 24.50, 24.58, 24.48, 24.54}. The normalized values are {0.074, 0.333, 0.000, 0.148, 0.370, 0.444, 0.704, 1.000, 0.630, 0.852}. If the reconstructed values (normalized version) from a lossy compression are {0.07, 0.33, 0.00, 0.14, 0.370, 0.44, 0.70, 1.00, 0.63, 0.85}, then the normalized errors are {0.004, 0.003, 0.000, 0.008, 0.000, 0.004, 0.004, 1.000, 0.000, 0.002}.

Normalization ensures error ranges of multiple time series to be in the same magnitude order, thus improves fairness among them. The error estimation using normalized error can follow the established methodologies [12].

3.2 Wavelet Coefficient Tree

The Haar wavelet transformation is an instance of wavelet decomposition. For a given time series, neighboring elements are averaged and their differences are also computed. The averages represent the trend of the time series and the differences reveal the details. The series of average values from the last transformation apply the same process until there is one average value left. The resultant values are called wavelet coef cients. An example is given in Table 1: the original time series is {2,6,5,11}, and the resultant wavelet coef cients are {6,-2,-2,-3}.

The Haar wavelet coef cients can be arranged into a tree structure [13, 9, 14]. A coef cient tree that is constructed based on the Haar wavelet coef cients for data set

Table 1. Haar Wavelet Transformation

Iteration	Average	Detail
1	[2,6,5,11]	-
2	[4,8]	[-2, -3]
3	[6]	[-2]

Table 2. Notations

Notation	Meaning
c_i	ith coef cient
d_i	ith element in the time series
path(i,j)	all the coef cients from c_i to the c_j
leaf(i)	all of leaf nodes of node i
left_leaves(i)	all the left leaves of i
right_leaves(i)	all the right leaves of i
parent(i)	all the coef cients that are c_i's parents

{3,4,3,2,6,8,9,7,2,3,1,2,10,8,7,9} is shown in Figure 1. The coef cien ts are labeled as $c(i)$, where i is its index. The values in the time series is labeled as $d(i)$.

The value d_i in the time series can be reconstructed using all the wavelet coef cients c_j along the path from root to d_i, using the notations in Table 2:

$$d_i = \sum_{c_j \in path(0,i)} \delta_{i,j} \cdot c_j, \quad (4)$$

where

$$\delta_{i,j} = \begin{cases} -1, i \in right\_leaves(j) \; or \; j = 0. \\ 1, i \in left\_leaves(j). \\ 0, otherwise. \end{cases}$$

3.3 Magnitude-Based Zeroing

The magnitude-based *zeroing* of wavelet coef cients is as follows. Given a threshold $\alpha \geq 0$, if a wavelet coef - cient c_i satis es $|c_i| < \alpha$, then the value of c_i is set as 0, i.e. the coef cient is cut off and does not participate in the reconstruction process.

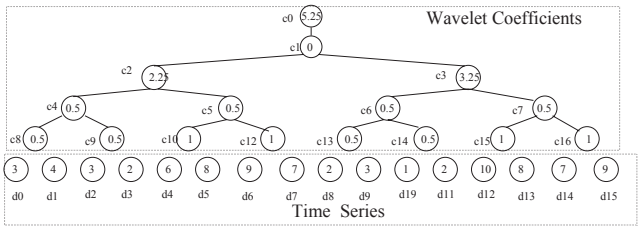

Figure 1. A sample coef cient tree

Wavelet coef cients zeroing techniques have been widely used in various applications for signal de-noising [15].

4 RACE Algorithm

This section outlines our Rate Adaptive Compression with Error bound (RACE) algorithm. We rst discuss the underlying Gradient Error Tree structure and error-based zeroing technique, and then give an overview of our RACE algorithm. We also present error additivity and patch-ability properties of RACE algorithm.

4.1 Gradient Error Tree

We adopt a tree structure which is similar to wavelet coef cient tree but with error value as nodal label. The error value associated with each node is evaluated as, if the sub-tree rooted from this node is zeroed, the maximum deviation between the reconstructed and the original values that are affected. The higher level a node is in the tree, the larger is the deviation. We term the error value as *error gradient*. A formal de n ition is as follows.

Definition *Error gradient*. The error gradient of the coef- cien t V, denoted as $G(V)$, is the maximum magnitude of the error that is incurred when the sub-tree rooted from V is zeroed, i.e.,

$$G(V) = max\{\forall i \in leaf(V) : |d_i - \sum_{c_j \in parent(V)} \delta_{i,j} \cdot c_j|\}. \quad (5)$$

Note that the error norm used in Equation 5 is the L^∞ in the absolution form. If the coef cients are normalized then the error norm becomes normalized error. We may de ne other variants of error gradients based on other error norms. For instance, error gradients based the L^2 error norm can be de ned as Equation 6.

$$G(V) = \sum_{\forall i \in leaf(V)} \sqrt{(d_i - \sum_{c_j \in parent(V)} \delta_{i,j} \cdot c_j)^2}. \quad (6)$$

In this paper, we use Equation 5 and normalized error to compress wavelet coef cients.

Figure 2 shows the gradient error tree which is corresponding to the wavelet coef cient tree in Figure 1. Note that each node i is labeled with its error gradient $G(i)$.

4.2 Error-Based Zeroing

With the development of gradient error tree structure, we now present our error based zeroing technique. A coef - cient and its children are zeroed if its error gradient is less

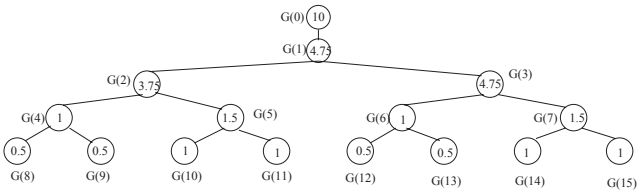

Figure 2. Error gradient

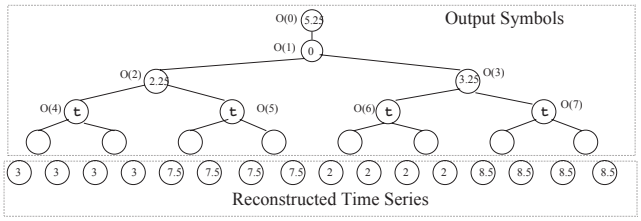

Figure 3. Threshold: 2

than the preset threshold. The zeroed sub-tree is encoded with a single tree symbol. Obviously the tree symbol fully embeds the position information of all the zeroed coef - cients. Because time series usually do not change significantly in a short duration, it is possible to further shrink these tree symbols using run-length encoding.

Now let us see some illustrations of error based zeroing. Take the wavelet coef cient tree depicted in Figure 1 for example. We prune it by using error-based zeroing with different error thresholds of 2 and 4. Figure 3 is the pruned coef cient tree using error threshold 2, where the zeroed sub-trees are labeled as "t", the numbers inside the bottom circles are derived from the zeroed coef cient tree. Note that the output symbol is denoted as $O(i)$, where i is the symbol's index in the output stream. The number of symbols needed to be encoded is 8, of which the last four "t"s can be further compressed using run length compression. Figure 4 is the pruned coef cient tree with the error bound of 4. The number of symbols to be encoded is 6.

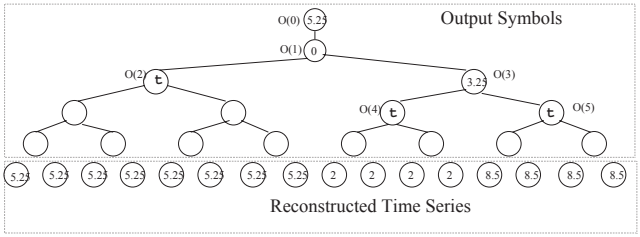

Figure 4. Threshold: 4

4.3 Overview of RACE Algorithm

RACE algorithm is built on top of error-based zeroing technique and it consists of a series of procedures which will be elaborated later in Section 5. Here we give a brief overview of it.

To compress a time series, we rst construct a wavelet coef cien t tree based on its wavelet decomposition (in this paper it is Haar Wavelet Transformation). We can generate an associated gradient error tree for the wavelet coef- cien t tree. We then compress the original time series by applying error-based zeroing to the wavelet coef cient tree with guidance from the gradient error tree. During multi-hop transmission in sensor networks, multi-pass compressions may be applied on the same time series from hop to hop if bandwidth limit is encountered. Thanks to the error bound additivity property (see Section 4.4) of our RACE algorithm, it is easy to estimate the overall error bound upon data reconstruction at the sink side.

The RACE compression algorithm is coupled with a patching mechanism that can narrow the error range and lower the maximum error whenever there is excess in bit rate. Formal description on patch-ability will be presented in Section 4.4, and details on this patching process will be presented in Section 5.4. Due to the patch-ability property of our RACE algorithm, its performance is less subject to the volatility of physical processes. This patching process has a very low overhead in encoding and matching the context information.

4.4 Properties of RACE Algorithm

In this subsection we discuss two important properties of RACE algorithm, namely, error bound additivity and patch-ability. With error bound additivity, we can estimate overall error bound for multi-pass compressions with ease. With patch-ability, RACE algorithm can smooth and minimize error range in an opportunistic manner.

Proposition 4.1 Error Bound Additivity. *Given a time series S, its wavelet coefficient tree is denoted as T. Using the gradient error tree algorithm, T is zeroed and transformed into T_1 when the threshold is t_1, T_1 is zeroed and transformed into T_2 using threshold t_2. Denote the reconstructed time series from T_2 as $S^{'}$. Then the deviation between $S^{'}$ and S is bounded by $t_1 + t_2$.*

Proof Take an arbitrary element $d_i \in S$. Denote d_i's reconstructed value in $S^{'}$ as d_i^2. The objective is to prove $|d_i - d_i^2| \leq t_1 + t_2$.

Denote d_i's reconstructed value from T_1 as d^1.

$|d_i - d_i^2| = |(d_i - d_i^1) + (d_i^1 - d_i^2)| \leq |d_i - d_i^1| + |d_i^1 - d_i^2|$.

Using the de n ition given by (5) in Section 4.1, we get

$$|d_i - d_i^1| = \begin{cases} G(j),\ if\ \exists j \in path(0,i) \\ 0,\ otherwise. \end{cases}$$

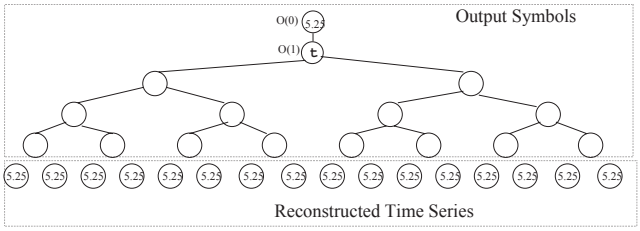

Figure 5. Stacked Thresholds: 2 and 4. Error bound: 6

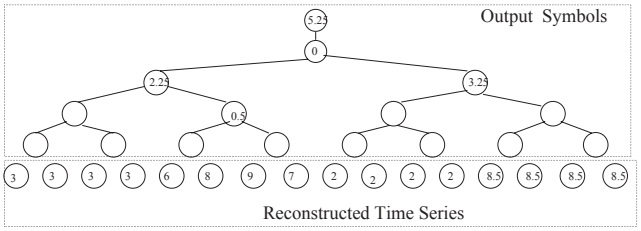

Figure 6. Patch-ability illustration

Similarly,
$|d_i^1 - d_i^2| = \begin{cases} G(l),\ if\ \exists l \in path(0,i) \\ 0,\ otherwise. \end{cases}$
Since $G(j) \leq t_1$ and $G(l) \leq t_2$, we have
$|d_i - d_i^2| \leq |d_i - d_i^1| + |d_i^1 - d_i^2| \leq G(j) + G(l) \leq t_1 + t_2$.
That proves the proposition.

Let us see an example of error bound additivity. If we apply a further compression with error bound of 4 to the wavelet coef cient tree in Figure 3, which has been compressed with error bound of 2, we can get the resultant tree as depicted in Figure 5. Comparing Figures 5 and 1, we observe that the error is bounded by the sum of the two thresholds, that is, 6.

The error bound additivity property makes it possible to estimate the error bound in a multi-hop network. Moreover, the stacked zeroing processes do not require the coef cient tree to be decoded to obtain the error bound; it operates directly on the wavelet coef cients, the computation complexity thus decreases at each relay hop.

Patching has been used to protect image quality against lossy compression in image coding eld (e.g, see [16]). We can utilize this technique to improve data quality in wireless transmission as well. We term this property of RACE algorithm as *patch-ability*.

Proposition 4.2 Patch-ability. *Given a Haar wavelet coefficient tree T and a threshold t. Denote $T_{\geq}$($T_{<}$) as the set of coefficients whose error gradients are greater than or equal to (less than) t. Obviously, $T_{<}$ is set of the coefficients zeroed by t. $T_{<}$ is sorted in descending order into $T_{<}^{'} = c_1^{<}, c_2^{<}, c_3^{<}, ...c_p^{<}$, based on the error gradients, i.e. if $1 \leq i < j \leq p$ then $G(i) \geq G(j)$. Denote first m coefficients as A_m, $A_m = c_1^{<}, c_2^{<}, c_3^{<}, ...c_m^{<} \subseteq T_{<}^{'}$. Then the error bound for the consolidated tree $T_{\geq} \bigcap A_m$ is $G(c_{m+1}^{<})$.*

Proof The proposition of patch-ability can be translated into an equivalent problem: When the n wavelet coef cients are sorted based on their error gradients, the reconstruction error using the rst m coef cients is then bounded by the $(m+1)$th coef cien t's error gradient. This is exactly how the zeroing process works. The proof is thus omitted.

An illustration of the patch-ability proposition is as follows. When the threshold is 4, the error gradients as depicted in Figure 2 is partitioned into the following sets: $T_{\geq} = c_0, c_1, c_3$ and $T_{<}^{'} = c_2, c_5, c_7, c_4, c_6, c_{10}, c_{11}, c_{14}, c_{15}, c_8, c_9, c_{12}, c_{13}$. The error gradients that correspond to the coef cients in $T_{<}^{'}$ are {3.75, 1.5, 1.5, 1, 1, 1, 1, 1, 0.5, 0.5, 0.5, 0.5}. Thus by de nition, $A_2 = c_2, c_5 = -2.25, -0.5$. Then $T_{\geq} \bigcap A_2 = c_0, c_1, c_3, c_2, c_5$. The consolidated tree is illustrated in Figure 6. The patch-ability property projects that the maximum error is bounded by $G(7) = 1.5$. Compare Figures 1 and 6, we observe that the maximum error occurs at d_{12}, so the deviation is 1.5. That con rms the error bound projection.

Because of the volatility of underlying physical processes, the errors of each compressed time series segment may differ signi can tly, which is con rmed in Section 6.1. In this occasion, the patch-ability property can utilize excess in bit rate when the compressibility of current segments is high, to compensate some signi cant coef cients for previous segments, and thus reduce overall error range.

5 Main Procedures of RACE Algorithm

This section presents the main procedures of RACE algorithm. We assume that the encoding of each symbol, whether it is a tree symbol "t" or numeric symbol of wavelet coef cient values, takes one unit space (e.g. octets).

5.1 Gradient Calculation

In order to apply error aware zeroing to the wavelet coef cient tree, we need to rst construct an associated gradient error tree. The process of gradient calculation is outlined in Procedure 1.

5.2 Rate Adaptive Compression

The process of compression is to select signi cant coef cien ts while trying to meet the bit rate constraint. The procedure is presented in pseudo code in Procedure 2. It compares the error gradients with the threshold and decides

Procedure 1 Gradient Calculation

Input: c: the coef cient to be evaluated.
$a[]$: Haar wavelet coef cients.
e: the accumulated error so far.
$size$: the number of the coef cients.
Local: left_child, right_child: left and right children of the current coef cient.
left_ret, right_ret: return values from the iterations using left and right children.
left_error, right_error: the error gradients for the left and right children.
Output: error gradient associated with the given coef - cient.

```
1: if c > size then
2:     return e
3: end if{if the end of the tree reached}
4: left_child ← c * 2 − 1
5: right_child ← c * 2
6: left_error ← e + a[c]
7: right_error ← e − a[c]
8: left_ret ← gradient(left_child, left_error)
9: right_ret ← gradient(right_child, right_error) {return the
   maximum error}
10: return max(right_ret, left_ret)
```

how the corresponding wavelet coef cients should be encoded: if the error gradient is greater than the threshold, the value of the coef cien t is put into the output stream, otherwise the coef cient and its children are zeroed and encoded as a tree symbol "t". If the resultant output volume exceeds the size constraint, the threshold is incremented and the coef cien ts are compressed iteratively until the size constraint is met.

Every time when threshold increases, it is only necessary to evaluate those coef cients that have not been zeroed during last iteration. This can save many comparison operations.

5.3 Multiple Time Series Compression

Compression of multiple time series that converge at a relay sensor is based on Procedure 2. All time series are compressed using the same threshold. If the aggregate of resultant data volume is still beyond bit rate requirement, they are trimmed again using an incremented threshold.

We may make some improvements to the above procedure. One is to explore the correlation among multiple time series and compress the principal components instead of equal treatment of all of them. Its effectiveness largely depends on the density of sensor placement and characteristics of the underlying physical processes. This can be studied in conjunction with sensor deployment and capacity planning.

Procedure 2 Rate Adaptive Compression

Input: NR: number of data entries.
$size$: maximum number of symbols that can be output.
$a[]$: Haar wavelet coef cients.
$g[]$: error gradients.
min: minimum error bound.
max: maximum error bound.
$step$: threshold increment.
Local: : i: the threshold value.
$index$: the moving cursor of the output array.
$count$: the number of symbols in the output array.
Output: $output[]$: the output symbol stream.

```
1: i ← min
2: while i ≤ max do
3:     index ← 0
4:     count ← 0
5:     for k = 1 to NR do
6:         if g[k] < i AND node k has not been evaluated then
7:             if output[index-1] ≠ "t" then
8:                 count ++
9:             end if{this sub tree can be zeroed}
10:            output[index++] ← "t"
11:            zero the sub tree rooted from k {if the last symbol is
               also "t", then run-length encoding can be used}
12:        end if
13:        if g[k] ≥ i AND node k has not been evaluated then
14:            count ++
15:            output[index++] ← a[k] {this coef cient is signi cant
               and is put into output array}
16:        end if
17:    end for
18:    if count ≤ size then
19:        break
20:    end if{the number of symbols can be compressed using
       designated space}
21:    i ← i + step
22: end while
23: return output
```

5.4 Error Smoothing and Minimization via Patching

The adaptive compression procedure may produce erratic error bounds over time. Relying on the property of patch-ability of RACE algorithm, we can use Procedure 3 to approximate sub-optimal error bound and narrow the error ranges. We assume that patching process needs at most three symbols to encode a zeroed coef cient: one to encode which tree the coef cient belongs to, the second to encode the coef cient's position within the tree, and the last one encodes the coef cient' s value.

The rationale behind the procedure is as follows. The compressibility of each time series segment varies over time, some of which can t the size constraint with lower errors than others. In many applications, it is desirable to improve the overall error bound for time series instead of

that for individual segments. So it is a natural idea to "steal" some bit rate space from highly compressible segments to compensate those ones that are less compressible and need more space to achieve the same error bound.

Procedure 3 Error Smoothing and Minimization via Patching

Input: NR: number of data entries.
$size$: maximum number of symbols that can be output.
$a[]$: Haar wavelet coef cients.
$g[]$: error gradients of each node.
avg: moving average of the previous maximum normalized error.
max: maximum error bound.
$r[]$: a sorted array of coef cient structures by the order of their corresponding error gradients.
Local: : i: the threshold value.
Output: $output[]$: the output symbol array.
$patch[]$: previously zeroed coef cients and their context.

```
1:  i ← avg
2:  while i ≤ max do
3:     index ← 0
4:     count ← 0
5:     for k = 1 to NR do
6:        if g[k] < i AND node k has not been evaluated then
7:           if output[index-1] ≠ "t" then
8:              count ++
9:           end if
10:          output[index++] ← "t"
11:          zero the sub tree rooted from k { if the last symbol is
             also "t", then run-length encoding can be used}
12:       end if
13:       if g[k] ≥ i AND node k has not been evaluated then
14:          count ++
15:          output[index++] ← a[k]
16:       end if
17:    end for
18:    if count ≤ size then
19:       break
20:    end if{the number of symbols can be compressed using
       designated space}
21:    i ← i + step;
22: end while
23: free_slots ← (size − count)/3
24: for j = 1 to free_slots do
25:    patch[j] ← r[j]
26:    remove r[j]
27: end for
28: return output + patch
```

6 Numerical Evaluation

We have evaluated the performance of RACE compression algorithm using real world data which was obtained from the Tropical Atmosphere Ocean project (TAO, `http://www.pmel.noaa.gov/tao/`). The data sets used in the evaluation date from January, 2001 to January, 2003. It consists of air temperature samples from a single mooring, subsurface temperature at different depths from a single mooring, and air temperature from multiple moorings. Without loss of generality, we assume that the sensors accrue every 512 samples to make a *segment*, which is taken as one unit for transmission. Please note that varying the number of samples in a segment does not affect the conclusions we draw in the following discussion.

6.1 Rate Adaptivity

In this evaluation we use a set of air temperature samples from a single mooring. Some of the statistical characteristics of the data set are summarized in Table 3. In order to evaluate the performance of rate adaptivity, we used three levels of size constraints for the compression of each segment: 256 symbols, 128 symbols, and 64 symbols. The corresponding compression ratios are 2:1, 4:1, and 8:1, respectively.

Category	Value
Number of samples	32,768
Max	29.94
Min	-9.99
Mean	27.57
Variance	6.33

Table 3. Air Temperature Data Characteristics

The normalized errors with respect to different size constraints are depicted in Figure 7. While RACE can compress the data into different sizes, we observe that data quality deteriorates more signi cantly when the available space size decreases. The higher compression ratio, the higher magnitude in maximum normalized error, which means lower quality of data and consequently more complicated in error estimation. More speci cally, the maximum normalized error for the compression ratio 2:1 (256 symbols) ranges between [1.0%, 3.4%]; while the most aggressive compression ratio 8:1 (64 symbols) has a error range between [2.0%, 19.1%].

We also observe that the normalized error follows similar uctuation patterns with different magnitudes under all compression ratios. This observation of erratic error ranges justi es our belief that some form of error smoothing is necessary.

6.2 Error Range Smoothing and Minimization

To evaluate RACE's ability of narrowing and smoothing the error range, we use the same data set as in Section 6.1. The results are plotted in Figure 8. We observe that

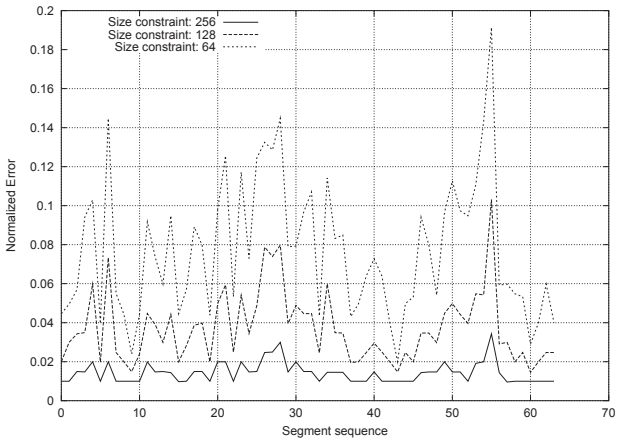

Figure 7. Maximum Normalized Errors

the error ranges are narrower and the maximum errors become smaller. In the cases of compression ratios 2:1 (256 symbols) and 8:1 (64 symbols), the error ranges become [0.9%, 1.7%] and [4.2% , 10.6%], respectively. Compared to the results in Figure 7, error ranges are greatly reduced by RACE's patching process.

For each compression ratio, we observe that the curve of error shows less uctuation, which justi es RACE's capability of smoothing error ranges.

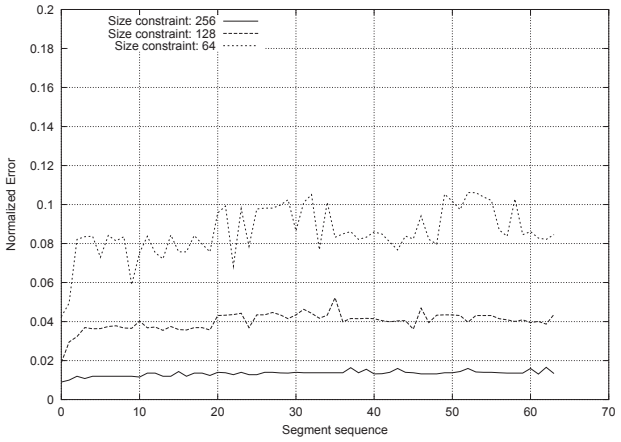

Figure 8. Smoothed Maximum Normalized Errors

6.3 Preservation of Statistical Interpretation

We evaluate RACE's capability of preserving statistical interpretations under different compression ratios. We use cross correlation as an instance to evaluate how the compression can preserve the multivariate correlationship. Cross correlation is the temporal correlationship between two variables. For example, cross correlation can be used to analyze the relation between the El Nino and La Nina phenomena. The cross correlation between two samples from variable x and y is de ned as $\frac{\sum(x_i-\widetilde{x})*(y_{i-d}-\widetilde{y})}{\sqrt{\sum(x_i-\widetilde{x})^2}*\sqrt{(y_{i-d}-\widetilde{y})^2}}$, where d is the temporal delay between x and y.

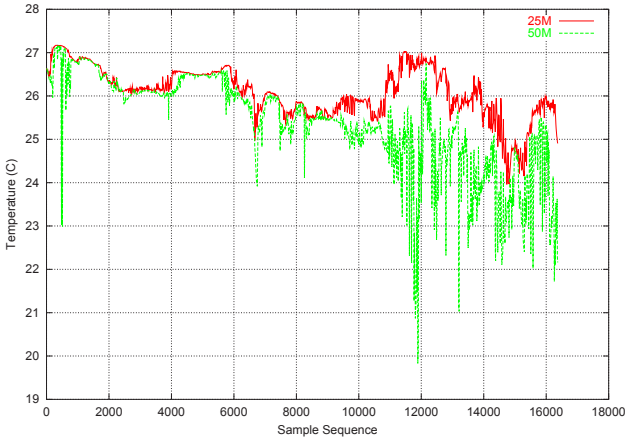

Figure 9. Subsurface Temperature at depths 25M and 50M

The data used in this evaluation are sea subsurface temperatures from depths 25M and 50M, which are plotted in Figure 9. In this evaluation, we use a wider compression ratio range from 4, 8, 16, 64, up to 256. Figure 10 plots the cross correlation variations with respect to different compression ratios. Because the correlation-curves of ratios 4 and 8 almost overlap with the one of uncompressed data, we only present the results for the higher compression ratios. We observe that the correlation between two time series are preserved with different levels of delity: the lower the compression ratio, the higher the d elity. We also observe that, even for very high compression ratio (256:1), RACE can still preserve signi can t details of the trendency. This observation con rms that our algorithm and choice of error norm can preserve statistical interpretation of time series very well.

7 Conclusion and Future Work

Continuous monitoring sensor networks have wide potential applications. We characterize this class of sensor networks as multivariate monitoring, multi-terminal data fusion, and multi-hop delivery. To address major challenges

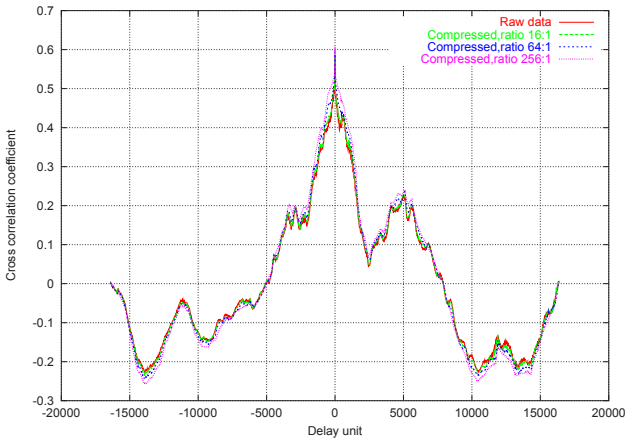

Figure 10. Cross Correlation between Subsurface Temperature at depths of 25M and 50M

of limited bandwidth and battery source in this class of networks, we propose a Rate Adaptive Compression with Error bound (RACE) algorithm that attempts to meet both network capacity and error bound requirements. Since RACE algorithm can always generate CBR or LBR output streams, it is bene cial for better transmission scheduling in sensor networks, which can reduce transmission energy consumption. When network capacity limit is encountered in multihop relaying, RACE can trade off data quality smartly for be tting bit rate. Because of the property of error bound additivity, it is easy to estimate overall error bound upon data reconstruction at the sink side. In order to exploit different compressibility of time series segments over the time, RACE algorithm adopts an opportunistic patching mechanism which can narrow and smooth error range. We have use real world data to evaluate RACE algorithm, and the results have veri ed that RACE can achieve our design goals.

One of our future work is to investigate how to incorporate spatial and temporal correlation of multiple time series into our compression algorithm.

References

[1] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," in *ACM MOBICOM '00*, 2000, pp. 56–67.

[2] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang, "A two-tier data dissemination model for large-scale wireless sensor networks," in *ACM MOBICOM '02*, Septermber 2002.

[3] D. Ganesan, D. Estrin, and J. Heidemann, "Dimensions: Why do we need a new data handling architecture for sensor networks?," in *Proceedings of the ACM Workshop on Hot Topics in Networks*, October 2002.

[4] G. J. Pottie and W. J. Kaiser, "Wireless Integrated Network Sensors," *Communications of the ACM*, vol. 43, no. 5, pp. 51–58, 2000.

[5] L. M. Feeney and M. Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment," in *IEEE INFOCOM*, 2001.

[6] I. Lazaridis and S. Mehrotra, "Capturing sensor-generated time series with quality guarantees," in *International Conference on Data Engineering (ICDE)*, March 2003.

[7] C. Olston and J. Widom, "Best-effort cache synchronization with source cooperation," in *ACM SIGMOD '02, Madison, Wisconsin*, May 2002.

[8] J.M. Shapiro, "Embedded image coding using zerotrees of wavelet coef cients," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3445–3462, 1993.

[9] M. Garofalakis and P. B. Gibbons, "Wavelet synopses with error guarantee," in *ACM SIGMOD*, June 2002.

[10] I. T. Jolliffe, *Principal Component Analysis*, Springer-Verlag, 1986.

[11] D. R. Hankerson, G. A. Harris, and P. D. Johnson, *Introduction to Information Theory and Data Compression, Second Edition*, Chapman and Hall, 2003.

[12] J. R. Taylor, *An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements*, University Science Books, 1997.

[13] K. Chakrabarti, M. Garofalakis, R. Rastogi, and K. Shim, "Approximate query processing using wavelets," *VLDB Journal: Very Large Data Bases*, vol. 10, no. 2–3, pp. 199–223, 2001.

[14] Y. Matias, J. S. Vitter, and M. Wang, "Wavelet-based histograms for selectivity estimation," in *Proc. of ACM SIGMOD '98*, 1998, pp. 448–459.

[15] C. S. Burrus, R. A. Gopinath, and H. Guo, *Introduction to Wavelets and Wavelets Transforms*, Prentice Hall, 1997.

[16] P. Greg Sherwood and Kenneth Zeger, "Progressive image coding on noisy channels," in *Designs, Codes and Cryptography*, 1997, pp. 72–81.