An Efficient General-Purpose Mechanism for Data Gathering with Accuracy Requirement in Wireless Sensor Networks

Ryo Sugihara* Andrew A. Chien*†

Abstract

A generic objective for a sensor network application is the gathering of data from a field of sensors. Because energy is often scarce in sensor networks, many techniques have been proposed to reduce data size within the network. These techniques either ignore the accuracy of the resulting data, or more often, provide no means for applications to control the resulting accuracy. However in many cases, applications have a quantitative requirement for sensor data accuracy, and the underlying system should meet that efficiently. In this paper, we describe a distributed algorithm that approximates and gathers data in an energy-efficient manner and strictly satisfies an application-provided accuracy requirement. This approximation is based on a hybrid data representation based on linear regression. A distinguishing feature of the proposed algorithm is that it absolutely does not require any models on statistical properties of data and noise, and needs only few general assumptions on sensor node topology. This feature enables the algorithm to serve as a general-purpose mechanism that can be widely used in many scenarios for data gathering-type applications. Simulation experiments with data traces from real environmental data show that it leverages the accuracy requirement to significantly reduce energy consumption.

1 Introduction

Sensor networks are a rapidly emerging research field in the academic and industrial world because of their vast range of potential application. One of the most common forms of sensor network applications is data gathering in which some data is sampled at a set of spatially distributed sensors and then collected together; perhaps at an uplink out of the sensor network. Examples of data gathering sensor network applications include habitat monitoring[13], environmental monitoring[12], target localization[11], structural monitoring[19], and a countersniper system[15]. Sensor management[21] is also one of the data gathering applications in a broader sense.

In general, applications desire fine data resolution (temporal and spatial) and high accuracy, which require high data volume and high bandwidth, to enable their higher level activities both within and without the network. On the other hand, sensor network systems that hosts one or several applications should conserve scarce energy and bandwidth. One

*University of California San Diego
†Intel Research

promising approach that resolves this conflict and supports both of these goals is to construct a compact data representation, enabling efficient manipulation and communication within the sensor networks. Techniques such as summarization, aggregation, approximation, statistical prediction, and their combinations all fall within this general approach. These methods each gain efficiency by representing only a portion of sensors and/or reducing the accuracy of sensor data passed on.

Specifically, we are interested in the construction of a custom data representation which meets an application's accuracy goals. We also pursue efficiency to meet the sensor network systems' goal as well. Generally, there is a trade-off relationship between energy cost and accuracy, as some papers point out[3, 17]. However, a significant difference of our approach is that we guarantee to satisfy a quantitative accuracy requirement, which is explicitly given by the application and the system is strictly required to meet.

In this paper, we first formulate the problem of data gathering with accuracy requirement. Then we propose a distributed algorithm for the problem which computes and uses custom data representations to achieve a specified accuracy requirement and saves energy by optimizing the representation to reduce communication cost. Our approach is a hybrid one, employing linear regression to compress data which is spatially close and thereby potentially correlated, and using a separate explicit representation for data as needed to satisfy the accuracy requirements. A distinguishing feature of the algorithm compared to others[5, 18] is that it absolutely does not require any prior knowledge or assumed models about the data's statistical properties. We believe this generality enables the algorithm to serve as a general-purpose mechanism that can be widely used in many scenarios for data gathering-type applications.

We evaluate the performance of our distributed algorithm, comparing it to a naive data gathering method, and also to a method analogous to "Distributed Regression"[6]. These experiments show that the proposed algorithm can exploit the cost-accuracy trade-off effectively, significantly reducing the size of representation, as well as the amount of communication in the sensor network.

Specific contributions of the paper include:

- formulation of an data gathering problem for sensor networks as a multiobjective optimization problem where application-specified accuracy requirement is a constraint,
- a distributed algorithm which computes a reasonable approximate solution to the problem using a custom hybrid data representation, without requiring any assumed models on statistical properties of data, and
- an evaluation of the proposed distributed algorithm showing that it derives a custom data representation while effectively exploiting the accuracy requirement for efficiency.

The remainder of the paper is organized as follows. In Section II, we formally define the problem. In Section III, we present our distributed algorithm for computing custom data representations. In Section IV, we evaluate the performance of the algorithm by simulation on the real environmental data. Some of the related work is presented in Section V. Section VI concludes the paper, pointing out some possible directions for future work.

2 Problem Statement

In this section we define the problem of data gathering with accuracy requirement more concretely. We first discuss what kind of accuracy requirements are suitable for sensor network applications, and then define data gathering as a multiobjective optimization problem.

2.1 Definition of Accuracy Requirement

2.1.1 What is Accuracy?

We define two types of accuracy: *measurement accuracy* and *system accuracy*. Measurement accuracy is the accuracy realized by measured data at each sensor. Each sampled data is merely an estimate of the reality and it usually contains error due to noise, sensing capability, faulty sensors, and so on. Measurement accuracy is out of the scope of this paper, since we cannot improve it without making further assumptions on the statistical properties of data, and we choose not to do that. In fact, a number of previous work[3, 18] make such assumptions to improve measurement accuracy; typically by adding more samples while assuming noise is independent and identically distributed. However, these assumptions are often difficult to validate and can be a source of another inaccuracy when they are incorrect.

On the other hand, system accuracy is related to the post-processes after measurement. Approximation and compression for the sake of efficiency can affect the system accuracy by introducing errors. It may be more understandable to think it as "degree of fidelity to sensor readings". In other words, we can achieve perfect system accuracy if we collect all measured data from sensors, but only with sacrificing the efficiency. In this paper, we will only focus on the system accuracy and refer to it simply as the accuracy.

2.1.2 Accuracy Metrics

MSE (mean square error) is one of the most frequently used accuracy metrics in the literature of modeling, approximation, and compression in sensor networks, due to its simplicity and theoretical tractability. However, one problem with MSE is that it is not very sensitive to outliers. Suppose a fire detection application that uses temperature sensors. We can imagine two extreme cases having the same MSE. The first case is that the gathered data are error-free[1] for all sensor nodes except the one that contains a huge error and a fire is not detected as a result. The other case is that the gathered data are slightly deviated from the sensor readings for all nodes, and thus the fire is successfully detected. Even though they have the same accuracy in terms of MSE, the quality (or practical significance) is totally different.

As an alternative metric, we propose maximum absolute deviation (MAD). Given measured data $(v_1, ..., v_n)$ and its approximation $(\hat{v_1}, ..., \hat{v_n})$, MAD is defined as $\max_i\{|\hat{v_i} - v_i|\}$. Different from MSE, MAD is sensitive to outliers. As seen in the above example of fire detection application, sensor network applications are often interested in deviating data points[2]. Sometimes these deviating data points may reflect the events of applications' interest, or

[1] With regard to *system accuracy* defined in 2.1. i.e. exactly same as the measured data.

[2] We will use the term "deviating data points" and avoid using "outliers". It is because outliers imply as if they are just a nuisance, whereas they may reflect interesting phenomena and should not be neglected.

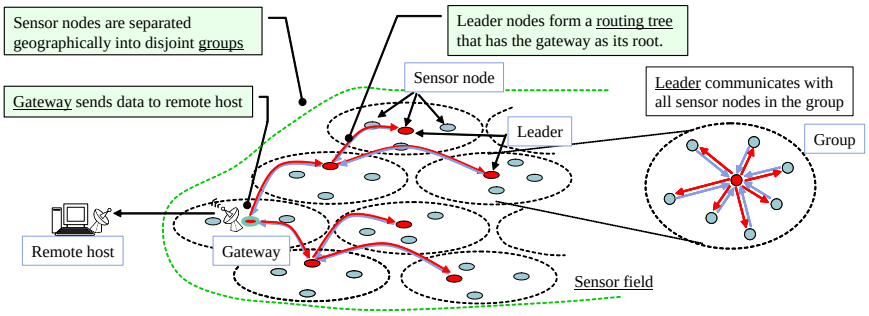

Figure 1: One common configuration of sensor network

they may be due to faulty sensors, which is also something applications need to know. These are the reasons why we believe MSE is not always a good metric of accuracy, specifically in the sensor networks context, and why we choose MAD instead.

2.2 Problem Definition

The ultimate goal of the problem is to construct and retrieve a data representation $\mathcal{R}$ that yields $\widehat{v}_i$, an approximation of v_i (measured value of i-th sensor) for all i. Here we first make some assumptions to define the problem space more specifically, and formulate it as a multiobjective optimization problem with a constraint and objective functions.

Assumptions

Figure 1 shows the configuration of the sensor network assumed in the problem. Sensor nodes are geographically separated into disjoint groups, each of which has a leader node. To define the group, we can use any topology control methods such as [1, 20][3]. One of the group leaders acts as a gateway to the remote host, which is outside of the sensor field. The gateway node communicates with the remote host via a wireless link.

Each node is static and knows its location in the global coordinate system. The leader of each group knows the locations of all the member nodes, and the remote host knows the locations of all nodes. Network topology is the same as "Cluster Tree Network" in ZigBee[22], where every sensor in the group has a link with its leader and the leaders consist a multi-hop routing tree rooted at the gateway. Each leader communicates with its parent and children in the routing tree. Non-leader nodes communicate only with their own leader.

All sensor nodes are battery-driven and communication is a dominant factor of energy consumption. The required energy for transmitting data is proportional to the size of data.

Constraints

The constraint is to satisfy the given accuracy requirement. As we discussed earlier, an application specifies the accuracy requirement in the form of maximum tolerable MAD ε. By using the notations introduced above, the constraint is described as $\forall i,\ |\widehat{v}_i - v_i| \leq \varepsilon$.

[3]There is no restriction on the size of each group, but it should be larger than three for the proposed algorithm to run more efficiently.

Objective Functions

The objective function of the problem is energy efficiency. We define two different metrics of efficiency: "size of data representation" and "amount of internal communication". Both of them are to be minimized. Size of data representation $|\mathcal{R}|$ is nearly proportional to the energy consumed at the gateway node. Amount of internal communication is the total communication within the network while constructing $\mathcal{R}$. It is related to the total energy consumed at all nodes. Note that these two metrics are not independent, but optimizing one does not necessarily optimize the other.

3 Data Gathering with Accuracy Requirement

In the previous section, we have formulated the problem of data gathering with accuracy requirement. However, it is an NP-hard problem even in a simpler version in which the objective is to solely minimize the size of data representation[7]. It motivates us to consider an approximation algorithm that yields reasonably good suboptimal solutions. In this section, we describe one of such algorithms that is distributed and uses a custom hybrid data representation based on linear regression.

3.1 Approach and Outline of the Algorithm

Figure 2 concisely shows the idea of our approach. As a medium of compacting the size of data representation, we use a plane to approximate multiple data points located in two dimensional space. The data points whose approximation errors fall within the accuracy requirement are represented by this plane. Other "deviating" data points are explicitly represented. The resulting data representation is a hybrid of those two.

A motivation for using planes is to capture spatially correlated structure, which is often the case, in the simplest possible way. However, more importantly, spatial correlation is not mandatory for the algorithm to work correctly. Even though the data points have no spatial correlation, the accuracy requirement is still satisfied. Similarly, sensor grouping and routing tree do not affect the correctness, either. All of them only affect the efficiency, depending on the underlying statistical property of the field which we neither know nor assume.

The proposed distributed algorithm constructs this hybrid representation in an efficient and scalable way. The outline is as follows. A regression plane is calculated at each group to approximate the data points in the group. As the coefficients of the planes are forwarded from the leaf groups toward the root, some of the planes may be combined together to a single plane, to make more compact representation. The resulting coefficients are sent back to each group to collect deviating data points, so that the representation can satisfy the application-specified accuracy requirement.

In the rest of the section, first we briefly describe linear regression, and then the detail of the algorithm follows.

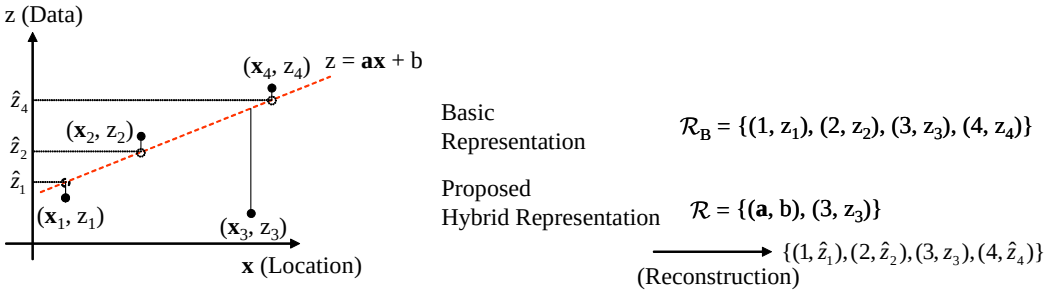

Figure 2: Illustration of our approach. In the proposed hybrid representation, data points $(\mathbf{x}_i, z_i)$ are approximated by $(\mathbf{x}_i, \hat{z}_i)$ except $(\mathbf{x}_3, z_3)$, which is a deviating data point and represented explicitly.

3.2 Linear Regression

Linear regression is a common statistical technique to model distributed data points by a simple linear equation expressed as $f(\mathbf{x}) = \mathbf{a}^T\mathbf{x}'$ where $\mathbf{x}' = [\mathbf{x}^T\ 1]^T$. Given the data set $\{\mathbf{x}_i, v_i\}$ $(1 \leq i \leq n)$, where $\mathbf{x}_i$ is the location and v_i is the measured value of sensor s_i, least squares estimator (LSE)[9] of the coefficient vector $\mathbf{a}$ is $\hat{\mathbf{a}} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{v}$ where $\mathbf{v} = [v_1\ ...\ v_n]^T$ and $\mathbf{H} = [\mathbf{x}'_1|...|\mathbf{x}'_n]^T$. For the sake of brevity, we assume two dimensional space and $\mathbf{x}'_i = [x_i\ y_i\ 1]^T$. In this case, $\hat{\mathbf{a}}$ is calculated by using the following matrix and vector

$$\mathbf{H}^T\mathbf{H} = \begin{bmatrix} S_{xx} & S_{xy} & S_x \\ S_{yx} & S_{yy} & S_y \\ S_x & S_y & n \end{bmatrix}, \quad \mathbf{H}^T\mathbf{v} = [S_{xv}\ S_{yv}\ S_v]^T \tag{1}$$

where $S_x = \sum_i^n x_i$, $S_{xy} = \sum_i^n x_i y_i$ and similarly for others.

One thing worth mentioning here is that all of the elements in these matrix and vector can be updated in an incremental manner by adding/removing data points, since they are simple linear and quadratic summations. Since $\mathbf{H}^T\mathbf{H}$ is symmetric, $\hat{\mathbf{a}}$ is determined by nine distinct values (six for $\mathbf{H}^T\mathbf{H}$ and three for $\mathbf{H}^T\mathbf{v}$) in the above case, or generally $\frac{d(d+1)}{2} + d$ values when $|\mathbf{x}'| = d$. We refer to these set of values as "SumSet". By this fact, to obtain a new regression plane by combining multiple ones, we only need the SumSet of each of them, instead of each individual data point. We extensively use SumSet for communications between groups during plane combination process, which we explain next.

3.3 Distributed Algorithm

The algorithm consists of three steps, all of which take place at the leader node of each group. The first step, "linear regression with filtering", calculates a plane that approximates the data points in each group. It is executed individually at each group and the leaf groups send the resulting planes to their parents. The second step, "combine planes", attempts to combine some of the planes into one to make the representation more compact. It is executed only at non-leaf groups upon receiving the results from all of the child groups. At the end of this step, the results are sent upwards and the parent group executes the same step. Those results are also sent downwards to trigger child groups to execute the third step, "collect

deviating data points". In the third step, each group finds deviating data points and sends them upwards to the root.

Here we explain each of these three steps in detail.

Linear Regression with Filtering

Locally at each group, the group leader collects all the data from its members and calculates a plane that approximates them. Along with the calculation, deviating data points are filtered out and reserved for separate, explicit representation.

The procedure is described by the following pseudocode:

LINEAR-REGRESSION-WITH-FILTERING

```
1  K ← {s_1 .. s_l}                          ▷ Sensors in the same group
2  repeat
3      f_K ← REGRESSION(K)                   ▷ Calculate regression plane
4      m ← arg max_i |f_K(x_i) − v_i|        ▷ Find maximum deviating point
5      d_m ← |f_K(x_m) − v_m|
6      if d_m > ε do                         ▷ ε: Application-specified error bound
7          K ← K\{s_m}                       ▷ Filter out s_m
8    until d_m ≤ ε
```

The procedure starts with linear regression for all data points (line 1). On each iteration, the point which deviates the most from the plane is identified (line 4). If the deviation is more than ε (line 6), the error bound specified by the application, the point is filtered out (line 7) and the regression plane is recalculated for the rest of the data points. Iteration finishes when all data points (except the ones already filtered out) deviate less than ε (line 8). In the end, all data points except the excluded ones are approximated by the plane within error ε.

Note that this filtering process is redone later in the "collect deviating data points" procedure, and so there is no problem if it is omitted here. Nevertheless, we include it here in order to let the resulting plane reflect the trend of the majority so that it can get more chance to be combined with adjacent ones in the "combine planes" procedure explained next.

Combine Planes

At each non-leaf group, upon receiving the planes (in the form of SumSet) from all children, the group leader attempts to combine multiple planes into a single one to make the resulting representation more compact. Figure 3 shows the idea.

The procedure is described by the following pseudocode:

COMBINE-PLANES($G_1 \ldots G_N$)

```
1  S ← {G_1 .. G_N}                          ▷ Children groups and myself
2  while S ≠ φ do
3      T ← S                                 ▷ T: Next subset of planes to be combined
4      repeat
5          f_T ← REGRESSION(T)
6          x_T ← CENTROID(T)                 ▷ Centroid of group leaders in T
7          n ← arg max_i |f_i(x_T) − f_T(x_T)| ▷ f_i: Plane for group G_i
8          d_n ← |f_n(x_T) − f_T(x_T)|
9          if d_n > ε do
10             T ← T\{G_n}                   ▷ Eliminate G_n
11     until d_n ≤ ε
12     S ← S\T                               ▷ Output T, continue for the remaining groups
```

The procedure repeatedly searches the subset of groups whose planes are "similar" and combines those planes. First we calculate a regression plane by combining all planes from child groups (line 5)[4]. Then one of the planes that deviates the most from the combined one is eliminated in an iterative manner. In order to evaluate the distance between planes in a computationally efficient yet practically sufficient way, we use the difference of values at the centroid of group leaders (line 6, 7). For the plane that deviates the most, if the difference is more than ε, it is eliminated from the subset T (line 10)[5].

After finishing this procedure, SumSet for the combined plane that contains the current group is sent upward to its parent for further attempts of combination. All the other combined planes are finalized at this point, and their coefficients are sent both upwards to the root (to be a part of the representation) and downwards (to collect deviating data points).

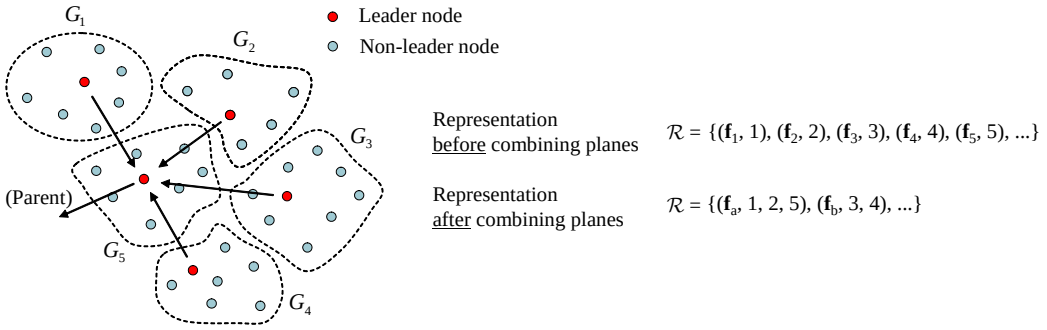

Figure 3: Combine planes. Each $\mathbf{f}_i$ is a representation of the plane, namely a list of coefficients, for group G_i. $\mathbf{f}_a$ and $\mathbf{f}_b$ are the ones for combined planes.

[4]This new plane can be calculated efficiently using the SumSet uploaded from each group.

[5]The elimination threshold does not need to be ε, but it is rather for simplicity. The optimal threshold value, or the optimal combination of planes, depend on data's statistical properties and group configuration, and finding them requires an exhaustive search. Note that the accuracy requirement is satisfied even for this presumably suboptimal criteria, because of "collect deviating data points" procedure.

Collect Deviating Data Points

At each group, after the plane for the group is finalized, the deviating data points that cannot be approximated by the plane within the error bound are collected. This procedure is triggered by receiving the coefficients, and consists of a simple iteration shown below:

COLLECT-DEVIATING-DATA-POINTS

1 $D \leftarrow \phi$ ▷ D: Set of deviating data points in the group
2 **for** $i \leftarrow 1$ **to** l **do** ▷ For each data point
3 $d_i \leftarrow |f'_K(\mathbf{x}_i) - v_i|$ ▷ Re-calculate the deviation with the new plane f'_K
4 **if** $d_i > \varepsilon$ **do**
5 $D \leftarrow D \cup \{v_i\}$ ▷ Add v_i to the set of deviating data points

After this procedure, the set of deviating data points D is sent upwards to the root.

3.4 Further Optimization

The proposed algorithm uses messages passed back and force between group leaders, trying to combine planes for more compact representation. However, it may not be efficient when the accuracy requirement is extremely tight that most of the data points cannot be approximated by the planes. To circumvent this problem, one idea is to give up fitting a plane when it represents only a small number of data points in the group[6]. For such "poor-fit" groups, all the data points are sent up to the root, just like treating all as deviating data points.

In this optimized version, "linear regression with filtering" procedure is followed by the step of deciding if the group is poor-fit or not. "Combine planes" procedure is skipped for poor-fit groups. By these modifications, we can remove exchange of SumSet and coefficients for poor-fit groups. It is expected to reduce the total amount of internal communication in case of tight accuracy requirement, without adversely affecting the size of data representation.

4 Evaluation

To evaluate the efficiency of the proposed algorithm, namely how well it exploits the cost-accuracy trade-off in realistic environments, and also to show it satisfies an accuracy requirement, we perform simulation experiments using actual environmental data traces.

4.1 Simulation Details

We use the data excerpted from "SMEX02 SSM/I Brightness Temperature Data, Iowa", publicly available from [16]. This data set provides brightness temperature data obtained by the Special Sensor Microwave/Imagery (SSM/I) satellite. For simulation experiments, we assume a sensor is located at each measured location in the data set. 155 sensor nodes are geographically divided into 16 groups and the node closest to the gravitational center of the

[6] In the experiment, we arbitrarily chose four as the threshold. Note that it should be at least three since any three points in three-dimensional space determine a plane.

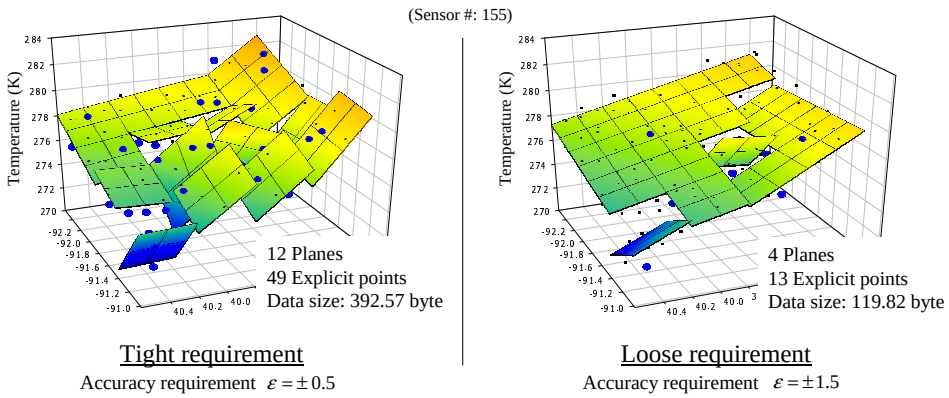

Figure 4: Visualization of proposed data representation. Small dots are measured data and approximated by the planes. Large dots are deviating data points and represented explicitly. The representation is more complex in the tight accuracy requirement (left) than in the loose one (right).

nodes in the group is chosen as the leader. Each group contains 7-10 sensor nodes. We also determine the multihop routing tree and its root node as the gateway to the remote host.

The performance of the proposed algorithm is compared to the naive method, in which all sensors send their own data to the root along the routing tree without any sophisticated data processing. We also evaluate "Regression only (w/o plane combination)" method that employs only linear regression and neither combines planes nor collects deviated data points. Since the accuracy is not controllable in this method, we obtained the resulting accuracy from the produced representation. We chose this method as one of the references since it is equivalent to what "Distributed Regression"[6] yields, which will be discussed later.

The metrics of the efficiency are the size of data representation and the amount of internal communication, as discussed in section 2.2. On evaluating the size of data representation, each of the integer values contained in the representation, which include ID of groups and sensors, are assumed to be 2 bytes. Other non-integer values are assumed to be 4 bytes for each. As for the amount of internal communication, we used "byte.length" as the unit, where 1 byte.length is equivalent to transmitting 1 byte to 1 unit length.

4.2 Results and Discussion

Figure 4 is a visualization of the resulting representation for two different accuracy requirements. When the requirement is tight (Fig.4, left), there are more planes used and larger number of explicitly represented data points, compared to the case of loose requirement (Fig.4, right). All approximated data points successfully satisfy the specified accuracy requirement.

Figure 5 shows the size of data representation when varying accuracy requirements. The size is constant in the naive method regardless of varying accuracy requirements. The proposed method is comparable to that when the absolute accuracy is needed ($\varepsilon = 0$), but it exhibits a significant reduction of size as the accuracy requirement gets looser. "Regression only (w/o plane combination)" appears as a point, which is located on $\varepsilon = 1.70$, since the

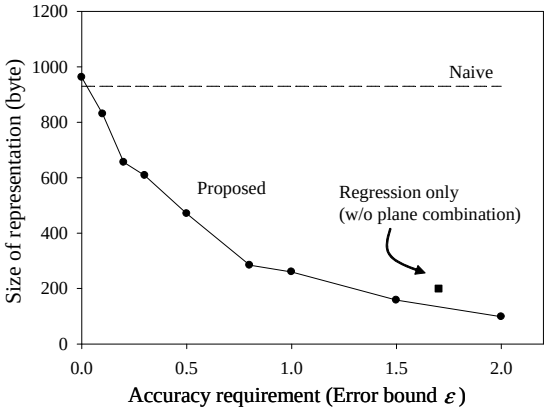

Figure 5: Size of data representation for varying accuracy requirements.

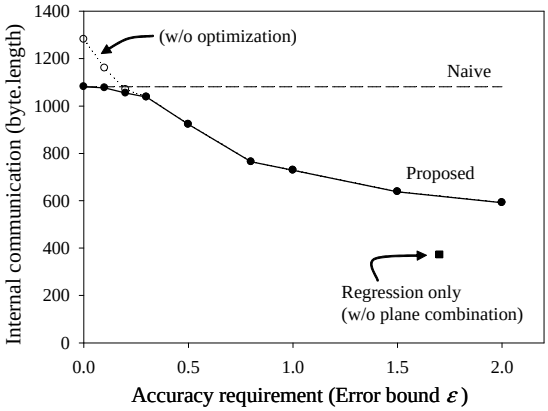

Figure 6: Amount of internal communication for varying accuracy requirements.

accuracy is not explicitly controllable in this method. For the data traces used for this simulation, the size of data representation by the regression-only method is larger than the one by the proposed algorithm with the equivalent accuracy requirement. This difference is mainly due to the redundancy among the regression planes since the regression-only method does not combine planes even if fewer number of planes could approximate the data points equally well.

Figure 6 shows the amount of internal communication. The naive method is shown as a flat line again by the same reason as above. Similarly as the size of data representation, the proposed method exhibited a reduction for looser requirements. However, it is very close to the naive method when the accuracy requirement is tight. In fact, as shown in the figure, it is worse than the naive method if the optimization described in section 3.4 are not applied. "Regression only (w/o plane combination)" is shown as a point again, but required only a small amount of internal communication, since it only needs one-pass to construct a representation, whereas the proposed method requires two-pass.

Note that the results shown here are only valid for this particular data set and particular sensor configurations. Since we do not make any assumptions on statistical properties of data and/or noise, and neither we do not have any strong restrictions about the sensor grouping and routing topology, different data set and different sensor configuration may yield significantly different results. For example, if the simulation were done on a totally random data set with no spatial correlation, the proposed algorithm probably cannot exploit the accuracy requirements effectively for efficiency, since we implicitly expect data to have some spatial correlation when we fit planes. However, such data cannot be gathered efficiently in any ways, and we claim the proposed algorithm to achieve reasonably good performance in any common cases in sensor network applications.

5 Related Work

The idea of using linear regression to generate efficient data representations has been introduced to sensor networks by Guestrin et al.[6], who propose a distributed algorithm of spatial data modeling. Their approach views accuracy as a side effect of how well the linear regression works, and in effect they cannot explicitly control accuracy. In contrast, in our problem definition, an application-specified accuracy requirement must be met in any cases, and the proposed algorithm realizes this. Some of the earliest work in data fusion protocols[8, 10] are also different from our work in this capability of explicit control of accuracy, though they share the idea with our work to conserve and/or balance energy consumption by gathering data to fusion centers where data compression possibly takes place.

There is a variety of previous work addressing approximation and summarization of data in sensor networks. For example, Considine et al.[4] propose an approximated aggregation algorithm in the presence of node/link failure. Shrivastava et al.[14] propose an efficient technique which obtains approximate quantiles such as the median, which usually require the collection of all data for the exact value. They also did a theoretical analysis and gave an upper bound to the approximation error. However, many of the studies focus on approximating aggregated data via operators such as SUM, AVG, and MIN/MAX. In contrast, our work focuses on controlled accuracy in representing the entire data set.

Adaptive sampling in the context of field estimation is also analogous to our work. Backcasting[18] is one of the algorithms that first collects small subset of data to get rough estimate of the field and then refines it by activating additional sensor nodes to achieve a target accuracy. However, it assumes zero-mean Gaussian noise and focuses on a limited class of field that contains a certain type of boundaries inside. Our work is more widely applicable since we have no assumption on noise and no limitation on the characteristics of the field. Fidelity driven sampling[2] is also similar to our work in requiring no prior knowledge about data and achieving explicitly specified accuracy goal, but they assume mobile sensor nodes that they can maneuver for the purpose of obtaining higher resolution from a certain area in the field.

Some of approximation and summarization schemes conserve energy by estimating the sensor values and thereby reduce the need for communication. In the BBQ system, Deshpande et al.[5] propose a centralized scheme that uses a statistical model in a central server to reduce the needed sensor network activity. In an extreme case, a query could be answered without any access to sensor nodes. As with our scheme, BBQ meets application-specified accuracy requirements, however, based on the validity of the statistical model. Boulis et al.[3] take a similar approach, but in a distributed manner in which each node constructs a statistical model. In both schemes, the accuracy of the results depends intimately on the validity of the statistical model itself. On the other hand, our standpoint is that it is unrealistic that we can assume and validate statistical models which strictly satisfy the specified accuracy requirement in any situations. Thus we chose not to assume such models in the proposed algorithm.

6 Conclusion and Future Work

We have formulated data gathering problem with accuracy requirement and presented a distributed algorithm. The proposed algorithm accepts an application-specified accuracy requirement and computes a compact data representation for spatially distributed sensor data to meet that requirement. Our simulation experiments using real environmental data set demonstrate that the algorithm meets the accuracy requirements, and does so efficiently in terms of the size of data representation and the amount of internal communication. The algorithm exhibits good characteristics in efficiently exploiting the cost-accuracy trade-off. Since we don't need any assumptions on statistical properties of data, we believe the algorithm can serve as a general-purpose mechanism for data gathering-type sensor network applications.

Possible future work includes further evaluation of the proposed algorithm for a range of realistic environments and data sets. Examples of possible broadening include more detailed analysis of efficiency by taking into account the communication overhead and hardware characteristics. It may also contain the reconsideration of efficiency metrics, such as the balance of energy consumption among the nodes. Another possible direction includes the improving fault resilience of the algorithm in case of communication failure and node failure. We can borrow robust algorithms for constructing groups and routing tree, but we need to more carefully design the algorithm to deal with failures in a graceful manner.

7 Acknowledgments

The authors are supported in part by the National Science Foundation under awards NSF Cooperative Agreement ANI-0225642 (OptIPuter), NSF CCR-0331645 (VGrADS), NSF ACI-0305390, and NSF Research Infrastructure Grant EIA-0303622. Support from the UCSD Center for Networked Systems, BigBandwidth, and Fujitsu is also gratefully acknowledged. The first author is also supported by IBM Japan.

References

[1] S. Bandyopadhyay and E. J. Coyle. An energy efficient hierarchical clustering algorithm for wireless sensor networks. In *Proceedings of IEEE INFOCOM*, pages 1713–1723, March 2003.

[2] M. A. Batalin, M. H. Rahimi, Y. Yu, D. Liu, A. Kansal, G. S. Sukhatme, W. J. Kaiser, M. Hansen, G. J. Pottie, M. B. Srivastava, and D. Estrin. Call and response: experiments in sampling the environment. In *Proceedings of ACM Second International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 25–38, 2004.

[3] A. Boulis, S. Ganeriwal, and M. B. Srivastava. Aggregation in sensor networks: An energy-accuracy trade-off. In *IEEE Workshop on Sensor Network Protocols & Applications*, 2003.

[4] J. Considine, F. Li, G. Kollios, and J. Byers. Approximate aggregation techniques for sensor databases. In *Proceedings of the 20th International Conference on Data Engineering (ICDE)*, pages 449–460, 2004.

[5] A. Deshpande, C. Guestrin, S. Madden, J. Hellerstein, and W. Hong. Model-driven data acquisition in sensor networks. In *30th International Conference on Very Large Data Bases (VLDB)*, August 2004.

[6] C. Guestrin, P. Bodik, R. Thibaux, M. Paskin, and S. Madden. Distributed regression: an efficient framework for modeling sensor network data. In *Proceedings of The Third International Symposium on Information Processing in Sensor Networks (IPSN)*, pages 1–10, May 2004.

[7] R. Hassin and N. Megiddo. Approximation algorithms for hitting objects by straight lines. *Discrete Applied Mathematics*, 30:29–42, 1991.

[8] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *HICSS '00: Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 8*, Washington, DC, USA, 2000. IEEE Computer Society.

[9] S. M. Kay. *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*. Prentice Hall PTR, 1993.

[10] S. Lindsey and C. S. Raghavendra. Pegasis: Power efficient gathering in sensor information systems. In *Proceedings of IEEE Aerospace Conference*, 2002.

[11] J. Liu, J. Liu, J. Reich, P. Cheung, and F. Zhao. Distributed group management for track initiation and maintenance in target localization applications. In *Proceedings of The Second International Symposium on Information Processing in Sensor Networks (IPSN)*, pages 113–128, 2003.

[12] J. Lundquist, D. Cayan, and M. Dettin. Meteorology and hydrology in yosemite national park: A sensor network application. In *Proceedings of The Second International Symposium on Information Processing in Sensor Networks (IPSN)*, pages 518–528, 2003.

[13] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson. Wireless sensor networks for habitat monitoring. In *ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*, 2002.

[14] N. Shrivastava, C. Buragohain, D. Agrawal, and S. Suri. Medians and beyond: new aggregation techniques for sensor networks. In *Proceedings of ACM Second International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 239–249, 2004.

[15] G. Simon, M. Maróti, A. Lédeczi, G. Balogh, B. Kusy, A. Nádas, G. Pap, J. Sallai, and K. Frampton. Sensor network-based countersniper system. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 1–12, 2004.

[16] The National Snow and Ice Data Center (NSIDC). SMEX02 SSM/I Brightness Temperature Data, Iowa. `http://nsidc.org/data/docs/daac/nsidc0200_smex_ssmi.gd.html`.

[17] M. Welsh and G. Mainland. Programming sensor networks using abstract regions. In *First USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI)*, March 2004.

[18] R. Willett, A. Martin, and R. Nowak. Backcasting: adaptive sampling for sensor networks. In *Proceedings of The Third International Symposium on Information Processing in Sensor Networks (IPSN)*, pages 124–133, May 2004.

[19] N. Xu, S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin. A wireless sensor network for structural monitoring. In *Proceedings of ACM Second International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 13–24, 2004.

[20] O. Younis and S. Fahmy. Distributed clustering in ad-hoc sensor networks: A hybrid, energy-efficient approach. In *Proceedings of IEEE INFOCOM*, pages 629–640, March 2004.

[21] J. Zhao and R. Govindan. Sensor network tomography. In N. Bulusu and S. Jha, editors, *Wireless Sensor Networks: A Systems Perspective*. Artech House, 2005.

[22] ZigBee Alliance. `http://www.zigbee.org/`.