Signal Compression in Wireless Sensor Networks

By Marco F. Duarte[1], Godwin Shen[2], Antonio Ortega[3], and Richard G. Baraniuk[4]

[1]Electrical and Computer Engineering, University of Massachusetts, Amherst, MA
[2]Northrop Grumman Corporation, Los Angeles, CA
[3]Electrical Engineering, University of Southern California, Los Angeles, CA
[4]Electrical and Computer Engineering, Rice University, Houston, TX

Signal compression is an important tool for reducing the communication costs and increasing the lifetime of wireless sensor network deployments. In this paper, we overview and classify an array of proposed compression methods, with an emphasis on illustrating the differences between the various approaches.

Keywords: Wireless Sensor Networks, Distributed Compression, Transform Coding, Wavelets, Compressive Sensing

1. Introduction

Research on wireless sensor networks (WSNs) has been motivated by important applications such as climate change tracking, critical infrastructure monitoring, and wide area surveillance. In many of these applications, relatively simple sensors gather data that is then transmitted to centralized nodes, or *sinks*, where it is analyzed and processed. Compressing data as it is transmitted to the sink can play a potentially important role in increasing the lifetime of the sensors by reducing the number of bits they need to transmit, thus lowering overall power consumption. Over the last few years, several techniques have been developed for data compression in WSNs. This paper summarizes the key characteristics of several representative approaches and provides an overview of recent progress.

While standard data compression techniques can be directly applied to WSNs, improved performance can be achieved by taking into consideration the spatially distributed nature of the application. In particular, if the goal is to minimize battery consumption, then one should consider the cost of generating and transmitting bits to represent the information captured, rather than simply count the number of bits generated. To understand the importance of this point, consider a situation where sensors capture spatially correlated signals. In this scenario one could, for example, use information from one sensor to predict information in another and, given high correlation, an overall lower communication rate could be achieved. This, however, would require communication between the sensors or prior knowledge of their correlation, i.e., the first sensor would have to convey information to the second in order to effect predictive coding. A key observation is that this communication may reduce the overall number of bits to represent the signal, *but may not lower the overall energy consumption* due to the new communication costs.

In this paper, we focus on the challenges posed by *spatial compression*, i.e., exploiting redundancy in data across neighboring nodes. Techniques that exploit temporal redundancy do not require additional coordination/communication among sensors and thus can be deployed in combination with other methods without concerns about transport costs. A key intuition in this paper is that exploiting spatial correlation requires non-processed or raw data to be transmitted from certain *raw nodes* so that correlations can be exploited at other *aggregating nodes*. While the specifics of each compression technique differ, a good indication of performance can be obtained by considering the relative number of raw and aggregating nodes. Roughly speaking, choosing to have fewer raw nodes leads to lower initial transport costs (less unprocessed data is transmitted), but also means that compression rates at aggregating nodes may not be as high, since aggregating nodes will receive information from more distant, and thus less correlated, raw nodes.

We explain below how existing techniques (including the Karhunen-Loève transform, wavelets, and compressive sensing) can be adapted to operate in distributed environments by introducing a taxonomy of compression schemes for distributed WSNs. We describe each method in some detail and additionally illustrate how each can be modified to take into consideration the distributed nature of the application. Such changes improve the overall energy consumption for a given representation quality.

This paper is organized as follows. Section 2 provides background on WSNs and compression. Section 3 introduces the taxonomy of compression algorithms. Section 4 overviews example algorithms, while Section 5 provides conclusions.

2. Background

(*a*) *Wireless sensor networks*

A wireless sensor network (WSN) is a collection of wireless sensor devices that (*i*) can communicate with each other (and a sink node) via wireless data transmission, (*ii*) can sense, process, transmit, and receive data, and (*iii*) are able to perform these tasks in a distributed manner. WSNs naturally involve the theory and practice of sensing, signal processing, communication and information theory, and networking. Since the sensors communicate wirelessly and since *in-network data processing* will often require sensors to exchange data, a full treatment of data processing for WSNs requires addressing the communication infrastructure.

Let the signal of interest be $f(x, y, z, t)$, where x, y and z denote the coordinates of three-dimensional space and t denotes time. Suppose there are J sensors denoted by index $j = 1, 2, \ldots, J$ and that each sensor samples the signal at some spatial position (x_j, y_j, z_j). Also let t_n, for $n = 1, 2, \ldots, N$, denote the discrete times at which each sensor measures the signal. Let $\mathbf{f}$ denote the sampled signal vector where the set of samples $\mathbf{f}[i] = f(x_i, y_i, z_i, t_i)$ comprising $\mathbf{f}$ is given by the particular problem instance. In some cases, $\mathbf{f}$ contains noisy samples of the signal of interest f.

The sensed signal might consist of point measurements taken at infrequent intervals (e.g., temperature in a room every minute), one-dimensional signals acquired at a moderate to high sampling rate (e.g., seismic, acoustic, radio signals), two-dimensional images, three-dimensional video, or beyond. As we move to higher dimensions (both in time and space), the total amount of data generated quickly becomes enormous, necessitating some kind of compression. The compression can span different domains of the data space (temporal, spatial, spectral, etc.). As mentioned above, designing effective spatial compression schemes will typically entail communication between sensors and thus must obey the constraints placed by WSN architecture. This major challenge must be addressed in any WSN design.

(*b*) *Signal compression*

In signal compression, the goal is to find an approximation $\widehat{\mathbf{f}}$ to the signal of interest $\mathbf{f}$, encoded using a budget of b bits, that minimizes the approximation distortion $d(\mathbf{f}, b) = \|\mathbf{f} - \widehat{\mathbf{f}}\|$ according to a given distortion metric. The *rate-distortion* (RD) function $D(\mathbf{s}, b)$ (Cover & Thomas, 1991; Cristescu *et al.*, 2005) provides a monotonic map between the size of the compressed signal and its fidelity to the uncompressed version. Often we can parametrize the approximation $\widehat{\mathbf{s}}_K$ by the number K of degrees of freedom involved in the approximation. One can then consider the distortion as a function of K with a fixed number of bits b, $d(\mathbf{s}, K)$, and spend, for example, b/K bits encoding each of the degrees of freedom.

(*c*) *Distributed compression*

In a WSN, the recorded signal is not immediately available at a centralized location; such collection would incur significant communication expense. Therefore, WSNs prefer distributed algorithms to perform compression. The main challenge in the design of a distributed compression algorithm is determining the extent of coordination required among the nodes to transmit and process the acquired signal in a distributed fashion, i.e., the partitioning of the network into raw and aggregating nodes. The goal of the algorithm is typically to reduce the amount of information sent to the sink while achieving the smallest distortion possible. Ideally, this is done so that the overall signal gathering cost (measured in terms of communication expense) is minimized for a given fidelity in reconstructing the signal obtained from each node.

As an additional consideration, it is often desirable to tailor the design of the compression algorithm to meet particular constraints on communication for the WSN, such as routing topology, latency, and power consumption (Scaglione & Servetto, 2005). In the case where these constraints are severe and rigid, the compression algorithm design must carefully balance these additional constraints with the resulting compression rate and distortion achieved.

Table 1. *Taxonomy of WSN compression algorithms. Notation: RAW: raw data; SO: spatial only; TO: temporal only; ST: spatio-temporal; SD: separate design; RC: routing over compression; CR: compression over routing; JO: jointly optimized; NC: no coding; DSC: distributed source coding; DTC: distributed transform coding; CS: compressive sensing; DPC: distributed predictive coding.*

Name	Model	Rout/Comp	Coding
Collection tree protocol (Ciancio *et al.*, 2006)	RAW	RC	NC
Adaptive temporal coding (Song *et al.*, 2010)	TO	SD	NC
Distributed source coding (Cristescu *et al.*, 2005)	SO	SD	DSC
Distributed KLT (Gastpar *et al.*, 2006)	SO, ST	SD	DTC
Tree KLT (Shen *et al.*, 2009*a*)	SO	RC	DTC
Tree-based wavelets (Wagner *et al.*, 2006)	SO	SD, CR	DTC
Graph-based wavelets (Shen & Ortega, 2010)	SO	RC	DTC
Distributed compressive sensing (Duarte *et al.*, 2006)	ST	SD	CS
Compressive wireless sensing (Bajwa *et al.*, 2006)	SO	CR	CS
Randomized gossiping (Rabbat *et al.*, 2006)	SO, TO, ST	CR	CS
Sparse random projections (Wang *et al.*, 2007)	SO, TO, ST	CR	CS
Localized compressive sensing (Lee *et al.*, 2009)	ST	JO	CS
T-DPCM (Pattem *et al.*, 2009)	SO	JO	DPC

3. WSN Compression Algorithm Taxonomy

We present a taxonomy for a number of representative WSN compression algorithms in Table 1. The algorithms will be discussed in detail in Section 4. Our taxonomy considers the type of compression performed and the integration between the routing and networking aspects of the WSN and the compression algorithm. Compression algorithms often consist of the combination of two parts: a *signal model* that provide energy compaction through linear transformations (a process known as *transform coding*) and a *coding scheme* that encodes the values of the transform coefficients using as few bits as possible. The taxonomy categories below explore three different dimensions involved in the design of a complete distributed compression system.

- *Signal model*: the signal model identifies the key characteristics of the signals to be exploited during compression (e.g., spatial correlation due to a physical model, or sparsity due to low event complexity).

- *Interplay between routing and compression:* the design of the compression algorithm may have to respect specific constraints on the communication architecture of the WSN (e.g., routing topology, communication scheme such as unicast vs. broadcast) during the design of the signal model, data collection, etc.

- *Coding tools*: we must select a specific coding and quantization algorithm. Several design parameters may be available for each option to decide how to exploit signal characteristics while operating under communication constraints.

(*a*) *Signal models for compression*

There are many possible types of correlation or structure that can be leveraged in linear transformations for data compression purposes. In the first class, the coding scheme acts directly on the raw data obtained at the sensors without the need to transform the data to encode the correlations between sensors. This *raw data* approach is appropriate, for example, when the sensors aim to report the presence of highly localized activity that is sensed only by few sensors, or when the sensors aim to report anomalies in sensing or operation, such as reduced battery power or device failure. The raw data approach is optimal in these cases due to the uncorrelated nature of the data communicated by the sensors.

In the second class, the compression scheme exploits only temporal correlations. In such *temporal-only* approaches, each sensor applies one of the coding schemes described in Section 3c to its own observation vector. These approaches do not require communication during the compression step, since the sensor has all the data needed for the algorithm. On the one hand, temporal-only models may be optimal when signals are temporally correlated and temporal sampling rates are high. They may also be best if a simpler system is preferred, since sensors can operate independently of each other (without having to coordinate). On the other hand, temporal-only models do not exploit the spatial correlations with other sensors in their local neighborhood that appear in many WSN applications. Furthermore, temporal-only models can lead to excessive latency due to the need to collect enough time samples to achieve meaningful compression rates.

In the third class, the compression scheme exploits the correlation among data from different sensors. In these *spatial-only* approaches, the compression is applied to the distributed observation vector $\mathbf{f}_{t_n} = [f(x_j, y_j, z_j, t_n)]_{1 \leq j \leq J}$. Spatial-only models are well-matched to applications that record data from smoothly varying fields or where sensor density is high. In contrast to the raw and temporal-only approaches, this class of models can impose a large communication burden on the WSN. Furthermore, spatial-only models do not exploit the temporal coherence that is present in many WSN applications, potentially causing significant redundancy during compression at successive times.

Ideally, a signal model for compression purposes will exploit both temporal and spatial correlations. Such *spatio-temporal* models consider all samples $\mathbf{f} = [f(x_j, y_j, z_j, t_n)]_{1 \leq j \leq J, 1 \leq n \leq N}$ and typically achieve the best tradeoffs in rate distortion. Not surprisingly, these methods face the challenges of both temporal-only and spatial-only models, including potentially high latency, increased communication burden, and increased coding complexity.

(*b*) *Interplay between routing and compression*

As mentioned in Section 2c, WSN compression schemes must carefully balance the goals of low-rate representation of the compressed data, low-distortion approximation, and suitability to the communication constraints and routing topology of the WSN. The simplest designs are *separate designs* that treat the communication and compression schemes independently.

When the communication and networking constraints are severe and rigid, the compression scheme must adhere to the resources available to the network, prioritizing *routing over compression.* Characteristics that can force rigid networking constraints include large scale deployments, limited power availability, and minimum network lifetime.

When the communication and networking constraints are more flexible, the compression scheme can take priority over networking during the search for the optimal rate-distortion tradeoff, i.e., we can prioritize *compression over routing.* In this case, it can be desirable to first design the compression scheme and then design the routing topology to optimize the performance of the compression algorithm. Such flexibility can be present when the sensor density is high and power and communication resources are less restricted; the high density of sensor nodes makes it more desirable to trade off additional expense of the available resources for a higher degree of compression.

Clearly the optimal approach is to *jointly optimize* the compression algorithm and routing scheme. However, the rich parameter space available in this class of applications increases not only the potential payoff but also the level of difficulty of the design process (Pattem *et al.*, 2004; Scaglione & Servetto, 2005; Luo & Pottie, 2007; Shen & Ortega, 2008*a*).

(*c*) *Coding tools*

We identify five main classes of coding algorithms that can be applied directly to the data or to its representation under a signal model. The first (trivial) class is *non-coding.* This is appropriate for data that exhibits no spatial or temporal correlation, since there is nothing to be gained by compressing un-correlated data. Note that even if spatial correlation is present in the data, non-coding may be competitive with spatial compression methods. For instance, this may be the case if spatial compression requires coordination through communication that is too costly for the nodes, or if the bitrate reduction achieved is not sufficiently significant (e.g., if the overhead bits due to packetization dominate the communication cost).

When a high degree of spatial data correlation exists, it will likely be beneficial to perform some form of distributed spatial compression. The idealized scenario is when the (statistical) spatial correlation structure is known; in this case, one can simply perform distributed source coding to "optimally" encode each sensor's data, where "optimality" is understood in the traditional RD sense (Cristescu *et al.*, 2005). However, since the spatial correlation structure is typically unknown (or is difficult to estimate), one must often resort to simpler spatial compression methods that do not rely on a model or can adapt to changes in the correlation structure. One simple way to model spatial correlation is via *distributed predictive coding.* Distributed predictive coding schemes operate by (*i*) predicting data at each node using previously encoded data from other nodes, and (*ii*) encoding and transmitting the prediction residuals.

While existing distributed predictive coding schemes provide a simple form of coding, they do not fully exploit the spatial data correlation. In order to achieve greater compression through spatial processing, it is necessary to develop more general *distributed (sparsity-inducing) transform coders.* These transforms operate linearly on data taken from multiple nodes and yield "transform coefficients". These coefficients are then encoded and transmitted to the sink. These methods often outperform non-coding, temporal-only, and distributed predictive coding methods. Extensions to spatio-temporal transforms have also been explored and been shown to yield even better performance.

Finally, in compressive sensing (CS) approaches, combinations of data from multiple nodes are generated and encoded, as in distributed transform coding; however, in contrast to transform coding, the number of linear combinations generated is smaller than the number of sensors in the network. CS approaches can exploit prior knowledge about the signal being sensed (e.g., smoothness) in order to resolve during signal reconstruction the ambiguities produced by undersampling.

4. Distributed Compression Algorithms for WSNs

We now review a representative set of distributed compression algorithms for WSNs that we classified under our taxonomy in Table 1.

(*a*) *Non-transform based data collection techniques*

(i) *Collection tree protocol*

The collection tree protocol (CTP)(CTP, 2007) is a well-known example of a routing-over-compression non-coding method. The basic idea is to choose communication routes according to a heuristic measure of communication cost called Expected Transmissions (ETX). In this algorithm, each node keeps a running average of the number of transmissions needed to send a packet to each of its neighbors; multiple transmissions are typically needed since packet losses occur often. Moreover, nodes that are one hop from the sink propagate their running averages to nodes that are two hops away from the sink. Then the "two-hop nodes" add the received running averages to their own running averages, forming a "two-hop" ETX value for the "two-hop" nodes. The two-hop nodes then send their two-hop ETX values to their "three-hop" neighbors, and so on. In this way, nodes can form a minimum ETX routing tree simply by choosing their parent as the one-hop neighbor with minimum ETX. This is optimal from a routing standpoint (in the minimum ETX sense), but is clearly suboptimal in terms of communication cost since the data is uncompressed.

(ii) *Adaptive temporal coding*

One way to improve upon non-coding protocols such as CTP is to perform compression at each node using its own samples collected over time, i.e., perform *temporal coding.* The majority of these methods employ adaptive coding over time followed by entropy coding (Santini & Romer, 2006; Song *et al.*, 2010). This particular technique can yield significant improvements over raw data collection. Note that these algorithms are temporal only and do not exploit spatial correlations, and so one could potentially improve performance by incorporating some form of spatial compression. However, since compression is done locally at each node, no inter-sensor communication is required to perform compression. In fact, temporal coding can be classified as a separately designed routing and compression scheme. This has obvious practical advantages, since sensors do not need to expend energy for inter-sensor communications in order to compress, nor is there any need to coordinate these communications, which can be a difficult problem if the communication structure changes frequently.

(iii) *Tree-DPCM*

When the data flowing towards the sink follows a routing tree, each node in the tree has access to all of the data transmitted by its descendants in the tree. Thus, a simple way to reduce the overall bitrate is for each node to compute a prediction based on data from its descendants and transmit to its parent only a residual with respect to this predictor. The parent node, in turn, can reconstruct data from its descendants, compute a prediction based on descendant data, and so on. An example of such an approach (Shen *et al.*, 2009*a*) demonstrates significant gains over raw data transmission, especially if adaptive prediction is performed.

(iv) *Distributed source coding*

In the event that the (statistical) spatial correlation structure is known, the best routing-over-compression method for the spatial-only model is distributed source coding (DSC) (Cristescu *et al.*, 2005). However, such methods require reliable estimates of the correlation structure. In practice, achieving such reliable estimates may require observing the data being sensed over impractically long intervals. For example, in a general case, if complicated correlations exist across sensors, then one may need to estimate a covariance matrix that is not necessarily sparse (so that a large number of measurements is required for reliability). Moreover, the correlation structure may change over time, so that data gathering without DSC will be needed from time to time in order to re-estimate it. Thus, any practical DSC scheme will need to operate at a rate greater than what theoretically can be achieved because (*i*) the higher rate will be needed to ensure some robustness with respect to mismatches between the estimated model and the actual correlation structure, and (*ii*) non-DSC techniques will have to be used from time to time in order to adapt the model parameters. These practical issues have so far not been fully studied, so that the degree to which DSC methods can be deployed in practice remains an open question.

(*b*) *Distributed transform coding*

In transform coding, the compression scheme acts on a linearly transformed version of the data. Specifically, for the sampled data $\mathbf{f}$, we typically use an orthonormal transformation matrix Ψ that yields a coefficient vector $\mathbf{w} = \Psi^{-1}\mathbf{f}$ (so that $\mathbf{f} = \Psi\mathbf{w}$) and achieve compression by encoding $\mathbf{w}$ using the schemes of Section 3c, dividing the bit budget amongst the entries of $\mathbf{w}$ according to their impact in signal distortion. For example, the bit budget may be partitioned among the entries of $\mathbf{w}$ according to their expected magnitudes; in certain cases, it might be optimal to discard small coefficients when the savings in bitrate outweigh the distortion incurred by ignoring them. The signal is then decompressed by decoding the bitstream and populating an approximate coefficient vector $\widehat{\mathbf{w}}$ with only the preserved coefficients (setting all others to zero), and then inverse transforming to obtain the estimate $\widehat{\mathbf{f}} = \Psi\widehat{\mathbf{w}}$. The amount of distortion introduced by this type of compression is due to the quantization error during coding and the magnitude of the discarded coefficients; thus, efficient transform coding relies on a suitable transform and coding scheme pair that minimizes these two sources of distortion.

When transform coding is implemented in a distributed setting, we must account for the communication cost of computing the coefficient vector $\mathbf{w}$. This will depend both on the type of signal model used (spatial and/or temporal) and on the characteristics of the entries of the matrix Ψ (e.g., if a given row of Ψ^{-1} has nonzero entries corresponding to samples of $\mathbf{f}$ taken by multiple sensors, then the sensors will need to communicate or pool their information at a sink to calculate the corresponding transform coefficient). To integrate knowledge of the routing and networking structure into transform coding, we must restrict the choices of matrices Ψ so that the communication required to apply Ψ^{-1} to the data matches the network constraints.

In this subsection, we will focus on the Karhunen-Loève Transform (KLT), where the signal $\mathbf{f} \in \mathbb{R}^{\mathbf{N}}$ is modeled as a multivariate Gaussian random vector with zero mean (for simplicity) and low-rank (or approximately low-rank) covariance matrix $\Sigma_{\mathbf{f}}$. Intuitively, this model represents signals as belonging to (or being close to) a K-dimensional subspace of $\mathbb{R}^N$ for a sufficiently large value of K. Under this model, the signal can be approximated simply by obtaining a basis Ψ for the aforementioned subspace and obtaining the K coefficients of the signal in that basis. The basis Ψ can be learned from representative datasets $\{\mathbf{f}\}$ using principal component analysis (PCA). However, the generic KLT does not pose particular constraints on the structure of the obtained basis Ψ.

(i) *Distributed KLT*

The distributed Karhuenen-Loève transform (DKLT) (Gastpar *et al.*, 2006) is an adaptation of the KLT to communication-constrained settings. While in the standard KLT the estimated basis Ψ is generally dense, when the WSN must restrict its communication to only local transmissions, the KLT coefficients should depend only on sensor readings from communication graph neighborhoods. That is, the matrix Ψ_D should have *block-diagonal form*, where each block in the diagonal of Ψ_D transforms a local neighborhood of samples from $\mathbf{f}$ to a subset of the coefficients $\mathbf{w}$.

While finding the optimal matrix Ψ_D is an open problem, it is possible to find locally optimal solutions by iterating over the blocks corresponding to the different

WSN neighborhoods using modified PCA approaches. Such approaches can account for knowledge of partial information about the rest of the network's compression scheme, e.g., knowledge of the local correlation matrices, partial knowledge of local KLT bases, or partial knowledge of the recorded signals. Depending on the composition of the vector $\mathbf{f}$, the DKLT can use a spatial-only or spatio-temporal model. DKLT also follows the routing-over-compression design scheme, since the communication network neighborhoods determine the constraints of the DKLT compression basis Ψ.

(ii) *Tree KLT*

The Tree KLT (T-KLT) method is a routing-over-compression, distributed transform coding method that is "optimal" for a spatial-only model (Shen *et al.*, 2009*a*). It is optimal among the class of "unidirectional transforms" in the sense that it decorrelates data along any sub-tree of the routing tree. First, data from "leaf nodes" (i.e., nodes with no children in the routing tree) is transmitted in raw form to their parents. Then, a KLT matrix is computed at each parent node and the KLT is applied to the raw data. From that point on, since each node receives KLT coefficients instead of raw data, each node (*i*) applies an inverse KLT to data from each of its children, (*ii*) computes a KLT for data from itself and its descendants, then (*iii*) transmits the KLT coefficients. This process is repeated until all data is received at the sink. The sink then performs an inverse KLT on data received from each of its children (i.e., its 1-hop neighbors). Note that, for this approach to work, either (*i*) nodes would have to transmit the KLT matrices they design to their parent nodes, or (*ii*) statistical knowledge of the correlation structure would have to be available at the nodes. More specifically, it would be necessary for each node to obtain estimates of the covariance matrix of the data on its own sub-tree and the covariance matrix of the sub-tree for each of its children. Since covariance matrices are generally not easy to estimate, this may not be the most practical routing-over-compression distributed transform coding method. Moreover, since many inverse transforms are needed to recover original data on each sub-tree, this method will be susceptible to quantization error propagation. Thus, this method can serve mostly as an upper bound to benchmark the performance of other unidirectional transform methods.

(*c*) *Wavelet transforms*

Transform coding can be extended from the KLT to models that are more elaborate than a low-dimensional subspace and that need not be signal-dependent. In particular, the sparsity model assumes that signals are within or close to one of the K-dimensional subspaces of $\mathbb{R}^N$ spanned by an arbitrary choice of K out of the N column elements of Ψ. Many classes of signals have been shown to exhibit sparsity in particular bases. For example, the Fourier basis provides sparse representations for smooth signals, while wavelet bases provide sparse representations for piecewise smooth signals.

In this subsection, we focus on the case of wavelet transforms, which are natural for WSN settings such as anomaly detection in smooth signal fields as well for providing compact representations for smooth fields with few discontinuities. In particular, we focus on the wavelet lifting implementation (Sweldens, 1996), which provides an algorithmic wavelet transform that is ideally suited for low computation constraints. In wavelet lifting, for each sample within the coefficient vector $\mathbf{f}$, we operate on neighborhoods of multiple sizes. Each neighborhood is partitioned into two sets (commonly called *even* and *odd*), and the transform coefficients are calculated by applying *prediction* and *update* filters to the even and odd neighbors of each odd and even sample, respectively. The neighborhood sizes vary according to the scale, providing a multi-resolution analysis of the vector $\mathbf{f}$; the presence of large wavelet coefficients at a given location and scale indicates the presence of discontinuities in the sensed field within the corresponding neighborhood of the studied sensor.

For wavelet compression in WSNs, a sink queries for sufficiently many encoded wavelet coefficients in order to reach a target compression distortion. Compression here results from the fact that the bitrate needed to encode wavelet coefficients will be smaller than the bitrate required by a standard data dump to the sink. The savings in coefficient coding have to be balanced with the additional communication expense required to calculate the wavelet coefficients. Thus, there generally exist efficiency breakpoints, dependent on the number of sensors in the network and the size of the field, in which the communication cost of calculating wavelet coefficients overweighs the savings in communications from wavelet compression. Note in particular that computing prediction coefficients requires sending raw data from even to odd neighbors in a neighborhood, so overall transmission cost may be significantly affected by the number of such “raw” nodes in the network.

(i) *Irregularly sampled wavelets*

The baseline wavelet lifting approach is designed for signals $\mathbf{f}$ that have been sampled on a uniform grid. Additionally, the prediction and update filter designs assume that the cost of computing a wavelet coefficient depends solely on the number of samples involved. This setup does not map very well onto a WSN, since in a WSN sampling is very often performed irregularly in the space domain according to the particular deployment of the sensors in the field. Nonetheless, it is possible to adapt wavelet transforms designed for regularly sampled data to irregular sampling (Wagner *et al.*, 2006; Baraniuk *et al.*, 2008). In this case, the definition of the neighborhoods must be modified. This approach utilizes location information to define local neighborhoods of nodes at multiple scales that exchange information, forming multiresolution geographical neighborhoods. This also forces the neighborhoods to be calculated in sequence from finest (smallest) to coarsest (largest) scales.

For a local neighborhood at a particular scale, the partition stage arbitrarily selects a node to be placed in the even subset and moves all nodes within a smaller geographical neighborhood of the new even node to the odd subset. This process is repeated with the nodes remaining in the neighborhood until each node has been labeled as even or odd. The prediction and update filters are designed to perform optimal approximation of the corresponding sampled value by fitting a plane to the readings from sensors in the even/odd neighborhood partitions, correspondingly (Wagner *et al.*, 2006). The prediction and update steps can be made robust to communication losses through simple updating of the weights for sensors involved in the dropped communication links and their immediate neighbors.

The irregular wavelet transform requires communication in local neighborhoods to compute scaling and wavelet coefficients. While the neighborhoods become larger as the wavelet scale becomes coarser, it has been shown experimentally that this transform provides a more balanced distribution of the communication expense needed to receive all sensed values, as compared to a data dump approach that stresses the communication bottleneck around the sink.

The irregular wavelet transform is well suited for settings in which the communication neighborhoods are small and multiple hops are involved in the paths to the sink, which increases the communication savings due to compression. This approach falls in the spatial-only class; additionally, the fact that the filters used for compression are chosen according to the geographical neighborhoods, which are not necessarily tied with the network topology of the WSN, means that the method can be placed in the compression-over routing or separate design classes.

(ii) *Graph-based wavelets*

While the irregular wavelet transform makes use of node proximity regardless of network topology (compression over routing), graph-based wavelets (Shen & Ortega, 2008*b*; Shen *et al.*, 2009*b*; Shen & Ortega, 2010; Narang *et al.*, 2010) use a lifting transform constructed on the network communications graph (routing over compression). The partitioning step chooses odd nodes that provide maximal decorrelation. A lifting transform can be inefficient in terms of the overall number of communications, especially if even nodes must first transmit raw data to odd neighbors and then wait for odd neighbors to transmit transform coefficients back to them before they can compute their own transform coefficients and forward them to the sink. This may produce significant communication overhead, since many nodes are transmitting data twice, and data may be transmitted *away* from the sink.

Graph-based wavelets strive for low communications cost by requiring nodes to transmit data just once and to do so in the direction of the sink. In fact, it is possible to transmit data only once by computing the transform as data is forwarded to the sink along a given routing tree. Such transforms have *unidirectional operation* (Shen & Ortega, 2008*b*, 2010), since each node only transforms its own data using data from neighbors along a one-dimensional routing path to the sink. This is achieved by constructing the transforms along an arbitrary routing tree while also utilizing data received via broadcast from neighbors not in the tree. The resulting transform is critically sampled (one coefficient per node) and exploits correlation across routing paths and broadcast links.

It is possible to optimize the even-odd splitting under unidirectionality constraints. An even-odd splitting can be viewed as a constrained set-covering problem where every odd node needs to be covered by at least one even node and the communications are constrained by the transmission ranges used in the routing tree; i.e., a node can only "cover" neighbors whose distance is less than the distance from itself to its parent in the routing tree. The heuristic constrained minimum set covering algorithm of Narang *et al.* (2010) has been shown to provide lower energy consumption than other graph-based wavelet methods for a fixed reconstruction quality.

(*d*) *Compressive sensing*

In some WSN applications, it is difficult or costly to learn or apply an appropriate decorrelating transform. For such applications, compressive sensing (CS) (Candès, 2006), which computes a dimensionality reducing linear projection of the measurement values, is a promising alternative to transform coding.

In CS, we assume that the signal $\mathbf{f}$ has a sparse or nearly sparse representation $\mathbf{w}$ in an orthonormal basis Ψ, where $\mathbf{w} = \Psi^{-1}\mathbf{f}$, with only K coefficients in $\mathbf{w}$ being large or significant. Under this assumption, it is possible to obtain a reduced-length representation $\mathbf{y} \in \mathbb{R}^M$, $M < N$, by using a measurement matrix Φ that provides $\mathbf{y} = \Phi\mathbf{f}$. When the matrix Φ preserves the structure of sparse signal vectors, the signal $\mathbf{f}$ can then be estimated accurately when $\mathbf{y}$ and Φ are known using sparse recovery algorithms. Interestingly, a large class of *random matrices* is suitable as long as $M = \mathcal{O}(K \log(N/K))$. More generally, we choose Φ to have rows that are *incoherent* with the elements of the transform Ψ, i.e., the rows of Φ should not align with elements of Ψ.

CS can reduce the total amount of communication required during the compression stage through dimensionality reduction when $K \ll N$. However, it might involve additional collaboration between sensors to calculate the measurements $\mathbf{y}$. The standard CS framework must also be modified to meet the WSN communication constraints described earlier.

(i) *Distributed compressive sensing*

Distributed compressive sensing (DCS) (Duarte *et al.*, 2006) is an extension of the standard CS acquisition framework to correlated signal ensembles. Let $\mathbf{f}_1, \ldots, \mathbf{f}_J$ be a set of sparse or nearly sparse signals in the time domain that are acquired by the nodes in a WSN. Since the different signals correspond to different recordings of the same event, we can expect significant structure to be present between the significant coefficients of their representation in some basis. For example, a *sparse common/innovations model* expresses each signal as the sum of a sparse common component, present at each node, and a sparse innovation component that is unique and different at each node. A practical situation that follows this model is environmental sensing in which light, humidity, etc. is approximately the same across a field, but with few small local fluctuations.

While it is possible to perform CS acquisition at each sensor and then perform separate recovery for each individual signal, this naive approach ignores the structure present between the signals in the WSN. DCS exploits this additional structure through specially tailored signal recovery algorithms. To reduce the communication required between the sensors during the measurement process, each sensor obtains measurements of its own signal $\mathbf{y}_j = \Phi_j \mathbf{f}_j$ independently; these measurements are then sent to a central location that is interested in recovering the signal. At the sink, the entire CS measurement process is fused through measurement vector and signal concatenation together with block diagonalization for the CS matrix to obtain a single standard CS measurement equation $\mathbf{Y} = \Phi\mathbf{X}$. We then can recover the signal ensemble from $\mathbf{Y}$ using the matrix Φ with modified CS recovery algorithms. It has been shown that the number of measurements needed from the network for accurate recovery of all of the signals in the WSN can be reduced from $M = \mathcal{O}(JK \log(N/K))$ (using separate CS on each signal) to just $M = \mathcal{O}(K)$ as $J \to \infty$, where K denotes the sparsity of each signal.

Since DCS exploits both the structure of each sensor's observed signal and across the signals observed by the network, we classify the algorithm as spatio-temporal. Additionally, since we do not rely on a particular communication scheme to collect the data at the sink, DCS falls under the separate design class of the routing/compression taxonomy.

(ii) *Compressive wireless sensing*

Compressive wireless sensing (CWS) performs CS over the sensing field by distributing the calculation of the inner products with the measurement vectors (Bajwa *et al.*, 2006). CWS assumes that a sink is interested in obtaining the recorded data from an ensemble of sensors. In this case, the vector $\mathbf{f}$ contains the J samples obtained by the sensors in the ensemble for a single discrete time instance. Denote by $\Phi(m, j)$ the entry of Φ at row m and column j. During the m^{th} discrete time instance, $1 \leq m \leq M$, the j^{th} sensor multiplies its measured value with the scalar $\Phi(m, j)$. The scalar products are then aggregated by a matched source/channel communications architecture: the scalar products are used to modulate the amplitude of a codeword waveform that is sent synchronously to the fusion center. The center then observes the sum of the scalar products aggregated through the communication channel, effectively obtaining the inner product $\langle[\Phi(m, 1)\ \ldots\ \Phi(m, J)]^T, \mathbf{f}\rangle$ corresponding to the measurement $\mathbf{y}[m]$.

The use of analog modulation in CWS requires a bound on the entries of the matrix Φ and the vector $\mathbf{f}$ to meet power constraints for the WSN. The constraints may require scaling down all measurements by a fixed multiplicative factor. Additionally, the wireless channel-based aggregation requires close synchronization and knowledge of channel state information. On the other hand, the use of synchronous communication bounds the required communication power required from the entire network for communication, improving network scalability (Bajwa *et al.*, 2006). Since CWS relies on a particular communication scheme, we place it under the jointly optimized class of the taxonomy. Additionally, CWS is labeled as an spatial-only compression scheme due to its focus on samples obtained at a single discrete time instance at each sensor.

(iii) *Randomized gossiping*

In contrast to the matched source/channel communication used in CWS, it is also possible to compute the projections of the data into measurement vectors through randomized gossiping (Rabbat *et al.*, 2006; Dimakis *et al.*, 2010). Randomized gossiping spreads the measurements among the sensors in the network in such a way that it suffices to query a small number M of sensors in order to obtain all measurements necessary for recovery. This approach makes most sense when communication to the sink is too expensive, or when it is desirable to have the CS measurements be available from any one of the nodes in the WSN.

Randomized gossiping assumes that each sensor j, $1 \leq j \leq J$, is aware of the j^{th} column ϕ_j of the CS matrix Φ. Each sensor multiplies its own measured value $\mathbf{f}[j]$ with the corresponding column ϕ_j to provide its initial "message" $\mathbf{m}_j(0) = \mathbf{f}[j]\phi_j$. For each communication instance t, a node j_1 is selected uniformly at random; the node then chooses a second node j_2 also uniformly at random, and the pair of nodes exchange their messages to calculate the update, which is equal to the average of the pair of messages. It is possible to show that as $t \rightarrow \infty$, the messages $\mathbf{m}_j(t) \rightarrow \mathbf{y}/J$ for all $1 \leq j \leq J$. In words, randomized gossiping enables both the computation and distribution of the CS measurements.

When it is infeasible to maintain a communication graph, e.g., due to mobility or low sensor reliability, randomized gossiping provides a more robust means of disseminating information at the cost of higher communication and routing complexity. More generally, the improvement afforded by randomized gossiping depends on the ratio between the transmission cost to the sink and the communication cost among nodes. Randomized gossiping can use a spatial and/or temporal signal model in the taxonomy of Table 1, depending on the samples of $f(x, y, z, t)$ producing the discretized representation $\mathbf{f}$. Additionally, we classify this method as a compression-over-routing method since the communication protocol is driven by the design of the measurement vectors.

(iv) *Sparse random projections*

To alleviate the communication expense associated with the computation of the CS measurements given in CWS and randomized gossiping, we can employ measurement matrices that are themselves sparse, which then implies that only small groups of sensors have to communicate with each other in order to calculate the measurement values (Wang *et al.*, 2007). In particular, we assume that each sensor j selects each different sensor independently with probability L/J, and collect the sensors selected by sensor j as a set $\Omega_j \subseteq \{1, \dots, J\}$. On average, we will then have $E[|\Omega_j|] = L$. Sensor j will then query the sensors $j' \in \Omega_j$ for their readings $\mathbf{f}[j']$. In this way, each sensor can calculate a measurement of the form $y_j = \sum_{j' \in \Omega_j \cup \{j\}} a_{j,j'} \mathbf{f}[j']$, where $a_{j,j'}$ are i.i.d. Rademacher (± 1) random variables. Such measurements can be rewritten as $\mathbf{y} = \Phi\mathbf{f}$, where each row of Φ contains nonzeros corresponding to the indices in $\Omega_j \cup \{j\}$, and the rows correspond to a subset $\Lambda \subseteq \{1, \dots, J\}$, $|\Lambda| = M$ of probed sensors.

The use of sparse random projections imposes a mild condition on the signal structure: the signal should be dense in its sampled representation $\mathbf{f}$ (in time and/or space) in order to provide guarantees on signal recovery via sparsity or compressibility under transform coding. Under this condition, it is possible to obtain a sufficiently accurate approximation of the coefficient vector as long as $M = \mathcal{O}\left(\frac{LK^2 \log J}{J}\right)$. For example, one can significantly reduce the communication cost of CS by letting $L = 1$, i.e., query just one sensor per measurement, when the signal is compressible using the discrete cosine transform or a standard KLT with a dense basis Ψ (Masiero *et al.*, 2009). The similarities between randomized gossiping and sparse random projections explain the matching classifications under the taxonomy.

(v) *Localized compressive sensing*

The CS schemes described so far reduce the data gathering cost by ensuring that each projection requires communication within a small group of sensors. However, if these projections are generated randomly, then the sensors involved could be far apart and the resulting transport cost might still be high. Recent work has shown that, when considering the trade-off between reconstruction quality and transport cost, *structured* localized CS approaches can outperform sparse random projections (Lee *et al.*, 2009). In particular, for a known WSN configuration it has been shown that one could select a routing tree to provide efficient data gathering and then define CS projection operators that are restricted to involve sensors within "sections" of the routing tree. For example, one can divide the sensor network into a series of quadrants centered at the sink and have routing and projection proceed independently within each of those quadrants.

Gains have been reported with simple tree building strategies and extended to provide a cost function for tree optimization (Lee & Ortega, 2010). The goal is then to choose a sectioning of the routing tree that drives the design of Φ, so as to minimize the coherence with the basis Ψ. The intuition for this approach is that when the elements of the basis Ψ overlap with a large number of sections of the network, the information is "spread" and can be efficiently measured in a localized fashion. In general, localized routing and projections lead to higher coherence for many common choices of basis Ψ (thus requiring more measurements), but result in a lower routing cost per measurement. With this metric, it is possible to search for localized routing leading to projections having lower coherence. For example, a simple tree building heuristic based on this metric leads to improved reconstruction over sparse random projections (Lee & Ortega, 2010). This approach, like the sparse random projection method, can be seen to use a spatial-only model, but unlike sparse random projections, routing is jointly optimized along with the transform.

5. Discussion and Conclusions

In this paper, we have reviewed a number of representative examples of distributed compression algorithms for WSN applications. The design constraints of WSN applications demand a compromise between the degree of compression achieved by the algorithm and the power and communication burdens imposed by the algorithm. The degree of balance between these two opposing goals yields a variety of types of approaches, depending on the degree of synergy between the coding and communication aspects of the algorithm, as well as on the types of signal models used during compression.

The degrees of freedom available for the design of distributed compression algorithms have motivated the algorithm taxonomy developed in this paper. The taxonomy provides not only a way of classifying current approaches, but also a roadmap for future research in this area. Important avenues not covered here include, for example, the development of distributed compression algorithms in networks of mobile and/or unsynchronized sensors. We expect the importance of data compression to increase over the coming decades due to the exploding amount of data acquired in ever more dense and ever more high resolution WSN deployments (Baraniuk, 2011). Clearly this is an exciting time for both WSNs and distributed compression.

We thank the anonymous reviewers for many helpful comments. RGB was supported in part by grants NSF CCF-0431150 and CCF-0728867. MFD was supported in part by NSF Supplemental Funding DMS-0439872 to UCLA-IPAM, P.I. R. Caflisch. GS and AO were supported in part by grants NASA AIST-05-0081 and NSF CCF-1018977.

References

2007 Collection tree protocol, tinyos-2. http://www.tinyos.net/tinyos-2.x/doc/.

Bajwa, W., Haupt, J., Sayeed, A. & Nowak, R. 2006 Compressive wireless sensing. In *Int. Conf. Info. Proc. in Sensor Networks (IPSN)*, pp. 134–142. Nashville, TN.

Baraniuk, R. G. 2011 More is less: Signal processing and the data deluge. *Science*, **331**(6018), 717–719.

Baraniuk, R. G., Cohen, A. & Wagner, R. 2008 Approximation and compression of scattered data by meshless multiscale decompositions. *Appl. Comput. Harmon. Anal.*, **25**(2), 133 – 147.

Candès, E. J. 2006 Compressive sampling. In *Int. Congress of Mathematicians*, vol. 3, pp. 1433–1452. Madrid, Spain.

Ciancio, A., Pattem, S., Ortega, A. & Krishnamachari, B. 2006 Energy-efficient data representation and routing for wireless sensor networks based on a distributed wavelet compression algorithm. In *Int. Conf. Info. Proc. in Sensor Networks (IPSN)*, pp. 309–316. Nashville, TN.

Cover, T. M. & Thomas, J. A. 1991 *Elements of information theory.* New York: Wiley.

Cristescu, R., Beferull-Lozano, B. & Vetterli, M. 2005 Networked Slepian-Wolf: Theory, algorithms, and scaling laws. *IEEE Trans. Info. Theory*, **51**(12), 4057–4073.

Dimakis, A., Kar, S., Moura, J., Rabbat, M. & Scaglione, A. 2010 Gossip algorithms for distributed signal processing. *Proc. IEEE*, **98**(11), 1847–1864.

Duarte, M. F., Wakin, M. B., Baron, D. & Baraniuk, R. G. 2006 Universal distributed sensing via random projections. In *Int. Conf. Info. Proc. in Sensor Networks (IPSN)*, pp. 177–185. Nashville, TN.

Gastpar, M., Dragotti, P. L. & Vetterli, M. 2006 The distributed Karhuenen-Loève transform. *IEEE Trans. Info. Theory*, **52**(12), 5177–5196.

Lee, S. & Ortega, A. 2010 Joint optimization of transport cost and reconstruction for spatially-localized compressed sensing in multi-hop sensor networks. In *Asia Pacific Signal and Info. Proc. Assoc. Summit (APSIPA)*. Singapore.

Lee, S., Pattem, S., Sathiamoorthy, M., Krishnamachari, B. & Ortega, A. 2009 Spatially-localized compressed sensing and routing in multi-hop sensor networks. In *Int. Conf. Geosensor Networks*, pp. 11–20. Oxford, United Kingdom.

Luo, H. & Pottie, G. J. 2007 Designing routes for source coding with explicit side information in sensor networks. *IEEE/ACM Trans. Networking*, **15**(6), 1401–1413.

Masiero, R., Quer, G., Munaretto, D., Rossi, M., Widmer, J. & Zorzi, M. 2009 Data acquisition through joint compressive sensing and principal component analysis. In *IEEE Global Telecomm. Conf. (GLOBECOM)*. Honolulu, HI.

Narang, S., Shen, G. & Ortega, A. 2010 Unidirectional graph-based wavelet transforms for efficient data gathering in sensor networks. In *IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 2902–2905. Dallas, TX.

Pattem, S., Krishnamachari, B. & Govindan, R. 2004 The impact of spatial correlation on routing with compression in wireless sensor networks. In *Int. Conf. Info. Proc. in Sensor Networks (IPSN)*, pp. 28–35. Berkeley, CA.

Pattem, S., Shen, G., Chen, Y., Krishnamachari, B. & Ortega, A. 2009 SenZip: An architecture for distributed en-route compression in wireless sensor networks. In *Workshop on Sensor Networks for Earth and Space Science Applications (ESSA)*. San Francisco, CA.

Rabbat, M., Haupt, J., Singh, A. & Nowak, R. 2006 Decentralized compression and predistribution via randomized gossiping. In *Int. Conf. Info. Proc. in Sensor Networks (IPSN)*, pp. 51–59. Nashville, TN.

Santini, S. & Romer, K. 2006 An adaptive strategy for quality-based data reduction in wireless sensor networks. In *Int. Conf. Networked Sensing Systems (INSS)*, pp. 29–36. Chicago, IL.

Scaglione, A. & Servetto, S. 2005 On the interdependence of routing and data compression in multi-hop sensor networks. *Wireless Networks*, **11**, 149–160.

Shen, G., Narang, S. K. & Ortega, A. 2009*a* Adaptive distributed transforms for irregularly sampled wireless sensor networks. In *IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 2225–2228. Taipei, Taiwan.

Shen, G. & Ortega, A. 2008*a* Joint routing and 2D transform optimization for irregular sensor network grids using wavelet lifting. In *Int. Conf. Info. Proc. in Sensor Networks (IPSN)*, pp. 183–194. Saint Louis, MO.

Shen, G. & Ortega, A. 2008*b* Optimized distributed 2D transforms for irregularly sampled sensor network grids using wavelet lifting. In *IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 2513–2516. Las Vegas, NV.

Shen, G. & Ortega, A. 2010 Transform-based distributed data gathering. *IEEE Trans. Signal Processing*, **58**(7), 3802–3815.

Shen, G., Pattem, S. & Ortega, A. 2009*b* Energy-efficient graph-based wavelets for distributed coding in wireless sensor networks. In *IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 2253–2256. Taipei, Taiwan.

Song, W.-Z., Shirazi, B., Huang, R., Xu, M., Peterson, N., LaHusen, R., Pallister, J., Dzurisin, D., Moran, S. *et al.* 2010 Optimized autonomous space in-situ sensor web for volcano monitoring. *IEEE J. Selected Topics in Applied Earth Observations and Remote Sensing*, **3**(4), 541–546.

Sweldens, W. 1996 The lifting scheme: A custom-design construction of biorthogonal wavelets. *Appl. Comput. Harmon. Anal.*, **3**(2), 186–200.

Wagner, R., Baraniuk, R. G., Du, S., Johnson, D. B. & Cohen, A. 2006 An architecture for distributed wavelet analysis and processing in sensor networks. In *Int. Conf. Info. Proc. in Sensor Networks (IPSN)*, pp. 243–250. Nashville, TN.

Wang, W., Garofalakis, M. & Ramchandran, K. 2007 Distributed sparse random projections for refinable approximation. In *Int. Conf. Info. Proc. in Sensor Networks (IPSN)*, pp. 331–339. Cambridge, MA.