Reordering for Better Compressibility: Efficient Spatial Sampling in Wireless Sensor Networks

Mohammadreza Mahmudimanesh, Abdelmajid Khelil, Neeraj Suri
Technische Universiät Darmstadt
Hochschulstr. 10, 64289 Darmstadt, Germany
{mahmudimanesh,khelil,suri}@cs.tu-darmstadt.de
Tel. +49-6151-163414, Fax. +49-6151-164310

Abstract—**Compressed Sensing (CS) is a novel sampling paradigm that tries to take data-compression concepts down to the sampling layer of a sensory system. It states that discrete compressible signals are recoverable from sub-sampled data, when the data vector is acquired by a special linear transform of the original discrete signal vector. Distributed sampling problems especially in Wireless Sensor Networks (WSN) are good candidates to apply CS and increase sensing efficiency without sacrificing accuracy. In this paper, we discuss how to reorder the samples of a discrete spatial signal vector by defining an alternative permutation of the sensor nodes (SN). Accordingly, we propose a method to enhance CS in WSN through improving signal compressibility by finding a sub-optimal permutation of the SNs. Permutation doesn't involve physical relocation of the SNs. It is a reordering function computed at the sink to gain a more compressible view of the spatial signal. We show that sub-optimal reordering stably maintains a more compressible view of the signal until the state of the environment changes so that another up-to-date reordering has to be computed. Our method can increase signal reconstruction accuracy at the same spatial sampling rate, or recover the state of the operational environment with the same quality at lower spatial sampling rate. Sub-sampling takes place during the interval that our reordered version of the spatial signal remains more compressible than the original signal.**

Compressive Wireless Sensing; Spatial Sampling; Reordering; Permutation; Compressibility; Compressed Sensing;

I. INTRODUCTION

Background: Sampling is the process of acquiring physical parameters and representing them in digital form. The result of sampling is a vast amount of raw data that can be stored or transmitted via a communication network for further processing [1]. The term signal is used to address the physical signals that can be acquired by a digital sensory system. For example, a sound wave illustrates a 1-dimensional signal which shows the change of sound amplitude over time. An image is a 2-dimensional spatial signal defined as the light intensity sensed by photocells of a digital camera. A Wireless Sensor Network (WSN) designed to sense the temperature of an environment, gives another spatial discrete signal containing thousands of temperature samples wirelessly reported by sensor nodes (SNs) to a base station, called the sink. Here we view the signal sensed by a WSN as a vector whose elements are the physical parameter values at the points where SNs are located. Therefore, WSN senses a 1-dimensional signal which can be mapped to a 2-dimmensional rectangular environment with the help of information from sensors localization [2][8].

A WSN has to sample the physical parameter(s) of its operational environment at discrete points where SNs are located. Therefore, in general WSN can be viewed as a distributed *spatially* discrete sampling problem (Note that the signal is also temporally discrete, because WSN samples the environment at discrete time instances). From now onwards, we only discuss discrete signals. We view a discrete signal sensed by a WSN as a vector consisting of n elements (n is the number of SNs). Every element is the value sensed by an individual SN. Under a suitable linear transform of a discrete signal such as the Fourier Transform (FT) [3] or Discrete Cosine Transform (DCT) [4], one can expect that most of the energy of the signal is concentrated on a few Fourier/DCT coefficients [5]. It is known that physical signals sampled from natural phenomena are band-limited, i.e., their spectral representation is sparse. Consequently, the spectral representation (such as representation under FT or DCT) of almost all physical signals has a few relatively large coefficients and many other small (nearly zero) coefficients (that is DCT or FT coefficients). This is the main logic behind compression techniques. A sparser representation of a signal is easier to be densely encoded. Because most of such compression techniques apply a linear transform such as FT or DCT at the very initial step before encoding, they are also called *transform-compression* techniques [6].

The Problem Area & Current Approaches: Transform-compression methods are very efficient, but they are useful only when all the samples are centrally available and can be accessed simultaneously to apply linear transformation. This makes transform-compression inefficient when acquiring the samples is cost-intensive. WSN is such an example where getting each sample requires not only the energy consumed by the active sensing node, but also all the SNs along the route from the source to the sink. As WSN nodes typically have very limited energy hence, it is critical to gather sensor data by the lowest sampling rate possible.

Compressive Sampling [7] or Compressed Sensing (CS) overcomes the limitations of transform-compression and proposes a sampling method which acquires the samples at a lower rate. CS tries to acquire only the samples that are necessary in constructing the original signal and to avoid extra sampling when the signal is sparse or compressible. CS states that it is possible to reconstruct a discrete signal from a set of randomly chosen values selected from a vector computed by a linear transform of the discrete signal vector. The reconstruction takes place by finding the solution to a convex optimization problem [11]. CS is a vital sampling method for situations where acquiring individual samples is very expensive such as in WSN.

Paper Contributions: In this paper, we propose a novel enhancement to CS in WSN. Our improvement does not change the basics of CS or its WSN implementation. Consequently, the proposed technique can apply to all the current achievements of CS [15][16] as a complementary or enhancement method. We show that if we view the signal vector under a *reordering (mapping) function*, it is possible to obtain a more compressible signal, i.e., a signal which is sparser in a certain domain such as FT or DCT. In brief our work offers a new approach to the WSN sampling problem by enhancing the CS basis. The main contributions of this paper can be categorized as follows: (I) A relatively fast and simple polynomial-time algorithm that finds a permutation of samples of the discrete signal vector *f*, so that the linear transform of *f* to frequency domain is sparser than the original ordering of samples, (II) proposing an enhanced Compressive Wireless Sensing (CWS) [16] model which is capable of adapting itself to the environment changes. When the state of the environment doesn't change quickly, our model is capable of reducing spatial sampling rate through constructing a more compressible view of the spatial signal, and (III) evaluation of our algorithm and enhanced CWS model on static and dynamic environments in different situations and varied WSN configurations. Our evaluations show that: (a) The improved permutation found for the spatial signal vector *f* representing the current state of the environment, keeps being superior to the original permutation in very next sampling intervals, and (b) our methodology is valid for many varied WSN instances.

The paper is structured as follows. Section II describes our system model. In Section III, we present a brief introduction to CS, define basic terms and formulate our objectives. Section IV refers to existing work related to our methodology. Precise problem formulation is described in Section V. Our novel ordering technique to enhance CS in WSN is presented in Section VI. We evaluate our solution using simulation and numerical experiments in Section VII. We conclude our work and give future directions in Section VIII.

II. SYSTEM MODEL

We consider the standard WSN model composed of *n* static SNs randomly distributed in the area of interest. SNs are equipped with short range radios, and rely on batteries and limited processing resources. SNs sense and report environmental phenomena such as the air temperature at their position. SNs send their samples via multihop routing to a dedicated node called the sink. SNs are located on WSN area using localisation methods [8], and their position is fixed through the sampling and virtual environment reconstruction process. Alternatively, individual SNs collectively define a discrete one-dimensional signal vector and the position of each SN maps the corresponding element of the vector to a point on the WSN area. This also corresponds to the CS view of the WSN.

The sink has two main responsibilities: 1) Reconstructing the original signal from compressively sampled data. and 2) Computing sub-optimal reordering of the SNs with sparser DCT (or FT) representation. We assume the sink to be powerful in processing, i.e., computation and storage. Reconstructing the original signal from undersampled data is done by performing a search in the *n*-dimensional vector-space that matches the sampled data vector [9]. Finding an improved view of the SNs through reordering also acquires plenty amount of processing. These computations should be performed in a reasonable amount of time to keep the WSN responsive enough.

III. ENHANCED CS THROUGH SPARSITY IMPROVEMENT

CS [7] states that if a discrete signal vector *f* of size *n* is sparse (having many near-zero samples and only a few meaningful relatively large elements), it is possible to reconstruct the signal from *m* randomly selected samples produced by a suitably-chosen linear transform Φ of *f* where $m << n$. In other words we can reconstruct signal $f_{n\times 1}$ from $y_{m\times 1} = \Phi_{m\times n} f$ if the signal *f* is sparse enough and Φ satisfies some preconditions dictated by the CS theory. Vector *f* is *S*-Sparse if it has at most *S* nonzero elements. In reality we usually deal with non-sparse (in time/space domain) signals. CS also shows that even if the signal itself is not sparse but can be sparsely presented in another domain named the Ψ-domain, it is still possible to reconstruct *f* with an overwhelming (asymptotically 1) probability from *m* samples [10] when:

$$m \geq C \cdot \mu^2(\Phi, \Psi) \cdot S \cdot \log(n) \tag{1}$$

where $C > 1$ is a small constant, and *S* is the number of non-zero elements in Ψf and μ is the *coherence* [10] between the sampling domain (Φ-domain) and the sparse representation domain (Ψ-domain):

$$\mu(\Phi, \Psi) = \max_{1 \leq k, j \leq n} \left| \varphi_k \cdot \psi_j \right| \tag{2}$$

where φ_k's and ψ_j's are basis vectors of Φ- and Ψ-domains respectively. Note that the Ψ-domain must be chosen so that the projection of *f* under Ψ-transform results in a sparse vector. In most cases signals sampled from the nature are quite sparse in DCT or FT domain [3]. The projection of a signal vector sampled from a natural phenomenon on to the frequency-domain may not be actually sparse; instead it may have many negligible coefficients. We define a *compressible* discrete signal *f* as a vector whose Ψ-transform has mostly near-zero elements.

The *m* samples are selected uniformly at random from $\Phi\Psi f$. It can be shown that if Φ is an orthogonal matrix uniformly sampled from unit sphere, then it has low coherence with any orthogonal representation basis Ψ [6]. Signal reconstruction from compressed samples is performed by solving the following convex optimization problem [11]:

$$\forall x \in \Re^n \ \min \|x\|_{l_1} \ \text{subject to} \ \forall k \in M \ \ y_k = \langle \varphi_k, x \rangle \tag{3}$$

where $M \subset \{1,2,3,\ldots,n\}$ and $\forall k \in M \;\; y_k = \langle \phi_k, \Psi f \rangle$ which the last term means the projection of Ψf on φ_k. It can be shown that this convex optimization problem can be cast to a linear program [12]. In simple words, the program (3) tries to find the vector x in Ψ-domain with the least l_1-norm that is consistent with the sampled vector y [13][14]. If the solution to program (3) is denoted by x^* then the recovered vector will be $f^* = \Psi^{-1} x^*$. If we define x_S as a sparse version of vector x which has only its largest S elements and all its other elements are set to zero, one can suffer no much more error than $\|x - x_S\|_{l_2}$ if m in Eq. (1) is determined assuming the signal is S-sparse – having only S non-zero elements.

In most distributed CS applications, sampling and sparse bases are determined prior to the deployment of the sensor network. Therefore, according to Eq. (2), μ in Eq. (1) can be supposed to be constant because Φ- and Ψ-domains likely do not change after network deployment. Therefore, S is the only parameter that is effective on the minimum number of required samples m. If we succeed to decrease S, then we can reconstruct the original signal from fewer samples, saving valuable bandwidth and SNs energy.

In some sampling problems such as WSN, the ordering of the samples is not dictated by an independent phenomenon. Mostly, we can assign the value sensed by one sensor to the first signal element and the other to the second one, and so on. This paper focuses on such conditions where we can determine the sampling order. It is important to differentiate the *position* and *order* of SNs here. Reordering only takes place at the sink and all its required computations are done outside the WSN. That's why our proposed model doesn't add overhead to the WSN nodes. In simple words, reordering is an alternative *view* of our WSN signal vector under which we can apply CS more efficiently.

Our enhancement works by improving compressibility, that means increasing sparsity of *f* under Ψ-transform, which means decreasing S in Eq. (1). Then, from Eq. (1) it implies that a view of the signal that makes it appear to be more compressible, leads to lower compressive sampling rate required for signal reconstruction. Lower compressive sampling rate means more efficient bandwidth usage and decreased energy consumption prolonging WSN lifetime. In simple words: Energy consumption ~ Data transmission rate ~ Compressive sampling rate m ~ S ~ 1/Compressiblity. Compressibility is *virtually* increased by finding a mapping under which f is sparser in Ψ-domain, which is our main contribution in this paper.

IV. Related Work

CWS [16] by Bajwa et al. discusses how to compressively sample the discrete signal vector *f*. Their approach does not address the adaptation of CS to dynamic phenomena and network conditions. In this work, we allow for CWS adaptation by designing our sub-optimal ordering technique.

According to Eq. (1), reordering the vector y in such a way that leads to a more compressible representation of the signal vector, decreases the minimum sampling rate to reconstruct the original signal vector *f*. In our initiative work [17] we have benchmarked our intuitive model for a small WSN with only eight sensors, for lack of an algorithm that finds an improved permutation in reasonable processing time. Our previous work using an exhaustive search for the optimal permutation showed that if the order of samples is not strict, we can reorder the samples to make it into a more compressible discrete signal vector. Compressibility is of great importance to get better results in signal reconstruction from compressively sampled signals. In this paper, we present a greedy algorithm that finds a sub optimal permutation in polynomial time. We will show that a good reordering of the samples, can drastically improve compressibility. This novel work also considers dynamic environments. With the help of our new polynomial-time sub-optimal reordering algorithm, we can evaluate our methodology in varied and more realistic conditions. Our new evaluations on dynamic environments justify our previous results in [17] while turning many of its hypotheses into solid principles.

Adaptive sub-Nyquist sampling techniques are not a new topic in WSN. Backcating [18] is a sampling technique for WSNs which adapts the spatial sampling rate (and hence energy consumption) to the current state of the operational environment and accuracy requirements of the user. This approach proposes a tree-shaped hierarchical sampling configuration with the SNs as leaves. Regions with more important data (higher spatial frequency) are densely populated by active SNs. On the other hand, fewer SNs located in regions with more redundancy (less spatial frequency) are activated. Data reported by SNs travel the hierarchical communication tree up to the sink. Although Backcasting is an adaptive technique according to region of importance and can reduce extra sampling, it limits itself to a fixed topology and may not well extend to random distribution of SNs. CS offers a more general method and is less constrained to preconditions as required by Backcasting. This makes CS quite universal and very independent of network and environment specifications. Therefore, our enhanced CS through compressibility improvement is quite extensible.

V. Problem Formulation

WSN can operate in two modes that we name *full* mode and *half* mode. In full mode, from our previous knowledge of the operational environment, we determine an expected value for sparsity of the signal's Ψ-transform, namely S_e with which the expected minimum CWS sampling rate (number of discrete samples) m_e is calculated (by substituting S with S_e in Eq. (1)). We acquire m_e samples according to the methods discussed in CWS [16]. In half mode, we calculate a sub-optimal reordering of the signal at the sink. Then the sink broadcasts a new version of the sampling matrix $\Phi_{m \times n}$ with fewer rows, as the minimum required samples (m) is reduced because signal's sparsity is improved through our reordering.

Without loosing generality, we only limit our configuration to Φ being an orthogonal Gaussian random basis and Ψ to be the Fourier domain. The conclusions of this work also apply to any other pair of orthogonal incoherent sampling and sparse bases. One can apply this method to other bases such DCT or wavelets. In fact the method discussed here applies to any other transformation (projection) bases which can be presented in matrix multiplication form. For example one can assign the DCT matrix to be Ψ, and find the sub-optimal reordering as described by our method detailed in this section. It is important to note that here we simplify a pure combinatorial problem (Subsection V.A) to a more tractable linear algebra problem and find a solution to the simplified translation (Subsection VB) of the original problem.

A. Combinatorial Problem Statement

Let π be a permutation function defined as below

$$\begin{aligned} &\pi : \{1,\dots,n\} \rightarrow \{1,\dots,n\} \\ &\forall i, j \in \{1,\dots,n\} : i = j \Rightarrow \pi(i) = \pi(j) \\ &\forall i \in \{1,\dots,n\} \, \exists \, j \in \{1,\dots,n\} \, \pi(j) = i \end{aligned} \tag{4}$$

We define *reordering* of vector f under function π as f_π:

$$\forall i \in \{1,\dots,n\} \;\; f_\pi(i) = f(\pi(i)) \tag{5}$$

Now let's define Π to be the set of all possible mutually different permutation functions π. The optimal permutation function π^* for vector f under a transformation matrix Ψ is the one that among all $n!$ possible π's leads to the sparsest representation of f in the Ψ-domain:

$$\forall \pi \in \Pi \;\; \pi \neq \pi^* : \quad \Psi f_{\pi^*} \text{ is sparser than } \Psi f_\pi \tag{6}$$

As mentioned before, we can apply our approximation method and assume compressible vectors to be sparse in Ψ-domain by zeroing negligible elements of f.

B. Condensing the Energy of the Signal

Fourier transformation projects a vector sampled in time- or space-domain to the frequency domain. It can be shown that Discrete Fourier Transform (DFT) is in fact a rotation and scaling of the discrete signal vector. Therefore, if we rearrange the signal vector so that its FT has much of the energy of the signal concentrated on few frequency coefficients we may expect that we can have a sparser representation of the signal in Fourier domain. The inverse projection of the unit vectors in the Fourier domain to the time/space domain, gives us some vectors in the sampling domain. If we find an order of the samples that best matches one of these basis vectors, we can expect that the FT of the optimally reordered sample vector has most of its energy concentrated on a few Fourier coefficients. If we assume Ψ to be the FT matrix, then the inverse projected basis vectors in the sampling domain will be the columns of Ψ^{-1} (this is also valid for any other transform which can be represented as matrix multiplication). For each basis vector we can find the optimal permutation of the samples, resulting in n different permutation functions for each of the n basis vectors. Among these we choose the permutation that matches a basis vector better than the others.

In our model f is the discrete spatial vector whose each element is the real value sensed by corresponding SN. Initially, the elements of f are ordered simply by SNs id's. Our objective is to find a permutation π^* under which Ψ-transform of f is sparser. Now for each basis vector ψ we construct *difference matrix* Δ as below:

$$\Delta_{n\times n} = \begin{bmatrix} |f_1 - \psi_1| & |f_1 - \psi_2| & \cdots & |f_1 - \psi_n| \\ |f_2 - \psi_1| & |f_2 - \psi_2| & \cdots & |f_2 - \psi_n| \\ \vdots & \vdots & \ddots & \vdots \\ |f_n - \psi_1| & |f_n - \psi_2| & \cdots & |f_n - \psi_n| \end{bmatrix} \tag{7}$$

We define two vectors with least difference as most matching. Therefore, finding π^* in Eq. (6) is equivalent to finding n elements of Δ with minimum sum, so that no two of them are on the same row or column.

VI. Sub-Optimal Reordering for Enhanced WSN CS

In the following, we introduce a simple greedy algorithm that finds such n elements in $O(n^2 \log n)$ which is slightly faster than Edmonds-Karp method, making it more desirable for running against many basis vectors. Next, we show how we use our algorithm to improve compressibility. Note that finding n constrained elements from Δ with the globally minimum sum requires $O(n^3)$ processing time [19]. Even for some hundreds of SNs, it takes much time to compute the optimal permutation. Remember that we have to find such n elements with minimum sum against many basis vectors. Our algorithm is simple to implement and requires less computation time. This allows us to run our simulations over and over and generally investigate the performance of our methodology. Here we focus only on the effect of reordering on CWS performance. Even this simple reordering algorithm enhances CWS drastically. As a future improvement we can work on better algorithms that find a globally optimal permutation in a practical amount of time.

A. Greedy Approximate Solution

Here we introduce our algorithm (Algorithm 1) for finding a sub-optimal permutation of the samples, called the *Sub-Optimal Permutation (SOPerm)*. Note that this algorithm only compares two unit vectors (or two vectors of the same scale) and outputs a sub-optimal permutation to project the first vector on the second. Here ψ's are the unit vectors of the sparse representation domain and f is our discrete signal vector. The basis vectors (ψ's) are fixed and we have to find a permutation for f which is sub-optimal. We implicitly suppose that f and ψ are of equal scale, i.e. $\|f\|_{l_2} = \|\psi\|_{l_2}$. Even if f and ψ are not equal in scale, we can use their normalized versions in our proposed method and find the optimal permutation. Obviously, an optimal permutation found for normalized versions of f and ψ is also optimal for the original vectors f and ψ. SOPerm first constructs the matrix Δ. Then, SOPerm runs n steps and at each step greedily selects the element which is not on a row or column of a previously selected element and the addition of that element to the sum of previously selected elements is minimum. This is equal to adding simply the smallest non-constrained element of Δ, because all the elements of Δ are positive real numbers.

In Algorithm 1, Lines 3-7 simply constructs the matrix Δ. Line 8 initializes the set of visited rows and columns (R and C respectively) to null. Our greedy approach is implemented in Lines 9-16. At each iterate of the for-loop an element at the ith row and jth column of Δ with minimum value is selected so that i doesn't exist in R and j doesn't exist in C. At Line 14 inside the for-loop the ith row and the jth column are marked as visited by appending them to the sets R and C respectively.

Algorithm 1: Sub-Optimal Permutation (SOPerm)

```
1   [π , σ] = Sub-Optimal_Permutation (f, ψ)
2   begin
3       for i ← 1 to n do
4           begin
5               for j ← 1 to n do
6                   Δ_{i,j} = |f_i − ψ_j|
7           end
8       C ← φ, R ← φ, σ ← 0
9       for k ← 1 to n do
10          begin
11              select i ∈ {1,2,3,...,n} − R  and
12                  j ∈ {1,2,3,...,n} − C  so that
13                  Δ_{i,j}  is minimum among all i,j;
14              R ← R ∪ {i},  C ← C ∪ {i}, σ ← σ + Δ_{i,j}
15              π_j ← i
16          end
17  end
```

Because the spatial FT/DCT of a natural environment is supposed to have only low frequencies, we expect to find the sub-optimal permutation among low frequency basis vectors. Our simulations justify this hypothesis, as searching in higher frequencies resulted no better permutations than the one found among low frequencies. Therefore, to decrease processing time our SOPerm algorithm is run against lower frequency basis vectors.

B. Using Reordering to Tune Distributed CS

So far we have presented a practical method to redefine a model of the WSN under which we can improve signal sparsity and hence decrease compressive sampling rate. The SOPerm algorithm will be implemented on the sink (outside the WSN). SOPerm tries to find a reordering of SNs that leads to a sparser discrete spatial signal. For simplicity, we consider that the spatial CS takes place on a regular basis every T time units (Adapting T to the dynamics of the signal and characteristics of the operational environment is meaningful but not the scope of this work). According to the CWS model by Bajwa et al., we assume that there are efficient methods to acquire the compressively sampled vector y from the original spatial signal f at each time. Vector y is acquired by all the SNs in a distributed fashion [15][16]. The sink reconstructs the original signal by solving program (3). Note that at the initial point, we set the minimum required samples for reconstruction, to be m_e calculated according to Eq. (1) with presumption of signal f being at most S_e-sparse. SOPerm is then run for the reconstructed signal f^* which is almost exactly equal to f. SOPerm calculates a sub-optimal permutation π^* under which vector f is S-sparse and $S < S_e$. For the next sampling round, we use this permutation and sample at a rate at least $m = C \cdot \mu^2(\Phi, \Psi) \cdot S \cdot \log n < m_e$. We believe that π is still a good reordering for next T time units because the environment doesn't change so quickly over time. We emphasize again that our model applies only to WSNs with fixed SNs in an environment with moderate changes over time. Detecting events (quick impacts of the environment) in WSN by CS is another topic [20] which is not in the scope of this paper. Our model enhances reconstruction techniques of the whole environment based on CWS. We can sample at rate $m + r$ with $r << m$ (that means sampling rate is little bit higher than m) to ensure that we acquire enough compressive samples to reconstruct the original signal. In the subsequent sampling rounds and with our new reordering, we can reconstruct the spatial signal f from fewer compressively sampled data vector y and repeat all this process for the following sampling round.

Figure 1 illustrates the overall architecture of our adaptive reordering model. Our model is in fact a closed-loop system that applies CWS every T time units and updates its internal model according to the changes in the operational environment (for example air temperature). Because m changes over time, the sampling matrix $\Phi_{m \times n}$ has to be broadcast to all SNs before each sampling round. This can be infeasible for very large networks, but we have to use broadcasting as our publishing method of sampling matrix because our system requires flexibility against environmental changes. As an alternative, one can find a way to publish only the integer number m and then SNs shall compute their own individual columns of Φ according to a predetermined seed for generating a pseudo-random sampling matrix which can be also computed exactly at the sink.

As illustrated in Figure 1, our approach starts with a presumption about the operational environment and begins its CWS operation normally in full mode. At first no reordering is performed, i.e., $\pi = (1,2,3,\ldots,n)$. After reconstructing signal from compressively sampled data vector, we apply SOPerm to improve signal sparsity under the new reordering π^*. This new permutation may lead to a sparser signal vector in Ψ-domain (FT or DCT domain). The parameter λ (Figure 1) determines how many unit vectors from Ψ are compared by the SOPerm algorithm. As discussed in Subsection VI.A, we can find a sub-optimal permutation among very first unit vectors of the Ψ-domain because the natural operational environment is expected to have only low frequency coefficients in its Ψ-transform. The lower the parameter λ, the faster our approach computes sub-optimal reordering. The higher the parameter λ, the higher is the chance to find a better permutation. However, in search for matching to unit basis vectors among high frequencies, it is less probable to find a better permutation than the one we have found among the lower frequency unit basis vectors. Parameter $\eta > 0$ is a threshold which controls that with how much precision we consider a signal to be sparse in Ψ-domain. The smaller the threshold η, the more accurate is our CWS reconstruction, but at the cost of higher sampling rate. In each cycle, the current reconstructed state of the environment is displayed to the user by mapping back the elements of f^*_π

under the inverse permutation indexing vector π^{-1}, i.e., $f^* = \left(f_\pi^*\right)_{\pi^{-1}}$.

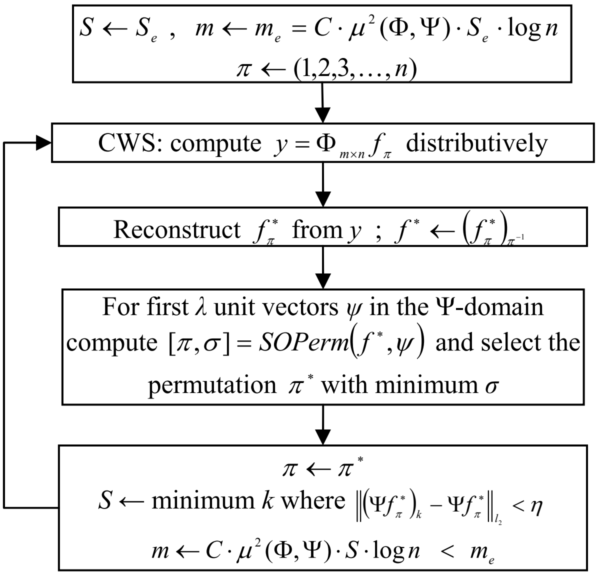


Figure 1. Enhanced CWS model with reordering

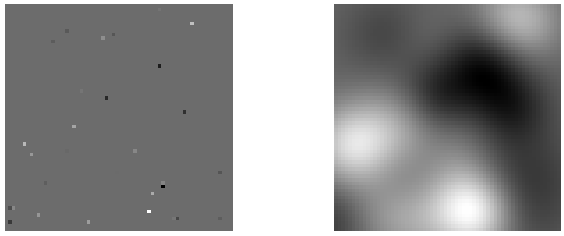
VII. Performance Evaluation

To evaluate our adaptive CS for WSN, we use simulations. After detailing the settings, we present the evaluation results of our reordering algorithms.

A. *Simulation Environment*

Our simulations are done in MATLAB. The simulation environment is defined as a rectangular area where a set of SNs are randomly distributed. Our simulation only evaluates the performance of reordering on sparsity of the signal and doesn't consider every details of the communication protocol. Our prototype scripts are in fact numerical experiments that show how the sparsity of the spatial signal is enhanced under reordering found by SOPerm.

Due to lack of appropriate real world values of real physical phenomena, we simulate the behaviour of a natural environment and work with the generated synthetic data. As a proof of concept, we consider the air temperature of individual points on a rectangular area. We can view the temperature map of the environment as a 2-D greyscale picture. To construct such a 2-D image of the temperature map that behaves much like a real temperature map we need to construct a picture whose pixel intensities don't vary sharply along x or y axis. The following steps show how we construct such a picture with smooth variations of pixels intensity. At the first step we paint all the pixels with intensity 0. The second step is to set some points of the image to random values other than zero. In our implementation of the environment simulator, we choose some points on the image, and set them a random real number chosen from a Gaussian random distribution with zero mean and variance 1.0. Finally, we apply a 2-D Gaussian filter on the resulting image. Figure 2 depicts our method for a 64×64 image. Figure 2(a) shows the random *guiding* points. Figure 2(b) shows the final simulated operational environment after applying a Gaussian filter on Figure 2(a).

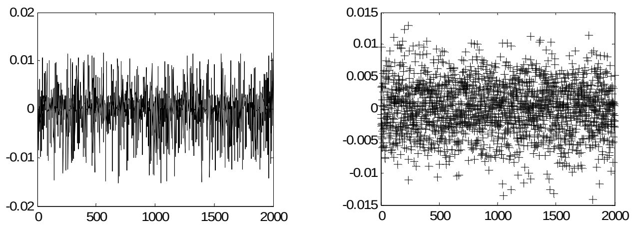


(a) Random points on the temperature map (b) Gaussian filter applied on (a)

Figure 2. Generation of synthetic data for the signal (here air temperature).

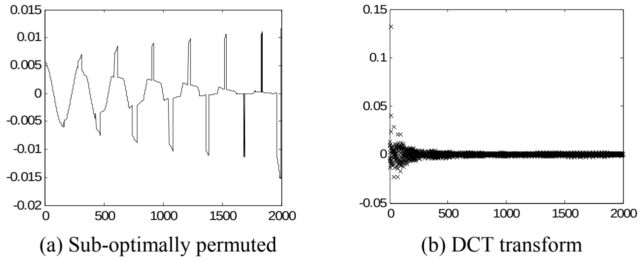
B. *Impact of Reordering on Signal Compressibility*

SNs are distributed randomly on the surface of a temperature map like Figure 2(b). Signal vector *f* is composed of the intensity of the pixels located at the points where SNs are situated. Now our SOPerm algorithm comes into play. The SOPerm algorithm gives us a reordering of *f* so that its DCT transform is sparser. Note that in our numerical experiments we have chosen DCT as our Ψ-domain for simplicity, to avoid dealing with imaginary parts of complex numbers in our charts and diagrams. In another simulation discussed later in this section we use FT as the transformation matrix which has complex values to evaluate SOPerm on transformation matrices with imaginary parts. Figure 3(a) shows the original discrete signal *f* of size 2000 composed of the sensed temperature values according to permutation of samples ordered by the original index (id) of the SNs. Figure 3(b) depicts the DCT transform of the actual vector *f*. Then, we apply SOPerm on *f* with regard to the first 20 unit basis vectors in the DCT system, and finally we choose the best permutation among the 20 permutations found by the SOPerm runs. Let's call the sub-optimal permutation as π^*. Figure 4(a) shows the sub-optimally reordered vector f_{π^*} computed by our SOPerm algorithm and Figure 4(b) depicts its DCT transform. It is apparent that f_{π^*} is much sparser in the DCT domain than *f*.



(a) Indexed by SN id's (b) DCT transform

Figure 3. Original signal



(a) Sub-optimally permuted (b) DCT transform

Figure 4. Signal after re-ordering

Sampling interval T depends on the configuration of our network and the characteristics of our environment. We have to consider also the time required to update sampling matrix Φ. With a rough estimate we may require about 100 seconds for updating and data gathering (performing one iteration of the closed loop represented in Figure 1) for network consisting of 1000 SNs (assuming we have a 256kbp broadcast channel). This may seem too long for catching real-time events. But as we have mentioned earlier our model only deals with reconstructing the state of environment with highest accuracy while consuming least energy. Event detection [20] can operate in parallel with our model by embedding another sampling matrix specialized for event detection, making our WSN into a hybrid state-monitoring and event-detecting system.

C. Impact of Reordering for Dynamic Physical Phenomena

Now, we show that sub-optimal permutation computed for a discrete spatial signal at a specific moment, can be still sub-optimal for upcoming moments in future, if the state of the environment doesn't change very quickly over time. To simulate such a dynamic environment, we have upgraded our synthetic environment described in Section VII.A to support changes in time. We have moved the random points (Figure 2(a)) over the image randomly to different directions. Simultaneously, we pass the image through a Gaussian filter to keep realistic-appearing distribution of temperature that changes over time. Figure 5 shows the outcome of such a synthesized animated image. In Figure 5, the points move around the rectangular image randomly in each direction to maximum 10 pixels away. This small steps causes that the environment doesn't change too quickly over time. In Figure 5 the environment changes in the sequence specified by the directed arrows. We have run the simulation for 64 time periods and the images shown in Figure 5 are actually 8 snapshots taken every 8 time units. During all this time, 200 SNs are compressively reporting the value of the physical parameter under the points they are located. Therefore, at each time instance we have a different spatial signal vector. For each vector, we run SOPerm with regard to DCT domain and find a sub-optimal permutation. We keep a history of previously computed sub-optimal permutations. Figure 6 shows how old permutations may still lead to sparser DCT transform of the signal at the current time. Equivalently, this means that a sub-optimal permutation at a specific time is still sub-optimal for the following time instances.

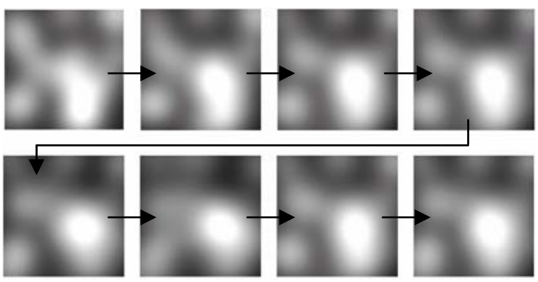


Figure 5. Dynamic synthesized simulation environment

Figure 6 shows that the sparsity of the signal with its normal order (denoted by □) stands always lower than sub-optimal permuted signal vector and the signal vector reordered according to previously calculated sub-optimal permutations. At each time instance t, SOPerm is applied on the signal vector f_t and resulted π_t as a sub-optimal permutation for the signal at instance t. Sparsity of $(f_t)_{\pi_t}$ is depicted with (*) on Figure 6. We have stored the indexing permutation vectors $\pi_1, \pi_2, \ldots, \pi_{64}$ and then applied them on the final signal vector f_{64}. This means we have calculated the sparsity of f_{64} under $\pi_1, \pi_2, \ldots, \pi_{64}$. Sparsity of the signal vectors $(f_{64})_{\pi_1}, (f_{64})_{\pi_2}, \ldots (f_{64})_{\pi_{64}}$ is depicted with $\triangle$ on Figure 6.

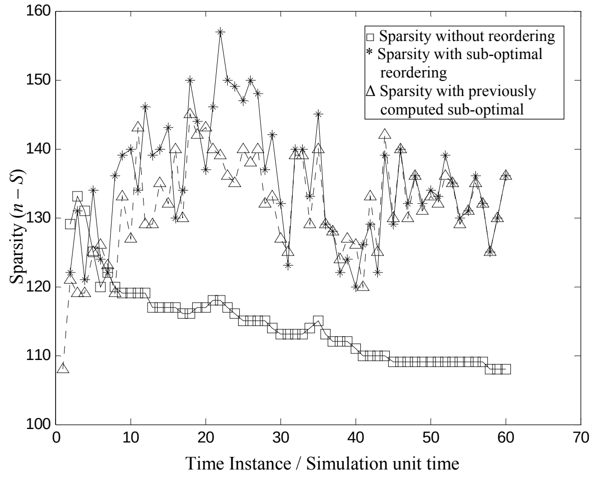


Figure 6. Change of signal sparsity in time

The interesting result is that previously computed sub-optimal permutations still make the signal sparser even for signal vectors at times later than the instance they have been computed. However, as it can be seen on the diagram, outdated permutations (previously computed π_t where $t < 30$) don't provide a good reordering anymore. As we approach closer to the present time ($t > 30$ till $t = 64$) we see that previously computed π_t's work as well as the SOPerm permutation which is especially calculated for f_{64}.

D. Impact of Reordering in different WSNs

So far we have run our simulations only for a fixed WSN. Now we vary the number of nodes ($200 \le n \le 800$), the most relevant WSN parameter for spatial sampling performance. Similarly, we first construct synthetic simulated operational environment. For each n, we randomly distribute n SNs over the 2D image representing our environment and acquire spatial signal f and then we apply SOPerm on f to see how much improvement in sparsity we can gain. For each of these WSNs, we run the same simulation 100 times and compare the averaged sparsity of discrete spatial signal f with and without using reordering by SOPerm. Figure 7 depicts the result of the experiment. The experiment is run with the Fourier domain as our Ψ-domain. This experiment also illustrates the validity of SOPerm algorithm for transformation matrices with imaginary elements. Because FT matrix contains both real and imaginary values, we can expect a decrease in sampling rate equal to half of the sparsity improvement illustrated in Figure 7. During the test, the threshold η is set to 0.01. With the reordering resulted from SOPerm algorithm, environmental spatial signal stands always sparser than the original signal without reordering. This overall experiment shows that our proposed model is safe to be used as a universal method to enhance CWS and decrease sampling rate and energy consumption while delivering the same quality of environmental signal reconstruction.

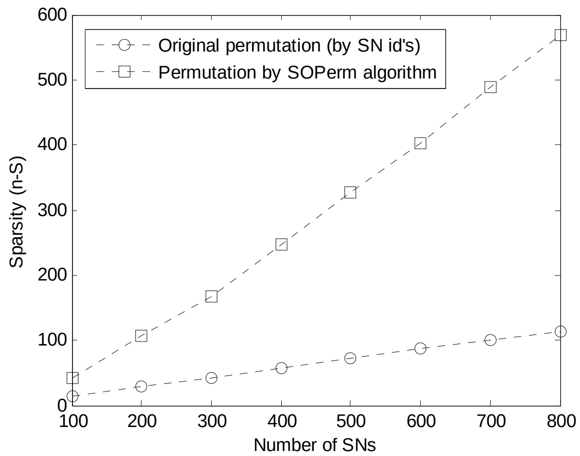


Figure 7. Comparing sparsity of spatial signal with and without reordering.

VIII. CONCLUSIONS AND FUTURE WORK

In this paper, we discussed how we can get an alternative view of a discrete signal which is more compressible. When we are not restricted to obey a predefined order of the samples, we can define an order of the samples which is more compressible (WSN senses a spatial signal vector f. But we are not forced to assume e.g. f_1 to represent the value sensed by SN with id #1 and so on for other SNs and f_i 's.). The more compressible the signal, the sparser its Fourier-, DCT-, wavelet-, etc. transform is. Compressibility is important when we want to apply CS in a sampling application where acquiring individual samples is very costly. Spatial distributed sampling such as in WSN is a suitable field to apply our method, since we can define the order of the samples arbitrarily and exploit a more compressible view of the spatial signal to gain better results from CWS. Note that in our model reordering takes place virtually at the sink rather than physical placement of SNs [21]. The permutation found by our proposed SOPerm algorithm provides sub-optimal solution for very next sampling rounds. Therefore, we can rely on the sub-optimal permutation found in the previous sampling round and apply CWS based on that permutation. We expect that we can reconstruct the signal with high quality from decreased compressive samples if the environment doesn't experience drastic changes in short time. SOPerm proved itself robust through our overall simulations and we can safely use it to enhance CWS for many different situations. The model presented in this work is a rather general framework that draws very solid but at the same time very basic initial path lines. More elaboration is needed to make it closer to a guideline for a physical implementation of an enhanced sampling methodology based on CS in WSN, optimal reordering and map-based WSN frameworks [22][23]. We are currently working on a more detailed model that measures the overall energy consumption and examines the robustness of our enhanced CS model in noisy environments.

ACKNOWLEDGMENT

Research supported in part by DFG GRK 1362 (TUD GKMM), EU INSPIRE, EU CoMiFin, and CASED (www.cased.de).

REFERENCES

[1] Elson, J. and Estrin, D. “Sensor Networks: A Bridge to the Physical World”, In Wireless Sensor Networks, C. S. Raghavendra, K. M. Sivalingam, and T. Znati, Eds. Kluwer Academic Publishers, 2004.

[2] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, “Wireless sensor networks: a survey”, Comput. Networks, vol. 38, no. 4 pp. 393-422, 2002.

[3] Bracewell, R. N. “The Fourier Transform and its Applications”, Second edition ed., New York: McGraw-Hill, 1978.

[4] N. Ahmed, T. Natarajan, and K.R. Rao, “Discrete Cosine Transfom”, IEEE Trans. on Computers, 23(1), 1974.

[5] Weaver, H. J. “Applications of Discrete and Continuous Fourier Analysis”, New York: John Wiley & Sons, 1983.

[6] E. Candès and M. Wakin, “An Introduction to Compressive Sampling”, IEEE Signal Processing Magazine, 25(2), 2008.

[7] E. Candès, “Compressive Sampling”, Int. Congress of Mathematics, 2006.

[8] G. Mao, B. Fidan, B. D.O. Anderson, “Wireless sensor network localization techniques, Computer Networks”, Volume 51, Issue 10, pp. 2529-2553, ISSN 1389-1286, 2007.

[9] E. Candès and J. Romberg, “l_1-magic: Recovery of Sparse Signals via Convex Programming”, Caltech (as Technical Notes for the L1-MAGIC software package), available online at http://www.acm.caltech.edu/l1magic/downloads/l1magic.pdf, 2005.

[10] E. Candès and J. Romberg “Sparsity and Incoherence in Compressive Sampling”, Inverse Problems, 23(3) ISSN 0266-5611, 2007.

[11] S. Boyd and L. Vandenberghe. “Convex Optimization”, Cambridge University Press, 2004.

[12] P. Bloomfield and W. Steiger, “Least Absolute Deviations: Theory, Applications, and Algorithms”, Birkhäuser, Boston, 1983.

[13] D. Donoho, “Compressed Sensing”, IEEE Trans. on Information Theory, 52(4), 2006.

[14] J. Haupt, R. Nowak, “Signal Reconstruction from Noisy Random Projections”, IEEE Trans. on Inform. Theory 52(issue?), 2006.

[15] M. F. Duarte, M.B. Wakin, D. Baron, and R.G. Baraniuk “Universal Distributed Sensing via Random Projections”, Int. Conf. on Information Processing in Sensor Networks (IPSN), 2006.

[16] W. Bajwa, J. Haupt, A. Sayeed, and R. Nowak, “Compressive Wireless Sensing”, Int. Conf. on Information Processing in Sensor Networks (IPSN), 2006.

[17] M. Mahmudimanesh, A. Khelil and N. Yazdani, “Map-Based Compressive Sensing Model for Wireless Sensor Network Architecture, A Starting Point”, In: First Int. Workshop on Wireless Sensor Networks Architectures, Simulation and Programming (WASP), 2009.

[18] R. Willett, A. Martin, and R. Nowak, “Backcasting: adaptive sampling for sensor networks”, Proceedings of the 3rd international Symposium on information Processing in Sensor Networks (IPSN '04), pp. 124-133, 2004.

[19] R. Burkard, M. Dell'Amico and Silvano Martello “Assignment Problems”, Society for Industrial and Applied Mathematics, Ch. 4, pp. 73-79, 2009.

[20] J. Meng, H. Li and Z. Han, “Sparse Event Detection in Wireless Sensor Networks using Compressive Sensing”, Conference on Information Sciences and Systems (CISS), 2009.

[21] A. Krause, C. Guestrin, A. Gupta and J. Kleinberg, “Near-optimal sensor placements: maximizing information while minimizing communication cost”, Proceedings of the 5th international conference on Information processing in sensor networks (IPSN'06) pp. 2-10, 2006.

[22] A. Khelil, F. K. Shaikh, B. Ayari and N. Suri “MWM: A Map-based World Model for Event-driven Wireless Sensor Networks”, In Proc. of The 2nd ACM International Conference on Autonomic Computing and Communication Systems (AUTONOMICS), 2008

[23] A. Khelil, F. K. Shaikh, A. Ali and N. Suri “gMAP: An Efficient Construction of Global Maps for Mobility- Assisted Wireless Sensor Networks”, In the Sixth Annual Conference on Wireless On demand Network Systems and Services (WONS), 2009.