Error-Resilient and Complexity-Constrained Distributed Coding for Large Scale Sensor Networks

Kumar Viswanatha, Sharadh Ramaswamy[*], Ankur Saxena[†] and Kenneth Rose[‡]
University of California - Santa Barbara
Santa Barbara, CA - 93106-9560, USA
{kumar,rsharadh,ankur,rose}@ece.ucsb.edu

ABSTRACT

There has been considerable interest in distributed source coding within the compression and sensor network research communities in recent years, primarily due to its potential contributions to low-power sensor networks. However, two major obstacles pose an existential threat on practical deployment of such techniques in real world sensor networks, namely, the exponential growth of decoding complexity with network size and coding rates, and the critical requirement for error-resilience given the severe channel conditions in many wireless sensor networks. Motivated by these challenges, this paper proposes a novel, unified approach for large scale, error-resilient distributed source coding, based on an optimally designed classifier-based decoding framework, where the design explicitly controls the decoding complexity. We also present a deterministic annealing (DA) based global optimization algorithm for the design due to the highly non-convex nature of the cost function, which further enhances the performance over basic greedy iterative descent technique. Simulation results on data, both synthetic and from real sensor networks, provide strong evidence that the approach opens the door to practical deployment of distributed coding in large sensor networks. It not only yields substantial gains in terms of overall distortion, compared to other state-of-the-art techniques, but also demonstrates how its decoder naturally scales to large networks while constraining the complexity, thereby enabling performance gains that increase with network size.

Categories and Subject Descriptors

E.4 [**Coding and information theory**]: Data compaction and compression, Error control codes; G.3 [**Probability and statistics**]: Probabilistic algorithms; I.4.2 [**Compression**]: Approximate methods

General Terms

Algorithms, Theory, Experimentation

Keywords

Distributed source-channel coding, Large scale sensor networks, Error resilient coding

1. INTRODUCTION AND MOTIVATION

Sensor networks have gained immense importance in recent years, both in the research community as well as in the industry, mainly due to their practicability in numerous applications. Sensors are typically low power devices and minimizing the number of transmissions is one of the primary objectives for a system designer. It is widely accepted that exploiting inter-sensor correlations to compress information is an important paradigm for such energy efficient sensor networks. The problem of encoding correlated sources in a network has conventionally been tackled in the literature from two different directions. The first approach is based on 'in-network compression' wherein the compression is performed at intermediate nodes along the route to the sink [8]. Such techniques tend to be typically wasteful in resources at all-but the last hop of the sensor network. The second approach involves 'distributed source coding' (DSC) wherein the correlations are exploited before transmission at each sensor [3].

The basic DSC setting involves multiple correlated sources (e.g., data collected by a number of spatially distributed sensors) which need to be transmitted from different locations to a central data collection unit/sink. The main objective of DSC is to exploit inter-source correlations despite the fact that each sensor source is encoded without access to other sources (see Fig. 1). The only information available before designing DSC is their joint statistics (e.g., a training dataset). Today the research in DSC can be categorized into two broad camps. First approach derives its principles from channel coding, wherein block encoding techniques are used to exploit correlation [1, 9, 18]. While these techniques are efficient in achieving good compression and error-resilience (using efficient forward error correcting codes), they suffer from significant delays and high encoding complexities, which make them unsuitable for several sensor network applications. The second approach is based on

[‡]This work was supported by the NSF under grants CCF-0728986, CCF-1016861 and CCF-1118075.

[*]S. Ramasway was with Mayachitra, Inc., USA, at the time of this work.

[†]A. Saxena is now with Samsung Telecommunications America, 1301 E. Lookout Dr., Richardson, TX, USA - 75082.

IPSN'12, April 16–20, 2012, Beijing, China.
Copyright 2012 ACM 978-1-4503-1227-1/12/04...$10.00.

source coding and quantization techniques, which introduce practically zero delay into the system. Efficient design of such zero delay DSC for noiseless systems has been studied in several publications including [4, 5, 12, 14], and will be more relevant to us in this paper.

However, two major obstacles have deterred these approaches from gaining practical significance in real world sensor networks. Firstly, the decoder complexity grows exponentially with the number of sources making these conventional techniques (typically designed for 2 - 3 sources) infeasible for large sensor networks. Surprisingly, very few researchers have so far addressed this important issue, e.g. [6, 10, 19, 20]. However most of these approaches suffer from important drawbacks which will be explained in detail in section 3.

The second important reason for inefficiency of current DSC methods is the fact that sensor networks usually operate at highly adverse channel conditions and codes designed for a noise-less framework provide extremely poor error-resilience. The design of such error-resilient DSC is a very challenging problem, as the objectives of DSC and channel coding are counter-active in the sense that one tries to eliminate dependencies, while the other tries to correct errors using the dependencies. On the one hand, the system could be made compression centric and designed to exploit inter-source correlations analogous to the noiseless framework. However, this reduces the dependencies among the transmitted bits leading to poor error-resilience at the decoder and eventually poor reconstruction distortions. On the other extreme, the encoders could be designed to preserve all the correlations among the transmitted bits which could be exploited at the decoder to achieve good error resilience. However, such a design fails to exploit the gains due to distributed compression leading to poor over all rate-distortion performance.

Motivated by these practical challenges, in this paper we address the problem of error-resilient and zero-delay distributed compression for large scale sensor networks. In a recent work [16], a new decoding paradigm for large scale DSC was proposed in case of noiseless networks, wherein the received bits were first compressed (transformed) down to an allowable decoding rate and the decoding was performed in the compressed space. In this paper, we build upon the work in [16] and propose an optimal method to compress the received bits which naturally builds error resilience into the system leading to a unified error-resilient and complexity constrained mechanism for distributed coding in large scale sensor networks. Essentially, we map every received index to a cloud center based on a minimum distance criterion leading to a classification of indices into decoding spheres. The reconstructions are purely based on the sphere to which the received index belongs. These spheres (cloud centers), when designed optimally, lead to an error-correcting code which serves the dual purpose of a source-channel decoder. We use design principles from source-channel coding for individual sources, and propose a global optimization technique based on deterministic annealing [13] to address the intricate nature of the design problem. As we will present in section 4, our methodology overcomes all the drawbacks with the conventional approaches, presented in section 3, and provides significant improvements in reconstruction distortion over state of the art methods for both synthetic and real world sensor network datasets.

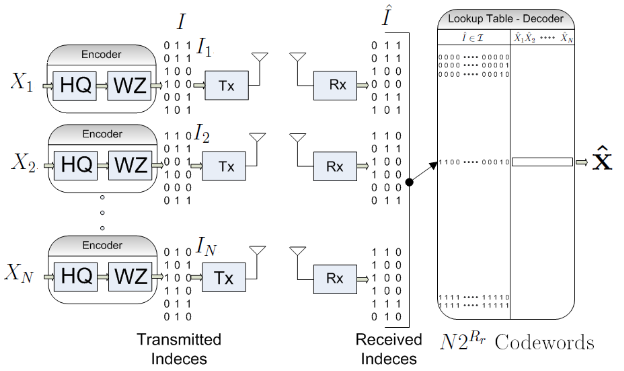

Figure 1: Basic DSC setup - Lookup table based decoder

The rest of the paper is organized as follows. In Sec. 2, we formulate the problem, introduce notation, and discuss the difficulties in the design of large-scale distributed source coding system. In Sec. 3, we review related work, and explain our proposed compression/classification based approach in Sec. 4. Sec. 5 describes the algorithm for system design with details about complexity in Sec. 6. Finally, results are presented in Sec. 7, followed by Conclusions in Sec. 9.

2. DESIGN FORMULATION

Before describing the problem setup, we state some of the assumptions made in this paper. Firstly, to keep the understanding simple, we only consider spatial correlations between sensors and neglect the temporal correlations. Temporal correlations can be easily incorporated using techniques similar to that in [15]. Secondly, in this paper we consider only channels with errors, noting that the methodology can be easily extended to incorporate erasures. We will briefly address this issue in section 8. Further, we assume that there exists a separate channel from every sensor to the central receiver, i.e., information is not routed in a multi-hop fashion. However, the method we propose is fairly general and is applicable to the multi-hop setting. Throughout this paper, we make the practical assumption that while the joint densities may not be known during the design, there will be access to a training sequence of source samples and channel errors during design. In practice this could either be gathered off-line before the deployment of the sensor network or could be collected during an initial phase after deployment.

We begin with the description of the conventional (zero delay) DSC setup. We refer to [6] for a detailed description. Consider a sensor network composed of N sensors (denoted by $s_1, s_2 \ldots, s_N$ respectively). The sensors communicate with a central receiver (denoted by S) at rates $(R_1, R_2 \ldots R_N)$ respectively over noisy channels as depicted in Fig. 1. At regular time intervals, each sensor observes some physical phenomenon (eg. temperature, pressure etc). These sensor observations are modeled as correlated random variables denoted by $(X_1, X_2 \ldots X_N)$. Sensor s_i encodes X_i using R_i bits and transmits it to the receiver. The central receiver attempts to jointly reconstruct $(X_1, X_2 \ldots X_N)$ using bits received from all the sensors. The objective is to design the encoders at each of the sensors and decoders (estimators) at the central receiver so that the overall distortion between the observations and the reconstructions is minimized.

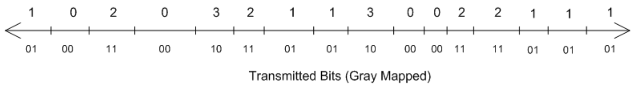

Figure 2: Example of a typical encoder (irregular quantizer). In this example, $N_i = 16$ quantization regions and $R_i = 2$ bits.

Encoding at each sensor is composed of two stages. At sensor s_i, the first stage is a simple high rate quantizer (labeled as "HQ" in Fig. 1), $\mathcal{H}_i$, which discretizes the real space into a finite number of non-overlapping regions N_i. Specifically, $\mathcal{H}_i$ is a mapping which assigns one of the quantization indices to every point in the real space, i.e.,

$$\mathcal{H}_i : X_i \in \mathcal{R} \to \mathcal{Q}_i = \{1 \dots N_i\} \tag{1}$$

Note, that the quantizers are *high rate* so as to exclude them from the joint encoder-decoder design. This is a practical engineering necessity, and the primary purpose of the high-rate quantizers is to discretize the sources. We refer to [14] for further details. The second stage of encoding, which we call, a 'Wyner Ziv map'/WZ-map[1](also called binning in some related work [20]), relabels the N_i quantization regions with a smaller number, 2^{R_i}, of transmission indices. Mathematically, the Wyner Ziv map at source i, denoted by $\mathcal{W}_i$, is the following function:

$$\mathcal{W}_i : \mathcal{Q}_i \to \mathcal{I}_i = \{1 \dots 2^{R_i}\} \tag{2}$$

and the encoding operation can be expressed as a composite function:

$$I_i = \mathcal{E}_i(x_i) = \mathcal{W}_i \left(\mathcal{H}_i(x_i)\right) \ \forall i \tag{3}$$

A typical example of a WZ-map is shown in Fig. 2. Observe that the WZ-map performs lossy-compression. In fact, some regions which are far apart are mapped to the same transmission index, and this makes the encoding operation at each source equivalent to that of an irregular quantizer. Although this operation might seem counter intuitive at first, if designed optimally, it is precisely these modules which assist in exploiting inter-source correlations without inter-sensor communication. Essentially, the design would be such that, it enables the decoder to distinguish between the possible quantization regions for the transmitted index of the particular source using the indices transmitted from other sources. It is fairly well known in the source coding literature (see [12, 14, 20] and the references therein) that these WZ-maps, if properly designed, provide significant improvements in the over all rate-distortion performance compared to that achievable by regular quantizers operating at the same transmission rates (see also section 7.4). It is important to note that the WZ-maps must be designed jointly before the sensor network begins its operation using the source-channel statistics or a training sequence of observations. Efficient design of these mappings for noiseless networks has been studied in several prior publications such as [6, 14].

The encoder at sensor s_i transmits the binary representation of I_i, determined by a standard Gray mapping, to the remote receiver using a standard BPSK modulation scheme. In this paper, we assume that the channels are independent additive white Gaussian noise and the receiver employs separate optimal detection. This makes the effective channel seen by each bit an independent Binary Symmetric Channel (BSC) whose cross-over probability depends on the variance of the noise. However, we note that the design principles presented in the paper are based on an available training set of source samples and channel errors and hence can be easily extended to more general modulation-demodulation schemes and channel error patterns. In particular, the method can be easily applied to the setting where bits are routed over multiple hops (in which case the channel errors are correlated), by collecting the corresponding training set of error samples and designing the system using the collected training sets. We denote the symbol obtained following optimal detection and inverse Gray mapping by $\hat{I}_i \in \mathcal{I}_i$ as shown in Fig. 1. We use the short-hand $I = (I_1, I_2 \dots I_N)$ and $\hat{I} = (\hat{I}_1, \hat{I}_2 \dots \hat{I}_N)$. Note that both I and $\hat{I}$ take values in $\mathcal{I} = \mathcal{I}_1 \times \mathcal{I}_2 \dots \mathcal{I}_N$.

Observe that the total number of bits received at the decoder is $R_r = \sum_{i=1}^{N} R_i$, of which a subset could be erroneous. The decoder reconstructs each source based on the received index $\hat{I}$. Formally, the decoder for source i is a mapping from the set of received index tuples to the reconstruction space and is given by:

$$\mathcal{D}_i : \mathcal{I} \to \hat{X}_i \in \mathcal{R} \tag{4}$$

Usually the decoder is assumed to be a lookup table, which has the reconstruction values stored for each possible received index as shown in Fig. 1. For optimal decoding, the lookup table has a unique reconstruction stored for each possible received index tuple. Hence the total storage at the decoder grows as $\mathcal{O}(N \times 2^{R_r}) = \mathcal{O}(N \times 2^{\sum_{i=1}^{N} R_i})$, which is exponential in N. We call the total storage of the lookup table as the *decoder complexity.* In most prior work, DSC was performed for a few (typically 2 - 3) sources, with the implicit assumption of design scalability with network size. But this exponential growth in decoder complexity for optimal decoding with the number of sources and transmission rates makes it infeasible to use the conventional setup in practical settings even with moderately large number of sources. Just to illustrate, consider a sensor network with 20 sources communicating at $R_i = 2$ bits per source. The decoder receives 40 bits of information and has to store a unique reconstruction for every received bit combination. This would require a decoder storage of over $20 \times 2^{\sum_{i=1}^{20} 2} \approx 175$ TeraBytes... In the next section, we describe some of the related work which has been done to address this huge exponential storage at the decoder.

It is worthwhile to note that the encoding operation in the above scheme involves a simple quantization of the source samples followed by a direct look up of the transmission index. The total storage at each encoder includes its high rate quantization codebook (of size $|\mathcal{Q}_i|$) and the corresponding WZ-map (of size $|\mathcal{Q}_i| 2^{R_i}$). For typical values of $|\mathcal{Q}_i|$ and R_i, the encoder complexity is significantly small and hence can be easily implemented on a physical sensor mote. This inherent advantage makes such approaches to distributed coding more viable in low cost practical sensor networks than the channel coding based methods, such as [9, 18], which require complex Slepian-Wolf coders at each source. Hence, hereafter, our concern will be only towards addressing decoder complexity, assuming that the encoders can be easily implemented on a physical sensor mote.

[1] The term 'Wyner-Ziv map' is coined after Wyner and Ziv [17] who first solved the lossy version of the side information setup in information theory

3. RELATED WORK

One practical solution proposed in the past to handle the exponential growth in decoder complexity is to group the sources based on source statistics [6] and to perform DSC within each cluster. By restricting the number of sources within each group, the decoder complexity is maintained at affordable limits. Evidently, even in the noiseless scenario, such an approach does not exploit inter-cluster dependencies and hence would lead to sub-optimal estimates. Moreover, when there is channel noise, the resilience of the decoder to channel errors degrades significantly as it is forced to use only a subset of received bits to correct any error. Also in most prior work, source groups are designed only based on the source statistics, completely ignoring the channel conditions. Indeed, it is a much harder problem to come up with good source grouping mechanisms which are optimized for both source and channel statistics.

It is worthwhile to mention that an alternate approach, other than the lookup table has been proposed in the literature to practically implement the decoder [6, 19, 20]. In this approach, the decoder computes the reconstructions on the fly by estimating the posterior probabilities for quantization index q_i as $P(q_i|\hat{I})$, when a particular $\hat{I}$ is received. Such an approach requires us to store the high rate quantization codewords at the decoder, which grow only linearly in N. However, to compute the posterior probabilities $P(q_i|\hat{I})$, using Bayes rule, we have:

$$P(\tilde{q}_i|\hat{I}) \;=\; \gamma \sum_{Q:q_i=\tilde{q}_i} P(\hat{I}\Big|I(Q))P(Q) \qquad (5)$$

where γ is a normalization constant, and $Q = (q_1, q_2 \ldots, q_N)$. The above marginalization requires an exponential number of operations to be performed at the decoder, let alone the exponential storage required to store the probabilities $P(q_1, \ldots q_N)$.

To limit the computational complexity, prior work such as [2, 6, 20] have proposed clustering the sources and linking the clusters using a limited complexity Bayesian network (or a factor graph), and thereby using message passing algorithms to find $P(\tilde{q}_i|\hat{I})$ with affordable complexities. These approaches provide significant improvement in distortion over simple source grouping methods at fixed transmission rates and channel SNRs as they exploit inter-cluster correlations efficiently. However, a major drawback of such techniques, which is usually overlooked, is that they require the storage of the Bayesian network/factor graph at the decoder. Though this storage grows linearly in N, it grows exponentially with the rate of the 'high rate quantizers'. To be more precise, if we choose $N_i = 2^{R_q} \forall i$, then the storage of the Bayesian network grows of the order of $\mathcal{O}(N2^{MR_q})$ where M is the maximum number of parents for any source node in the Bayesian network. Typically (see for example [12, 14, 6]), in source coding, R_q is chosen as $R+3$ or $R+4$ for the Wyner-Ziv maps to exploit the inter source correlations efficiently. This makes the Bayesian network based techniques less efficient as the gains in distortion obtained by introducing the Bayesian network are superseded by the excess storage required to store the Bayesian network. We will show in our results that the Bayesian network based methods under-perform even the source grouping techniques even for moderate values of N at a *fixed storage*. Hence, though it is counter-intuitive at first, it is indeed beneficial to group more sources within a cluster instead of connecting the clusters using a Bayesian network.

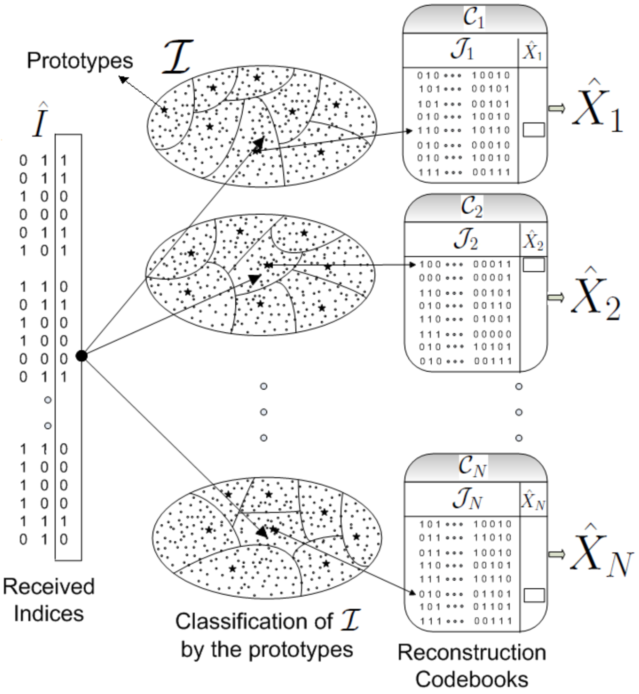

Figure 3: Prototype based bit-mapper approach to decoding

We note that, the storage required for transition probabilities in the Bayesian network can be significantly reduced if the joint densities of source distributions are parameterized (for example as multivariate Gaussian). However, such approximations are highly prone to estimation inaccuracies and could lead to sub-optimal designs for more general/real world source and channel statistics as has been observed in [20]. We next describe our proposed classification-based approach for decoding which overcomes these drawbacks and achieves a unified approach to error resilient and complexity constrained distributed compression.

4. THE CLASSIFICATION/COMPRESSION BASED APPROACH TO DECODING

Recall that the decoder receives $R_r = \sum_{i=1}^{N} R_i$ bits of information of which part could be erroneous. The look up table at the receiver cannot store a unique reconstruction for every possible received combination of bits. Hence, to decode source s_i, we first find an optimal classification scheme which groups the set of all possible received index tuples, $\mathcal{I}$, into K_i groups. We then assign a unique reconstruction for all received combinations which belong to the same group. Essentially, we decompose the monolithic decoder, which was a simple look-up table, into a compressor/classifier/bit-mapper followed by a look-up table of reconstructions. Note that the classification could possibly be different for decoding each source. This would bring down the total storage required for codebooks from $N2^{R_r}$ to $\sum_{i=1}^{N} K_i$, which can easily be controlled by varying the number of groups.

However, a generic bit mapper would require us to store the class information for every possible received index which entails a storage exponential in N, defeating the purpose of classification. Hence we impose the structure of a 'nearest neighbor classifier' or a 'vector quantizer' for the bit-mapper which enforces each received index tuple to be clustered to one of the cloud centers based on a minimum distance criterion as shown in Fig. 3. Such a modification to the bit-mapper, though leads to some loss in optimality, provides a two fold advantage. *On one hand, it dramatically reduces the storage overhead required to store the bit-mapper, as it requires us to store only the cloud centers. On the other hand, it builds error resilience into the system as it essentially implements an error correcting code at the decoder by assigning the same codeword to nearby received indices.* If the probability of channel error is not large, we would expect $\hat{I}$ to be sufficiently close to I and hence belong to the same decoding sphere (group) as I. If the prototypes, encoders and reconstruction codebooks are optimally designed for the given source-channel statistics/training sequences, such an approach would assist error correction leading to improved end-to-end reconstruction distortion.

These cloud centers are called 'prototypes' in the literature [13] and the structure is normally termed 'nearest prototype classifier'. Technically, these prototypes can be defined in any sub-space (for example $\mathcal{R}^N$) with an appropriate distance metric defined between the received index tuples to the prototypes. However, we require these prototypes to entail minimal excess storage, but at the same time provide enough diversity for achieving good error-resilience. Hence we enforce each prototype to belong to $\mathcal{I}$ and choose the corresponding distance metric to be the Hamming distance between the binary representations of the received indices and prototypes. Given a set of prototypes $\mathcal{J}_i = \{S_{i1} \dots S_{iK_i}\}$, $S_{i,j} \in \mathcal{I}\ \forall i, j$, the bit-mapper can mathematically be written as:

$$\mathcal{B}_i(I) : \arg\min_{S \in \mathcal{J}_i} d_i(I, S) \tag{6}$$

where $d_i(\cdot,\cdot)$ denotes the Hamming distance. We note that, the design methodology is applicable for prototypes chosen from any generic sub-space.

In the next stage of decoding, each prototype is associated with a unique reconstruction codeword. We denote this mapping by $\mathcal{C}_i(S_{i,j})$. Hence, if the received index is $\hat{I}$ and if the nearest prototype to $\hat{I}$ is $S_{i,j}$, then the estimate of source i is $\hat{x}_i = \mathcal{C}_i(S_{i,j})$, i.e., the composite decoder can be written as:

$$\hat{X}_i(\hat{I}) = \mathcal{D}_i(\hat{I}) = \mathcal{C}_i(\mathcal{B}_i(\hat{I})) \tag{7}$$

5. ALGORITHM FOR SYSTEM DESIGN

As mentioned in section 2, we assume that a training set of source and channel samples is available during design. Hence, given a training set, $\mathcal{T} = \{(\mathbf{x}, \mathbf{n})\}$, of source and noise samples, our objective in this section is to find the encoders, prototypes and reconstruction codebooks which minimize the average distortion on the training set, which is measured as:

$$D_{avg} = \frac{1}{N|\mathcal{T}|} \sum_{(\mathbf{x},\mathbf{n}) \in \mathcal{T}} ||\mathbf{x} - \hat{\mathbf{x}}||^2 \tag{8}$$

Note that in the above equation, we have assumed the distortion metric to be the mean squared error (MSE) and given equal weightings to all the sources similar to [6, 20]. However, the design methodology is applicable to any other general distortion measure.

We first note that the high rate quantizers are designed separately using a standard Lloyd-Max quantizer design technique to minimize the respective average squared error. The challenging part is to design the Wyner-Ziv maps jointly with the prototypes and the reconstruction codebooks to minimize D_{avg}. We note that readers who are not particularly interested in implementing the system, can conveniently skip the rest of this section.

Note that the design of such 'nearest prototype classifiers' or 'generalized vector quantizers' has been studied earlier in the context of source-channel coding for a single source and is known to be a very challenging problem [11]. The main challenge arises due to the fact that, unlike the standard quantizer design problem where the objective is to minimize the average quantization distortion, here the classifiers/quantizers are to be designed to minimize the distortion in the reconstruction space. One straight forward design approach is to employ a greedy-iterative descent technique which reduces D_{avg} in each iteration. Such an algorithm would initialize the Wyner-Ziv maps, the prototypes and the reconstruction codebooks *randomly* and then update the parameters iteratively, reducing D_{avg} in each step, until convergence. As the number of possible Wyner-Ziv maps and prototypes is finite, convergence is guaranteed to a local minimum for any initialization.

However, in (8), the prototypes are present inside a highly non-convex function which makes the greedy approach likely to get trapped in very poor local minima (even with multiple random initializations), thereby leading to sub-optimal designs. Finding a good initialization for such greedy iterative descent algorithms, even for problems much simpler in nature than the one at hand, is known to be a very difficult task. Hence in the following section, we propose a global optimization technique based on deterministic annealing (DA) which provides significant gains by avoiding poor local minima. Also note that the design approach we propose, optimizes all the system parameters for the given source *and* channel statistics. However the design approaches proposed in most prior work such as [6, 20] optimize the WZ-maps for the noiseless scenario (without the knowledge of the channel) and then design only the decoder codebooks for the given channel statistics. We particularly study the gains due to this optimal design later in section 7.

5.1 Deterministic Annealing Based Design

A formal derivation of the DA algorithm is based on principles borrowed from information theory and statistical physics. Here, during the design stage, we cast the problem in a probabilistic framework, where the standard deterministic bit-mapper is replaced by a random mapper which associates every training sample to all the prototypes in probability. The expected distortion is then minimized subject to an entropy constraint that controls the "randomness" of the solution. By gradually relaxing the entropy constraint we obtain an annealing process that seeks the minimum distortion solution. More detailed derivation and the principle underlying DA can be found in [13].

Specifically, for every element in the training set, the received index tuple, $\hat{I}$, is mapped to all the prototypes, $\mathcal{J}_i$, in probability. These probabilities are denoted by $P_i(j|k)$ $\forall i \in (1, \dots, N)$, $j \in (1, \dots, |\mathcal{J}_i|)$, $k \in (1, \dots, |\mathcal{T}|)$, i.e., the received index tuple for training sample k is associated to prototype j in $\mathcal{J}_i$ with probability $P_i(j|k)$. Hence, the average distortion is:

$$D_{avg} = \frac{1}{N|\mathcal{T}|} \sum_{k=1}^{|\mathcal{T}|} \sum_{i=1}^{N} \sum_{j \in \mathcal{J}_i} P_i(j|k) \left(x_i(k) - \hat{x}_i(j)\right)^2 \qquad (9)$$

where $x_i(k)$ is training sample k of X_i and $\hat{x}_i(j) = \mathcal{C}_i(S_{ij})$. Note that this includes the original hard cost function as a special case when probabilities are *hard*, i.e.,:

$$P_i(j|k) = \begin{cases} 1 & \text{if } \arg\min_{j'} d_i(S_{ij'}, \hat{I}(k)) = j \\ 0 & \text{else} \end{cases} \qquad (10)$$

It is important to note that these mappings are made *soft* only during the design stage. Of course, our final objective is to design hard bit-mappers which minimize the average distortion.

Further, we impose the ‘nearest prototype’ structural constraint on the bit-mapper partitions by appropriately choosing a parametrization of the association probabilities. Similar methods have been used before in the context of design of tree-structured quantizers [13], generalized VQ design [11] and optimal classifier design [7]). It can be shown using the principle of entropy maximization that (refer to [13]), to impose a ‘nearest prototype’ structure, at each temperature, the association probabilities must be governed by the Gibbs distribution:

$$P_i(j|k) = \frac{e^{-\beta_i\left(d_i\left(\hat{I}(k), S_{ij}\right)\right)}}{\sum_j e^{-\beta_i\left(d_i\left(\hat{I}(k), S_{ij}\right)\right)}} \qquad (11)$$

Observe that this parametrization converging to the ‘nearest prototype classifier’ as $\beta_i \to \infty$.

These mappings introduce randomness into the system measured by the Shannon entropy as:

$$H = \frac{1}{N|\mathcal{T}|} \sum_{k \in \mathcal{T}} \sum_{i=1}^{N} \sum_{j \in \mathcal{J}_i} P_i(j|k) \log P_i(j|k) \qquad (12)$$

DA algorithm minimizes D_{avg} in (9), with a constraint on the entropy of the system, (12), where the level of randomness is controlled by a Lagrange parameter (usually called the temperature in the literature due to its roots in statistical physics), T as:

$$J = D_{avg} - TH \qquad (13)$$

Initially, when T is set very high, our objective is to maximize H and hence all the β_i are very close to 0. This leads to a very fuzzy system where all the received indices are mapped to every prototype with equal probability. Then at each stage, the temperature is gradually lowered maintaining the Lagrangian cost at its minimum. β_i gradually raises as T reduces, thereby making the association distribution less fuzzy. Finally as $T \to 0$ all the $\beta_i \to \infty$ and we obtain hard mappings where every received index maps to the closest prototype. As $T \to 0$ our Lagrangian cost becomes equal to D_{avg} and our original objective is realized. At each temperature, we minimize J with respect to $\mathcal{W}_i$, $\mathcal{J}_i$, β_i and $\mathcal{Q}_i$ $\forall i$. This minimization is achieved using a standard gradient descent method with update rules given below.

5.1.1 Wyner-Ziv Map Update

At fixed T, the WZ-map update rules are given by:

$$\mathcal{W}_i^*(m) = \arg\min_{l \in \mathcal{I}_i} J(\mathcal{W}_i(m) = l) \qquad (14)$$

$\forall i \in (1, \dots, N), m \in \mathcal{Q}_i$ where $J(\mathcal{W}_i(m) = l)$ denotes the Lagrange cost obtained on the training set when $\mathcal{W}_i(m)$ is set to l with all the remaining parameters unchanged.

5.1.2 Prototype Update

Note that each prototype can take values in the set $\mathcal{I}$ and the size of the set $|\mathcal{I}| = 2^{\sum_{i=1}^{N} R_i}$, which grows exponential in N. Hence, for large sensor networks, it is infeasible to find the best prototype in each iteration from the set $\mathcal{I}$. Hence, in each step, we find an incrementally better prototype among the neighboring prototypes, which are at a Hamming distance of one. Mathematically, for fixed Wyner-Ziv maps and reconstruction codebooks, the update rule for prototypes is:

$$S_{ij}^* = \arg\min_{s \in N(S_{ij})} J(S_{ij} = s) \qquad (15)$$

where $J(S_{ij} = s)$ is the Lagrange cost obtained by setting $S_{ij} = s$ with all the remaining parameters unchanged and $N(S_{ij})$ denotes all neighboring prototypes of S_{ij}.

5.1.3 β_i Update

As β_i are real values, we find the gradient of J with respect to β_i for fixed Wyner-Ziv maps, prototypes and reconstruction codebooks and employ a standard gradient descent operation to update β_i. The gradients of J with respect to β_i $\forall i$ is given by:

$$\frac{\delta J}{\delta \beta_i} = \frac{1}{N|\mathcal{T}|} \sum_{k,j} \Big\{ (x_i(k) - \hat{x}_i(k))^2 + T \log(2P_i(j|k)) \\ P_i(j|k) \Big(\sum_{j'} P_i(j'|k) d(\hat{I}(k), S_{ij'}) - d_i(\hat{I}(k), S_{ij}) \Big) \Big\} \qquad (16)$$

Then the update rule for β_i is given by:

$$\beta_i^* = \beta_i - \triangle \frac{\delta J}{\delta \beta_i} \qquad (17)$$

where $\triangle$ is the step size for descent.

5.1.4 Reconstruction Codebook Update

Note that, J is a convex function of the reconstruction values and hence the optimum codebook which minimizes J for any fixed encoders, prototypes and β_i is given by:

$$\hat{x}_i(j) = \mathcal{C}_i(j) = \frac{\sum_k P_i(j|k) x_i(k)}{\sum_k P_i(j|k)} \qquad (18)$$

The complete steps for DA are shown as a flowchart in Algorithm 1[2]. T is initialized to a very high value and β_is are set very low. All the Wyner-Ziv maps and the reconstruction codebooks are initialized randomly. The prototypes are set to the median of the received indices so as to minimize the average Hamming distance. Temperature is gradually lowered using an exponential cooling schedule, $T^* = \alpha T$. In all our simulations, we used $\alpha = 0.98$. At each temperature, all the system parameters are optimized using Eqns. (14), (15), (17) and (18) till the system reaches equilibrium. This

[2] The simulation code is available at: http:www.scl.ece.ucsb.edu/html/database/Error_Resilient_DSC.

equilibrium is perturbed and used as an initialization for the next temperature. These iterations are continued till T approaches zero. In practice, the system is 'quenched', i.e, T is set to zero and the bit-mapper is made hard, once the entropy becomes sufficiently small. Note that the optimization steps at $T = 0$ are same as that for the greedy approach. However, instead of a random guess, the equilibrium at the previous temperature is now used as the initialization. We further note that under certain conditions on continuity of phase transitions in the process, DA achieves the global minimum [13], but its ability to track the global minimum as we lower the temperature depends on a sufficiently slow cooling schedule (i.e., α sufficiently close to 1). However in practice α is restricted based on the available design time. In our simulations, we observed that using $\alpha = 0.98$ achieves significantly better solutions compared to the greedy descent approach.

Algorithm 1. ***DA Approach for System Design***

```
————————————————————————————————
Inputs: $N_i$ (Number of high rate quantization indices),
$R_i$ (Transmission rates),
$R_{d_i}$ (Decoding rate, i.e., $|\mathcal{J}_i| = K_i = 2^{R_{d_i}}$),
$\mathcal{T}$ (Training set), $T_{max}(\sim 1-10)$, $T_{min}(\sim 10^{-5}-10^{-4})$,
$\beta_{min}(\sim 0.1-0.2)$, $H_{min}(\sim 0.1-0.2)$, $\alpha < 1$ (Cooling Rate),
$\triangle(\sim 0.1-0.2)$.
Outputs : $\mathcal{H}_i$ (High rate quantizers),
$\mathcal{W}_i$ (WZ-maps),
$\mathcal{J}_i$ (Prototypes),
and $\mathcal{C}_i$ (Reconstruction codebooks)
————————————————————————————————

1. Design the high rate quantizers individually using a
   standard Lloyd-Max algorithm.
2. Initialize: $T = T_{max}$, $\beta_i = \beta_{min}$, Initialize WZ-maps randomly,
   set $S_{ij} = Median(\hat{I}(\mathbf{x}),\ \mathbf{x} \in \mathcal{T})\ \forall i \in (1,\ldots,N), j \in (1,\ldots,\mathcal{J}_i)$.
3. Compute: $P_i(j|k)$ using (11) and $\mathcal{C}_i(j)$ using (18).
4. Update:
   • WZ-maps using (14).
   • Prototypes using (15).
   • $\beta_i$ using (17), and then compute $P_i(j|k)$ using (11).
   • $\mathcal{C}_i(j)$ using (18).
5. Convergence: Compute $J$ and $H$ using (13) and (12) respectively.
   Check for convergence of $J$. If not satisfied go to step (4)
6. Stopping: If $T \leq T_{min}$ or $H \leq H_{min}$, set $P_i(j|k)$ as (10)
   and perform last iteration for $T = 0$. Then STOP.
7. Cooling:
   • $T^* \leftarrow \alpha T$.
   • Perturb prototypes: $S^*_{ij} \leftarrow s \in Neighborhood(S_{ij})$,
     where $s$ is chosen randomly.
   • Perturb $\beta^*_i \leftarrow \beta_i + \delta$ for small $\delta > 0$ generated randomly.
   • Go to (4)
```

5.2 Note on Design Complexity

The design complexity for the proposed setup, either using the greedy approach or using DA grows as $\mathcal{O}(R_r^3|\mathcal{T}|)$. The DA approach has a larger constant and requires more computations compared to greedy approach for a single initialization. However, as the greedy approach has to be run over multiple random initializations to achieve a good solution, the exact comparison of design complexities is difficult and depends on the actual source-channel distributions. A generally accepted and observed fact (see [13]) is that for a given design time, DA provides far better solutions compared to that achieved by greedy approaches over multiple random initializations for such complex non-convex optimization functions.

6. OPERATIONAL COMPLEXITY

In this section we compare the computational and storage complexities during operation of all the three approaches for large scale DSC described earlier. For comparison purposes, we assume that every source sends information at rate R and all the high rate quantizers operate at rate $R_q \geq R$. Also, we assume that the decoding rate is $R_{d_i} = R_d$ ($R_d \leq NR$) for all sources. For the source grouping approach, this means that the maximum number of sources in any cluster is R_d/R; for the Bayesian network approach, this implies that the maximum number of parent nodes for any source node is R_d/R and for the proposed approach, this implies that the number of prototypes for decoding any source is $|\mathcal{J}_i| = |K_i| = 2^{R_d}\forall i$.

6.1 Computational Complexity

Firstly, we note that the computational complexity during operation of all the three approaches is polynomial in N. It is easy to observe that the decoder in the source grouping method has literally no computations to make, i.e. the complexity is a constant, $\mathcal{O}(1)$. The decoder in the Bayesian network approach has to implement a message passing algorithm for every received combination of indices. This leads to a computational complexity which grows as $\mathcal{O}(N2^{R_q R_d/R})$. On the other hand, the proposed prototype based bit-mapper approach finds the closest prototype for every received index tuple, which requires $\mathcal{O}(2^{R_d}N\log N)$ bit comparisons. Note that, though the complexity grows slightly faster than N, it requires only bit comparisons, and will incur much lesser machine cycles than required for implementing each iteration in the Bayesian network approach. As all the three methods can be implemented in practice with affordable computational complexities, we hereafter assume they are 'equivalent' with respect to computations and focus only on their storage requirements.

6.2 Storage Complexity

Table I shows the order of growth in storage as a function of N,R, R_q and R_d for all the three approaches. Here, F denotes the bits required to store a real number or the floating point accuracy. In all our simulations, we use $F = 32$ bits.

Storage due to	Codebook	Module
Source grouping	$N2^{R_d}F$	$N\log_2(\frac{NR}{R_d})$
Bayesian network	$N2^{R_q}F$	$N2^{(R_qR_d/R)}F$ $+N\log_2(N)$
Prototype based bit-mapper	$N2^{R_d}F$	$N^2R2^{R_d}$

Table 1: Order of growth in storage complexities

The codebook storage in all the three settings are considerably easier to derive. For example in the prototype approach, there is a unique codeword associated with every prototype. There are 2^{R_d} prototypes for decoding each source and hence the total storage for the reconstruction codebooks is $N2^{R_d}F$. Similar arguments lead to the codebook storage for the other approaches as given in Table I.

For analysis of module storage, we first begin with the source grouping method. It requires us to store the group labels for each source. As there are at least NR/R_d groups, we need at least $N\log_2(\frac{NR}{R_d})$ bits to store the source groupings. For the Bayesian network approach, we require an order of $N\frac{R_d}{R}\log_2(N)$ bits to store the parent node information for each source. However, there is an additional storage required to store the transition probabilities which grows as $N2^{R_qR_d/R}F$. The prototype based bit-mapper approach requires us to store all the prototypes at the decoder. Each prototype requires NR bits to store and there are $N2^{R_d}$ such prototypes leading to a total storage of $N^2R2^{R_d}$.

A first look at Table I suggests that the prototype based bit-mapper approach entails a module storage which grows as N^2 in the number of sources and hence should entail a very high overhead due to module storage. However, for typical values of these parameters, (i.e., $N \sim 10-500$ sources, $R \sim 1-10$ bits, $R_q \sim (R+2)-(R+4)$ bits and $R_d/R \sim 2-4$) the storage overhead of the proposed approach is not very significant and the distortion gains obtained overhaul the minimal loss due to excess storage[3]. However, in these typical ranges, the Bayesian network approach entails a storage which is significantly higher than the other two methods and hence leads to higher distortions at a fixed storage. Note that, the values in Table I indicate the order of growth of storage complexity and hence are accurate only upto a constant. In all our simulations, we consider the exact storage required and not the values derived from Table I.

7. RESULTS

To test the performance of the proposed approach, we used 3 different datasets:

1) **Synthetic dataset:** A toy dataset consisting of 10 synthetic sources, randomly deployed on a square grid of dimensions 100 m × 100 m was generated according to a multivariate Gaussian distribution. All sources were assumed to have zero mean and unit variance. The correlation was assumed to fall exponentially with the distance. Specifically, we assumed $\rho = \rho_0^{d/d_o}$, $\rho_0 < 1$. For all our simulations with this dataset we set $d_o = 100$. The training set generated was of length 10000 samples. All results presented are on a test set, also of the same length, generated independently using the same distribution.

2) **Temperature sensor dataset :** The first real dataset we used was collected by the Intel Berkeley Research Lab, CA [4]. Data were collected from 54 sensors deployed in the Intel Berkeley Research Lab between February 28 and April 5, 2004. Each sensor measured temperature values once every 31 s [5]. We retained data from top 25 sensors that collected highest number of samples. Times when subset of these sensors failed to record data were dropped from the analysis. The data were normalized to zero mean and unit variance. Samples collected till March 18th, 2004 were used to train the system and the remaining were used as the test set.

3) **Rainfall dataset :** As a second real dataset, we used the rainfall dataset used in[8] [6]. This data-set consists of the daily rainfall precipitation for the Pacific northwest region over a period of 46 years. The measurement points formed a regular grid of 50km x 50km regions over the entire region under study. The first 30 years of data were used for training and the remaining to test the system. Note that the inter-source correlations in such 'large area' datasets are considerably lower. However, performance evaluation using such diverse real world datasets is important to validate the efficiency of the proposed setup.

We note that, all our results are in terms of the crossover probability of the effective BSC seen by each bit. We denote the cross over probability (error probability) by P_e, ie., $P(1|0) = P(0|1) = P_e$. Note that P_e is directly related to the channel SNR (CSNR) as $P_e = Q(\sqrt{CSNR})$. In all our simulations, we generated a training sequence of channel errors of the same size as the training set. The average distortion of the test set over 100 random (i.i.d.) channel realizations is used as the performance metric.

7.1 Complexity-Distortion Trade-off

Fig. 4 shows the total storage (complexity) versus the distortion trade-off for all the three datasets. For these simulations, the transmission rate was set to $R_i = 1$ bit. This allows us to compare the performances with the minimum distortion achievable using full complexity decoding. We will present results at higher transmission rates in section 7.4. The decoding rate was varied from 1 to 5 bits to obtain the distortion at different complexities. We plot the total storage, which includes both codebook and module storage, versus the distortion to obtain a trade-off curve. We show results obtained using all the three decoding methods - source grouping where the grouping is done using source optimized clustering approach described in [6], Bayesian network as described in [20] and the prototype based bit-mapper approach proposed in this paper. For fairness, we design the WZ-maps for the given channel statistics for all the approaches. However, note that, in most prior work the channel statistics were ignored while designing the WZ-maps [20]. We study the gains due to this optimal design in the following section. For comparison, we also include the performance obtained for designs using greedy-iterative descent approach (opti-

[3] Note that if $N >> 500$, then the optimal approach would be to group ~ 500 sources within each cluster and to perform decoding based on the proposed approach at affordable complexities within each cluster, instead of directly grouping at the allowed complexity

[4] Available at http://db.csail.mit.edu/labdata/labdata.html

[5] Note that the sensors also measured humidity, pressure and luminescence. However, we consider only the temperature readings here

[6] Available for download at http://www.jisao.washington.edu/data sets/widmann

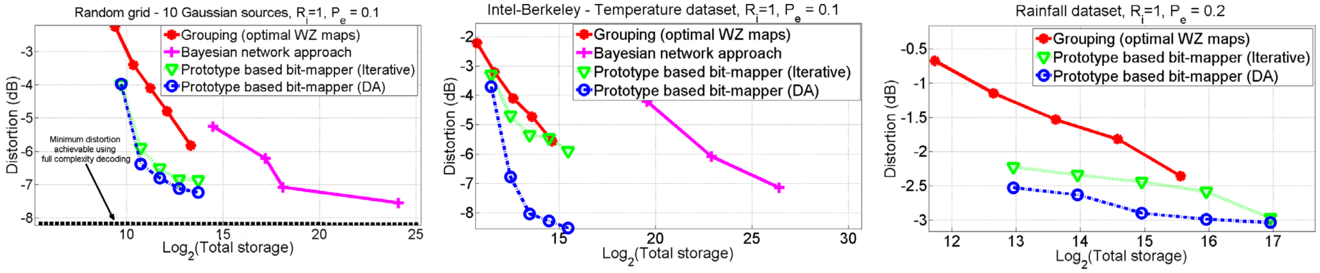

Figure 4: Total storage Versus Distortion for 3 different datasets. (a) Synthetic dataset, $R_i = 1 \forall i$ and $P_e = 0.1$ (b) Temperature sensor dataset, $R_i = 1$, $P_e = 0.1$ (c) Rainfall dataset, $R_i = 1$, $P_e = 0.2$

mized over upto 25 random initializations) along with that achieved using DA.

Fig. 4(a) shows the result obtained for the synthetic dataset using $\rho_0 = 0.9$ and $P_e = 0.1$. We see gains of over 2 dB in distortion compared to the source grouping technique at a fixed storage. Alternatively, the total storage can be reduced by $10X$ times while maintaining the same distortion. We also see that the performance of the prototype based bit-mapper approaches the optimal 'full complexity' decoder significantly faster than the source grouping method. However, observe that, though the Bayesian network based decoder gains substantially over source grouping approach in distortion at fixed decoding rates, the excess storage required to store the Bayesian network offsets these gains, leading to much higher storage at fixed distortions. Note that, in this case, the greedy approach also provides similar performance as DA, as the probability of getting trapped in local minima is low after 25 runs for smaller networks.

Figures 4(b) and 4(c) show the performance obtained for the temperature sensor dataset and the rainfall dataset at P_e of 0.1 and 0.2 respectively. As the temperature sensor dataset has considerably higher correlations, we see gains of over 2.5 dB in distortion at fixed storage over source grouping approach. Due to lower correlations in the rainfall dataset, we choose a higher P_e. Here, gains of about 1dB in distortion are obtained. In general, higher correlations assist the bit-mapper as it uses all the received bits to correct errors, unlike the grouping approach which is forced to use only the bits within each group. From 4(b), it also follows that the overhead required to store the Bayesian network aggravates at higher N and the performance degrades further, making the Bayesian network approach impractical for very large networks[7]. Also, for these datasets, observe that the performance of the greedy-iterative descent method is considerably poorer than that using DA. Hence, hereafter, we only show results for DA, noting that the greedy approach leads to poor designs for large networks.

In what follows, we compare the distortion performance of the prototype based bit-mapper and the source grouping approaches by varying the network and design parameters at a fixed decoding rate. As the total storage is not reflected in these plots, we do not consider the performance of the Bayesian network approach hereafter, noting that, the storage required to achieve good distortion performance is significantly higher.

7.2 P_e Versus Distortion

In this section, we show the performance gains when P_e is varied. We restrict P_e to be in the range $0 - 0.2$ (i.e, CSNR > -1.5 dB). For all the simulations, we have chosen $R_i = 1$ and $R_d = 3$. Fig. 5(a) shows the distortion obtained as a function of P_e for the synthetic dataset. For the source grouping approach, we plot 2 curves. The first curve shows the performance when the WZ-maps are optimized jointly with the decoder for the given channel statistics. The second curve shows the performance when the WZ-maps are designed without the knowledge of channel statistics (instead designed to minimize reconstruction distortion at zero noise). However, after the design of the WZ-maps, the reconstruction codebooks are designed for the given channel statistics. Clearly, optimal design of the WZ-maps for the given channel provides about 0.5dB improvement in distortion. Further, major improvements of over 2 dB, is due to the error-resilience provided by the proposed decoder structure. We see similar behavior even for the two real world datasets in figures 5(b) and 5(c). The higher error-correction capability of the nearest prototype structure is further reflected as the gains improve when P_e increases (CSNR decreases). Again observe that the gains in case of the rainfall dataset are smaller due to lower correlations in the dataset.

7.3 Performance with Network Size

In this section we study how the gains vary with the size of the network. As random deployment makes it hard to compare, we consider a uniformly placed, linear grid of sensors between two fixed points. We increase the number of sensors from 6 to 90 while keeping the transmission and decoding rates fixed. We assume a correlation model which falls off exponentially with the distance and assume P_e to be 0.2 throughout. Fig. 6 compares the results obtained for the source grouping approach and the proposed bit-mapper approach. We see that the gains keep increasing with the network size. This is because, as the number of sources increase, the decoder receives several more correlated bits which are efficiently used by the proposed approach to correct errors. On the other hand, the inefficiency of the the source grouping method is directly evident as it uses only bits within each cluster.

7.4 Performance as a Function of Other Design Parameters

In the following, we show results only for the synthetic dataset described in the beginning of this section. We vary different design parameters and study the performance gains.

[7] For the rainfall dataset, the storage required for the Bayesian network approach was significantly larger and hence we do not plot it along with the other curves

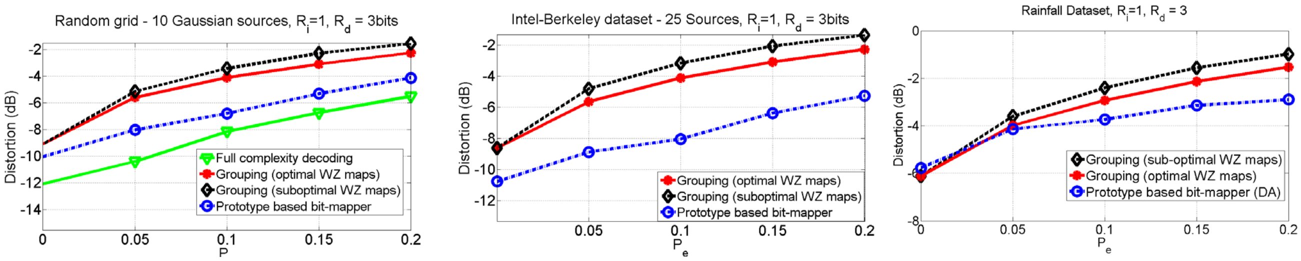

Figure 5: P_e versus Distortion for 3 different datasets. For all the plots, we have used $R_i = 1$ and $R_d = 3$ (a) Synthetic dataset (b) Temperature sensor dataset (c) Rainfall dataset

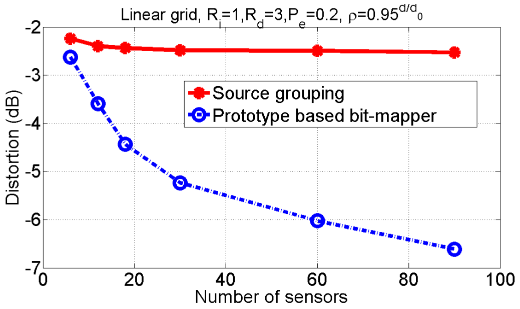

Figure 6: Variation of reconstruction distortion with the number of sources deployed on a linear grid placed uniformly along a length of 10 Kilometers. Correlation model is assumed to be $0.95^{dist(Km)}$, $R_i = 1$bit and $P_e = 0.2$

7.4.1 Correlation (ρ_o)

Fig. 7(a) shows the distortion as a function of ρ_o. The plot shows the results for the source grouping method, the proposed approach and the optimal full complexity design which uses all the received bits. 3 dB improvement of the proposed approach over the grouping method at very high correlations provides further evidence of improved error resilience.

7.4.2 Transmitted Bits (R_i)

In this section, we compare the performances when the transmission rates are increased. We consider 3 different transmission rates, $R_i = 1, 2$ and 4. However, we fix the decoding rate at 4 bits. We see that the gains increase radically to over 6dB, at higher transmission rates. This is primarily because of two reasons. Firstly, as R_i increases, the decoder has access to more correlated bits which can be used efficiently for correcting more errors. Secondly, the decoder for any source has the freedom of selectively giving importance only to a subset of bits sent from a different source. However, the source grouping approach does not exploit either of these advantages and hence suffers significantly more at higher transmission rates. However, the problem with operating at very high transmission rates is that the proposed design complexity grows as $(\sum_{i=1}^{N} R_i)^3$ and hence it requires sophisticated computing capabilities for efficient design.

7.4.3 Rate of Quantizers (R_q)

All results so far have focused on the decoder structure. One might be curious to know the importance of the encoder structure/WZ-maps. Figure 7(c) shows the decrease in distortion when the rates of $\mathcal{H}_i$ are increased from $R_q = 1$ to 4 bits, while keeping the transmission rate fixed at $R_i = 1$. Note that $R_q = 1$ is equivalent to having no WZ-maps (i.e., each encoder is a simple scalar quantizer). Results show over 2.5dB gains for the bit-mapper approach and about 1.5 dB improvement for the source grouping approach when R_q is increased from 1 to 4 bits. Such improvements (see also [20]) demonstrate the crucial role played by WZ-maps in exploiting inter-source correlations. Also note that the proposed structure for the decoder provides about 1dB improvement over source grouping method even when $R_q = 1$ (i.e., when there is no distributed encoding, for example see [2]). This result is particularly useful in practical sensor networks wherein the sensors employ standard scalar quantization.

8. DISCUSSION

8.1 Extension to Handle Erasures

It is critical to develop robust distributed source coding techniques for networks with bit/packet erasures - in fact, erasures are seen more often in low powered sensor networks than errors. In this section we briefly address this issue and describe how the proposed technique can be easily extended to handle erasures. In the erasure setting, it is assumed that a subset of the transmitted bits are lost due to sensor/channel failures and the decoder reconstructs all the sources based only on the received bits. The objective is to design the encoders (at each source) and decoders (for each bit erasure pattern) to minimize the average distortion at the decoder. In the most general setting, the decoder has an independent codebook for each possible erasure pattern and an estimate for each source is made by looking at the corresponding codebook when a subset of the bits are received. Quite evidently, for optimal decoding, the total number of codebooks grows exponentially with the number of sources and transmission rates, let alone the exponential growth in the number of estimates (codewords) within each codebook. It is easy to verify that the total storage at the decoder (the decoder complexity) for optimal decoding grows as $\mathcal{O}(3^{NR})$ if $R_i = R \forall i$.

In this paper, we describe one possible approach to extend the classifier based decoding paradigm to handle erasures. We note that there are several other possible methods to extend it and their performance comparisons will be performed as part of future work. Recall that, to build error resilience, the decoder mapped the received index tuple to one of the cloud centers based on a minimum distance cri-

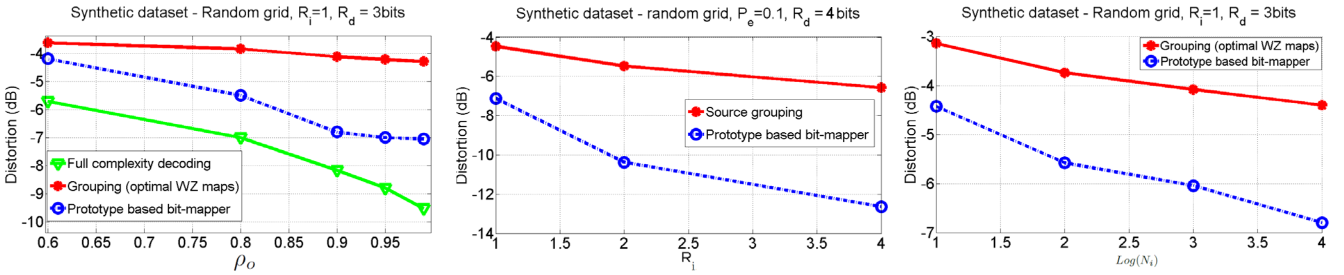

Figure 7: All the three plots are for the Synthetic dataset generated for a random grid of sensors (a) Performance gains with varying correlation coefficient (b) Performance gains as a function of R_i (c) Performance gains with the number of high rate quantization levels

terion leading to the classification of the index tuples into decoding spheres. The reconstructions were purely based on the sphere to which the received index belongs. In the current setting, however, a subset of the transmitted bits are not received at the decoder. The received index tuples are now mapped to one of the cloud centers only based on the bits that are received. The closest cloud center is chosen based on the Hamming distance between the received bits and the corresponding bits in the cloud centers. In other words, since the missing bits can be 0 or 1, we assume the corresponding value to be 1/2 - a value that is equidistant from 0 and 1. Subsequently, the distance (now the absolute value of the difference) is computed between the cloud centers and the received index tuple, with every missing bit replaced by a 1/2, and the source reconstruction is decided based on the nearest center. It is important to note that as a result of the prior {0,1,1/2} subterfuge, the received index tuples are now mapped to one of the cloud centers only based on the bits that were actually received.

The proposed approach essentially mimics an erasure code at the decoder which attempts to recover the lost bits using the correlation across the sources. Observe that the method naturally provides better robustness to channel erasures as it uses all the *received* bits to correct erasures, unlike the source grouping method, which would have estimated the sources only using the received bits within corresponding subsets. The cloud centers and the reconstruction codebooks can be designed using an approach similar to that described in section 4 using a training sequence of source samples and erasure patterns to minimize the expected reconstruction distortion. Also note that, using the same principles, the proposed technique can be easily applied to networks which suffer from a combination of bit errors and erasures.

8.2 Handling Non-Stationary Statistics

In the proposed approach, the system parameters are designed using a training sequence of source and channel samples before deployment. Essentially, this design assumes that the source and channel statistics are stationary in time. This assumption is of course not always valid, and the purpose of this subsection is to briefly outline some options for adapting the proposed approach to non-stationary settings, so as to reap its benefits in such applications. One possible approach to handle time varying statistics is to design the system (collect raw training data) at regular intervals of time and to adapt the system parameters to the new statistics. This entails some additional overhead due to system training and could lead to faster depletion of network resources if the statistics are highly non-stationary. An alternate approach approach is to store multiple sets of system parameters, designed for different statistics, and to use a particular set of parameters by estimating the current average statistic at the sink. The possible implications of these directions on practical deployment of sensor networks will be evaluated as part of our future work.

9. CONCLUSIONS

In this paper, we proposed a new coding approach to large scale distributed compression which is robust to channel errors/erasures. In the proposed approach, the set of possible received index tuples is first classified into groups and then a unique codeword is assigned for each group. This results in low complexity, practically realizable decoders that are scalable to large networks. The classification is achieved using a 'nearest prototype classifier' structure which assists in achieving good error-resilience. We also presented a deterministic annealing based global optimization algorithm for design, which enhances the performance by avoiding multiple poor local minima on the cost surface. Simulation results show that the proposed scheme achieves significant gains as compared to other state-of-the art techniques.

10. REFERENCES

[1] J. Bajcsy and P. Mitran. Coding for the Slepian-Wolf problem with turbo codes. *Proceedings of IEEE GLOBECOM*, 2:1400–1404, 2001.

[2] J. Barros and M. Tuechler. Scalable decoding on factor graphs - a practical solution for sensor networks. *IEEE Trans. on Communications*, 54(2):284–294, February 2006.

[3] R. Cristescu, B. Beferull-Lozano, and M. Vetterli. Networked slepian-wolf: Theory, algorithms and scaling laws. *IEEE Trans. on Information Theory*, 51(12):4057–4073, Dec 2005.

[4] M. Fleming, Q. Zhao, and M. Effros. Network vector quantization. *IEEE Trans. on Information Theory*, 50:1584–1604, Aug 2004.

[5] T. J. Flynn and R. M. Gray. Encoding of correlated observations. *IEEE Trans. on Information Theory*, 33(6):773–787, 1987.

[6] G. Maierbacher and J. Barros. Low-complexity coding and source-optimized clustering for large-scale sensor networks. *ACM Transactions on Sensor Networks*, 5(3), Jun 2009.

[7] D. Miller, A. Rao, K. Rose, and A. Gersho. A global optimization technique for statistical classifier design. *IEEE Trans. on Signal Processing*, 44(12):3108 –3122, dec 1996.

[8] S. Pattem, B. Krishnamachari, and R. Govindan. The impact of spatial correlation on routing with compression in wireless sensor networks. *IEEE Trans. on Sensor Networks*, 4(4), 2008.

[9] S. S. Pradhan and K. Ramchandran. Distributed source coding using syndromes (DISCUS): Design and construction. *IEEE Trans. on Information Theory*, 49:626–643, Mar 1999.

[10] S. Ramaswamy, K. Viswanatha, A. Saxena, and K. Rose. Towards large scale distributed coding. In *Proc. of IEEE ICASSP*, pages 1326 – 1329, Mar 2010.

[11] A. Rao, D. Miller, K. Rose, and A. Gersho. A deterministic annealing approach for parsimonious design of piecewise regression models. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 21:159–173, Feb 1999.

[12] D. Rebollo-Monedero, R. Zhang, and B. Girod. Design of optimal quantizers for distributed source coding. In *Proceedings of IEEE DCC*, pages 13–22, Mar 2003.

[13] K. Rose. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. *Proceedings of IEEE*, 86(11):2210–2239, Nov 1998.

[14] A. Saxena, J. Nayak, and K. Rose. Robust distributed source coder design by deterministic annealing. *IEEE Trans. on Signal Processing*, pages 859 – 868, Sep 2009.

[15] A. Saxena and K. Rose. Distributed predictive coding for spatio-temporally correlated sources. *IEEE Trans. on Signal Processing*, 57:4066–4075, Oct 2009.

[16] K. Viswanatha, S. Ramaswamy, A. Saxena, and K. Rose. A classifier based decoding approach for large scale distributed coding. In *Proc. of IEEE ICASSP*, pages 1513–1516, May 2011.

[17] A. D. Wyner and J. Ziv. The rate-distortion function for source coding with side information at the decoder. *IEEE Trans. on Information Theory*, 22:1–10, Jan 1976.

[18] Z. Xiong, A. Liveris, and S. Cheng. Distributed source coding for sensor networks. *IEEE Signal Processing Magazine*, 21(5):80–94, 2004.

[19] P. Yahampath. Joint source decoding in large scale sensor networks using markov random field models. In *IEEE ICASSP*, pages 2769 – 2772, Apr 2009.

[20] R. Yasaratna and P. Yahampath. Design of scalable decoders for sensor networks via bayesian network learning. *IEEE Trans. on Comm.*, pages 2868–2871, Oct 2009.