Error/Erasure-Resilient and Complexity-Constrained Zero-Delay Distributed Coding for Large-Scale Sensor Networks

KUMAR VISWANATHA, SHARADH RAMASWAMY, ANKUR SAXENA,
and KENNETH ROSE, University of California - Santa Barbara

There has been considerable interest in distributed source coding (DSC) in recent years, primarily due to its potential contributions to low-power sensor networks. However, two major obstacles pose an existential threat to practical deployment of such techniques: the exponential growth of decoding complexity with network size and coding rates and the critical requirement of resilience to bit errors and erasures, given the severe channel conditions in many wireless sensor network applications. This article proposes a novel, unified approach for large-scale, error/erasure-resilient DSC that incorporates an optimally designed, nearest neighbor classifier-based decoding framework, where the design explicitly controls performance versus decoding complexity. Motivated by the highly nonconvex nature of the cost function, we present a deterministic annealing-based optimization algorithm for the joint design of the system parameters, which further enhances the performance over the greedy iterative descent technique. Simulation results on both synthetic and real sensor network data provide strong evidence for performance gains compared to other state-of-the-art techniques and may open the door to practical deployment of DSC in large sensor networks. Moreover, the framework provides a principled way to naturally scale to large networks while constraining decoder 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

Additional Key Words and Phrases: Distributed source-channel coding, large-scale sensor networks, error-resilient coding, complexity-constrained decoding

ACM Reference Format:
Kumar Viswanatha, Sharadh Ramaswamy, Ankur Saxena, and Kenneth Rose. 2014. Error/erasure-resilient and complexity-constrained zero-delay distributed coding for large-scale sensor networks. ACM Trans. Sensor Netw. 11, 2, Article 35 (December 2014), 33 pages.
DOI: http://dx.doi.org/10.1145/2663352

This work is supported by the National Science Foundation, under grants CCF-1016861, CCF-1118075, and CCF-1320599. The material in this article was presented in part at the ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), 2012, Beijing, China [Viswanatha et al. 2012] and at the IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP) 2011, Prague, Czech Republic [Viswanatha et al. 2011].
Authors' addresses: This work was done when the authors were with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA, 93106; K. Viswanatha is currently with Qualcomm Technologies Inc., Corporate Research and Development Division, San Diego, CA, 92121; email: kumar@ece.ucsb.edu; S. Ramaswamy is currently with Google, Inc., 1600 Amphitheatre Pkwy, Mountain View, CA, 94043; A. Saxena is currently with Samsung Research America, 1301, E. Lookout Drive, Richardson, TX, 75082; email: ankur@ece.ucsb.edu; K. Rose is with the Electrical and Computer Engineering Department, University of California, Santa Barbara, CA, 93106; email: rose@ece.ucsb.edu.

DOI: http://dx.doi.org/10.1145/2663352

1. INTRODUCTION AND MOTIVATION

Sensor networks have gained immense importance in recent years, both in the research community and in the industry, mainly due to their practicability in numerous applications. It is essential to operate them at low power to extend network lifetime. Since communication consumes significant power, the primary objective of the system designer is to operate the sensor nodes/motes at the lowest bit rate possible. It is widely accepted that exploiting intersensor 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 recompression is performed at intermediate nodes along the route to the sink [Pattem et al. 2008]. Such techniques tend to be 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 [Cristescu et al. 2005].

The basic DSC setting involves multiple correlated sources (e.g., data collected by a number of spatially distributed sensors) that need to be transmitted from different locations to a central data collection unit/sink. The main objective of DSC is to exploit intersource correlations despite the fact that each sensor encodes its source without access to the other sources. The only information available during the design of DSC is the joint statistics (e.g., extracted from a training dataset). Current research in DSC can be categorized into two broad camps. The first camp derives its principles from channel coding, wherein block encoding techniques are used to exploit correlation [Bajcsy and Mitran 2001; Pradhan and Ramchandran 2003; Xiong et al. 2004]. While these techniques are efficient in achieving good compression and resilience to channel errors and packet losses (using efficient error-correcting codes), they suffer from significant delays and high encoding complexities, which make them unsuitable for an important subset of sensor network applications. The second approach is based on source coding and quantization techniques, which introduce low to zero delay into the system. Efficient design of such low-delay DSC for noiseless systems has been studied in several publications [Fleming et al. 2004; Cardinal and Assche. 2002; Flynn and Gray 1987; Rebollo-Monedero et al. 2003; Saxena et al. 2010] and will be more relevant to us in this article.

This article is motivated by two of the major obstacles that have deterred these approaches from gaining practical significance in real-world sensor networks. First, the decoder complexity grows exponentially with the number of sources and the coding rates, making these conventional techniques (typically demonstrated for two to three sources) infeasible for large sensor networks. As an illustrative example, consider 20 sensors transmitting information at 2 bits per sensor. The base station receives 40 bits and using these it reconstructs estimates of the signals perceived by the 20 sensors. This implies that to fully exploit all information, the decoder has to maintain a codebook of size 20×2^{40}, which requires about 175 Terabytes of memory. In general, for N sensors transmitting at R bits per sensor, the total decoder codebook size would be $N2^{NR}$. Some prior research has addressed this important issue (e.g., [Maierbacher and Barros 2009; Ramaswamy et al. 2010; Yahampath 2009; Yasaratna and Yahampath 2009]). However, methods to date suffer from significant drawbacks, which will be explained in detail in Section 3.

The second important reason for the inefficiency of current DSC methods is the fact that sensor networks usually operate at highly adverse channel conditions and codes designed for noise-free settings provide no resilience to channel errors and erasures. The design of DSC that is robust to channel errors and erasures is highly challenging, as the objectives of DSC and channel coding are counteractive in the sense that one

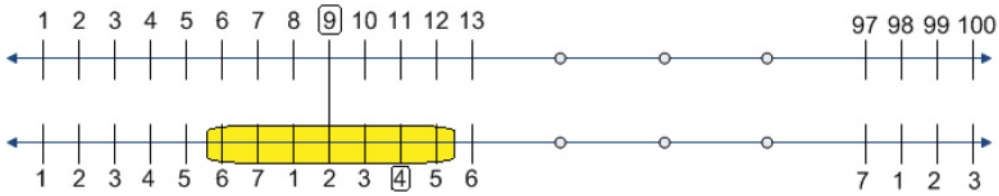

Fig. 1. An example to demonstrate the operating principles of DSC: two sensors observe temperatures that cannot differ by more than 3°. The simple DSC approach is to send the first sensor's temperature (T_1) specifying one of the 100 symbols and the second sensor's temperature (T_2) modulo 7 (specifying one of the seven symbols). The decoder knows that T_2 must be in the range $T_1 - 3$ to $T_1 + 3$ and hence decodes T_2 losslessly.

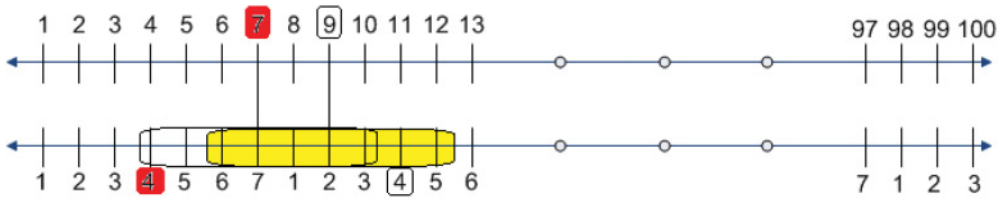

Fig. 2. The sensitivity of DSC to channel errors: for the example shown in Figure 1, a small error in the decoded value of T_1 causes considerable error in T_2.

tries to eliminate intersource dependencies while the other tries to correct errors using the redundancies.

1.1. An Illustrative Example: The Pitfalls of Distributed Quantization

We illustrate the underlying challenge in designing error/erasure-resilient DSC using a pedagogical example. Consider a simple network with two temperature sensors communicating with a base station. The two sensors observe integral temperatures in the range of 1 to 100 and their objective is to convey their respective temperatures to the base station, losslessly. Suppose that the sensors are located sufficiently close to each other so that the difference between their measured temperatures is never greater than 3°. If the two sensors were allowed to communicate with each other, one approach to exploit correlation is to allow the first sensor to communicate its temperature to the base station (i.e., specify one of the 100 symbols) and then transmit the difference from the value measured by the second sensor (i.e., specify one of seven symbols). Distributed source coding provides an elegant approach to perform compression at these rates, even if the two sensors are not allowed to communicate with each other. Let the first sensor send its temperature as before, using one of the 100 transmit symbols. Now, the second sensor sends its temperature modulo 7, which specifies one of seven symbols, and does not require knowledge of the first sensor's measurement. Let us consider the decoding operation. Suppose that the channel is noise-free; that is, the decoder receives error-free information from both the sensors. Let the two sensors' temperatures be denoted by T_1 and T_2, respectively. As the decoder knows T_1 exactly, it knows that T_2 must fall in the range $T_1 - 3$ to $T_1 + 3$. It is easy to verify that all these temperatures lead to different modulo 7 values for all T_1. Hence, the decoder succeeds in decoding T_2 without error, even though the encoder at the second sensor has no knowledge of T_1. Figure 1 depicts the decoding operation when there are no channel errors.

However, now let us consider the same encoding scheme when the channel is noisy. Suppose that T_1 is 9 and T_2 is 11; that is, the sensors transmit 9 and 4, respectively, as shown in Figure 2. Further, let us suppose that T_1 is corrupted by the channel

and received at the decoder as 7 instead of 9. It is easy to verify that the decoder would decode T_2 as 4 instead of 11, when the information from the second sensor is received error-free. Essentially, the nonlinearity introduced at the encoder, to exploit the correlation between the sensors, makes the system highly susceptible to channel errors/erasures. The challenge underlying this example is precisely what makes the design of such distributed source codes, which are robust to channel noise, a very important and challenging problem. On the one hand, the system could be made compression centric and designed to fully exploit intersource correlations. However, this reduces the dependencies across transmitted bits, leading to poor error resilience at the decoder. On the other extreme, the encoders could be designed to preserve all the correlations among the transmitted bits that could be exploited at the decoder to achieve good error resilience. However, such design fails to exploit the gains due to distributed compression, leading to poor overall rate distortion performance. The design approach proposed in Section 5 optimally trades compression efficiency for error resilience to achieve minimum end-to-end distortion at a fixed decoder complexity.

1.2. Contributions of This Article

Motivated by these practical challenges, we address the problem of error/erasure-resilient and zero-delay distributed compression for large-scale sensor networks. We propose a new decoding paradigm, wherein the received bits are first compressed (transformed) down to an allowable decoding rate and the reconstructions are estimated in the compressed space. A nearest neighbor vector quantizer structure is imposed on the compressor, 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 partitioning of the received index space into decoding spheres. The reconstructions are purely based on the sphere to which the received index belongs. These spheres (clouds), when designed optimally, lead to an error/erasure-correcting code that serves the dual purpose of a source-channel decoder. We use design principles from source-channel coding for point-to-point communication and propose a design strategy, based on greedy iterative descent, to learn the system parameters and show that it provides significant improvement in performance over conventional techniques. Motivated by the nonconvex nature of the cost function, we also propose a deterministic annealing-based design algorithm that provides further gains by avoiding poor local minima and by approaching the globally optimal solution. As we will present in Section 4, our methodology overcomes the drawbacks of conventional approaches, enumerated 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.

The contributions of the article are summarized as follows:

—We propose a new decoding structure for large-scale DSC wherein the monolithic decoder is decomposed into a compressor/bit mapper followed by a lookup table of reconstructions/codebooks. The number of cloud centers in the compression/bit mapper is explicitly controlled during the design, thereby allowing for a scalable approach to DSC.
—We impose a nearest neighbor/vector quantizer structure for the compressor/bit mapper that naturally builds resilience to channel errors and erasures.
—We propose a simple greedy iterative descent approach for the joint design of the system parameters based on an available training set of source and channel samples.
—Motivated by the highly nonconvex nature of the underlying cost function, we propose a technique based on deterministic annealing (DA) for the joint design, which

approaches the global optimal solution and further enhances the performance over the naive greedy iterative descent approach.
—We perform extensive simulation tests involving both synthetic and real-world sensor network datasets to demonstrate the advantages the proposed approach exhibits over other state-of-the-art techniques.

We note that preliminary versions of the results in this article have appeared in part in Viswanatha et al. [2012] and Viswanatha et al. [2011]. In this article, in addition to the results in Viswanatha et al. [2012] and Viswanatha et al. [2011], we propose and analyze the important extension of the proposed approach to bit erasures. We provide simulation results, both on real and synthetic datasets, as evidence for significant gains over state-of-the-art techniques in case of bit erasures. Another important issue addressed in this article is the robustness of the proposed approach to time-varying source and channel statistics. We show that the proposed framework is highly robust to mismatch in estimated statistics. Further, we provide a detailed analysis of the design and operational complexity of the proposed approach and provide important practical variants of the scheme that significantly reduce the design complexity.

The rest of the article is organized as follows. In Section 2, we formulate the problem, introduce notation, and discuss the design difficulties of large-scale distributed source coding systems. In the first part of the article, we focus on channel bit errors and postpone the treatment of erasures to simplify the problem formulation. In Section 3, we review related work, and in Section 4, we explain our proposed compression/classification-based approach. Section 5 describes the algorithm for system design, and the associated design complexities are derived in Section 6. The operational complexity of the proposed approach is compared with other state-of-the-art techniques in Section 7, followed by the results in Section 8. In Section 9, we present a methodology to incorporate bit erasures into the proposed framework and demonstrate the gains achievable. We also show in Section 10 the robustness of the proposed technique to mismatch in assumed source and channel statistics. We conclude in Section 11.

2. DESIGN FORMULATION

Before describing the problem setup, we state some of the assumptions made in this article. First, for simplicity, we only consider spatial correlations between sensors and neglect temporal correlations. Temporal correlations can be easily incorporated using techniques similar to that in Saxena and Rose [2009]. Second, in this article, we assume that there exists a separate channel from every sensor to the central receiver; that is, information is not routed in a multihop fashion. However, the method we propose is fairly general and is applicable to the multihop setting. Throughout this article, 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 of-line before deployment of the sensor network or be collected during an initial phase after deployment. During the first half of the article, we consider only channels with bit errors and demonstrate the working principles of the proposed approach. We extend the methodology to incorporate bit erasures in Section 9.

We begin with the conventional (zero-delay) DSC setup. We refer to Maierbacher and Barros [2009] for a detailed description. Consider a network of N sensors (denoted $s_1, s_2 \ldots, s_N$, respectively). The sensors communicate with a central receiver (denoted S) at rates $(R_1, R_2 \ldots R_N)$, respectively, over noisy channels. At regular time intervals, each sensor makes an observation (e.g., temperature, pressure, etc.). These sensor observations are modeled as correlated random variables $X_1, X_2 \ldots X_N$. Sensor s_i encodes X_i using R_i bits for transmission to the receiver. The central receiver attempts

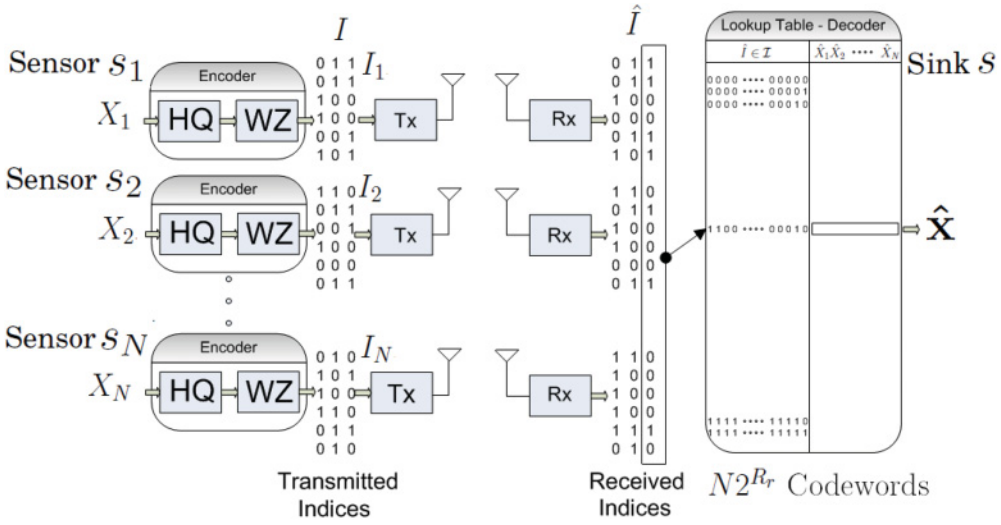

Fig. 3. Basic DSC setup: lookup-table-based decoder.

to jointly reconstruct X_1, $X_2 \ldots X_N$ using bits received from all sensors, as depicted in Figure 3. 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 their reconstruction is minimized.

Encoding at each sensor is performed in two stages. At sensor s_i, the first stage is a simple high-rate quantizer (labeled as "HQ" in Figure 3), $\mathcal{H}_i$, which discretizes the real space into a finite number of nonoverlapping regions N_i. Specifically, $\mathcal{H}_i$ is a mapping that assigns one of the quantization indices to every point on the real space, that is,

$$\mathcal{H}_i : \mathcal{R} \rightarrow \mathcal{Q}_i = \{1 \ldots N_i\}. \tag{1}$$

Note that these quantizers are *high rate* so that they only discretize and may be excluded from joint encoder–decoder design. This is a practical engineering choice and we refer to Saxena et al. [2010] 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 [Yasaratna and Yahampath 2009]), 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:

$$\mathcal{W}_i : \mathcal{Q}_i \rightarrow \mathcal{I}_i = \{1 \ldots 2^{R_i}\}, \tag{2}$$

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

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

A typical example of a WZ map is shown in Figure 4. Observe that the WZ map performs lossy compression. In fact, some regions that 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 counterintuitive at first, if designed optimally, it is precisely these modules that enable exploiting intersource correlations, without intersensor communication. Essentially, the design would

[1]The term "Wyner-Ziv map" is coined after Wyner and Ziv [Wyner and Ziv 1976], who first solved the lossy version of the side information setup introduced by Slepian and Wolf [1973].

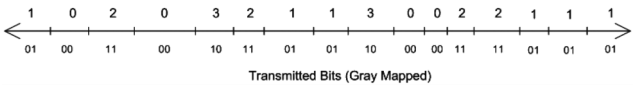

Fig. 4. Example of a typical encoder (irregular quantizer). In this example, $N_i = 16$ quantization regions and $R_i = 2$ bits.

be such that the decoder can distinguish between the distinct quantization regions corresponding to a transmitted index, by accounting for the indices received from other, correlated sources. It is fairly well known in the source coding literature (see Rebollo-Monedero et al. [2003], Saxena et al. [2010], and Yasaratna and Yahampath [2009] and the references therein) that these WZ maps, if properly designed, provide significant improvements in overall rate distortion performance compared to that achievable by regular quantizers operating at the same transmission rates (see also Section 8.4). It is important to note that the WZ maps must be designed jointly prior to operation, using available 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 Maierbacher and Barros [2009] and Saxena et al. [2010].

The encoder at sensor s_i transmits the binary representation of I_i to the remote receiver using a standard binary symmetric modulation scheme. In this article, we assume that each channel introduces an 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 article 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 by $\hat{I}_i \in \mathcal{I}_i$ as shown in Figure 3. In vector notation, $I = (I_1, I_2 \dots I_N)$ and $\hat{I} = (\hat{I}_1, \hat{I}_2 \dots \hat{I}_N)$, where 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

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

Usually the decoder is assumed to be a lookup table, which has the reconstruction values stored for each possible received index as shown in Figure 3. 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 refer to the total storage of the lookup table as the *decoder complexity*. In most prior work, DSC principles have been demonstrated with a few (typically two to three) sources. 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 networks even with a moderately large number of sources. In the next section, we describe some related prior work to address this huge exponential storage at the decoder.

It is worthwhile to note that the encoding operation in the previous scheme involves a simple quantization of the source samples followed by a direct lookup of the transmission index. The total storage at each encoder includes its high-rate quantization

codebook (of size $|\mathcal{Q}_i|$) and the WZ maps (of size $|\mathcal{Q}_i|R_i$). For typical values of $|\mathcal{Q}_i|$ and R_i, the encoder complexity is sufficiently modest 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 Pradhan and Ramchandran [2003] and Xiong et al. [2004], which entail significant encoding delay and complexity. Hence, hereafter, our concern will be only toward addressing decoder complexity, assuming that the encoders can be easily implemented on a physical sensor mote.

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 degree of correlation [Maierbacher and Barros 2009] and to separately 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 intercluster dependencies and hence would yield suboptimal performance. 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 the received bits to correct errors. Also, in most prior work, source grouping is designed only based on the source statistics while ignoring the impact of channel conditions. Indeed, it is a much harder problem to devise good source grouping mechanisms that 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 for practical implementation of the decoder [Maierbacher and Barros 2009; Yahampath 2009; Yasaratna and Yahampath 2009]. In this approach, the decoder computes the reconstructions on the fly by estimating the posterior probabilities, $P(q_i|\hat{I})$, for quantization index q_i given received index $\hat{I}$. Such an approach requires the storage of the high-rate quantization codewords at the decoder, which grow linearly in N. However, to compute the posterior probabilities, $P(q_i|\hat{I})$, using Bayes rule, we have

$$P(\tilde{q}_i|\hat{I}) = \gamma \sum_{\boldsymbol{Q}:q_i=\tilde{q}_i} P(\hat{I}\big|I(\boldsymbol{Q}))P(\boldsymbol{Q}), \tag{5}$$

where γ is a normalization constant, $\boldsymbol{Q} = (q_1, q_2 \ldots, q_N)$, and $P(\hat{I}\big|I(\boldsymbol{Q}))$ is the conditional PMF of the channel. The previous marginalization requires an exponential number of operations to be performed at the decoder, as well as exponential storage required to store the probabilities $P(\boldsymbol{Q})$.

To limit the computational complexity, prior work (e.g., [Barros and Tuechler 2006; Maierbacher and Barros 2009; Yasaratna and Yahampath 2009]) has proposed clustering the sources and linking the clusters in a limited complexity Bayesian network (or a factor graph), thereby using message passing algorithms to find $P(q_i|\hat{I})$ at affordable complexity. These approaches provide significant improvement in distortion over simple source grouping methods at fixed transmission rates and channel SNRs, as they exploit intercluster correlations. However, a major drawback of such techniques is that they require the storage of the Bayesian network/factor graph at the decoder. While this storage grows linearly in N, it grows exponentially with the rate of the high-rate quantizers. Specifically, if $N_i = 2^{R_q} \forall i$, then the storage of the Bayesian network grows as $\mathcal{O}(N2^{MR_q})$, where M is the maximum number of neighbors for any source node in the Bayesian network. Typically [Rebollo-Monedero et al. 2003; Saxena et al. 2010; Maierbacher and Barros 2009], R_q is chosen as $R+3$ or $R+4$ for the Wyner-Ziv maps to exploit the intersource correlations effectively. This impacts the efficiency of the

Bayesian network approach as the gains in distortion, due to introducing the Bayesian network, come at the cost of the excess storage required to store the Bayesian network. We will show in our results that Bayesian network-based methods may even underperform source grouping techniques for moderate values of N at a *fixed storage*. Hence, though counterintuitive at first, it is sometimes beneficial to group more sources within a cluster instead of connecting the clusters using a Bayesian network.

We further note that the storage required for transition probabilities in the Bayesian network can be significantly reduced if the joint source densities are parameterized (e.g., as multivariate Gaussian). However, such approximations are restrictive and prone to estimation inaccuracies and hence lead to suboptimal designs for more general/real-world source and channel statistics, as has been observed in Yasaratna and Yahampath [2009]. We next describe our proposed classification-based approach for decoding that 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, which may have been corrupted by the channel. The lookup 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 that 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 that belong to the same group. Essentially, we decompose the monolithic decoder, which was a simple lookup table, into a compressor/classifier/bit mapper followed by a lookup table of reconstructions. Note that the classification need not be the same 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 fully general bit mapper would require us to store the class information for every possible received index that 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 that clusters the received index tuples into clouds based on a minimum distance criterion, as shown in Figure 5. Such a modification to the bit mapper may incur some loss in optimality but provides a twofold 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 effectively implements an error-correcting mechanism at the decoder by assigning the same codeword to several nearby received indices.* If the probability of channel error is not excessive, 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 [Rose 1998] and the structure is normally termed "nearest prototype classifier." Technically, these prototypes can be defined in any subspace (e.g., $\mathcal{R}^N$) 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 restrict the prototypes to the binary hypercube, $\mathcal{I}$, and choose as distance metric the Hamming distance between the binary representations of the received indices and prototypes. Recall that the Hamming distance between two binary tuples is defined as the number of positions at which the

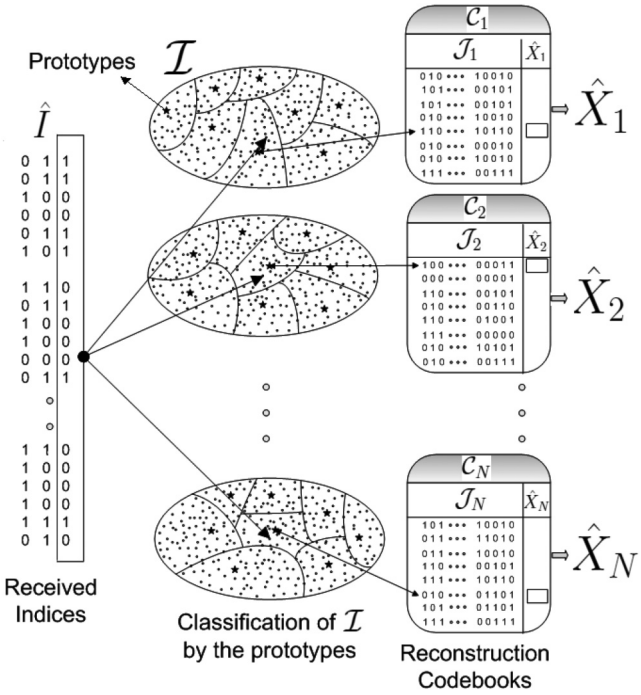

Fig. 5. Prototype-based bit mapper approach to decoding.

corresponding bits differ. 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

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

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

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 $\hat{x}_i = \mathcal{C}_i(S_{i,j})$; that is, 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. SYSTEM DESIGN ALGORITHM

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 that 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 previous equation, we have assumed the distortion metric to be the mean squared error (MSE) and given equal weightings to all the sources similar to Maierbacher and Barros [2009] and Yasaratna and Yahampath [2009]. We note that

Table I. Summary of Notation Used

Notation	Description
N	Number of sources
R_i	Rate of transmission of source i
$\mathcal{T}$	Training set
$\mathcal{J}_i$	Set of prototypes for decoding source i
S_{ij}	jth prototype associated with decoding source i
I and $\hat{I}$	Transmitted and received indices
$\mathcal{I}$	Set of all possible indices
D_{avg}, H, J, T	Average distortion, entropy, Lagrangian, and temperature
$P_i(j\|k)$	Probability of associating the kth training sample
	to the jth prototype for decoding source i
β_i	Inverse pseudo temperatures

the MSE distortion metric measures the average energy of the error in reconstruction and hence is a measure of the quality of reconstruction. Nevertheless, the design methodology we propose is extendable and applicable to any other general distortion measure. A summary of the notation used in the rest of the section is listed in Table I.

We first note that the high-rate quantizers are designed separately using a standard Lloyd-Max quantizer design technique [Gersho and Gray 1991] to minimize the respective average squared errors. The challenging part is to design the Wyner-Ziv maps jointly with the prototypes and the reconstruction codebooks to minimize D_{avg}.

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 [Rao et al. 1999]. The main challenge arises because, 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, which is different from the space where quantization is performed. One straightforward design approach is to employ a greedy iterative descent technique that 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 Equation (8), the prototypes are parameters in a highly nonconvex function, which makes the greedy approach likely to get trapped in poor local minima (even if multiple random initializations are tried), thereby leading to suboptimal 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 an optimization technique based on deterministic annealing (DA) that 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, in contradistinction with recent source-channel design approaches such as Maierbacher and Barros [2009] and Yasaratna and Yahampath [2009], which optimize the WZ maps for the noiseless scenario (without the knowledge of the channel) and then optimize the decoder codebooks for the given channel statistics. We particularly study the gains due to this jointly optimal design later in Section 8.

We note that the computational complexity of the DA-based design scheme may be deemed moderately high for certain applications. We emphasize that the design is normally performed only once, offline, and hence design complexity tends to be a

less critical constraint for most practical applications. Nevertheless, should the design complexity be strictly constrained, one may use the greedy iterative descent algorithm, with very few initializations. We note that the proposed framework performs better than conventional techniques even when a greedy iterative descent technique is employed to optimize the system parameters. Deterministic annealing offers further gains over the greedy iterative descent technique. We next present the DA-based algorithm for the joint design of the system parameters. We omit explicitly stating the steps for the greedy iterative descent scheme as the optimization steps for the last stage of DA coincide with the greedy iterative descent approach. However, instead of a random guess, DA uses the equilibrium solution of an annealing process as an initialization for the iterative descent.

5.1. Design by Deterministic Annealing

A formal derivation of the DA algorithm is based on principles borrowed from information theory and statistical physics. Here, the design problem is cast in a probabilistic framework, where the standard deterministic bit mapper is replaced by a random mapping that maps a 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 principles underlying DA can be found in Rose et al. [1992] and Rose [1998].

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 mapping probabilities are denoted by $P_i(j|k)$ $\forall i \in (1, \ldots, N)$, $j \in (1, \ldots, |\mathcal{J}_i|)$, $k \in (1, \ldots, |\mathcal{T}|)$; that is, the received index tuple for training sample k is mapped to prototype j in $\mathcal{J}_i$ with probability $P_i(j|k)$. Hence, the average distortion is

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

where $x_i(k)$ is training sample k of sensor 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*, that is,

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

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 that minimize the average distortion.

Further, we impose the nearest prototype structural constraint on the bit mapper partitions by appropriately choosing a parameterization of the association probabilities. Similar methods have been used before in the context of design of tree-structured quantizers [Rose 1998], generalized VQ design [Rao et al. 1999], and optimal classifier design [Miller et al. 1996]. The principle of entropy maximization can be used to impose a nearest prototype structure, leading, at each temperature, to association probabilities that are governed by the Gibbs distribution [Rose 1998]:

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

where β_is are called the inverse pseudo-temperatures. Observe that this parameterization converges to the nearest prototype classifier as $\beta_i \to \infty$.

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

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

The DA algorithm minimizes D_{avg} in Equation (9), with a constraint on the entropy of the system, Equation (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

$$J = D_{avg} - TH. \tag{13}$$

Initially, when T is set to a very high level, our objective is to maximize H and hence all the β_is are very close to 0. This leads to a very random 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. All β_is gradually increase as T reduces, thereby making the association distribution less random. Finally, as $T \to 0$, all the $\beta_i s \to \infty$ 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 next.

5.1.1. Wyner-Ziv Map Update. At fixed T, the WZ map update rules are given by

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

$\forall i \in (1, \ldots, N), m \in \mathcal{Q}_i$, where $J(\mathcal{W}_i(m) = l)$ denotes the Lagrange cost obtained for 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 exponentially in N. Hence, for large sensor networks, it is impractical to exhaustively search for the best prototype in the set $\mathcal{I}$, in every iteration. Therefore, in each step, we find an incrementally better solution among all the neighboring prototypes, which are at a Hamming distance of one from the current solution. Mathematically, for fixed Wyner-Ziv maps and reconstruction codebooks, the prototype update rule is

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

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 the prototype neighborhood about S_{ij}. For an illustrative example, consider three sensors transmitting at 2 bits/sensor. Let the current prototype configuration associated with decoding the first source be 110100. Then, during the prototype update step, the best prototype is chosen among the following possible solutions:

$$\{110100, 010100, 100100, 111100, 110000, 110101, 110101\}$$

Observe that this low-complexity update step requires us to compute the Lagrange cost associated with $\sum_{i=1}^{N} R_i + 1$ prototypes (seven in this example), as opposed to $2^{\sum_{i=1}^{N} R_i}$ (64 in this example) in case of an exhaustive search.

5.1.3. β_i Update. As β_is take 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 $\forall i$ are given by

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

Then the update rule for β_i is given by

$$\beta_i^* = \beta_i - \triangle \frac{\delta J}{\delta \beta_i}, \tag{17}$$

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 that minimizes J for any fixed encoders, prototypes, and β_is is given by

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

The complete set of steps for DA is summarized in Flowchart 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 Equations (14), (15), (17), and (18) until the system reaches equilibrium. This equilibrium is perturbed and used as an initialization for the next temperature. This annealing procedure is continued until T approaches zero. In practice, the system is "quenched" once T is small enough; that is, T is set to zero and the bit mapper is made "hard," once the entropy becomes sufficiently small. Note that the optimization steps, at $T = 0$, coincide with 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 the continuity of phase transitions in the process, DA achieves the global minimum [Rose et al. 1992; Rose 1998], 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). In practice, however, α is restricted based on the available design time. In our simulations, we observed that using $\alpha = 0.98$ is enough to achieve significantly better solutions, compared to the greedy descent approach.

6. DESIGN COMPLEXITY

This section describes the design complexity for the proposed algorithm. We study the complexity of each of the design stages independently. These steps are integral parts of both the greedy and the DA-based design algorithms. Throughout this section, we assume that every source sends information at rate R and all the high-rate quantizers operate at rate $R_q \geq R$. Further, we assume that the decoding rate is $R_{d_i} = R_d$ ($R_d \leq NR$)

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

Flowchart 1: DA Approach for System Design

Inputs : N_i (Numberof high-rate quantization indices),
R_i (Transmission rates),
R_{d_i} (Decoding rate, i.e., $|\mathcal{J}_i| = K_i = 2^{R_{d_i}}$),
$\mathcal{T}$ (Trainingset), $T_{max}(\sim 1-10)$, $T_{min}(\sim 10^{-5}-10^{-4})$,
$\beta_{min}(\sim 0.1-0.2)$, $H_{min}(\sim 0.1-0.2)$, $\alpha < 1$ (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 $S_{ij} = \text{median}(\hat{I}(\mathbf{x}),\ \mathbf{x} \in \mathcal{T})\ \forall i \in (1,\ldots,N), j \in (1,\ldots,\mathcal{J}_i)$.
(3) *Compute:* $P_i(j|k)$ using Equation (11) and $\mathcal{C}_i(j)$ using Equation (18).
(4) *Update:*
— WZ maps using Equation (14).
— Prototypes using Equation (15).
— β_i using Equation (17), and then compute $P_i(j|k)$ using Equation (11).
— $\mathcal{C}_i(j)$ using Equation (18).
(5) *Convergence:* Compute J and H using Equations (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 Equation (10) and perform last iteration for $T = 0$. Then STOP.
(7) *Cooling:*
— $T^* \leftarrow \alpha T$.
— Perturb prototypes: $S^*_{ij} \leftarrow s \in \text{neighborhood}(S_{ij})$, where s is chosen randomly.
— Perturb $\beta^*_i \leftarrow \beta_i + \delta$ for small $\delta > 0$ generated randomly.
— Go to step (4).

for all sources. This implies that the number of prototypes for decoding any source is $|\mathcal{J}_i| = |K_i| = 2^{R_d} \forall i$. We denote the training set by $\mathcal{T}$ and the size of the training set by $|\mathcal{T}|$.

6.1. Design of the High-Rate Quantizers

Recall that the high-rate quantizers are designed independently at each source, to minimize the respective average squared error, using a standard Lloyd-Max quantizer design algorithm [Gersho and Gray 1991] before the joint design of the remaining system parameters. During each iteration, all the training samples are clustered to the nearest codeword and then the codewords are updated to the centroid of the training samples within each cluster. Hence, the design complexity of the high-rate quantizers grows as $\mathcal{O}(N2^{R_q}|\mathcal{T}|)$. This is a low-complexity step as compared to the joint design of all the other system parameters. We note that there are techniques to accelerate the scalar quantizer design, but these will not be considered here.

6.2. Wyner-Ziv Map Update

During each iteration, the search for the optimal WZmap involves finding $2^R|\mathcal{T}|$ distortion values for each source, calculated as $\sum_{i=1}^{N}\sum_{j\in\mathcal{J}_i} P_i(j|k)(x_i(k)-\hat{x}_i(j))^2$. Each training sample is then assigned to one of the 2^R transmit indices that minimizes the average reconstruction distortion. The average entropy of the system does not depend on the WZ maps and hence need not be recomputed during the WZ map update. Therefore, the complexity cost during each iteration of the design of WZ maps grows as $\mathcal{O}(N^2 2^{R+R_d}|\mathcal{T}|)$.

6.3. Prototypes Update

Recall that during each iteration of the design, an incrementally better prototype is chosen among all the neighboring prototypes that are at a Hamming distance of one from the current solution. There are overall $N2^{R_d}$ prototypes, each having NR neighbors at a Hamming distance of one. To find the complexity of the prototype update step, we first derive the complexity of evaluating the Lagrangian cost for any fixed configuration of the prototypes.

The average distortion is computed using Equation (9), which requires computations on the order of $\mathcal{O}(2^{R_d}|\mathcal{T}|)$ per source. Note that to compute the total entropy of the system, we need to first find the probabilities $P_i(j|k)$ using Equation (11). This step requires us to find the distances between the prototypes and the received index tuples in the training set. As the received tuples are NR bit vectors and there are 2^{R_d} prototypes, finding these distances requires $\mathcal{O}(NR2^{R_d})$ computations and hence the total complexity for finding $P_i(j|k)$ grows as $\mathcal{O}(NR2^{R_d}|\mathcal{T}|)$ per source. The total entropy of the system is computed using Equation (12), which entails a complexity of $\mathcal{O}(N2^{R_d}|\mathcal{T}|)$ per source. Therefore, the complexity for evaluating the Lagrangian cost is dominated by the step that involves finding $P_i(j|k)$, whose complexity grows as $\mathcal{O}(NR2^{R_d}|\mathcal{T}|)$.

From this analysis, it appears as though the computational complexity required to update the prototypes during each iteration grows as $\mathcal{O}(N^3R^24^{R_d}|\mathcal{T}|)$, which is cubic in N. However, certain tricks allow us to reduce the complexity further, as illustrated next.

Let b_1, b_2, and c be three NR bit vectors. Let b_1 and b_2 differ in exactly one position. If the Hamming distance between b_1 and c is given, then we require only two additional computations to find the distance between b_2 and c. This property allows us to reduce the number of computations required to find $P_i(j|k)$. Specifically, let the distances between the received tuples in the training set and the prototypes be given, for the current solution. To find an incrementally better prototype configuration, these distances must be recomputed for all the prototypes in the neighborhood of the current solution. However, this step requires just two additional computations per prototype, as compared to the NR computations, which would be necessary with no prior knowledge. It is easy to verify that the total complexity required to update the prototypes now reduces to $\mathcal{O}(N^2R4^{R_d}|\mathcal{T}|)$, which is only quadratic in N.

6.4. β_i Update

The β_is are updated using a gradient descent technique according to Equation (17). It is clear from Equation (17) that the number of computations required to find each of the gradients grows as $\mathcal{O}(4^{R_d}|\mathcal{T}|)$ and hence the total complexity cost for updating all the β_is is $\mathcal{O}(N4^{R_d}|\mathcal{T}|)$.

6.5. Reconstruction Codebook Update

Updating the reconstruction codebooks is a fairly simple operation. During each iteration, the codebooks are updated using Equation (18), and it is easy to show that the codebook update involves $\mathcal{O}(N2^{R_d}|\mathcal{T}|)$ operations.

It is clear from the previous description that the design complexity is primarily dominated by two stages: the WZ map update and the prototype update. Therefore, the total complexity for the proposed design algorithm grows as $\mathcal{O}(N^2(R4^{R_d} + 2^R2^{R_d})|\mathcal{T}|)$, which is quadratic in N. It is important to note that the order of growth in design complexity remains the same for both the DA-based design scheme and the greedy iterative design algorithm. In practice, the DA approach has a larger constant and requires more computations compared to the greedy approach for a single initialization. However, as the greedy approach has to be run over multiple random initializations

Table II. Order of Growth in Storage Complexities

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	$N2^{R_d}F$	$N^2R2^{R_d}$

to avoid poor solutions, the exact comparison of design complexities is difficult and depends on the actual source-channel distributions. A generally accepted and observed fact (see Rose et al. [1992] and Rose [1998]) 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 nonconvex optimization functions.

7. OPERATIONAL COMPLEXITY

In this section, we compare the computational and storage complexities during operation of all 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 $R_q \geq R$. 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 implies 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. For the proposed approach, this implies that the number of prototypes for decoding any source is $|\mathcal{J}_i| = |K_i| = 2^{R_d} \forall i$.

7.1. Computational Complexity

First, we note that the computational complexity during operation of all three approaches is polynomial in N. It is easy to observe that the decoder in the source grouping method has literally no computations to perform; that is, 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 $\mathcal{O}(N2^{R_qR_d/R})$. On the other hand, the proposed prototype-based bit mapper approach finds the closest prototype for every received index tuple, which requires $\mathcal{O}(R_dRN^2)$ bit comparisons. Note that, although the complexity grows as N^2, it requires only bit comparisons and will incur much fewer machine cycles than that required for implementing each iteration in the Bayesian network approach. Additionally, the complexity for the proposed approach grows only linearly in R and R_d as opposed to the exponential growth in these rates for the Bayesian network approach. As all three methods can be implemented with affordable computational complexities in practice, we hereafter assume they are feasible as far as computational complexity is concerned and focus on their storage requirements.

7.2. Storage Complexity

Table II summarizes the order of growth in storage as a function of N,R, R_q, and R_d for all 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.

The codebook storage required for each of the three approaches is considerably easier to derive. For the source grouping approach, R_d bits are used to decode each source and hence the total number of codewords to be stored is $N2^{R_d}$. Similarly, for 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 $N2^{R_d}F$. For the Bayesian network approach, it is sufficient to store the high-rate quantization codewords for all the sources. This entails a complexity of $N2^{R_q}$, leading to a total codebook storage as indicated in Table II.

As for module storage, the source grouping method 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. The Bayesian network approach requires complexity that grows as $\mathcal{O}(N\frac{R_d}{R}\log_2(N))$ bits to store the parent node information for each source. However, additional storage is required to store the transition probabilities, which grow as $\mathcal{O}(N2^{R_q R_d/R}F)$. The prototype-based bit mapper approach requires the storage of 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 glance at Table II would suggest that, since the prototype-based bit mapper approach requires a module storage that grows as N^2 in the number of sources, it entails a very high overhead. 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 for the proposed approach is modest and the distortion gains obtained more than compensate for the minimal loss due to excess storage.[3] On the other hand, within these typical ranges, the Bayesian network approach entails a storage requirement that is significantly higher than the other two methods and hence yields higher distortion at a prescribed storage. Note that the values in Table II indicate the order of growth of storage complexities and hence are accurate only up to a constant. In all our simulations, we consider the exact storage required and not the order expressions from Table II.

8. RESULTS

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

1) **Synthetic dataset:** A toy dataset consisting of 10 synthetic sources, randomly deployed on a square grid with dimensions of 100m $\times$ 100m, was generated according to a multivariate Gaussian distribution (the grid is shown in Figure 6). All sources were assumed to have zero mean and unit variance. The correlation was assumed to decay 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 10,000 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-world dataset we used was collected by the Intel Berkeley Research Lab, CA.[4] Data were collected from 54 sensors they deployed between February 28 and April 5, 2004. Each sensor measured temperature values once every 31 seconds.[5] We used data from the top 25 sensors that collected the highest number of samples and retained time samples when all sensors recorded data. Half the dataset was used to train the system and the remaining half was used as the test set.

3) **Rainfall dataset:** As a second real-world dataset, we used the rainfall dataset of Pattem et al. [2008].[6] This dataset consists of the daily rainfall precipitation for the

[3]Note that if $N >> 500$, then the optimal approach would be to group $\sim$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.

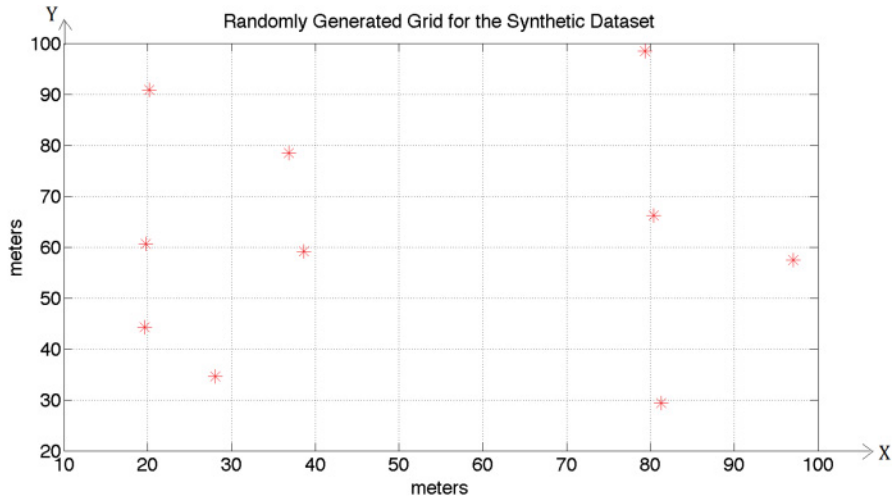

Fig. 6. Randomly generated synthetic sensor grid. Stars denote the sensor motes.

Pacific Northwest region over a period of 46 years. The measurement points formed a regular grid of 50km x 50km cells over the entire region under study. Again, half the data were used for training and the remaining to test the system. Note that the intersource 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 the performance depends on the crossover probability or the error probability of the effective BSC seen by each bit. We denote the crossover probability by P_e, that is, $P(1|0) = P(0|1) = P_e$. Note that P_e is directly related to the channel SNR (CSNR) as $P_e = Q(\sqrt{CSNR})$; that is, $P_e = 0.1$ corresponds to a CSNR of about $2.15dB$. 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.

8.1. Complexity–Distortion Tradeoff

Figures 7, 8, and 9 show the total storage (complexity) versus the distortion tradeoff for the three datasets, respectively. 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 for the synthetic dataset. We will present results at higher transmission rates in Section 8.4. 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 tradeoff curve. We show results obtained using all three decoding methods: source grouping where the grouping is done using the source-optimized clustering approach described in Maierbacher and Barros [2009], Bayesian network as described in Yasaratna and Yahampath [2009], and the prototype-based bit mapper approach proposed in this article. 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 [Yasaratna and Yahampath 2009]. We study the gains due to this optimal design in the following section. For comparison, we also include the performance obtained for designs using the greedy iterative descent

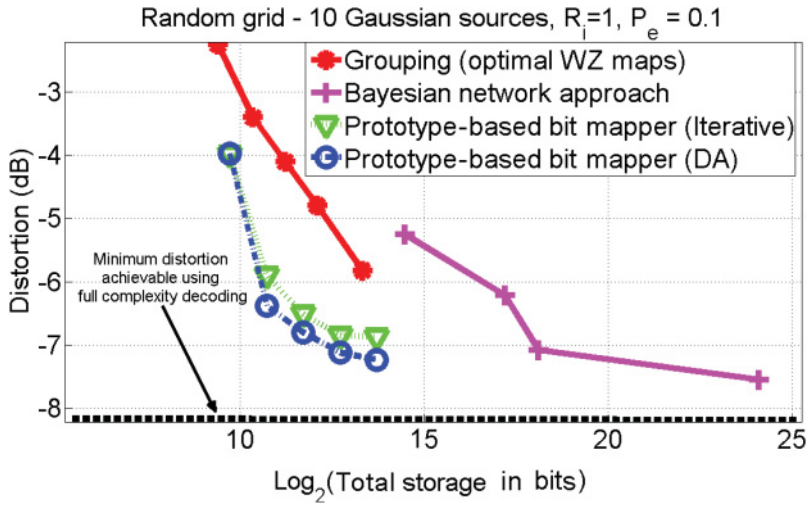

Fig. 7. Total storage versus distortion for the synthetic dataset, $R_i = 1 \forall i$ and $P_e = 0.1$.

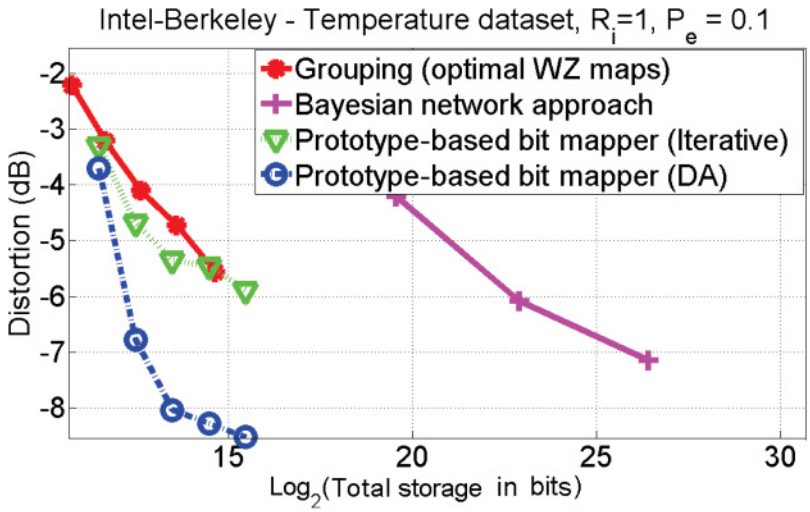

Fig. 8. Total storage versus distortion for the temperature sensor dataset, $R_i = 1$, $P_e = 0.1$.

approach (optimized over up to 25 random initializations) along with that achieved using DA.

Figure 7 shows the result obtained for the synthetic dataset using $\rho_0 = 0.9$ and $P_e = 0.1$. We see improvements of over 2dB in distortion compared to the source grouping technique at a fixed storage. Alternatively, the total storage can be reduced by a factor of 10 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 while the Bayesian network-based decoder gains substantially over the 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

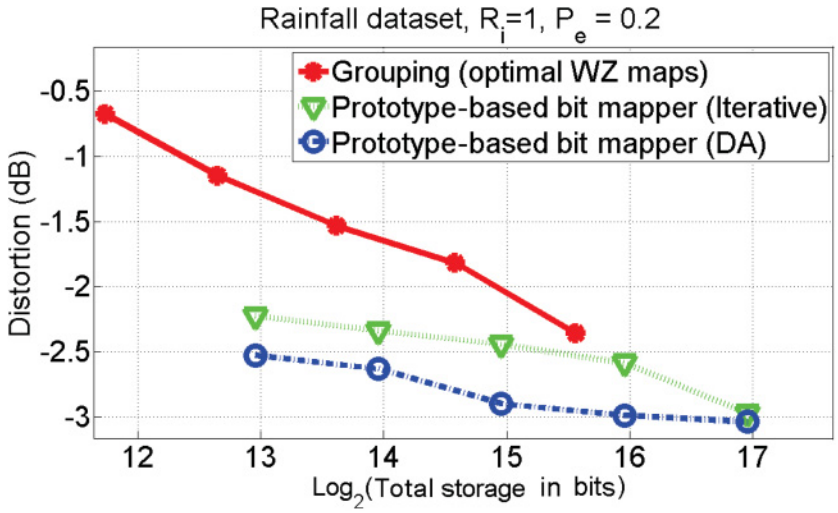

Fig. 9. Total storage Versus Distortion for the Rainfall dataset, $R_i = 1$, $P_e = 0.2$.

as DA, as the probability of getting trapped in local minima is low after 25 runs for smaller networks.

Figures 8 and 9 show the performance obtained for the temperature sensor dataset and the rainfall dataset at P_e of 0.1 and 0.2, respectively. For the temperature sensor dataset, we see gains of over 2.5dB in distortion at fixed storage over the source grouping approach. For the rainfall dataset, gains are around 1dB in distortion. 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. We therefore see more gains in the temperature sensor dataset as opposed to the rainfall dataset. Figure 8 also illustrates 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] Observe that for such large networks, the DA-based design significantly outperforms the greedy iterative descent approach, which in itself performs better than the source grouping and the Bayesian-network-based techniques. This clearly demonstrates the highly nonconvex nature of the optimization function and the susceptibility of greedy iterative descent techniques to poor local minima. Unless the *design* complexity is severely constrained, it is better to design the system using DA. Hence, hereafter, we only show results for DA, noting that the greedy approach, although it outperforms the conventional techniques, is susceptible to poor local minima 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 size 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.

8.2. 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 ≥ -1.5dB). For all the simulations, we have chosen

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

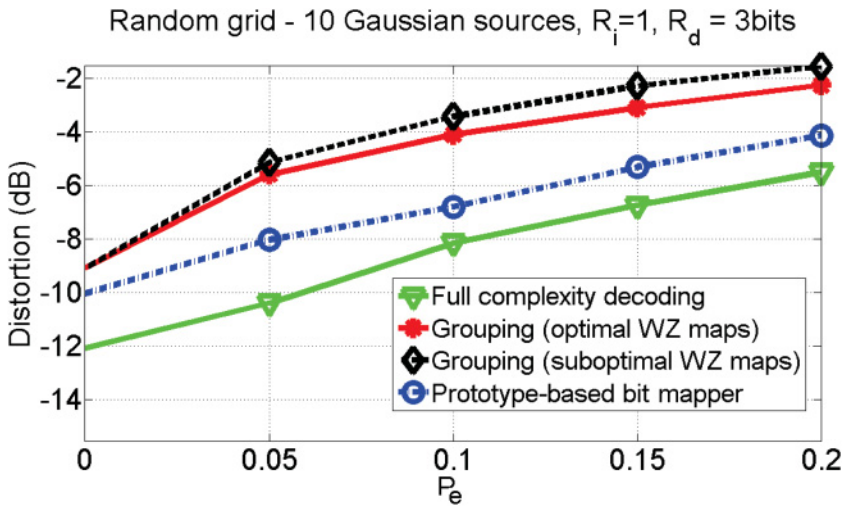

Fig. 10. P_e versus distortion for the synthetic dataset, $R_i = 1$ and $R_d = 3$.

$R_i = 1$ and $R_d = 3$. Figure 10 shows the distortion obtained versus P_e for the synthetic dataset. For the source grouping approach, we plot two 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 when the knowledge of channel statistics is absent, that is, 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. Recall that in all prior work related to DSC design for large networks, the system was designed assuming that the channel statistics are unknown. Clearly, optimal design of the WZ maps for the given channel provides about a 0.5dB improvement in distortion. Further, major improvements of over 2dB are due to the error resilience provided by the proposed decoder structure. We see similar behavior even for the two real-world datasets in Figures 11 and 12. 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.

8.3. Performance Versus Network Size

In this section, we study how the gains vary with the size of the network. To eliminate the impact of specific random realizations of the deployment grid, we consider a uniformly placed, linear grid of sensors between two fixed points. We increase the number of sensors from six to 90 while keeping the transmission and decoding rates fixed. We assume a correlation model that falls off exponentially with the distance and assume P_e to be 0.2 throughout. Figure 13 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 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 only uses bits within the given cluster.

8.4. Performance Dependance on Other Design Parameters

In the following, we vary different design parameters and study the performance gains on the synthetic dataset described in the beginning of this section.

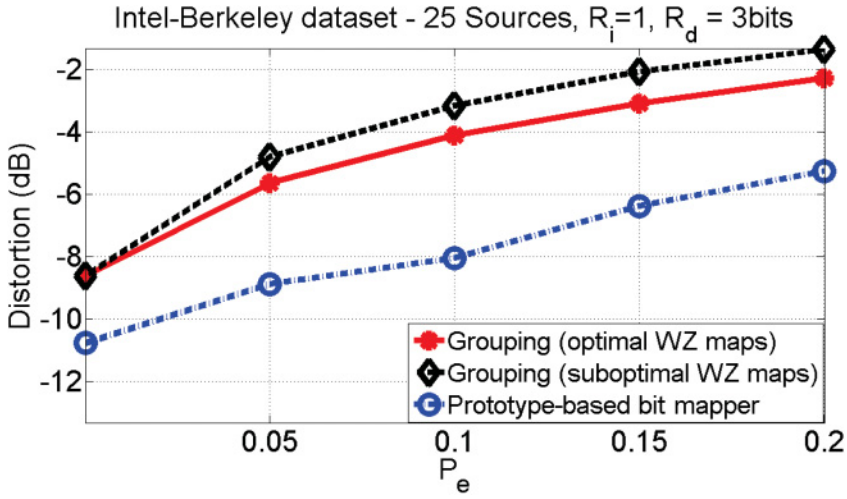

Fig. 11. P_e versus distortion for the temperature sensor dataset, $R_i = 1$ and $R_d = 3$.

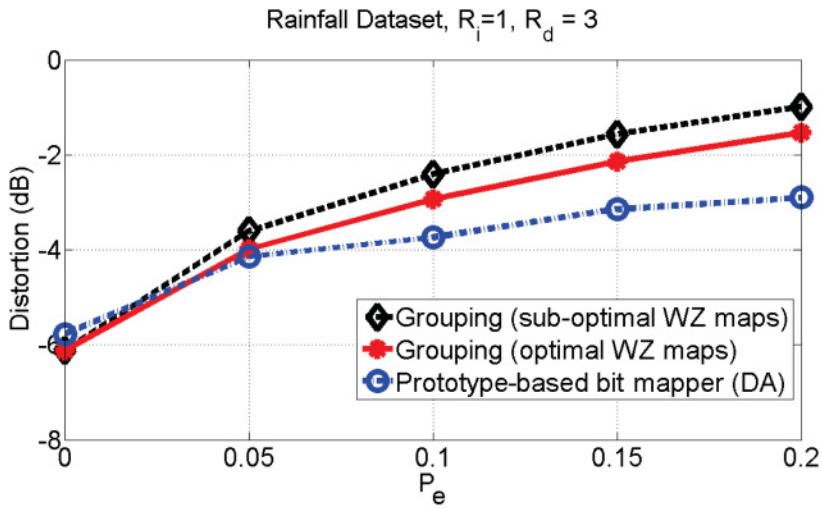

Fig. 12. P_e versus distortion for the rainfall dataset, $R_i = 1$ and $R_d = 3$.

8.4.1. The Correlation Parameter (ρ_o). Figure 14 depicts the distortion as a function of ρ_o. The plot shows results obtained by the source grouping method, the proposed approach, and the optimal full-complexity design, which uses all the received bits. A 3dB improvement of the proposed approach over the grouping method at very high correlations provides further evidence of improved error resilience.

8.4.2. Transmitted Bits (R_i). In this section, we compare the competitor's performances over a range of transmission rates. Figure 15 shows the comparison plots. We consider three 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. First, as R_i increases, the decoder has access to more correlated bits, which can be used efficiently for correcting more errors. Second, 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

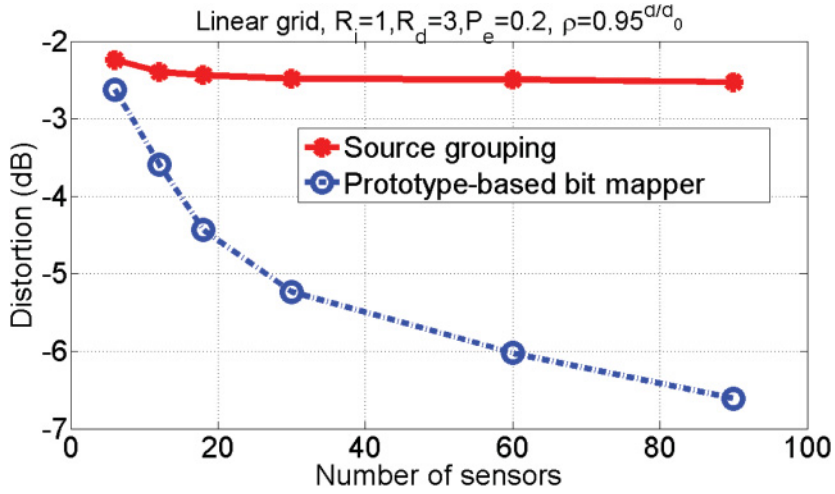

Fig. 13. Variation of reconstruction distortion with the number of sources deployed on a linear grid placed uniformly along a length of 10 kilometers. Correlation model is assumed to be $0.95^{dist(Km)}$, $R_i = 1$ bit and $P_e = 0.2$.

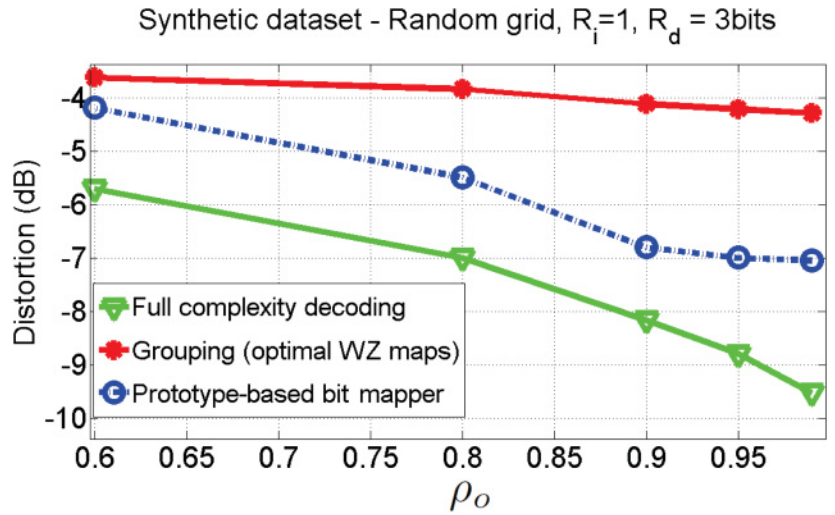

Fig. 14. Performance gains with varying correlation coefficient for the synthetic dataset.

approach does not exploit either of these advantages and hence suffers significantly more at higher transmission rates. It must, however, be emphasized that the difficulty arises at very high transmission rates as the proposed design complexity grows as $\mathcal{O}(N^2(R4^{R_d} + 2^R 2^{R_d})|\mathcal{T}|)$, that is, exponentially in the transmit and decoding rates.

8.4.3. Rate of Quantizers (R_q). All results so far have focused on the decoder structure. It is also of interest to consider the importance of the encoder structure/WZ maps. Figure 16 shows the decrease in distortion as the rates of $\mathcal{H}_i$ are increased from $R_q = 1$ to 4 bits while keeping the transmission rates 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 a 1.5dB improvement for

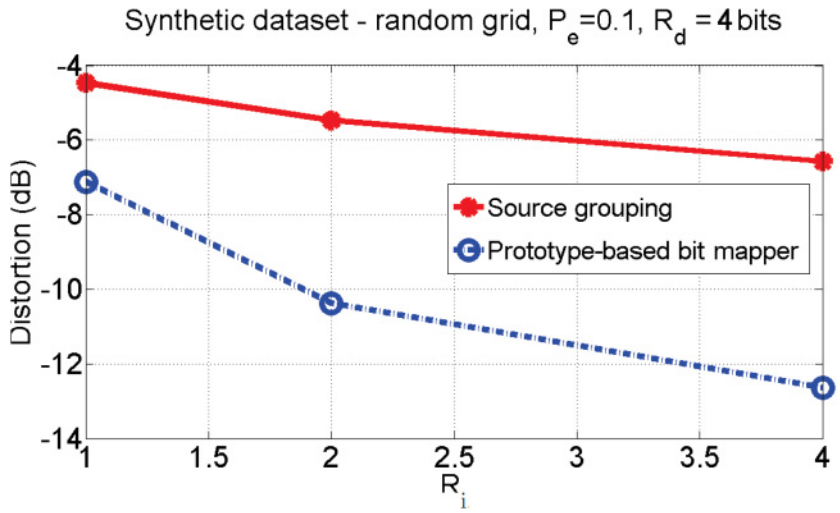

Fig. 15. Performance gains as a function of R_i for the synthetic dataset.

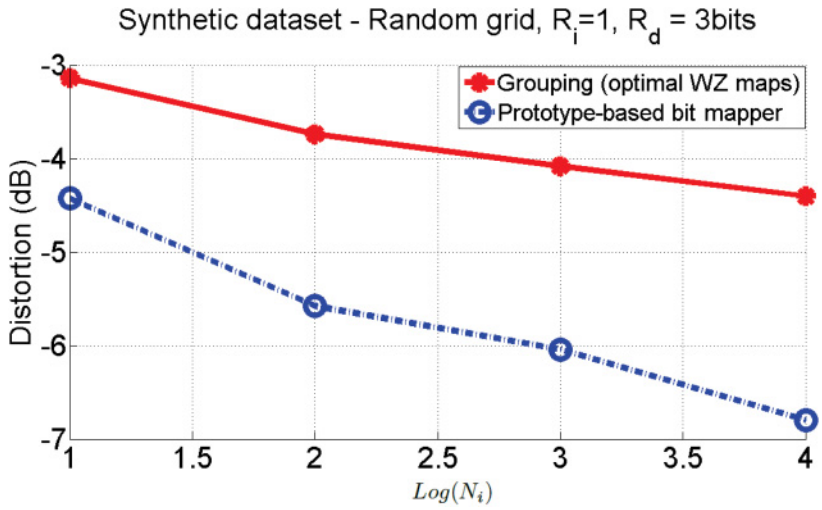

Fig. 16. Performance gains with the number of high-rate quantization levels for the synthetic dataset.

the source grouping approach when R_q is increased from 1 to 4 bits. Such improvements (see also Yasaratna and Yahampath [2009]) demonstrate the crucial role played by WZ maps in exploiting intersource correlations. It is important to note that while achieving these gains, we have not increased the transmit rate or the decoder complexity of the system. These gains are achieved only at the cost of a very minimal increase in the encoder complexity. This shows that the use of WZ maps to exploit intersource correlations is very critical in practical sensor networks.

Note, moreover, that the proposed structure for the decoder provides about 1dB improvement over the source grouping method even when $R_q = 1$, that is, when distributed source encoding is not used. Such an approach could be useful in certain applications (see, e.g., Barros and Tuechler [2006]) due to its lower design complexity. This result shows that, even in the absence of its DSC advantage, the proposed decoder

structure provides significant gains over the source grouping method due to the inherent error-correcting capability.

9. LARGE-SCALE DSC DESIGN FOR ERASURES

It is critical to develop robust distributed source coding techniques for networks with bit/packet erasures, as these are encountered often in low-powered sensor networks. In this section, we address this issue in detail and describe how the proposed technique can be extended to handle erasures. In the erasure setting, it is assumed that a subset of the transmitted bits is 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.

For optimal decoding, the decoder would, in principle, have an independent codebook stored for each possible bit erasure pattern, where an estimate for each source is obtained from the codebook corresponding to the particular subset of bits received. Obviously, the total number of codebooks in this case grows exponentially with the number of sources and transmission rates and is further compounded by the exponential growth in the number of codewords within each codebook. It is easy to show that, if the optimal decoder is implemented by such a naive approach, the total storage at the decoder (the decoder complexity) grows as $\mathcal{O}(N3^{NR})$ if $R_i = R \forall i$. However, we will later show that certain properties of the optimal codebooks, under the MSE distortion measure, enable implementation using reduced complexity that grows as $\mathcal{O}(N2^{NR+1})$, which is still exponential in NR.

In this article, we describe one possible approach to extend the classifier-based decoding paradigm to handle erasures and demonstrate by simulation results that the proposed technique significantly outperforms the source grouping technique in providing robustness against bit erasures. In the proposed approach, the received index tuples are mapped to one of the cloud centers only based on the bits that are received; that is, we assume that erased bits are equally likely to be a 0 or a 1. We describe the technique more formally in Section 9.2. The proposed approach effectively mimics an erasure code at the decoder, which attempts to recover the lost bits using the correlation across the sources. There are several other possible approaches to extend the underlying principles to handle erasures, which will be studied in future work. We also note that, using the same principles, the proposed technique can be easily applied to networks that suffer from a combination of bit errors and erasures. We omit the details here for brevity.

The rest of the section is described as follows. We first begin with the description of the optimal decoder in conjunction with bit erasures and derive the corresponding decoder complexity. We then consider an intuitive heuristic scheme that provides close to optimal performance at half the decoding complexity. We use both these techniques, designed within clusters, as competitors for the proposed approach. We then describe the proposed methodology to extend the classifier-based decoding paradigm to handle erasures.

9.1. Optimal Decoder for Bit Erasures

Recall the description of the conventional DSC system in Section 2. In the erasures setting, the decoded index tuple, denoted by $\hat{I}$, is a subset of the transmitted bits, I, that is, $\hat{I} \in 2^{\mathcal{I}}$, where we employ the notation, $2^{\mathcal{S}}$, to denote the power set (the set of all subsets) of a given set $\mathcal{S}$. The decoder reconstructs each source based on the received index $\hat{\mathcal{I}}$. Formally, the decoder for source i is given by the mapping

$$\mathcal{D}_i : 2^{\mathcal{I}} \rightarrow \hat{X}_i \in \mathcal{R}. \tag{19}$$

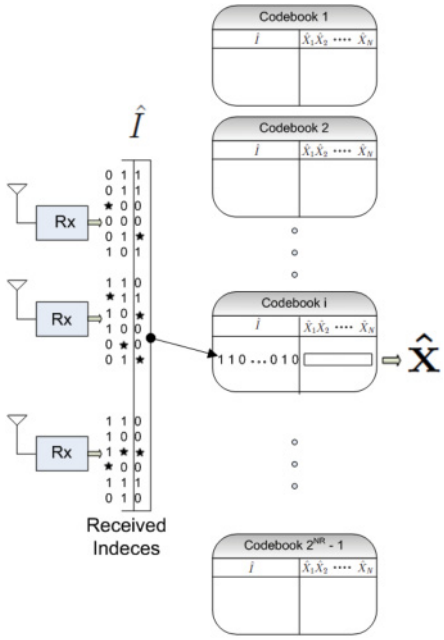

Fig. 17. The optimal decoding mechanism when there are bit erasures. The stars denote bits that are lost. Note that there is a unique codebook for each possible bit erasure pattern.

A straightforward way to implement the decoder is as a simple lookup table that stores a unique reconstruction for each source and for each possible received index tuple, as shown in Figure 17. The total number of possible received index tuples is given by $\sum_{k=1}^{R_r} 2^k$, and hence the number of reconstruction codewords to be stored for optimal decoding is given by $N \sum_{k=1}^{R_r} 2^k$, which grows as $\mathcal{O}(N3^{R_r})$. However, a neat trick allows us to reduce the total storage to $N2^{(R_r+1)}$ when the distortion measure is MSE, as illustrated next.

Recall that when the distortion measure is MSE, the optimal reconstruction for any source given a received index tuple is obtained by conditional expectation:

$$\hat{X}_i = E(X_i|\hat{I}). \tag{20}$$

Let $\mathcal{K}$ denote the bit positions that have been received reliably at the decoder, that is, $I(\mathcal{K}) = \hat{I}$. Then, the previous expression can be rewritten using the property of iterated expectations as follows:

$$\hat{X}_i \;=\; E(X_i|\hat{I}) = \sum_{I \in \mathcal{I}, I(\mathcal{K})=\hat{I}} P(I)E(X_i|I). \tag{21}$$

This allows us to compute the reconstructions on the fly given the subset of the bits that were received, using precomputed reconstructions $E(X_i|I) \forall I \in \mathcal{I}$ and the probabilities $P(I)$. This simplification reduces the total storage at the decoder for optimal decoding to $N2^{(R_r+1)}$, albeit with the overhead of an exponential number of computations to be performed if very few bits are received.

As a first competitor to the proposed approach, we consider a source-grouping-based scheme where optimal distributed source codes are designed within each group. The decoder complexity for this approach grows as $\mathcal{O}(N2^{(R_d+1)})$, where R_d denotes the

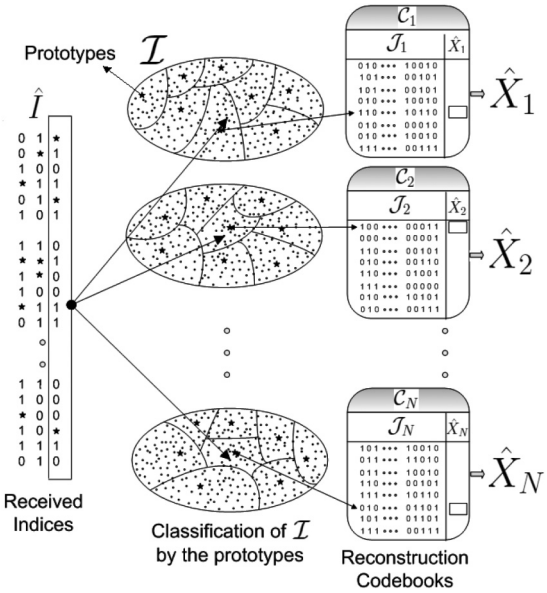

Fig. 18. The proposed decoding mechanism when there are bit erasures. Stars in the receiver indices denote bits that are lost. Now, the received index tuple is mapped to the nearest prototype based only on the bits that are received.

maximum number of bits used for decoding within a cluster. As another competitor to the proposed approach, we consider a heuristic scheme where the sources are grouped and the reconstructions are estimated using Equation (21), assuming that all the transmit indices are equally probable, that is, $P(I) = \frac{1}{2^{R_r}}$. A priori, such an approach compromises some of the performance in distortion, but it entails a decoder complexity that grows as $\mathcal{O}(N2^{R_d})$, which eventually yields an improved complexity–distortion tradeoff curve.

9.2. Proposed Approach to Handle Bit Erasures

Recall that, to build error resilience, the decoder mapped the received index tuple to one of the cloud centers based on a minimum distance criterion 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 is 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, as shown in Figure 18. 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 may equivalently assume the corresponding missing values 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. Clearly, the received index tuples are effectively mapped to one of the cloud centers only based on the bits that were actually received.

Formally, if $\mathcal{K}$ denotes the subset of bit positions that are received at the decoder reliably, the bit mapper for decoding source i is given by the following mapping:

$$\mathcal{B}_i(I, \mathcal{K}) : \arg\min_{S \in \mathcal{J}_i} d_i(I, S, \mathcal{K}), \tag{22}$$

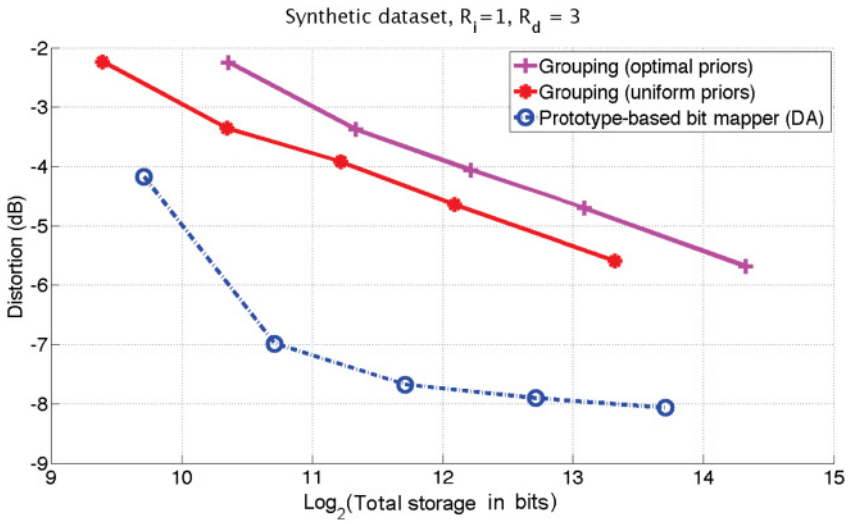

Fig. 19. Total storage versus distortion for the synthetic dataset, $R_i = 1 \forall i$ and $P_e = 0.1$.

where $d_i(I, S, \mathcal{K})$ denotes the Hamming distance between vectors I and S at positions $\mathcal{K}$.

Observe that the method naturally provides better robustness to channel erasures as it uses all the available *received* bits to correct erasures, unlike the source grouping method, which estimates the sources only using the received bits within relevant subsets. The cloud centers and the reconstruction codebooks can be designed using an approach similar to that described in Section 4 given a training sequence of source samples and channel erasure patterns to minimize the expected reconstruction distortion. The design and the operational complexities remain the same. We omit the details here for brevity.

9.3. Results

We again consider all three datasets mentioned in Section 8 for evaluating the proposed approach to handle bit erasures. Figures 19, 20, and 21 show the performance gains of the proposed approach for the three datasets, respectively. Here, P_e denotes the probability of a bit erasures. We assume $P_e = 0.1$ for the synthetic and the temperature sensor datasets and $P_e = 0.2$ for the rainfall dataset. As discussed earlier, we compare the proposed technique with two approaches, both based on source grouping. The first curve performs optimal decoding within each cluster based on Equation (21) using optimal priors $P(I)$, estimated using the training set. The second curve corresponds to a heuristic approach where suboptimal uniform priors are used for all $P(I)$. We observe that despite some expected loss due to approximating the prior to be uniform, which is largely negligible, the heuristic approach achieves a significantly better complexity–distortion tradeoff curve, as it requires half the decoder storage required by the "optimal priors" scheme. It is evident from the results that the proposed approach gains significantly over both the techniques based on source grouping. For the synthetic dataset, the proposed technique outperforms the source-grouping-based methods by over 3dB in distortion at fixed decoder complexities. Similarly, for the temperature and rainfall datasets, the gains are over 3dB and 1.5dB, respectively, in distortion at fixed complexities. As with the case of channel errors, these results demonstrate that the proposed technique provides better erasure resilience and recovers the lost bits more efficiently than the source grouping method, thereby providing improved end-to-end complexity–distortion tradeoff.

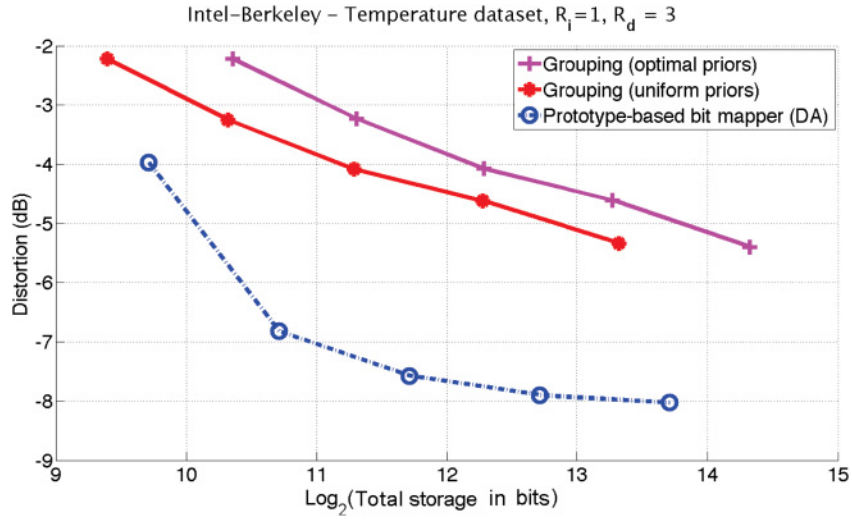

Fig. 20. Total storage versus distortion for the temperature sensor dataset, $R_i = 1$, $P_e = 0.1$.

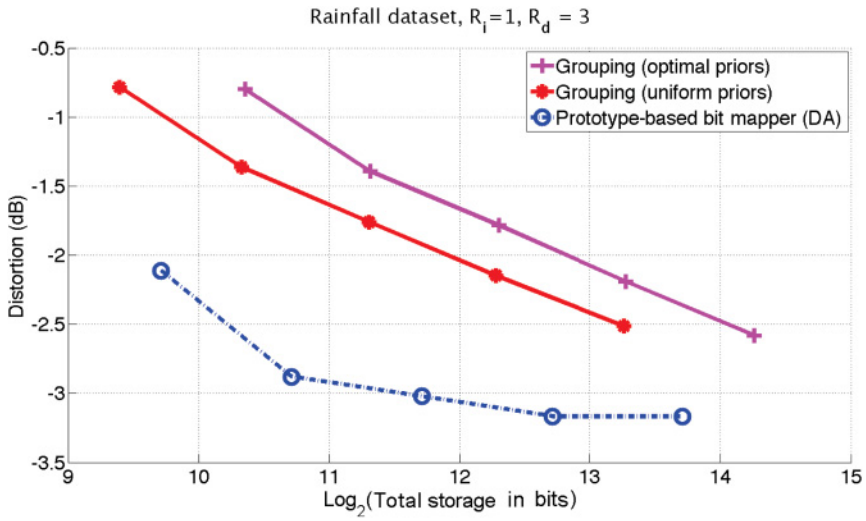

Fig. 21. Total storage versus distortion for the rainfall dataset, $R_i = 1$, $P_e = 0.2$.

10. ROBUSTNESS TO MISMATCH IN SOURCE AND CHANNEL 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 known and are stationary in time. However, in practice, the source and channel distribution often vary significantly with time, leading to a mismatch in statistics between training and operation. In this section, we study the robustness of the proposed approach to mismatch in source and channel statistics and show that the method offers a gradual degradation in performance as the statistics diverge.

We begin with mismatch in channel statistics for the same source distribution. We consider the 10-sensor Gaussian synthetic dataset described in Section 8 for the

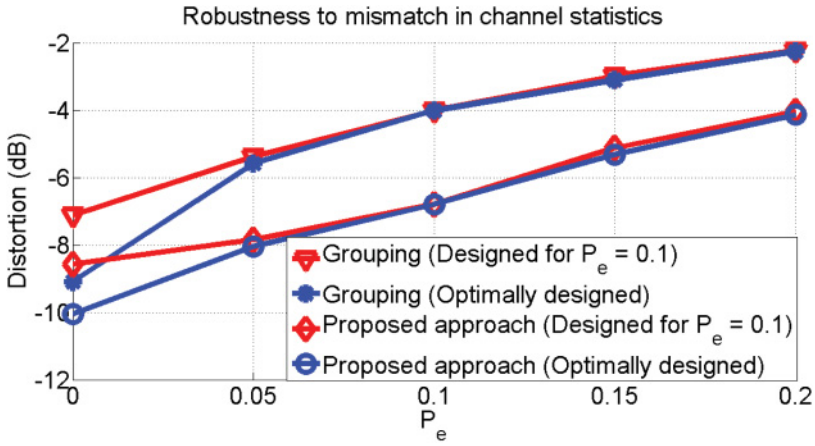

Fig. 22. Channel statistics mismatch.

analysis. For both the source grouping and the proposed approaches, we design all the system parameters assuming $\rho_0 = 0.9$ and $P_e = 0.1$. We then test the system by varying P_e from 0 to 0.2, without adjusting the system parameters to match the actual P_e. Figure 22 shows the performance curves obtained for the two approaches. For comparison, we also plot the performance obtained when the system was designed for the true channel statistics. We observe that the performance of both the source grouping and the proposed approaches deteriorates by a negligible amount as the channel statistics vary. Noticeable degradation in performance is seen only at $P_e = 0$. The degradation in performance is relatively similar for both techniques, and the proposed approach continues to gain significantly over the source grouping method, even with large deviations in channel statistics. This shows that the proposed approach, although designed optimally for the given channel statistics, continues to offer robustness against errors in channel estimation.

We next consider mismatch in source statistics, assuming that the channel statistics remain the same during operation. We again consider the 10-sensor Gaussian synthetic dataset described in Section 8. We design the system parameters assuming $\rho_0 = 0.9$ and $P_e = 0.1$ and then test the system by varying ρ_0 from $0.7-0.95$. The performance curves are shown in Figure 23. We observe that small deviations in statistics do not affect the performance of either technique. For large deviations in source statistics, the loss in performance of the proposed approach is marginally higher than that for the source grouping technique. However, the proposed approach offers a graceful degradation in performance as the source statistics deviate from the training set. Nevertheless, the gains over the source grouping technique continue to be significant for all correlation values.

We next briefly outline some options for adapting the proposed approach to significant variations in source statistics so as to reap its benefits in such highly nonstationary applications. One possible approach to handle such variations 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 nonstationary. An alternate approach is to store multiple sets of system parameters, designed for representative statistics, and to select a particular set of parameters, at a given time, by estimating the current statistics at the sink. The possible implications of

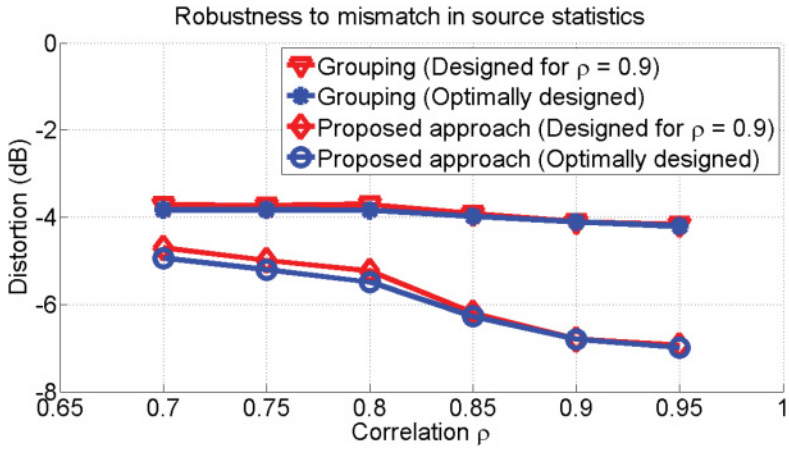

Fig. 23. Source statistics mismatch.

these directions on practical deployment of sensor networks will be evaluated as part of our future work.

11. CONCLUSIONS

In this article, we proposed a new coding approach to achieve large-scale distributed compression that is robust to channel errors/erasures. In the proposed approach, the set of possible received index tuples is first partitioned into subsets, each of which is assigned a unique codeword. The approach enables low-complexity, practically realizable decoders that are scalable to large networks. The index space partition is achieved using a nearest prototype classifier structure, which effectively provides good resilience to channel errors and erasures. We first proposed a design strategy based on greedy iterative descent, which itself provides significant gains over conventional techniques. We also presented a deterministic annealing-based optimization algorithm for the design, which provides further gains in performance by avoiding poor local minima that riddle the cost surface. Simulation results show that the proposed scheme achieves significant gains over state-of-the-art techniques.

REFERENCES

J. Bajcsy and P. Mitran. 2001. Coding for the Slepian-Wolf problem with turbo codes. In *IEEE GLOBECOM*. Vol. 2. 1400–1404.

J. Barros and M. Tuechler. 2006. Scalable decoding on factor graphs—a practical solution for sensor networks. *IEEE Trans. Commun.* 54, 2, 284–294.

J. Cardinal and G. Assche. 2002. A generalized VQ method for combined compression and estimation. In *IEEE International Symp. Information Theory*. 63.

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

M. Fleming, Q. Zhao, and M. Effros. 2004. Network vector quantization. *IEEE Trans. Inf. Theory* 50, 1584–1604.

T. J. Flynn and R. M. Gray. 1987. Encoding of correlated observations. *IEEE Trans. Inf. Theory* 33, 6, 773–787.

A. Gersho and R. Gray. 1991. *Vector Quantization and Signal Compression*. Springer.

G. Maierbacher and J. Barros. 2009. Low-complexity coding and source-optimized clustering for large-scale sensor networks. *ACM Trans. Sensor Networks* 5, 3.

D. Miller, A. Rao, K. Rose, and A. Gersho. 1996. A global optimization technique for statistical classifier design. *IEEE Trans. Signal Process.* 44, 12, 3108–3122.

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

S. Pradhan and K. Ramchandran. 2003. Distributed source coding using syndromes (DISCUS): Design and construction. *IEEE Trans. Inf. Theory* 49, 3, 626–643.

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

A. Rao, D. Miller, K. Rose, and A. Gersho. 1999. A deterministic annealing approach for parsimonious design of piecewise regression models. *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)* 21, 159–173.

D. Rebollo-Monedero, R. Zhang, and B. Girod. 2003. Design of optimal quantizers for distributed source coding. In *IEEE Data Compression Conference*. 13–22.

K. Rose. 1998. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. *Proceedings of IEEE* 86, 11, 2210–2239.

K. Rose, E. Gurewitz, and G. Fox. 1992. Vector quantization by deterministic annealing. *IEEE Trans. Inf. Theory* 38, 4, 1249–1257.

A. Saxena, J. Nayak, and K. Rose. 2010. Robust distributed source coder design by deterministic annealing. *IEEE Trans. Signal Process.* 58, 2, 859–868.

A. Saxena and K. Rose. 2009. Distributed predictive coding for spatio-temporally correlated sources. *IEEE Trans. Signal Process.* 57, 4066–4075.

D. Slepian and J. K. Wolf. 1973. Noiseless coding of correlated information sources. *IEEE Trans. Inf. Theory* 19, 471–480.

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

K. Viswanatha, S. Ramaswamy, A. Saxena, and K. Rose. 2012. Error-resilient and complexity constrained distributed coding for large scale sensor networks. In *ACM/IEEE Conference on Information Processing in Sensor Networks*.

A. D. Wyner and J. Ziv. 1976. The rate-distortion function for source coding with side information at the decoder. *IEEE Trans. Inf, Theory* 22, 1–10.

Z. Xiong, A. Liveris, and S. Cheng. 2004. Distributed source coding for sensor networks. *IEEE Signal Process. Mag.* 21, 5, 80–94.

P. Yahampath. 2009. Joint source decoding in large scale sensor networks using Markov random field models. In *IEEE ICASSP*. 2769–2772.

R. Yasaratna and P. Yahampath. 2009. Design of scalable decoders for sensor networks via Bayesian network learning. *IEEE Trans. Comm.* 2868–2871.

Received December 2013; revised July 2014; accepted August 2014