Dense Micro sensor Network with distributed compression Techniques

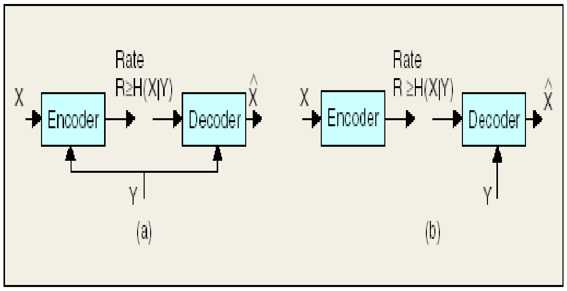
#G.Naveen kumar[1] E mail:gogikar.556@gmail.com
#Vanam. sravan kumar[2] sravan512@gmail.com
#Dr. Jayadev Gyani[3]**, HOD**, Dept of computer science and Engineering
E mail**:** jayadevgyani@yahoo.com
# **Jayamukhi Institute of Technological Sciences, Warangal, INDIA**

Abstract

The sensor networks are fundamentally constrained by the difficulty and energy expense of delivering information from sensors to sink. Currently if sensor system designers want to compress acquired data they must either develop application specific compression algorithms or off the shelf algorithms not designed for resource –constrained sensor nodes. The distributed nature of the sensor network architecture introduces unique challenges and opportunities for collaborative networked signal processing techniques that can potentially lead to significant performance gains. Many evolving low-power sensor network scenarios need to have high spatial density to enable reliable operation in the face of component node failures as well as to facilitate high spatial localization of events of interest. This induces a high level of network data redundancy, where spatially proximal sensor readings are highly correlated. In this article, we propose a new way of removing this redundancy in a completely distributed manner, i.e., without the sensors needing to talk to one another. Our constructive framework for this problem is dubbed DISCUS (distributed source coding using syndromes) and is inspired by fundamental concepts from information theory. In this article, we review the main ideas, provide illustrations, and give the intuition behind the theory that enables this framework.

1. Introduction

We are currently in the midst of a "distributed" revolution, where distributed ways of communicating, processing, sensing, and computing are dislodging more traditional centralized architectures.



The trend is to go away from a centralized, super-reliable, single-node platform to a dense and distributed multitude of cheap, lightweight, and potentially individually unreliable components that, as a group, are capable of far more complex tasks and inferences than any individual super-node. A classical example of this is in distributed sensing, where it is desirable to have high sensor density for reliability, accuracy, and cheaper employment. Advances in device technology, networking, and information processing have allowed the emergence of wireless sensor network technology: highly reliable, modular, ubiquitous devices that can form a network. In the paradigm investigated by Smart Dust, hundreds or thousands of sensor nodes of cubic-millimeter dimension are scattered about an environment of interest. Each node has the capability to sense elements of the environment, make computations, and communicate with other nodes or a centralized observer. The major constraint to individual node performance is energy, which is consumed primarily by sensing and communications operations [2].

The need for a spatially dense sensor network is driven by two requirements: i) reliable decision-making in the face of unreliable individual components and ii) superior spatial localization of transient events of interest. This can lead to considerable system redundancy, however, in the "ambient" mode. The need to strip this redundancy is underlined by a couple of additional factors. First, there is typically only a single radio channel available to the sensor nodes for communication, making efficient bandwidth utilization critical. Second, in a multihop network, the benefits of data compression are magnified as energy savings are incurred at each transmission and reception along the route. Motivated by this, our article addresses an important component of the communication fabric underlying sensor networks: namely, an efficient framework for minimizing the amount of inter node communication while preserving the resolution of the data gathered. The goal is to compress sensor data from individual nodes while requiring minimal (or no) inter sensor communication. One way of removing this spatial redundancy is through joint processing based on an elaborate inter sensor information exchange. However, the communication protocol associated with this exchange can itself be expensive.

This raises the interesting question about the tradeoff that minimizes the system energy. More specifically, what is the loss in overall compression efficiency should there be no inter sensor communication? If the joint distribution quantifying the sensor correlation structure is known, the surprising answer is that there is theoretically no loss in performance under certain conditions. The caveat, however, is that this is only in theory, as it is based on asymptotic and random coding arguments from information theory (under the name of the Slepian-Wolf coding theorem [3], [4] and its extensions).In this article, we are not interested in asymptotic bounds, but rather in the formulation of a constructive, systematic framework that can approach the bounds promised by information theory. Indeed, our work is motivated by the following quote from a key article in the 50th year Commemorative Special Issue of the *IEEE Transactions on Information Theory* [5] which laments that "despite the existence of potential applications, the conceptual importance of (Slepian-Wolf) distributed source coding has not been mirrored in practical data compression." We accordingly describe a constructive algorithmic framework that involves an interesting interplay of signal processing (source coding), communications (coding theory), and estimation theory. In the interests of clarity and to provide tutorial value, we will deliberately aim to keep the treatment simple and intuitive, rather than detailed and rigorous, referring the reader to appropriate references.

In addition to the application of distributed sensor networks, other potential applications of the material described here include stereo and multi camera vision systems, compression of hyper spectral imagery, distributed database systems, surveillance systems, and simulcast of digital and analog television [6]. Furthermore, there are some very interesting dualities and links between the distributed compression problem addressed in this article and other multi-user problems, including broadcast, multicast, inter symbol interference cancellation, and information-hiding/watermarking, that make the methods described in this article highly relevant tools for the toolkit needed to tackle those problems as well.

2. Distributed Compression

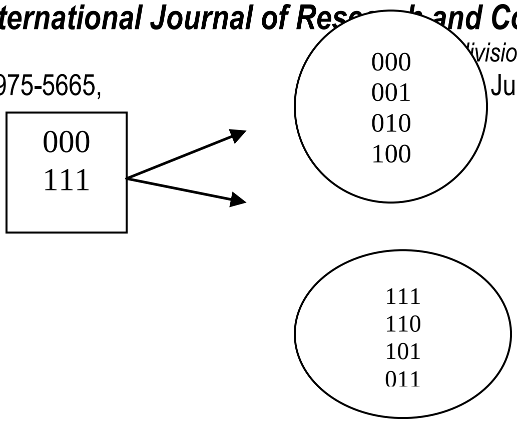
Let us consider the problem of compressing an information source in the presence of side information present only at the decoder in the form of another correlated source. The goal is for the decoder to reconstruct the original source using this side information as well as the

bitstream sent by the encoder. For clarity, we first consider discrete sources.

2.1 Example of Binary Sources

Let us consider the following riddle to get in sight into this problem. Suppose *X* and *Y* are equiprobable 3-bit binary words correlated in the following sense: the Hamming distance between *X* and *Y* is no more than one. If *Y* is available to both the encoder and the decoder, clearly it is wasteful to describe *X* using 3 bits, as there are only 2 bits of uncertainty between *X* and *Y*(the modulo-two binary sum of *X* and *Y*: {000,001,010,100}, which can be indexed and sent). Now what if *Y* were revealed *only* to the decoder but not the encoder: could *X* still be described using only 2 bits of information? A moment's thought reveals that the answer is indeed yes. The solution consists in realizing that since the decoder knows *Y*, it is wasteful for *X* to spend any bits in differentiating between {*X* =□000and *X* =□111}, since the Hamming distance between these two words is three, whereas *Y* is known to be within Hamming distance 1 of

X. Thus, if the decoder knows that either *X* =□000 or *X* =□111, it can resolve this uncertainty by checking which of them is closer in Hamming distance to *Y* and declaring that as the value of *X* . Note that the set {000 111,} is a 3-bit repetition code with a Hamming-distance of 3. Likewise, in addition to the set {000 111,}, the following three sets for *X*: {100 011,}, {010 101,}, and {001 110, }are composed of pairs of words whose Hamming distance is three. Further, these four sets cover the complete space of all possible binary 3-tuples that *X* can assume Thus we send the index of the coset containing *X* , thus requiring 2bits. This is illustrated in the figure to the right Recall that a channel code is specified by its 3-tuple (, ,)*n k d* , where *n* is code length, *k* is the message length and *d* is the minimum distance of the code. In the above example, we considered the cosets of the linear (, ,)3 1 3 repetition code. In channel coding jargon, these cosets are associated with a unique syndrome of the code. The syndrome **s** associated with a linear channel code is defined as **s Hx**, where **H** is the parity-check matrix of the code, and **x** is any valid codeword. The syndrome corresponding to all valid codewords is the zero-vector, since by defi-nition all valid codewords are in the null-space of **H**. A nonzero syndrome vector signals symptoms of an erroneous reception (hence the term syndrome



Example of binary source with when X=000 Or X=111 it belongs to same cosset the Corresponding outstanding sets are disjoint

3. Discrete Sources

Consider first the problem where X and Y are correlated discrete-alphabet independent identically distributed (i.i.d.) sources, and we have to compress X losslessly, with Y being known at the decoder but not at the encoder. To elaborate, if Y were known at both ends (see Fig. 1(a)), then the problem of compressing X is well understood: one can compress X at the theoretical rate [3] of its conditional entropy (conditional entropy, $H\ X\ Y(\ |\ )$ is a measure of probabilistic uncertainty in X given Y) given Y, $H\ X\ Y(\ |\ )$. But what if Y were known only at the decoder for X and not at the encoder (see Fig. 1(b)) The surprising answer is that one can still compress X using only $H\ X\ Y(\ |\ )$bits, the same as the case where the encoder does know Y. That is, by knowing just $p\ X\ Y(\ ,\ )$, the joint distribution of X and Y, without explicitly knowing Y, the encoder of X can perform as well as an encoder which explicitly knows Y (n theory, only $H,\ X$ $Y(\ |\ )$ needs to be known at the encoder, not even $p\ X\ Y(\ ,\ )$). This is known as the Slepian-Wolf coding theorem [4]. The Slepian-Wolf theorem has been extended to the lossy encoding of continuous-valued sources by Wyner and Ziv [7]-[9], who showed that a similar result holds in the case where X and Y are correlated i.i.d. Gaussian random variables. If the decoder knows Y, then whether or not the en coder knows Y, the rate-distortion performance for coding X is identical. (The only caveat is that Y has to be known losslessly at the decoder.) As in the lossless case, the result is asymptotic and non constructive. Although this is a source coding problem, in this work we propose a framework resting heavily on channel coding principles. Let us consider the case of binary sources as considered in "Example of Binary Sources," where we give an example to illustrate this connection to channel coding. The key concept to note here is that we partition the space of all outcomes of the source X into sets (called cosets) such that the minimum distance between any two code vectors in any coset is "large" enough. The encoder saves rate by sending only the index of the coset containing the outcome. The decoder recovers the outcome of X by searching through the coset whose index is received. The search is for that code vector which is "closest" (in the right metric) to the outcome of Y. This concept can be generalized to encoding of more general discrete sources as well as continuous alphabet sources as considered next.

3.1 General Scalar Sources

Here we remove the constraint that X, Y belong to a binary or even discrete alphabet and consider the continuous-valued case (defined on the real line R). In this article we consider a simple correlation structure between the source and the side information to illustrate the key con-cepts. The approach presented here can be extended to capture more elaborate correlation structures. We consider the specific case (there has been some work on more general correlation structures with a source coding perspective such as in [10]) where the correlation between X and Y is captured as follows: Y is a noisy version of *X*: i.e., $Y= X+ N$□□ , where *N* is also continuous valued (defined on the real line R), i.i.d., and independent of *X*. As before, the setup is that the decoder alone has access to the *Y* process, and the task is to optimally compress the *X* process. We will consider without loss of generality (WLOG) the case where *X* and *N* are zero-mean Gaussian random variables with known variances: our approach can be generalized to arbitrary distributions for *X* and *N*. The goal is to form the best approximation, *X^* , to *X* given an encoding bit budget of *R* bits per sample. We consider reconstruction with a fidelity criterion as given below. Let ρ(.) be a function ρ :R R X R□→R^{+}□ . We want to minimize *E[*ρ(*X* ,*X^*)]where *E*(.)is the expectation operator. This problem can also be posed as minimizing the rate of transmission *R* such that the reconstruction fidelity is less than a target distortion *D*. This involves an intricate interplay of source coding, channel coding, and estimation theory. An example dealing with scalar quantizers is given later on. Let us analyze the components of the problem, one by one .

3.2 Source Coding

Due to the finite rate constraint on the information transmitted, the source *X* has to be quantized. For a target reconstruction fidelity, a source code has to be designed, which involves the following:

- ***Partition of the source space:*** the scalar input source space is partitioned into $2R_s$disjoint regions, where $R_s$is de fined as the source rate in bits/sample.

- ***Codebook:*** Each region in the above partition is associated with a representation codeword, where the set of representation codewords comprises the source codebook.

The source is quantized to one of the source codewords, and the index of the quantized codeword is made available to the decoder errorlessly. This involves a transmission rate of $R_s$bits/sample. The representation codeword to which *X* is quantized is referred to as an active source codeword. The active source codeword is denoted by *U*. The decoding further involves a component which deals with the estimation of the source based on both the quantized source and the correlated side information *Y*.

Estimation

The decoder gets the best estimate of *X* (minimizing the fidelity criterion) conditioned on the outcome of the side information and the source space region containing *X*. The source rate $R_s$is chosen such that the final estimation error is within the target fidelity criterion.

4. Channel Coding

By exploiting the correlation between *X* and *Y*, we make the decoder recover (within a tolerably small probability of error) the index of the active source codeword with a lower rate of transmission than R_s. The active source codeword *U*, characterizing the quantized representation, is correlated to *X* and in turn correlated to the side information *Y*. (See Fig. 2.) This induces a fictitious channel *P Y U*(|) between *U* and *Y*. The input of the channel is observed by the encoder, and the output is observed by the decoder. We propose to build a “channel code” for this channel on the space of *U*. Let$2^{R}c$ denote the number of codewords in the de- signed channel code [11] where $R_c$is defined as channel rate (not to be confused with actual channels used for the transmission of information). Suppose, for a given realization, the active source codeword belongs to this channel code and this is known at the decoder, then we do not need to send any information to the receiver, as it can recover the intended codeword index by observing *Y* (by de coding *Y* in the channel code). Since any codeword in the source codebook can be an outcome of the quantization with a finite probability, we partition the space of source codebook into cosets of the designed channel code.

The encoder computes the index of the coset of the channel code containing the active source codeword. This index is transmitted errorlessly with a rate of transmission of R_s−□$R_c$bits/sample to the decoder. The decoder recovers the active source codeword in the given coset by finding a codeword which is closest (in some metric) to the observed side information. This approach involves occasional decoding error, where the side information is decoded to a wrong representation codeword which is not the active source codeword. The probability of decoding error can be made arbitrarily small by designing a channel code with a large minimum distance. The design [12] involves the following:

- Source quantization and estimation for the desired distortion performance.
- The representation codebook to maximize the correlation between *U* and *Y*.
- The channel code (and each of its cosets) to have a large achievable rate $R_c$with minimum probability of decoding error, on the space of the source codebook. The source codebook is partitioned into the cosets of this channel code.
- Efficient rule for decoding side information in a given coset of the channel code.The encoder and decoder are schematically shown in Fig. 2.

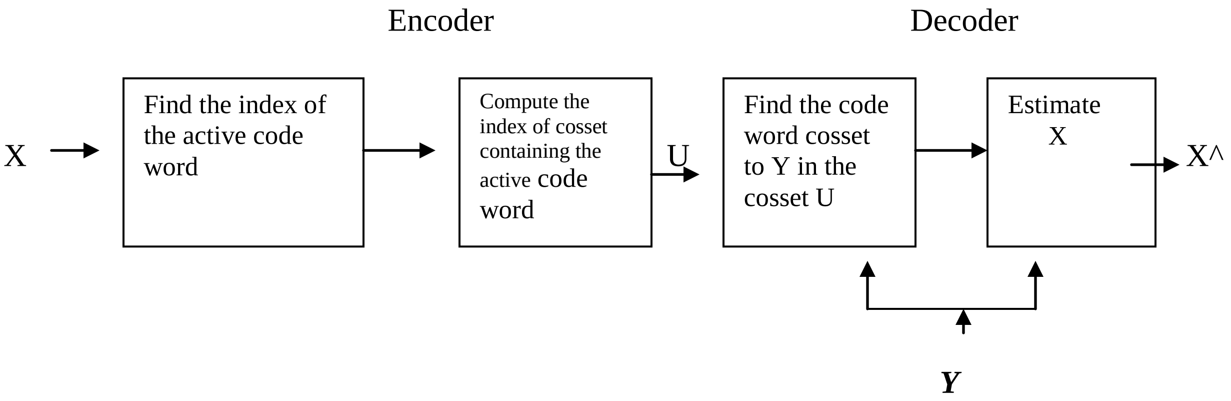

Encoder
Decoder
X
Find the index of the active code word
Compute the index of cosset containing the active code word
U
Find the code word cosset to Y in the cosset U
Estimate X
X^
Y

Fig 2 Efficient rule of decoding example

4.1 Scalar Partitioning Example

Consider first a simple fixed-length (length-*V*) scalar quantizer [13] designed for the probability density function of *X*. Let *V* Δ 8 for ease of discussion. Let Δ { , , ,$_0$1^{K}r_V−1$^{\}}$be the set of reconstruction levels as shown in the Fig. 3. Note that Δ □partitions the real line into *V* intervals each associated with one of the reconstruction levels. Thus the source codebookS Δ and R_s Δ 3 bits/sample. If we use this quantizer to encode *X*, we need to pay the price of 3 bits/sample. We would like to expend less rate (say 1 bit/sample) by exploiting the correlation between the source X and the side information Y while still using the same quantizer. One way to do this is the following. We partition the set Δ □into M V( )≤ cosets. For illustration, let M Δ 2. We group r r r_0, , ,and$_2$4 $r_6$into one coset. Similarly r r r_1, , , and$_3$5 $r_7$are grouped into another coset. The channel code C Δ { , , , }r r r $r_0$2 and R_c Δ 2 bits/ sample and the rate of transmission is 1 bit/sample. In this illustration we have taken the representation codeword $r_i$to be the centroid of the disjoint region Δ $_i$. The encoding can be described as follows:

- Find the codeword from the set which is closest (in terms of minimizing the desired distortion measure) to the source sample X. Call this the active codeword.0 1 K M −1} of the coset of Cin S
- Send the index U Δ { , , , containing the active codeword.

The decoder deciphers the active codeword by finding the codeword which is closest in some metric to Y in the coset whose index is sent by the encoder. After finding the codeword (say r_k), the decoder estimates X using all the available information. We wish to minimize the expected value of the distortion ρ(,X X^), where X^ is the estimate of X. As discussed before, there is always a finite probability of decoding failure. The probability of decoding failure (see Fig. 3) can be made sufficiently small with more efficient coset constructions. Thus for this case, the source codebook and the channel codebook are both memoryless. For a given rate of transmission R bits/source sample, we choose a scalar quantizer with 2 R_s levels and partition it into 2Rcosets each containing 2 R_c codewords.

4.2 Trellis Partition

The previous section is an example of an uncoded system based on scalar quantizers. We now describe a more sophisticated coded system based still on scalar quantization but now having a trellis-coded system having memory for the coset construction. We emphasize that we still use fixed-length scalar quantizers for{ }X_i $_i$$n$

but the cosets are built on the space Δ□n . Consider the space Δ □n, and let V Δ 8. In this space there are totally 2 3 n distinct sequences. The task is to partition this sequence space into cosets in such a way that the minimum distance between any two sequences in a coset is made as large as possible, while maintaining symmetry among the cosets.We consider a trellis-based partitioning based on convolutional codes and set-partitioning rules as in Ungerboeck's trellis-coded-modulation (TCM) [14].

Note that this is not to be confused with the concept of trellis-coded-quantization (TCQ) in source coding.We consider a trellis code where a bit stream with R_c bits/unit time is used to partition 2 R_c□1 codevectors (for the case R_s Δ □R_c□1) taking values in R. The set Δ □is partitioned into four subsets (for the sake of clarity) as before.We use Ungerboeck's four-state trellis with the above set partitioning rules. The trellis on this set is shown in Fig.

4(a) (which we call the principal trellis). Let Q:A3→□ Δ be the one-to-one mapping from 3-tuple binary data onto Δ according to the following rules: Q( ) Δ □ Δ □r Δ where Δ A^3
is the binary representation of Δ.Using this, we can partition the space ∇^ninto2^ncosets, each containing 22 n sequences. Let **H(t)** be the parity check matrix polynomial of the convolutional code used in the structure.

Let□□be any sequence in Δ □*n* , thus **Q** $-1_{(\,)}$ Δ
Δ ^{3}*n* . Let**S Q** $-1_{(}$Δ$_{)}$.Thus the function**H(t)S(t)** maps any Δ □belonging to Δ ^{n}into A^{n}. We are computing the syndrome of the given codevector Δ: this is precisely what the encoder needs to send to the decoder.

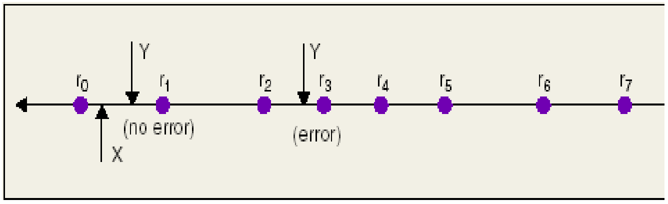


Fig.3 Example of code vector

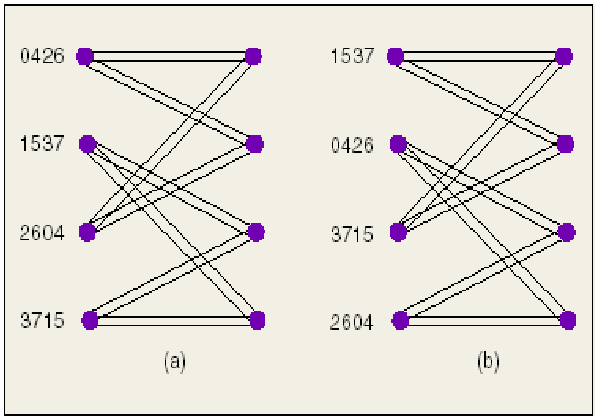
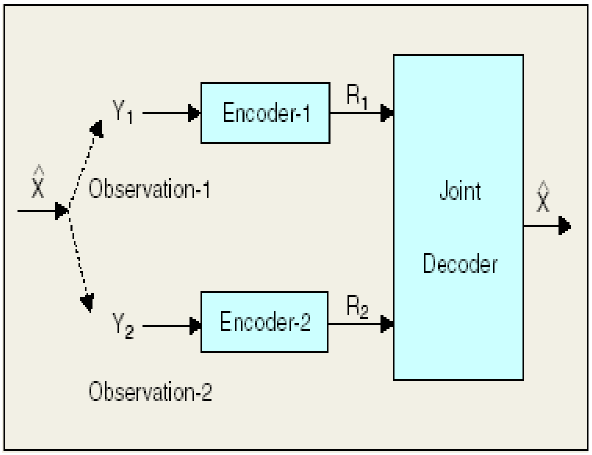


Fig4. structure of code vectors

4.3 Decoder Structure

The decoder has access to the process *Y* in addition to the syndrome sequence sent by the encoder. In the present example, it receives *n* bits of syndrome and *n* samples of the process *Y*. Once the decoder gets the syndrome sequence, it recognizes the coset (containing 2 2 *n* sequences) containing the active codeword sequence. We need a computationally efficient algorithm for searching through this list. The search is for that codeword sequence which is closest to

the sequence



(,*y* $y_1$2 ,...,)$y_n$in terms of the given distortion measure. If the syndrome were the all-zero sequence, then we can use the Viterbi algorithm for this search in the principal coset. Here we need to modify the Viterbi decoding algorithm which is suitable for any syndrome sequence. Consider the *k*th stage of the four-state Ungerboeck trellis [14] as shown in Fig. 4(a). This is the trellis for the coset with an all-zero syndrome (referred to as the principal coset). Here each edge connecting one of the four nodes at the(*k*−) *k*th stage has a label associated with it. At each of the four nodes at the (*k*−) path (which is the distance between partially received sequences) is maintained. At the *k*th stage, for each node, we need to compute the metrics of all the paths leading to that node and choose that path with the least metric. If the *k*th bit of the syndrome sequence is one rather than zero, we need to modify the labels on each edge at the *k*th stage. As discussed earlier, for the convolutional code under consideration, the sequence **Q 0 0 s(t) [** | |]*T*is one of the codeword sequences in the coset whose syndrome is **s(t)**. Thus at the *k*th stage of decoding, if the *k*th bit of **s(t)** is one rather than zero, we need to shift from the principal coset to the complementary coset (there are only two trellises in the given example; see Fig. 4). This can be done at every stage in a computationally elegant way [12].

Preliminary results [12] validate the power of the DISCUS framework. A typical instance of our simulation results involves distributed coding of correlated i.i.d. Gaussian sources that are noisy versions of each other with correlation signal-to-noise ratio (quantifying the ratio of the strength of the signal to the strength of the correlation noise in dB) in the range of 12 to 20 dB. For this instance, using very simple scalar quantization and trellis codes as coset channel codes, the DISCUS approach attains performance gains of 7 to 15 dB in the signal reconstruction fidelity over the theoretical performance bounds of coding systems (promised by Shannon [3], involving infinite-complexity coding systems) that ignore the correlation at the decoder. At the same time, our results indicate that we are within about 3-4 dB of the theoretical performance attainable if there were perfect communication between the sources. This gap can be lowered with more sophisticated source and channel codes than the simple methods used in our preliminary work [12] and are part of ongoing work. This shows the untapped potential of these concepts for significant gains in removing network data redundancy. Accurate statistical sensor models will be needed to extend the results from the Gaussian models used in the preliminary studies and are part of ongoing work. Note that in the system the probability of occurrence of the elements in a given coset are not the same. To capture this lack of uniformity we propose an approach based on periodization of the probability density function of the source *X*. This is illustrated in “Periodization of Probabil ity Density Function.” These systems give good gains on scalar sources when compared with the case when the side information is ignored while encoding.

Symmetric Encoding of Correlated Sources

So far, we have studied the asymmetric version of DISCUS where one of the sources sends partial information while the other sends full information (present in the form of side-information). In practice, it may be desirable to have flexibility in the transmission rates and generalize DISCUS to the case of symmetric encoding,
where all sensors send only partial information to the decoder. One solution to this is to do time-division-multiplexing [17], [18] between the sensors so that at any time, one of the sensors will be acting as a primary source. This requires synchronism between the sensors and the encoders need to switch between these operating modes, which can be cumbersome and unnecessary. Fortuitously, the asymmetric DISCUS framework can be extended to the symmetric case, which can be shown to incur no performance loss with respect to the asymmetric version, and this can be done at the same computational complexity. We will not detail this here and instead refer the reader to [19]. In addition to the conventional symmetric distributed compression problem, a problem of interest for sensor networks involves optimal sensor fusion under bandwidth constraints, which we now consider. Consider the sensor communication [20] system shown in Fig. 5. Here, a number of sensors observe an event, characterized by the signal *X*. The sensors observe independent noisy versions of this event (we restrict ourselves to this setup, though more complex models can also be treated), represented by the signal set{ }$Y_i$for *i* sensors. The individual sensors have rate constraints{ }$R_i$to a central decoding unit, which desires to optimally fuse this information to form an optimal estimate of *X*. It has been shown in theory

[21] that the optimal multisensor fusion problem under rate constraints exactly involves the DISCUS framework for coding and estimation.

5. Another Perspective

We address the problem of how to best allocate rates by using coded modulation to get the best performance. We first observe that there are two factors that contribute to the MSE of the system:

▲ ***Quantization error***: The quantization of the observation will induce distortion on the observation.

▲ ***Coset decoding error***: When the decoder selects the incorrect member of the chosen coset, this error will induce a (large) distortion on the observation.

We now interpret the DISCUS functionality from the familiar perspective of unequal error protection (UEP) channel codes. Recall that in a typical correlation scenario between X and Y, where X Y N□□□ , the LSBs of X and Y are least correlated, and the MSBs most correlated. Accordingly, DISCUS dictates that we spend more bit rate as we go from MSB to LSB. Qualitatively, as we approach the LSB region, we cannot extract any gains from side information, and we will have to pay a bit for a bit. As we approach the MSB region, we get more gains from the side-information and can use a family of unequal strength codes to extract this gain, needing weaker-strength codes (which cost us less and less) as we approach the MSB. Beyond a certain threshold, the MSBs are free.

This is illustrated in Fig. 6. Note the three markers: beyond the top (MSB) marker, the bit plane correlation permits no data needing to be sent. Beyond the bottom (LSB) marker, the bit rate budget will not permit further bit plane resolution. Between these two markers is the "syndrome" marker, which separates the "full price" zone from the "discount" zone. Of course, one can use multiple syndrome markers to reflect different shades of discount. These markers need to be optimized based on problem constraints. We draw parallels from this framework to that of multilevel coding in error correcting codes [22]. Note however that the analogy between DISCUS and the use of UEP codes for data transmission is completely opposite: in the latter, it is the MSBs that need higher strength codes!

5.1 Deployment in a Sensor Network

We illustrate through a simple example the power of DISCUS in the context of a sensor network. For simplicity, we consider the simple tree topology as given in Fig. 7.Further suppose that we use the following correlation structure in the tree to illustrate our concepts: the readings at all nodes are 3-bit binary values, and each child node is correlated with its parent node in the manner that the Hamming distance between child node readings and their parent node reading is no more than 1 bit. This exactly mirrors the example of "Example of Binary Sources."

Suppose the "central station" Node A wants to collect the readings from all other nodes in the network. This scenario often occurs in an ad-hoc network, when a certain node broadcasts a request for readings from other nodes. The "naive" solution would be to have the child nodes C and D send their 3-bit readings to their parent B, which would then relay these to A along with its own 3-bit reading. As a way of quantifying the amount of work done by the network, suppose each tree link is 1 m long. A metric that is used to measure the amount of energy expended in the network is bit-meters, referring to the number of bits times the distance traveled by the bits. Using this metric, a naive solution that ignores the correlation structure would expend 3 bit-meters each from Nodes C and D to Node A. Then A would expend 9 bit-meters to communicate with A (its own 3 bits plus the 6 bits of C and D) for a total of 15 bit-meters. Now let us consider the role of DISCUS in exploiting the correlation structure. Recall from "Example of Binary Sources" that nodes C and D can reliably communicate their 3-bit readings to their correlated parent node B using 2-bit syndromes. Node B can relay these messages along to node A, along with its own 2-bit syndrome with respect to node A. Node A invokes a successive decoding framework by first decoding its correlated node B based on its own reading and then decoding the readings of C and D relative to the decoded reading of B. As each 3-bit message in the original picture has been replaced by its corresponding 2-bit syndrome, the DISCUS-based scenario involves a reduction from 15 bit-meters to 10 bit-meters.

This toy example conveys the potential of DISCUS in a sensor network scenario. We can also illustrate a few other features. One drawback of the above example might be that Node A has to do all the decoding work. However, this can be alleviated by having Node B do the DISCUS decoding of C and D and relay to A their "differences" with respect to its own reading (in this case, the mod-2 sum). Due to the correlation structure, there are only four error or difference patterns {000, 100, 010, 001}, which can be indexed using 2 bits. The total network cost is still 10 bit-meters as before, but now there is some amount of "load balancing" between nodes A and B that might be desirable. Likewise, consider another scenario where node A wants to know the reading of node C only, rather than that of all the nodes. As in our example, there is no corre-lation structure between nodes A and C (as they are not a parent-child pair), this would cost 6 bit-meters. However, if node B can be used as a "trans coder," we can reduce this to 5 bit-meters by saving 1 bit-meter through DISCUS for the link between C and B. These examples illustrate that the DISCUS concept can be useful in a number of application scenarios depending on the network topology and correlation structure, leading to the promise of significant network energy savings.

6. Other Applications

The idea of coding with side information can enable a large range of applications. The coset coding framework that we have developed here can be used to enable various scenarios which make use of the "Slepian-Wolf binning" concept. We list out several promising applications and
their results:

***Multimedia transmission**:* We can use this framework to optimally upgrade an existing analog transmission by sending digital information. We treat the received analog signal as side information at the decoder. This work requires the graduation to more realistic models and is pre-sented in [23].The same framework can also be used for error resilient multimedia transmission by way of multiple description coding for lossy packet networks. We also address the uncertainty at the encoder about the actual packet losses [25].

***Blind Watermarking and Multi-user Communication**:* It can be shown that there are duality connections to another important problem of blind digital watermarking of signals as has been pointed out in [26] and [27]. Here we need to transmit messages by minimally perturbing

(water marking) some known signal such as speech or image or video. The decoder wishes to decode the message after the watermarked signal goes through some attack channel.

The broadcast channel, where a sender is communicating to many receivers, is intimately related to the blind watermarking scenario. We consider the signals of the previous users as the host signal and use the same watermarking framework to add more users. It has been shown that this is superior to using TDMA, FDMA, or CDMA [28].

7. Conclusions

We have presented a new domain of collaborative information communication and processing through the framework on distributed source coding. This framework enables highly effective and efficient compression across a sensor network without the need to establish inter-node communication, using well-studied and fast error-correcting coding algorithms.

8. References

[1] J.M. Kahn, R.H. Katz, and K.S.J. Pister, “Mobile networking for smart dust,” in *Proc. ACM/IEEE Int. Conf. Mobile Computing and Networking*, Seattle, WA, Aug. 1999.

[2] L. Doherty, B.A. Warnake, B. Baser, and K.S.J. Pister, “Energy and performance considerations for smart dust,” in *Int. J. Parallel and Dtstributed Sensor Networks*, 2001.

[3] T.M. Cover and J.A. Thomas, *Elements of Information Theory*. New York:Wiley, 1991.

[4] D. Slepian and J.K. Wolf, “Noiseless coding of correlated information sources,” *IEEE Trans. Inform. Theory*, vol. IT-19, pp. 471-480, July 1973.

[5] S. Verdu, “Fifty years of Shannon theory,” *IEEE Trans. Inform. Theory*, vol.44, pp. 2057-2078, Oct. 1998.

[6] S. Shamai, S. Verdu, and R. Zamir, “Systematic lossy source/channel coding,” *IEEE Trans. Inform. Theory*, vol. 44, pp. 564-579, Mar. 1998.

[7] A.D. Wyner, “Recent results in the Shannon theory,” *IEEE Trans. Inform.Theory*, vol. IT-20, pp. 2-10, Jan. 1974.

[8] A.D. Wyner, “On source coding with side information at the decoder,” *IEEE Trans. Inform. Theory*, vol. IT-21, pp. 294-300, May 1975.

[9] A.D. Wyner and J. Ziv, “The rate-distortion function for source coding with side information at the decoder,” *IEEE Trans. Inform. Theory*, vol. IT-22, pp. 1-10, Jan. 1976.

[10] Q. Zhao and M. Effros. “Broadcast system source codes: A new paradigm for data compression,” in *Proc. 33rd Asilomar Conf. Signals, Systems and Computers*, Pacific Grove, CA, 1999, pp. 337-341.

[11] R. Zamir and S. Shamai, “Nested linear/ lattice codes for Wyner-Ziv encoding,” in *Proc. IEEE Information Theory Workshop*, Killarney, Ireland, 1998, pp. 92-93.

[12] S.S. Pradhan and K. Ramchandran, “Distributed source coding using syndromes (DISCUS): Design and construction,” in *Proc. IEEE Data Compression Conf.*, Snowbird, UT, Mar. 1999.

[13] A. Gersho and R.M. Gray, *Vector Quantization and Signal Compression* .Norwell, MA: Kluwer, 1992.

[14] G. Ungerboeck, “Channel coding with multilevel/phase signals,” *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 55-67, Jan. 1982.

[15] M.W. Marcellin and T.R. Fischer, “Trellis coded quantization of memoryless and Gauss-Markov sources,” *IEEE Trans. Commun.*, vol. 38, pp. 82-93, Jan. 1990.

[16] J. Kusuma, L. Doherty, and K. Ramchandran “Distributed compression for sensor networks,” in *Proc. IEEE Int. Conf. Image Processing*, Thessaloniki, Greece, Oct. 2001.

[17] F.M.J. Willems, “Totally asynchronous Slepian-Wolf data compression,”*IEEE Trans. Inform. Theory*, vol. IT-34, pp. 35-44, Jan. 1988.

[18] B. Rimoldi and R. Urbanke, “Asynchronous Slepian-WoIf coding via source-splitting,” in *Proc. IEEE. Symp. Info. Theory*, Ulm, Germany, July 1997, p. 271.

[19] S.S. Pradhan and K. Ramchandran, “Distributed source coding: Symmetric rates and applications to sensor networks,” in *Proc. IEEE Data Compression Conf.*, Snowbird, UT, Mar. 2000, pp. 363-372.

[20] T.J. Flynn and R.M. Gray, “Encoding of correlated observations,” *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 773-787, Nov. 1987.

[21] Y. Oohama, “The rate-distortion function for the quadratic Gaussian CEO problem,” *IEEE Trans. Inform. Theory*, vol. 44, pp. 1057-1070, May 1998.

[22] U. Wachsmann, R.F.H. Fischer, and J.B. Huber, “Multilevel codes: Theoretical concepts and practical design rules,” *IEEE Trans. Inform. Theory*, vol. 45, pp. 1361-1391, July 1999.

[23] S.S. Pradhan and K. Ramchandran, “Enhancing analog image transmission systems using digital side information: A new wavelet-based image coding paradigm,” in *Proc. IEEE Data Compression Conf.*, Snowbird, UT, Mar. 2001, pp. 63-72.

[24] V.A. Vaishampayan, “Design of multiple description scalar quantizers,”*IEEE Trans. Inform. Theory*, vol. 3, pp. 821-834, May 1993.

[25] S.S. Pradhan, R. Puri, and K. Ramchandran, “(n,k) source-channel erasure codes: Can parity bits also refine quality?,” in *Proc. Conf. Information Science and Systems (CISS)*, Johns Hopkins Univ., Baltimore, MD, 2001.

[26] J. Chou, S.S. Pradhan, and K. Ramchandran, “On the duality between distributed source coding and data hiding,” in *Proc. 33rd Asilomar Conf. Signals, Systems and Computers*, Nov. 1999, pp. 1503-1507.

[27] M. Costa, “Writing on dirty paper,” *IEEE Trans. Inform. Theory*, vol.IT-29, pp. 439-441, May 1983.

[28] J. Kusuma and K. Ramchandran, “Coset codes for broadcast,” in *IEEE Symp. Information Theory*, 2002, submitted for publication.