AFRL-IF-RS-TR-2005-295
Final Technical Report
August 2005

DISTRIBUTED SIGNAL PROCESSING IN WIRELESS SENSOR NETWORKS

University of Southern California at Los Angeles

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. M279

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

STINFO FINAL REPORT

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2005-295 has been reviewed and is approved for publication

APPROVED: /s/

RICHARD C. BUTLER II
Project Engineer

FOR THE DIRECTOR: /s/

WARREN H. DEBANY, JR., Technical Advisor
Information Grid Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	AUGUST 2005	Final Aug 01 – Aug 04

4. TITLE AND SUBTITLE
DISTRIBUTED SIGNAL PROCESSING IN WIRELESS SENSOR NETWORKS

5. FUNDING NUMBERS
C - F30602-01-2-0549
PE - 61101E
PR - M279
TA - SE
WU - NS

6. AUTHOR(S)
Cauligi S. Raghavendra and
Viktor K. Prasanna

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Southern California
Department of Electrical Engineering Systems
EEB 2562
Los Angeles California 90089-2562

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Defense Advanced Research Projects Agency
3701 North Fairfax Drive
Arlington Virginia 22203-1714

AFRL/IFGA
525 Brooks Road
Rome New York 13441-4505

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TR-2005-295

11. SUPPLEMENTARY NOTES
AFRL Project Engineer: Richard C. Butler II/IFGA/(315) 330-1888/ Richard.Butler@rl.af.mil

12a. DISTRIBUTION / AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 Words)***
Sensor nodes forming a network and using wireless communications are highly useful in a variety of applications and scenarios. Such a wireless sensor network can be used to collect and process information from the field in military surveillance, building security, in harsh physical environments, and for scientific investigations on other planets. Sensor networks can perform distributed computations in the network for a number of interesting applications, including signal and image processing to detect object signatures, track detected objects in the field, statistical computations, etc. The objective of this research project is to develop time efficient distributed algorithms over wireless networks for some well known signal processing applications. Specific research problems include detection and tracking of objects using sensor data, application driven signal processing, and collective communications in wireless sensor networks.

14. SUBJECT TERMS
Distributed Sensor Networks, Energy Efficient Wireless Networks, Low Power Processing, Low Power Communications, SensIT Nodes, Distributed Algorithms, Signal Processing, Tracking, Data Compression, Data Correlation, Distributed Detection Algorithms, Energy Conservation in Wireless Networks, Two-Tier Sensor Network

15. NUMBER OF PAGES
58

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Table of Contents

List of Figures

List of Tables

1 Project Summary

Distributed processing applications in wireless sensor networks need to consider resource constraints. With high energy cost for wireless communications, application algorithms should be designed to reduce communication cost. Sensor readings are highly correlated spatially and temporally; furthermore, this correlation is extended to multi-modality sensor readings. Effective exploitation of these correlations can reduce data communication cost significantly. In most sensor applications, events are not happening in a consecutive manner; therefore, it is a waste of energy to keep nodes active for periods when no events are happening. In this project, we investigated energy efficiency and power awareness in wireless sensor networks. We have devised signal processing algorithms and networking protocols to support energy efficient operations in wireless sensor networks and have verified the effectiveness and energy efficiency of these algorithms and protocols using field data sets and experiments. The areas we focused on include: (i) Compression of sensor readings, (ii) Correlation of sensor readings, (iii) Detection of events in a sensor field. Following is the list of our major accomplishments:

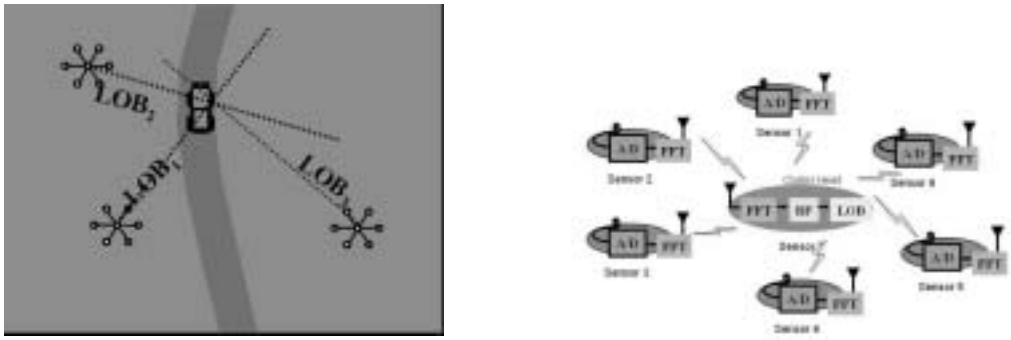
- Integer packing and run-length based coding of sensor readings without taking into account spatial correlation was done for applications where lossless compression with simple processing is desired.
- A compression scheme was proposed to exploit the spatial correlation which select coding parameters based on a joint energy, rate, and distortion cost function.
- A compression scheme was proposed to exploit spatio-temporal correlation. It uses a set-partitioning technique to organize large coefficients so that the spatial correlation can be effectively exploited at the level of significant coefficients.
- A power aware joint coding scheme was proposed for compression of correlated sensor readings and it also takes into account transmission power consumption and signal reconstruction fidelity.
- A scheme to quantify spatial correlation of sensor readings was proposed. This scheme uses linear predication to establish initial correlation and tracking based on Kalman filter to fine-tune the correlation.
- A wavelet based detection of events for a sensor field was proposed. It uses energy distribution as the criterion to detect false alarms.
- Tripwire cueing in sensor networks, which uses detection results to cue signal processing nodes to wake up only when necessary for advanced processing.

2 Introduction

Sensor networks became widely available as a variety of inexpensive COTS sensors capable of significant computations and wireless communication became available [45,9,7,13]. Sensor networks are becoming important in a number of areas, including target detection, biomedical engineering, environmental awareness and security surveillance. They are generally employed in a field for collection of signal data to collaboratively make decisions. Lightweight wireless enabled sensors are becoming more attractive due to their low cost, ease of deployment and high efficiency. These types of sensors have limited CPU and memory resources.

Sensors in this context are mostly powered by batteries; therefore, it is also important to efficiently use this limited energy supply by a node and by a network as a whole. As sensors have been built with more intelligence and move toward miniature size, the power issue is becoming the key obstacle to exploit the full potential of wireless sensor networks. Energy conservation by sensor nodes is therefore critical for many applications of sensor networks.

One typical application of wireless sensor networks is the Automatic target Tracking and Recognition (ATR), which uses a number of sensor arrays to classify or track targets in a sensor field using acoustic, seismic or infrared sensors. There are a few steps including (1) issuing an alarm of a potential target, (2) confirming the alarm, (3) classifying the target and/or tracking the target. At all these stages, carefully designed protocols and algorithms can help applications to conserve energy so that a network can work in a field with a longer lifetime. Figure 2.1(a) shows three sensor arrays tracking one target and Figure 2.1(b) shows the processing inside an array.



(a) Target Tracking

(b) Processing inside an Array

Figure 2.1: Automatic Target Recognition using Sensor Networks

Since not all components of a sensor node are used by an application at all times, one way to save energy is to turn off those components which are not needed, and switch them back on when a task execution needs them. When costs of switching components are taken into account, there can still be significant energy savings when a component is idle for a sufficiently long period. This idea has been

well exploited in hardware design of sensor nodes and has been proven to be quite successful, e.g. a deep-sleep mode introduced in LUTONIUM [43], and a node design in PASTA architecture [11]. The same idea can be extended to a sensor network. In a sensor network, a portion of the network can be switched off when events are far off and switched on when an event is imminent or is actually happening. To enable this idea in a sensor network for energy conservation, there are a few problems that need to be addressed. Namely, a network has to have a mechanism of identifying events and discriminating them, and most importantly, this mechanism does not incur high energy cost in itself.

Since radio communication is deemed to consume comparatively high power [52,61], it is equally important to reduce communication energy dissipation. Unlike in cellular wireless communications, sensor nodes are normally deployed on the ground with an antenna close to the ground, and they are equipped with low power radios. Due to these facts, ground scatter and reflection can seriously affect the radio signal. Therefore, per bit communication cost in wireless sensor networks is high.

Sensor readings collected by nodes in a sensor network are correlated. There are three types of correlations. First, since nodes are deployed in close proximity, sensor readings are spatially correlated. Second, since sensor readings are sampled in a short sampling interval, these readings are temporally correlated. Third, different sensing modalities of a sensor node also see these spatial and temporal correlations, and readings from different sensor modalities are correlated. In sensor network applications, there are in general two types of communication required: local communication and global communication. In-network processing is deemed to save communication; however local communication is required by many algorithms for achieving robustness and exploiting diversity for result accuracy. Therefore, data compression is useful to reduce communications cost among local sensors.

In many sensor network applications, sensor nodes collect acoustic, seismic or infrared data, and sensors collaboratively process signal data to detect/classify/track targets. In one scenario, a network consists of a number of sensor arrays, with each array having several homogeneous sensors, and each array is equipped with a radio card for wireless communications. Sensors in an array are geometrically close to each other, in the range of 100 meters, while inter-array distances could be much farther. (Fig. 2.2 shows a field test network with 4 sensor arrays and each has 3 sensors.) One sensor in each array is elected as the head, which aggregates the data from the members; e.g. nodes with line-of-bearing (LOB) shown in Fig. 2.2 and the head sensor node also observes the same event. The head may also perform tasks like information fusion.

There are a number of advantages in having signal processing performed inside a sensor array or a local region, and these advantages include low communication cost and low latency. Further information fusion at the inter-array level may be needed. Only local processing results, which can be represented by fewer bits, would need to be transmitted among sensor arrays. Figure 2.3 shows a target tracking application where one sensor array tracks one period of the target course. In it, there are three arrays (labelled as SA-1, SA-2 and SA-3). SA-1 tracks the target in the first segment (marked in dark gray), and SA-2 tracks the target in the second segment (marked in black) while SA-1's results are used as the initial position, and SA-3 starts to track the target once it moves into the third segment (marked in light gray). Further tracking is done by successive arrays based on the earlier tracking results. This example

Figure 2.2: Network of Sensor Arrays

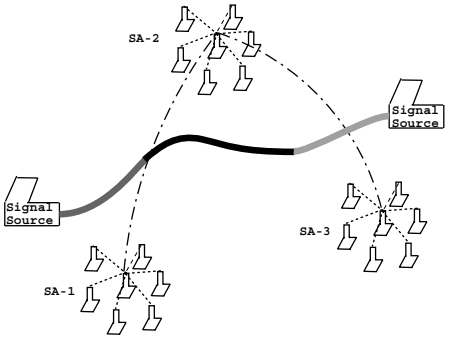


Figure 2.3: Target Tracking using Sensor Arrays

shows one common application characteristic of sensor networks, which uses in-network processing to save resources, e.g. energy savings due to less communication and memory activities.

There are many research activities to minimize the energy cost for the above scenario from both network and hardware standpoints [72,68,11,7]. These approaches may not provide sufficient energy savings for the scenarios of interest especially when the number of events is small and events happen infrequently. A sensor node must be kept awake all the time in order not to miss any event. Even though they can be put in low duty cycle mode when no event happens for a long period time, the energy dissipated may still be substantial due to various hardware constraints [10] and state switch cost. Furthermore, there will be delay in event processing since the state switch delay also needs to be taken into account. In addition, radio or sensing components must be awake at all times. For example, a PXA255 CPU consumes about 10 mW in sleep mode in order to keep its oscillators running and needs 10 ms to switch from sleep mode to active mode and the long-haul data radio consumes more than 100 mW when idle.

A two-tier sensor network could help conserve energy in wireless sensor networks. In this architecture, there are two types of sensors deployed, namely, trackers and tripwires. A tripwire consumes much less power compared to a tracker at a low duty cycle. For example, picoRadio can operate at less than 1 μW [45] as compared to the $10mW$ ORiNOCO radio. Tripwires are deployed along with trackers, and form

a network to monitor a sensor field so that trackers can be kept in sleep mode to conserve energy when no events are happening nearby. Tripwires wake up periodically to check the field status and wake up trackers for various signal processing tasks only when necessary. Since there is at least an order of magnitude difference in power consumption between a tripwire and a tracker, by using tripwires, a significant amount of energy can be saved. Figure 2.4 shows a field example with two targets whose trajectories are marked by white curves, where the star shaped objects are tripwires and "L" shaped objects are trackers. Different colors are used to indicate different power states of tripwires and trackers.

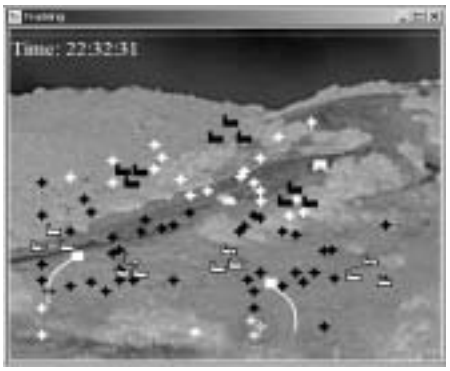


Figure 2.4: Two targets in a Two-Tier Wireless Sensor Network

Our contributions on compression of spatio-temporally correlated sensor data include a new scheme, called Embedded Set-Partitioning Iteratively Hierarchical Tree (ESPIHT), with practical bitplane coding for high compression ratio. To the best of our knowledge, this is the first scheme which deals with Slepian-Wolf and Wyner-Ziv problems on wavelet transformed data. The benefits include significantly better compression ratio with little processing overhead compared to other types of compression schemes. This scheme can further be made power aware by adapting the coding to channel conditions so that an optimal power level can be selected. This scheme combined with power control can reduce communication cost by more than 60% compared to a non-coded case.

For correlation analysis, we have developed a scheme which does not require sensor data calibration and is able to predict the correlation of spatial correlations in sensor data. It applies Discrete Kalman Filter (DKF) to further track correlation. The proposed DKF approach is an extension to the steepest descent method since it degrades to the gradient method when the state transition matrix in the state-space model is set to the identity matrix. This technique can be useful for many sensor network applications. We have examined applications such as the distributed source coding problem, sensor data storage systems and sensor data abstract.

For detection, we are focused on how to devise algorithms with sensors collaborating to reduce processing overhead and latency. In particular, we have developed a wavelet based distributed detection scheme. This scheme uses energy distribution in subbands to identify false alarms. It has an efficient search scheme and the transformation cost is split among a group of microsensors. A similar scheme based on FFT segmentation was also developed and tested. It fits well to more powerful sensor nodes

which can perform FFT transforms of small length. In both schemes, a fusion scheme was developed to generate detection predicates for node wake-up purposes. Combined with a two-tier heterogeneous sensor network, this scheme can bring more than 45% energy savings compared to the duty-cycling approach.

3 Background and Related Work

Radio communication dissipates significantly more energy compared to computation in sensor nodes. As sensor networks are severely constrained by energy, it is important to reduce communication energy. Compressing data to reduce the number of bits to be sent to meet application demands is a good approach to reduce communication energy dissipation in sensor networks.

In general, sensor readings from near-by sensors are correlated in space. Rather than straight data compression, spatial correlation can be exploited to achieve improved data compression. The distributed source coding problem of correlated sources has been studied extensively [53,44,25,22]. In [25], a two-stage iterative approach is devised followed by an index-reuse which is aimed to exploit spatial correlation. In [44], a data compression scheme, called DISCUS that exploits spatial correlation is presented. The idea is to use cosets to partition the code space so that a coset index in the form of syndrome is sent and the recipient will infer the correct codewords through the syndrome and side information. A trellis based code is used to partition code space in DISCUS. DISCUS has been shown effective when the sensors are densely deployed and is suitable for data collection type of applications, since encoding complexity is very simple while decoding is done in a node of sufficient power supply. However, this technique only exploits spatial correlation. In [22], an interesting problem of correlated coding and transmission has been studied and the rate allocation at sensing nodes is another degree of freedom in optimization of data gathering problems. There are two theorems which give lower bounds of rate for the lossless and lossy compressions. The Slepian-Wolf [53] correlated coding theorem addresses the case of lossless compression while the Wyner-Ziv rate-distortion theorem [66] addresses lossy compression of correlated sources.

The Slepian-Wolf problem refers to the lossless distributed correlated source coding problem and the sum of the achievable rates is lower bounded by the joint entropy, where encoding is done independently of each other and decoding is done jointly. For a two-source case X, Y, the achievable rate regions which can independently describe X and Y are determined by $R_X + R_Y \geq H(X;Y)$, $R_X \geq H(X|Y)$ and $R_Y \geq H(Y|X)$, where R_X is the rate to describe X and R_Y is the rate to describe Y. In this region, two corner points are corresponding to the cases where one source is described by its entropy $H(X)$ ($H(Y)$) and the other source is described by the conditional entropy $H(Y|X)$ ($H(X|Y)$). Practical codes close to these corner points have been designed as shown in Tab. 3.1. There are two general design methodologies: syndrome partitioning and random binning. Syndrome partitioning is a deterministic binning approach. Different codes are employed by these designs including Trellis codes, Turbo codes and low density parity-check codes. In [51], a general design method is proposed to approach any point on the lower boundaries of the achievable region. It uses LDPC codes and partitions the parity-check matrix for different points on the region boundary.

The Wyner-Ziv coding problem refers to the lossy version of the Slepian-Wolf problem in which the achievable rate of source to be coded is specified by the Wyner-Ziv rate-distortion function. Only for the correlated Gaussian sources of the Wyner-Ziv problem, the achievable rate is shown to be the same regardless whether the encoder has the knowledge of the side information [66]. For the case of doubly

Table 3.1: Slepian-Wolf Code Designs

Design	Codes Used	Binning Method
[44]	Trellis Code	Syndrome Partitioning
[12]	Turbo Code	Random Binning
[21]	Turbo Code	Syndrome Partitioning
[27]	Turbo Code	Random Binning
[39]	LDPC Code	Syndrome Partitioning

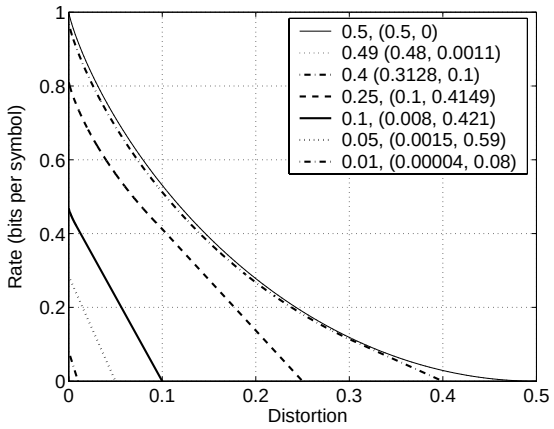


Figure 3.1: Rate Region of Doubly Symmetric Binary Source for Wyner-Ziv Coding Problem

symmetric binary sources (i.e. the source and side information can be considered to form a binary symmetric channel), without communication with the source as the side information during encoding, a rate between $R_{X|Y}(d)$ and $h(p(1-d)+(1-p)d)-h(d)$ is achievable, where $h(.)$ is the entropy function; p is the crossover probability, and $R_{X|Y}(d)$ is the minimum rate of X when it is encoded with knowledge of Y. More specifically, the achievable rate is determined by a lower convex envelope of the following set:

$$\{(d, h(p(1-d)+(1-p)d)-h(d))|0 \leq d < p\} \cup \{(p,0)\}.$$

Figure 3.1 shows a few such curves. From Fig. 3.1, the tangent points, as shown in the legend, could be arbitrarily close to both extreme points, i.e. $(0,0)$ and $(0.5,0)$. This tells us that the achievable rate at distortion d is almost always $h(p(1-d)+(1-p)d)-h(d)$ when p is close to 0.5, and that is almost always along the straight line as $\frac{(p-d)H(X|Y)}{p}$ when p is close to 0. Known approaches, e.g. [38,44], to the Wyner-Ziv problem use the above methodologies with bin dilution for a smaller rate with a possible error probability bounded away from 0, even when the codeword length asymptotically approaches infinity.

These schemes can be used for exploitation of spatial correlation in sensor readings. However, care should be taken when applying these schemes since the correlation structure presented in raw sensor readings may not meet the assumptions of these schemes. This results in many schemes applicable only to dense sensor networks where spatial correlation is high to compensate the overhead of these schemes

[44]. The challenge is how to organize data so that distributed source coding schemes can effectively exploit spatial correlation. Besides this, a practical scheme should also consider that a proper coding rate is always used to control the signal fidelity to meet application SNR requirements. This is important in sensor network applications due to field dynamics. Therefore, the challenge is to devise a scheme which is flexible enough to provide different levels of compression and fidelity for calibrating data so that the codec in question always uses the right rate.

In wireless sensor network applications, sensor readings are not only correlated in space, but also correlated in time. Wavelet transforms tend to compact signal energy into few subbands and a few large wavelet coefficients can represent the signal to a certain level of fidelity. By devising a clever way to organize these large coefficients and coding them appropriately, it seems reasonable to expect good compression performance in terms of exploitation of temporal correlation.

Additional gains can be achieved by exploiting both spatial and temporal correlation which lead us to the development of a novel spatial and temporal compression scheme for sensor networks. An interesting observation by our experimental study reveals that spatio-temporal, spatial-only and temporal-only schemes can apply to different types of sensor network applications and our proposed spatio-temporal compression fits well in all correlation scenarios via some simple codec parameter adjustments.

Correlation analysis is useful in many applications, for example, compression applications [57,20], beamforming for target tracking [19], and storage systems in sensor networks [26] [28], and routing problems with compression [55]. Since spatial correlation in these applications will be high and evolves slowly, a linear prediction [42] approach to predict future correlation is normally effective, and it can also be made efficient. In [20], an algorithm based on linear prediction and gradient method has been applied for the compression of sensor data and a zero-pole model of linear prediction is used. An implicit assumption in [20] is that sensor readings are already calibrated so that the single-zero prediction model could work to exploit the spatial correlation (the temporal correlation is done via an all-pole model in [20]).

Let $\mathcal{H}$ be the vector space of random variables (one random variable is a vector), and define the following inner product and the norm ($\mathcal{H}$ is therefore a Hilbert space):

$$< X, Y >= \mathcal{E}[X, Y], \qquad \|X\| = \sqrt{< X, X >}.$$

In other words, the inner product of X and Y is their covariance and the norm of X is the standard deviation of X.

For a given index i, the prediction error to x_i (unknown random variable) based on $m+l+1$ samples from random variable Y can be represented as $\sqrt{| < x_i, Y' > |}$ where Y' denotes the random variable derived from the $m + l + 1$ samples where $m + 1$ is the order (lag) of the forward prediction and $l + 1$ is the order (lag) of the backward prediction. Under the linear prediction model, we have

$$Y' = \sum_{j=0}^{m} \alpha_j y_{i+j} + \sum_{j=1}^{l} \beta_j y_{i-j},$$

and $< x_i, Y' >=$

$$\sum_{j=0}^{m} \alpha_j < x_i, y_{i+j} > + \sum_{j=1}^{l} \beta_j < x_i, y_{i-j} > . \tag{3.1}$$

In order to minimize the prediction error, we need an orthogonal basis of the subspace spanned by vectors $y_{i-l}, \cdots, y_i, \cdots, y_{i+m}$. By the orthogonality principle, which states that the error vector is uncorrelated to its input vector [33], if replacing the vectors in eq.(3.1) by the orthogonal basis, the unique decomposition of x_i corresponding to the basis has a set of coefficients which are exactly the coefficients which minimize the error of the linear predictor. Fortunately, under the stationary assumption, the order recursive Levinson-Durbin algorithm [24] up to order $m + l + 1$ can efficiently find these coefficients in time in $\mathcal{O}((l + m)^2)$ and space in $\mathcal{O}(l + m)$.

The linear prediction model is effective for certain applications and it is based on the observation that prediction coefficients change in a slower manner than the time series samples. These coefficients can be estimated periodically. However, the computation is heavy for low power sensor nodes. Therefore, linear prediction is used in our initial estimate.

The Levinson-Durbin algorithm described above can be used to solve a Linear Prediction (LP) system as follows: where P is the coefficient matrix and it is defined as follows:

$$Pz + b = 0 \tag{3.2}$$

$$P = \left(\sum_{i=1}^{L} y_{i+k} y_{i+j} \right)_{k \times j},$$

for $-l \leq k, j \leq m$ and

$$b = \left(\sum_{i=1}^{L} x_i y_{i-l}, \cdots, \sum_{i=1}^{L} x_i y_i, \cdots, \sum_{i=1}^{L} x_i y_{i+m} \right)^{\tau}.$$

Note that $\mathcal{F}_Y$ sequence starts from y_{-l}, where l is the lag on S-forward prediction. Matrix P is symmetric and almost always invertible. The solution to eq.(3.2) is $T = P^{-1}b$.

It takes time on the order of $\mathcal{O}((m + l)^3)$ to directly compute the inverse of P of eq. (3.2). P is an estimate of a standard covariance matrix and it is a symmetric and positive definite Toeplitz matrix in practice. When L (i.e. the sample length used in the prediction model) is large enough, this estimate approaches the covariance matrix of Y and we can simply use the Levinson-Durbin algorithm to solve eq.(3.2) for the LP coefficients. Note that in this case, the constant vector of the $m + l + 1$ simultaneous linear equations is different from that in the regular case which requires it is identical to the first column of P shifted by one element and with opposite signs. The Levinson-Durbin algorithm used in the research is a modified algorithm based directly on the orthogonality principle above. There is also a "superfast" algorithm [15] for the modified system of equations of eq.(3.2) taking time on the order of $\mathcal{O}(n \log^2(n))$.

The overhead can be manageable in a microsensor node especially when the algorithm is only executed periodically.

Many techniques have been explored for energy savings for signal processing in sensor networks. Previous endeavors focused on how to schedule trackers to go to sleep or to wake up trackers based on detection outcomes. On the one hand, all nodes wake up according to a predefined schedule so that only involved nodes are kept active for the exact duration of a task execution. Recently a new MAC - S-MAC [70] has been proposed, and it enables nodes to sleep not only for a scheduled period, but also for other periods for which it can infer by observing neighbor's behavior. This scheduling approach only fits well for certain applications and can save significant energy when a good network structure is known and a schedule which matches task execution behaviors can be set. However, it can incur heavy initation and coordination cost when these conditions are not met. On the other hand, tripwire (i.e. lightweight sensor node) is used to monitor the field and trackers (i.e. signal processing sensor node) are blindly woken up whenever there are some high energy activities in a field. Some of them could be caused by noises. One topic investigated in this research is devising a framework for effectively utilizing these tripwires.

An alternative approach to a two-tier network is to directly use sensor nodes to monitor the field and wake up other components whenever necessary, e.g. the radio module for wireless data communication. This would avoid the problem in a two-tier network (i.e. tripwire network tier and tracker network tier) of increased latency of notification. The energy-latency trade-off has been studied in [37] for the data gathering application. It is also an interesting problem in the context of two-tier sensor networks and it deserves to be examined.

A tracker waking-up scheme has been presented in [54] where the focus is on how to select a set of optimizing trackers from the application point of view. Although the work there is orthogonal to the tracker cueing scheme we proposed, an alternative cueing scheme (denoted by INIT) can be constructed for that algorithm by the following steps: (1) any tripwire initially constructs a local database of trackers; (2) online optimization algorithm proposed in [54] selects a set of trackers to execute an application. This INIT scheme however, does not solve the so-called exposed alarm problem and false alarm problem. The exposed alarm problem refers to the wake-up of inappropriate nodes which could be caused by tripwire energy threshold detection. The algorithm proposed in [54] can be executed by a set of woken trackers to select an optimal set of trackers for the task in question.

Efficient and effective detection algorithms are critical for energy savings in sensor networks. Only when events are detected, should a processing on it start. Receiver Operating Characteristics (ROC) based approach for narrowband signal detection has been well studied in sonar, acoustic and imagery applications [56]. There are limited works of detection on far-field wideband signal detection. In [41], an adaptive threshold detection algorithm has been proposed and it uses field noise to adjust the energy threshold so that a Constant False Alarm Rate (CFAR) can be achieved. For a narrowband far-field signal, the amplitude envelope with Gaussian noise can be modelled by the Rician distribution as shown in eq. 3.3.

$$f(r) = \frac{r}{\sigma^2}\exp\left(\frac{r^2+S^2}{2\sigma^2}\right) I_0\left(\frac{Sr}{\sigma^2}\right), \tag{3.3}$$

where,

$$I_0(y) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \exp(y\cos(t))dt,$$

and σ is the real noise variance or imaginary noise variance and S is the narrowband signal amplitude. The noise amplitude follows the Rayleigh distribution as shown in eq. 3.4, which is for a random variable as a square root of sum squares of two Gaussian random variables.

$$f(z) = \frac{z}{\sigma^2}\exp\left(-\frac{z^2}{2\sigma^2}\right). \tag{3.4}$$

The false alarm rate at an amplitude threshold T is given by eq. 3.5.

$$P_{fa} = \int_{T}^{\infty} f(z)dz. \tag{3.5}$$

Likewise, the false positive rate with threshold T can be obtained by eq. 3.6.

$$P_{fp} = \int_{T}^{0} f(r)dr. \tag{3.6}$$

However, when event signals are wideband in many acoustic target tracking applications, above closed-form formulae for false positive and false alarm rates seem hard to obtain. Therefore, an alternative approach is needed to identify false alarms in this situation. In [62], a Distributed False Alarm Detection algorithm called DFAD in wavelet domain is proposed. It uses a group of tripwires to perform a false alarm detection once an alarm is received or generated by itself. The decision fusion is used in the end to generate a predicate to determine whether or not to wake up trackers.

4 Methods, Assumptions and Procedures

We have worked on developing coding algorithms, correlation analysis algorithms and detection algorithms for signal processing in wireless sensor networks. We proposed several algorithms to exploit spatial and spatio-temporal correlations, and for energy savings. We also developed efficient detection algorithms to facilitate node control so that nodes are active only for periods in which they are needed.

4.1 Compression in Wireless Sensor Networks

In a specific ATR scenario with acoustic data from ground vehicles, sensor nodes collect data at the rate of 1024 samples per frame. In this application, sensor nodes need to send their collected data to a central node. Here, there is high spatial correlation as all sensors collect data from the same target(s). Therefore, an efficient coding scheme that can exploit the spatial correlation will reduce the number of bits transmitted, and hence energy. From the ATR field data, we apply our proposed coding scheme and achieve a factor of 8 reduction in bits transmitted without losing quality in target detection.

In this work, our focus is on energy efficient coding schemes for wireless sensor networks. We first applied integer packing and run-length coding which is able to reduce the rate to 3 or 4 bits per sample where as the sensor ADC precision is 16 bits. However, our analysis has shown that this type of compression scheme gives poor performance when energy consumption for encoding and decoding processing overheads are also considered. We then describe a new coding scheme called EEADSC which minimizes the Lagrangrian cost function $R + \lambda_D D + \lambda_E E$ where R is the bit rate, D is the distortion, E is the energy and λ_D, λ_R are the Lagrangrian coefficients. The proposed scheme fully exploits spatial correlation in wireless sensor networks and is adaptive according to the Received Signal Strength Indication (RSSI). We evaluated the proposed scheme using a data set from an ATR application which achieved up to a factor of 8 data compression. EEADSC uses TCQ quantization and trellis encoding to represent a 16 bit data value by as few as 2 bits. With the scheme, we reduce the overall energy cost for communication for this application by a factor of 2.53, including the overhead cost for processing in encoding/decoding. Table 4.1 shows the compression energy gain and processing overhead for different schemes. The proposed scheme gives the best performance when the above cost function is considered.

Table 4.1: Energy Profile of Coding Schemes

	Unco.(Tx)	Unco.(Rx)	EEADSC(Tx)	EEADSC(Rx)	Pack(Tx)	Pack(Rx)
Comp.	0.0	0.0	331.5 mJ	177.29 mJ	230.5 mJ	245.0 mJ
Comm.	969.2 mJ	840.8 mJ	107.99 mJ	93.86 mJ	219.34 mJ	189.56 mJ
Total	1810 mJ		710.65 mJ		884.40 mJ	

4.1.1 Power-aware coding for Spatio-temporally Correlated Sensor Data

By noticing the fact that sensor readings are correlated not only spatially but also temporally, we have proposed a new data compression scheme, called ESPIHT [57], that exploits this spatio-temporal correlation present in sensor networks to reduce the amount of data bits transmitted in a collaborative signal processing application. The proposed ESPIHT seamlessly embeds a Distributed Source Coding (DSC) scheme with a SPIHT based iterative set partitioning scheme to exploit both spatial and temporal correlation.

ESPIHT is based on the well-known set-partitioning principle of wavelet coefficients. A new feature of ESPIHT is that distributed source coding is performed while coding significant coefficients during a refinement phase. We have implemented a new scheme similar to that of a Set-Partitioning Iteratively Hierarchical Tree (SPIHT) [50], which is aimed to exploit temporal correlation in sensor readings. We highlight our modifications to the general SPIHT as follows. The general SPIHT is used for image coding based on a 2-D wavelet transform. We have developed a 1-D version of SPIHT which is suitable for compression of sensor readings (we use SPIHT to refer to our implementation for sensor applications hereafter).

Under the set partitioning rule of SPIHT, wavelet coefficients are partitioned in each iteration into three types. They are denoted as ISO, LIP and LSP, and defined as follows: LIP type of coefficients are not significant in an iteration; furthermore, none of their descendants are significant; ISO type of coefficients are isolated from further processing (i.e. they are parents of some LIP nodes) and are insignificant at an iteration, but they could become significant in further iterations; LSP type of coefficients are significant in an iteration. At a refinement phase, a bitplane deduced from an LSP queue is output. Changes made to regular SPIHT are as follows: (1) use of a stack structure instead of queue structure to reduce memory requirement, (2) directly outputting of residuals to reduce the number of bit operations in the refinement phase, (3) use of a dynamic programming approach with deepest-first tree traversal to identify significant coefficients, (4) use of two mark bits to facilitate the search in (3) with one bit for coefficient significance and the other for subtree significance which is rooted at the visiting node (a binary subtree is significant if there is one or more significant coefficients in it).

After computing the wavelet transform, wavelet coefficients are arranged into wavelet coefficient trees (i.e. a speudo-binary tree with a node corresponding to a coefficient according to its subband position and offset in a subband) with roots in the coarsest subbands following a specific zigzag scanning order of these roots. SPIHT iterates on bit positions of these coefficients. Assume there are $q + 1$ iterations, and it will stop at the g-th bit position, then the largest coefficient has $g + q$ bits. The iteration starts at the $(g + q)$-th bit position. During the i-th iteration, SPIHT traverses further down the tree to search for any significant coefficient which is greater than or equal to 2^i as the quantization threshold. For each coefficient in the LIP stack (LIP stack is a stack data structure to hold LIP type of coefficients), the codec records path bits and mark bits if a significant coefficient is found (the decoder needs this information to identify the position in the wavelet tree) and pushs any ISO coefficient into an ISO stack. Figure 4.1 shows one example of a wavelet decomposition tree. The dark nodes in Fig. 4.1 are significant coefficients after the current iteration; gray ones are left to successive iterations, and the circle nodes are in LIS after

the current iteration. Note that there is also an ISO node in Fig. 4.1 as shown.

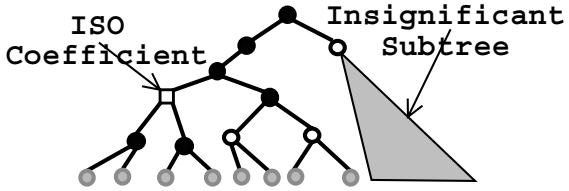


Figure 4.1: Search of Significant Coefficients

Instead of immediately outputting identified significant coefficients, SPIHT pushes them into an LSP stack (the LSP stack holds any newly identified significant coefficients, and the ISO stack is defined likewise). Coefficients in the ISO stack are all checked at the end of the current iteration. Any significant coefficient will also be pushed into the LSP stack. After all significant coefficients are found, these coefficients in the LSP stack are popped up with the most significant bit and sign bit removed. The rest of the bits are then encoded into a bitstream using a non-binary distributed source coding algorithm. There is also a stack called LIS which is defined as the placeholder for subtrees which need to be checked for significance for the next iteration. SPIHT outputs three types of bits: path and mark bits, LSP bits from the LIP stack and LSP bits from the ISO stack. These types of bits are named as path bits, fidelity bits and refinement bits. Path bits cannot tolerate any error. Fidelity and refinement bits may introduce distortion at different levels.

Fidelity bits and refinement bits are further compressed by a distributed source coding codec which exploits the spatial correlation between these significant coefficients of the sender and the corresponding coefficients (which of course may not be significant at the same iteration) of the receiver. Our codec is based on the syndrome partitioning method as in DISCUS [44] and [71], where syndromes are used to partition the space of fidelity and refinement bits into cosets, and coset indices in the form of syndromes are transmitted. Selection of DISCUS here is due to its simple decoding algorithm. Alternative schemes using Turbo codes [27] or LDPC codes [67] can be exploited, as well.

Decoding of ESPIHT consists of three steps: (1) decoding of fidelity and refinement bits using syndrome to select trellis at each step so that the codeword could generate an all-zero syndrome under it, (2) recovering of wavelet coefficients, and (3) reconstructing the time domain samples. Steps (2) and (3) follow the reverse steps of encoding explained above. In (1), to generate side information for decoding, coefficients in the reference frame at the decoder are identified based on the path bits encoded in the bitstream. A trellis based decoding scheme using the coset indices and these reference coefficients are actually used to recover these bits. The decoder needs to transform its frame using the same wavelet basis and the same number of levels as used during encoding of the source frame. Since the position information of the significant coefficients at the encoder is embedded in the path and mark bits segment, these corresponding coefficients search in the reference frame and can be found in a straightforward manner.

As having been pointed out earlier, in wireless sensor networks, radio communication cost is high and the channel is significantly inferior to other types of networks, e.g. the Internet. Compressed data is more sensitive to bit errors. However, too much protection on all the compressed bits may lose the benefits

of source coding due to the high ratio of redundancy in channel coding. If the source coder can provide information on the importance of compressed bits, different levels of protection can be provided so that critical information bits is well protected while the application can tolerate some errors on less critical information bits. There are three ways to support this type of protection: (1) channel coding, (2) high radio transmission power and (3) an ARQ protocol. In order to provide different levels of protection on the compressed bits, we take into account all three factors and derive a formulation to minimize system energy consumption. The resulting coding scheme is optimal in terms of energy conservation.

For practical systems, it has been shown that joint source and channel coding may reduce distortion, improve efficiency as well as reduce delay [29,40]. In this work, our targeted system is a wireless microsensor network with applications running on it. We use joint source-channel coding and power control on the radio communications to reduce the system energy consumption while maintaining a required SNR gain of the reconstructed signal. The proposed approach applies the ARQ protocol to the output of the code and combines a source coding scheme, a rate adaptable channel code and power control [58]. We have used both the temporal-only SPIHT [50] and the spatio-temporal ESPIHT [57] source coding schemes in our research. The proposed method is not limited to these source coding schemes. Any source coding scheme will work as long as it can separate compressed bits into different levels of importance. The resulting code can provide an Unequal Error Protection (UEP) on source coded bits. In [18], the importance of symbols are discriminated by designing modulation scheme with different signal distance on the constellation. However, in this work, we use different Rate Compatible Punctured Convolutional Codes (RCPCC) for symbols with a different level of importance. The ARQ adopted in this work is Type-I hybrid ARQ protocol [47], and is based on the ITOH algorithm [69]. In our analysis, we have derived a lower bound on the energy savings and an upper bound on the SNR loss of the proposed scheme.

Rates used for sensor data should adapt based on its importance. An unnecessarily low rate (high bit redundancy) on channel coding will consume energy without providing extra SNR gain or boosting accuracy of application results. In this subsection, we shall first work on a rate selection of $\mathcal{U}_h$ packets, then derive rates for $\mathcal{U}_m$ and $\mathcal{U}_l$ packets using RCPC codes.

Assume events of any packet delivery (repeated packet or original packet) are independent, and also assume one ARQ request is sent by a receiver if one packet has erroneous bit(s) upon channel decoding using ITOH algorithm. Furthermore, there is no loss of ARQ packets. Let $\epsilon(\lambda)$ denote the probability for a packet to be requested indicated by an ARQ with code rate $\frac{1}{\lambda}$, n be the number of packet, and $n+k$ be the number of packet transmission trials for these n packets. The Probability Mass Function (PMF) $p(k,\lambda)$ for k with code rate $\frac{1}{\lambda}$ is as follows:

$$p(k,\lambda) = \begin{cases} \sum_{i=1}^{k} \binom{n}{i} P_i^k \epsilon(\lambda)^k (1-\epsilon(\lambda))^n & k \leq n \\ \sum_{i=1}^{n} \binom{n}{i} P_i^n \epsilon(\lambda)^k (1-\epsilon(\lambda))^n & k > n \end{cases} \tag{4.1}$$

where P_i^k satisfies the following recursive formula:

$$P_i^k = \sum_{j=i-1}^{k-1} P_{i-1}^j \tag{4.2}$$

where $P_j^j = 1$, and $P_1^j = 1$ for $\forall j \geq 0$

From the application standpoint, delivery of n packets has to be done in a timely fashion, so the maximal number of repeat requests is fixed. Let k_0 be the maximal number of repeated requests of packets, and p_0 be the required packet delivery probability. Rate selection based on an energy criterion is formulated as the following Integer Programming Optimization Problem (IPOP):

$$(\tilde{\lambda}, \tilde{\tau}) = \underset{\lambda,\tau}{\operatorname{argmin}}(\lceil(\lambda - 1)n\rceil + k)\tau$$

subject to:

$$\lambda > 1 \text{ and is rational (implementable);} \tag{4.3a}$$

$$\sum_{i=0}^{k} p(i, \lambda) \geq p_0 \tag{4.3b}$$

$$\sum_{k=0}^{k_0} p(i, \lambda) \geq P_{sig} \tag{4.3c}$$

$$\tau_0 \leq \tau \leq 1 \tag{4.3d}$$

where P_{sig} is the minimal packet delivery probability and close to 1; τ is an antenna power factor between τ_0 and 1, and τ can take one of a fixed number of discrete values. The number of antenna power levels is fixed and the power on each level is also fixed. Constraint (4.3c) of IPOP is derived from an application latency requirement. The objective function in IPOP consists of two parts, namely, $\lceil(\lambda - 1)n\rceil\tau$ and $k\tau$. The first part is redundancy cost introduced by channel coding and the second part is repetition cost of some packets introduced by the ARQ protocol. Analytical optimal solution for the above problem is not always obtainable since the channel is not stationary. Due to constraints on hardware implementation, certain coding rates may be too costly and end up with too much circuitry power dissipation. One solution could be to solve IPOP with a feasible solution space with a limited pre-selected set of rates (the space in question is presumably smaller).

In order to see how transmission power affects the objective function value with the coding rate fixed at $\frac{1}{\lambda}$, denote $\Delta k = k_2 - k_1$, where k_1 is the number of retransmissions after power adjustment; k_2 is the number of retransmissions before power adjustment; τ is the power factor before power adjustment. Suppose the power after adjustment is $\tau + \Delta\tau$. In order to save energy, the following must hold:

$$\Delta k > (\lceil(\lambda - 1)n\rceil + k_1)\frac{\Delta\tau}{\tau}$$

Note that $k_1 < k_2$ in general if $\Delta\tau > 0$

To select an appropriate power level, the following formula [60] is used:

$$p_i = \alpha d_i^n + \beta$$

For example, for an Orinoco WaveLAN radio in a free space at a noise floor of -119dBm, $n = 2$, $\alpha = 14.8mW$, and $\beta = 11mW$. The path loss estimate uses the following formula (see Chapter 3 and Appendix B of [46]):

$$\overline{PL} = \overline{PL}(d_0) + 20\log\left(\frac{d}{d_0}\right) + X_\sigma$$

where $\overline{PL}$ is the average path loss in dB at distance d away from the transmitter in a free space; d_0 is a close-in distance, and X_σ is a random variable with zero-mean and variance σ. If the close-in distance is selected so that $\overline{PL}(d_0) = 0$, signal strength at d can be computed in dB as follows:

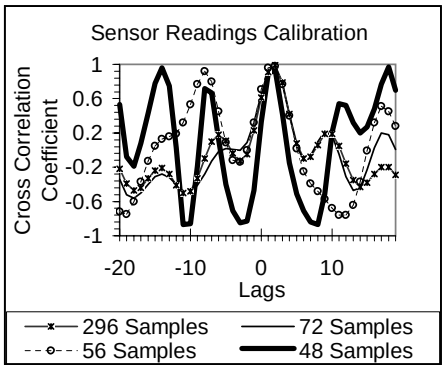
$$w = -20\log\left(\frac{d}{d_0}\right) + X_\sigma$$

Thus, using the above formula, a received signal SNR can be derived from its associated noise figure.

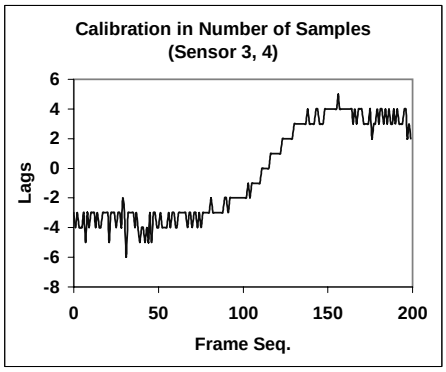
We select a code for path-bit packets via solving IPOP with a list of feasible channel code rates. This is done offline using an exhaustive search based on training data sets. Once a code is determined for path bits, we are able to determine codes for refinement bits and fidelity bits. In practice, an exhaustive search is possible since both the number of RCPC codes and the number of power levels are small. (We set both numbers to 10 in the experimental study and the search space is only of size 100.) However, if the granularity is too fine, IPOP turns out to be intractable.

4.2 Correlation Analysis and Its Applications

In general, sensor readings can be aligned when the polarity difference of the sensing signals is not too big from two sensors. Readings from close sensors are not skewed far apart. For example, they are off by -20 to 20 sampling intervals in a near field at 1 KHz sampling rate for moving objects at speeds of up to 60 mph. Figure 4.2 shows that spatial correlation exists in sensors' readings and node calibration varies from frame to frame. Figure 4.2(a) shows the cross correlation coefficients of two frames gathered at the same time from two nearby sensors. The correct lag should be 3, which can be computed with at least 72 samples every second. When only 56 samples are available, the ambiguity is apparent by noticing the two peaks at -8 and 3, and no correct lag can even be derived when only 48 samples are available. Figure 4.2(b) shows the calibrated lags over a period of 200 seconds of two nearby sensors. From Fig. 4.2(b), as it is known that sensor correlation changes over time, node calibration has to be done periodically. This incurs serious overhead for many applications, and calibration does not take into account the temporal correlation of sensor readings either. Therefore, its use in sensor network applications is rather limited.



(a) Frame Correlation



(b) Node Calibration

Figure 4.2: Correlation Lags in Sensor Networks

We have devised two different tracking algorithms and studied their performances. One approach is to track the correlation once the alignment of two sensors are established. This can significantly reduce the number of bits sent. The other approach is to use a Discrete Kalman Filter to perform the tracking using appropriate input control in each prediction update, once a linear predictor generates an initial alignment between two sensors. To improve the tracking accuracy, periodical re-initialization can align two sources. Trade-off is played out between tracking accuracy and update frequency. Processing overhead is manageable since the main overhead, i.e. periodical re-alignment, is done infrequently due to high tracking accuracy.

We assume the sensors' clocks are partially synchronized. By that, we mean that the time difference between two samples recorded at the same time by two different sensors is within one sampling period. For the case of 1 KHz sampling rate, the time difference for the above is within 1 ms. Note that the proposed algorithm will work with a coarser clock synchronization. However, the processing overhead will increase. There are some efficient synchronization schemes developed for sensor networks, e.g. [31]. We note that the synchronization requirement is not necessary at a network scale, but rather in the direct vicinity of a node with which to correlate. This requirement is much easier to fulfill than a global synchronization. The proposed tracking algorithm is shown by the following steps:

(S1) Each sensor samples the signal independently.

(S2) A sender sends a few samples which are only temporally coded in order to establish the initial coefficients.

(S3) The correlating node estimates the LP coefficients based on the received samples with the aid of offline computed parameters.

(S4) The correlating node applies DKF to track the correlation by performing the following steps:

(S4-A) Compute gradient as input control;

(S4-B) Compute Kalman gain matrix and steady state error covariance;

(S4-C) Compute the innovation and update state vector as the correlation coefficients;

(S4-D) Repeat from S4-A for a given number of steps.

(S5) To reconstruct approximate frames, the correlating node uses a steepest descent method based on the coefficients from the estimated coefficients of the DKF.

(S6) The correlating node loops back on S4 after each timeout period.

(S7) Based on application fidelity requirements, a new round of analysis can start by looping back to S2.

The initial estimation requires some fixed number of samples to be transmitted over to the correlating node, and this number of samples is determined by the LP orders. In general, when a fixed set of signals are present, this initial estimate can be done offline and occurs only once. For some acoustic signals in our experiments, this needs to be done periodically and the period of estimate depends on underlying signal characteristics.

This technique can be applied to a broad spectrum of applications in sensor networks. We explain two different ways of utilizing this technique: coding and sensor storage systems.

For two correlated sources X and Y, each sample is drawn from an independent, identical distribution. For the lossless compression case, the achievability of the Slepian-Wolf theorem [53] gives the lower bound of the transmission rate. The achievability of the Slepian-Wolf theorem states that for two iid sources, a rate close to the joint entropy $H(X,Y)$ can encode the two sources and the decoding error can be made arbitrarily small when the numer of samples, n, is large enough. The achievable rate provided by the Slepian-Wolf theorem is simply $nH(Y|X)$ by the following identities (the chain rule of joint entropy):

$$H(X,Y) = H(X) + H(Y|X) = H(Y) + H(X|Y),$$

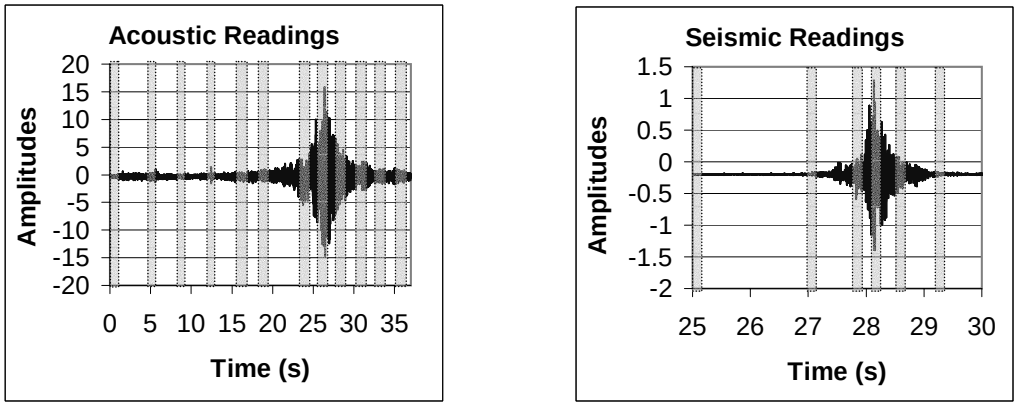
where $H(Y|X)$ and $H(X|Y)$ are the conditional entropy, and $H(X)$ and $H(Y)$ are the entropies of source X and source Y, respectively. Methods on approaching the Slepian-Wolf limit with practical compression complexity have been proposed [30,44]. For the lossy compression case, the Wyner-Ziv rate-distortion function [66] tells that the minimum achievable rate $R^*(d)$ at distortion level d with side information available at the decoder is as follows:

$$R^*(d) = \min_{p \in \mathcal{M}(d)} I(X, Z|Y),$$

where, $\mathcal{M}(d)$ is the set of codecs which meets distortion level d; Z is a transition random variable derived from a codec and $I(.,.)$ is the mutual information function. Furthermore, a transmission rate lower than $nH(Y|X)$ is possible when some distortion is allowed. The Wyner-Ziv rate-distortion function requires that the side information be accurate. In applications of sensor networks, side information facilitating compression may be presented as an approximation to the codec, and a good correlation analysis scheme is certainly necessary for a codec to achieve a lower transmission rate for a given distortion level.

Based on the above discussion, one important issue in the sensor storage and distributed source coding of correlated sources in wireless sensor networks is to estimate the encoder's values from the data available at the decoder. This is critical for many resource constrained sensor networks. Correlation tracking is therefore important in these types of applications, and it directly affects a codec and storage systems performance. With a high performance correlation analysis scheme, spatial correlation can be effectively exploited to its maximum. On the one hand, instead of using a decoder's frame for reference during decoding, the decoder uses a reconstructed frame. A good correlation analysis scheme can reconstruct the encoder's frame with a very low RMS error. This can greatly improve the performance of a codec. On the other hand, a high performance codec can further improve the analysis results due to the fact that more accurate estimates of source frames are available to adjust estimation parameters. This is the reason why the proposed analysis scheme can significantly improve the compression ratio of a codec.

Assuming a dynamic partition structure for a codec, any codeword can be correctly decoded as long as there is a "correct" reference at the decoder within the desired partition. To apply the proposed model to a codec, a coarser partitioning (i.e. larger average coset size) is used during Kalman filter tracking, and a measurement vector from the encoder can be confidently used in the estimation so we can weight more on the measurement. After the Kalman filter update-steps, the normal partition structure is restored and the steepest descent method is used for a continuous tracking update. With this interleaving between an update based on DKF and an update based on the steepest descent method, we are able to have a comparable SNR performance over other schemes.



(a) Acoustic Signal Case

(b) Seismic Signal Case

Figure 4.3: Illustration of Interleaving of Estimation and Tracking.

Figure 4.3 shows two plots of amplitude readings of one acoustic sensor and one seismic sensor over a period of 37 seconds where seismic readings (see Fig. 4.3(b)) are zoomed in. The data in these plots come from the sitex00 data set. Here only a segment of a single vehicle (AAV-O) case is shown to illustrate how estimation and tracking can be interleaved for the trade-off of accuracy and efficiency. In Fig. 4.3, correlation is estimated periodically using the steepest descent method on frames marked by a dotted

rectangle. Note that the reference frame is not shown in this plot. Frames not highlighted in Fig. 4.3 are tracked directly by the proposed DKF method. Each sample is decoded using corresponding tracked samples other than the sample in the reference frame. Figure 4.4 shows the decoding performance with and without correlation analysis. We selected ESPIHT [57] and DISCUS [44] for this comparison, and these schemes using correlation tracking are denoted as ESPIHT-T and DISCUS-T, respectively. When the SNR is set at a practical value of 20 dB, based on three different data sets, the bit rate can be reduced on average by at least 0.5 bits per sample because of correlation analysis. This extra gain is due to the fact that a large coset size can be set so that less bits are required to code coset indices during spatial coding.

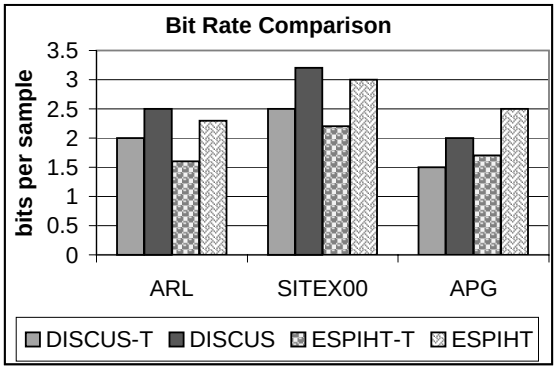


Figure 4.4: Bit Rate Comparisons of Codecs with/out Correlation Analysis.

Storage systems have been critical for many sensor network applications because of the resource constraint in sensor networks. High performance storage systems not only reduce the system memory requirement, but also facilitate network operations and conserve energy. Promising approaches have been reported, e.g. [26,28]. These approaches also exploit the temporal, spatial and multi-modality correlations for efficiency and storage space reduction. The proposed scheme can be used to facilitate online execution of these storage systems.

Summarization is one way to help application level queries. With data summary, coarser information can be extracted for a query, and only if necessary, should further data communications continue. With the proposed scheme, a data collecting node only needs to store $l_0 + m_0 + 1$ LP coefficients for a sequence of frames of another node. Compared to DIMENSIONS [26], where wavelet transform is done on a node closer to the source where source data can be collected easily, the proposed scheme requires much less information to store, very limited source data transmission in the local region and no data transmission on a large scale. The main feature can be recovered by online tracking using the proposed tracking algorithm. If further detail is needed, more data can be first compressed using the spatio-temporal compression algorithm and sent over from another data collecting node which is presumably closer to the source.

The data can also be organized in a hierarchical fashion with multiple layers, where each layer would have a different granularity of information. As we know, the coefficient corresponding to the principal index

conveys the most information about corresponding frames of another node, and further refinement information is contained in a number of auxiliary coefficients. Data can be organized in such a way that nodes closer to the source have the principal coefficient and more auxiliary coefficients. A query could be done in an iterative fashion with finer detail.

4.3 Two-Stage Detection of Events

To facilitate sensor node control in a sensor network, efficient and effective detection algorithms are essential. There are two types of detections, namely, alarm detection and false alarm detection. Alarm detection is usually based on a decision threshold which corresponds to an energy level of potential events. This type of detection is simply called threshold detection here. When a threshold detection algorithm issues an alarm, this could be an event or false alarm due to noise.

Tripwire threshold detection suffers from *False Alarm Problem* (FAP). FAP refers to a situation in which noise strength is large enough to trigger an alarm when no event is actually happening. There are many sources of false alarms in sensor networks and their applications: (1) ambient noise; (2) meaurement of thermal noise; (3) truncation error since a low precision Analog-to-Digital Convertor (ADC) is usually used; (4) scatter and/or reflection of sensing signals; (5) multi-path and multi-source interference. (4) and (5) usually cause false positives. These factors are not negligible in sensor network applications, especially when a sensor deployment is constrained by physical limitations of sensors, e.g. acoustic sensors are very close to the ground, and these limitations could cause severe ground reflection and further degrade algorithmic performance of event identification.

By noticing the fact that a signal with a wide spectrum and with energy evenly distributed among these spectra is less likely to be anything interesting, we can detect false alarms by measuring energy in different bands. Event signals of interest may be superimposed by noises, and this makes it difficult to identify an event. False Alarm Detection (FAD) can be done in a wavelet domain by first decomposing the signal samples using a wavelet transform, and then iteratively checking the energy distribution of subbands to detect false alarms. In Fig. 4.5, the signal energy as shown in dark is concentrated in some frequency bands and the noise energy is spread across the Nyquist frequency spectrum. Therefore, the signal can be identified as an event of interest using the iterative detection by noticing the significant difference between two subbands in iteration I3a (three iterations in Fig. 4.5 as labelled by the thin vertical lines). One of the desired properties of a wavelet based detection is that it makes an iterative detection possible and an iterative detection is advantageous in terms of processing overhead and flexibility.

An alternative approach could be based on a short-time windowed Fourier transform. Some drawbacks of this technique come from the need to select the window function and window length. A Fourier transform based approach can be viewed as a uniform division of the temporal-frequency resolution which makes it hard to devise an iterative algorithm which is flexible in trading-off accuracy with processing overhead. There are other factors which could impair the performance and incur overhead, including energy leakage over adjacent frequency bins during energy estimation, when signals are not perfectly aligned with discrete frequency bins. On the other hand, the wavelet based approach using

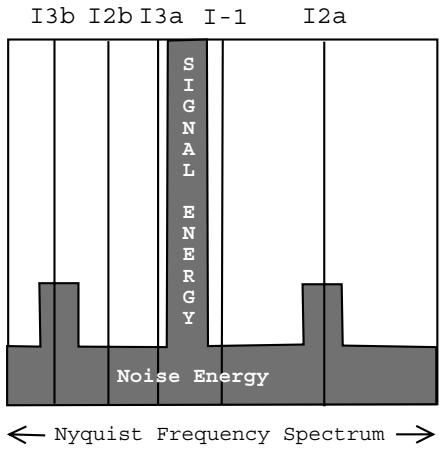


Figure 4.5: Iterative detection of an event of interest using Wavelets

tree decomposition provides a natural way of iterative design on detection. We may not need to do a full subband decomposition (either dyadic or pyramidal) and stop whenever the result is satisfactory. The computational gain of this approach is significant for low power devices. The idea of using wavelet transformation for false alarm detection stems from the feature extraction and signal classification using Wavelets packet decomposition [36]. It has been shown to be promising with efficient processing and accurate results.

In this study, we use the following formula *average squared amplitude* in dB to compute signal energy of a frame of fixed length in time domain for threshold detection:

$$W(k) = 10 \log \left(\frac{1}{n} \sum_{i=0}^{n} \left(x_i^k\right)^2 \right), \tag{4.4}$$

where, k is a frame index; n is the frame length and x_i^k is the i-th sample amplitude of k-th frame. For false alarm detection, the formula for computing a subband energy in wavelet domain is as:

$$W(u) = \sum_{i=0}^{n_u} \left(a_i^u - \overline{a^u}\right)^2, \tag{4.5}$$

where u denotes a subband; n_u is the length of subband u; $\overline{a^u}$ is the average amplitude of subband u; a_i^u denotes the i-th coefficient in wavelet subband u. Notice that the proposed algorithm works on a zero-mean signal. In the subband energy formula, we do not divide the sum by $n_u - 1$ since we are interested in an energy ratio instead of subband energy (see (4.6)).

Threshold detection is done on a frame by frame basis. During the k-th frame period, a tripwire computes a frame energy using (4.4) to the end of this frame. It then compares the energy with a predefined threshold T_f. If $W(k) \leq T_f$, the tripwire continues to monitor the field. Otherwise, it enters into the false alarm detection stage.

In order to explain the technique for false alarm detection, let u be a non-leaf node in a pre-assigned

wavelet subtree for a tripwire, and let us also denote two corresponding high-pass and low-pass subband transformed from u as u_2 and u_1, respectively. We define the energy ratio as follows:

$$\mathcal{R}(u) = \frac{\left|\gamma_1 W(u_1) - \gamma_2 W(u_2)\right|}{\left|\gamma_1 W(u_1) + \gamma_2 W(u_2)\right|}, \tag{4.6}$$

where $\gamma_1 W(u_1)$ and $\gamma_2 W(u_2)$ are the energy for low-pass and high-pass subbands, respectively, of node u; γ_1 and γ_2 are the normalization factors for a given transform for low-pass and high-pass subbands, respectively. When a signal has a mean of zero, by the Parseval theorem, the denominator in eq. (4.6) actually equals the energy of subband u. By Parseval equality, for the orthogonal basis, as that is the case for S-Transform with Haar basis, the following holds:

$$W(u) = \gamma_1 W(u_1) + \gamma_2 W(u_2). \tag{4.7}$$

The normalization factors only
depend on the basis and how the transform is actually performed (for S-Transform, γ_1 is $\sqrt{2}$ and γ_2 is $0.5\sqrt{2}$).

The false alarm detection processing uses a greedy search method on a wavelet decomposition tree. At each node of level i of wavelet decomposition (the decomposition indices start from 1), it first computes the energy ratio for a node using (4.5), and it then selects the node with a larger energy ratio to process in the next level. The detection processing iterates until either (1) a search reaches a predefined lowest level or (2) one node whose energy ratio is greater than threshold T_s is found. For case (1), a claim of false alarm denoted by H_a is generated, and for case (2), a claim of event denoted by H_0 is generated. Figure 4.6 shows an example of case (2) where nodes in black are processed and it stops at the black node of the third level of the subtree. In Fig. 4.6, many nodes (in gray) are not processed since

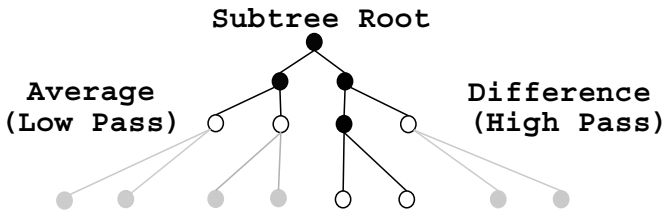


Figure 4.6: Iteration Path of an H_0 Predicate

its energy ratio is smaller than the other child of the same parent. Nodes in circle are evaluated of the subband energy, but no energy ratio is computed. In the greedy search, it looks ahead by one more level in order to determine the greater energy ratio. For a case of (1), the search stops at one of the leaves. The memory required in this processing is not larger than half of original frame length, and memory required in the next iteration is reduced by half.

For the above algorithm, the false alarm rate and the false positive rate can be estimated using Bayes formula. For convenience of notation, we use H_0 to denote a detection predicate of an event and H_a to denote a

detection predicate of a false alarm. There are four combinations of real cases and detection predicates:

(1) $C_0 : H_0 \longleftrightarrow H_0$,

(2) $C_1 : H_0 \longleftrightarrow H_a$,

(3) $C_2 : H_a \longleftrightarrow H_0$,

(4) $C_3 : H_a \longleftrightarrow H_a$,

where $Z : X \longleftrightarrow Y$ means the case Z outputs claim X while the truth is Y. C_0 and C_3 are correct while C_1 and C_2 (false positive) are false.

We can compute the error probability of type C_1 as follows,

$$P(C_1) = \sum_{path} \prod_{u} P(H_a \big| \mathcal{D}_u < \varepsilon_1), \tag{4.8}$$

where the summation is over all paths of the binary tree of the pyramidal decomposition of a frame. Similarly, we can compute the error probability of type C_2 as follows,

$$P(C_2) = \prod_{u} P(H_0 \big| \mathcal{D}_u > \varepsilon_2). \tag{4.9}$$

Since each $P(H_a \big| \mathcal{D}_u < \varepsilon_1)$ normally is much less than 0.5, $P(C_1)$ in eq. 4.8 strictly decreases as the number of iterations increases. The same holds for $P(C_2)$ in eq. 4.9. By Bayes rule, we have

$$P(H_a \big| \mathcal{D}_u < \varepsilon_1) = \frac{P(\mathcal{D}_u < \varepsilon_1 \big| H_a) P(H_a)}{P(\mathcal{D}_u < \varepsilon_1)}, \tag{4.10}$$

and

$$P(H_0 \big| \mathcal{D}_u > \varepsilon_2) = \frac{P(\mathcal{D}_u > \varepsilon_2 \big| H_0) P(H_0)}{P(\mathcal{D}_u > \varepsilon_2)}. \tag{4.11}$$

The estimates of $P(\mathcal{D}_u < \varepsilon_1 \big| H_a)$ in eq. (4.10) and $P(\mathcal{D}_u > \varepsilon_2 \big| H_0)$ in eq. (4.11) can be taken as $1 - \beta_1(\varepsilon_1 - \mathcal{D}_u)$ and $1 - \beta_2(\mathcal{D}_u - \varepsilon_2)$, respectively, where β_1 and β_2 can be pre-estimated offline based on training datasets, so do ε_1 and ε_2. These probabilities determine the confidence probabilities of detections.

4.4 Distributed Tripwire Cueing

Once an event is detected, many signal processing nodes (i.e. trackers) are eager to wake up to process it. There is a so-called exposed alarm problem with tripwires. In this project, we developed a protocol so that tripwires can cue trackers when an event is detected by tripwires. In this study, maximization of tracker sleep time, therefore, energy savings on trackers, is the main goal.

A high-level description of the proposed Multi-Hop Cueing (MHC) protocol is as follows. It uses three different types of beacons, namely: T-WAKEUP, T-QUENCH and T-ALARM. T-WAKEUP is sent

Table 4.2: Characteristics of Tripwire and Tracker Radios

	bitrate	typical freq.	power	range
Beacon Radio	< 1 kbps	900 MHz	< 1 mW	~ 10 m
Data Radio	> 5 kbps	2.4 GHz	> 100 mW	> 100 m

Table 4.3: State Machine of Tripwires

mW	MCU	RADIO	SENSOR	ADC
TM	$\sim$	$\sim$	on	$\sim$
TE	on	off	on	off
TL	off	on	off	off

by a tripwire to wake up a tracker; T-QUENCH is sent by trackers to stop tripwires from sending T-WAKEUP and notifies tripwires that at least one tracker has been woken up; T-ALARM is sent by trackers to notify near-by tripwires to resume its monitoring task. In order to investigate this cueing problem, we have defined a suitable radio model and waking-up channel.

A tracker can be woken up by a wake-up beacon. A tracker also has a tripwire unit built-in, i.e. it turns on when its associated tracker is asleep. There are two radios in a tracker where one long-haul radio, called a data radio, is for data communication of task execution and one tripwire radio, called a beacon radio for beacon communication. The tripwire radio in a tracker uses a channel to communicate with independent channels, and its long-haul radio uses a separate channel for data communication. The tripwire network forms a so called "wake-up" channel for trackers. The proposed protocol uses this wake-up channel. In this study, all tripwire radios and long-haul radios have fixed transmission power. However, this assumption is only for clarifying the explanation, and the proposed scheme does not depend on this. Table 4.2 summarizes the characteristics of tripwire and tracker radios. Tracker waking-up is performed as follows: a near-by tripwire wakes up the tripwire unit of a tracker, and this tripwire unit then wakes up other components of the tracker in which it resides. However, the wake-up initiating tripwire may be some hops away from a tracker. With multi-hop tripwire cueing, at least one tracker is woken up. There are two undesirable cases: (1) more than the required number of trackers for a task execution are woken by tripwires and (2) less than the required number of trackers are woken. Since T-WAKEUP floods, Case (2) cannot happen unless some deployment problem causes an insufficient number of trackers in a certain region of a field which is not considered in this proposal. Trackers which are awake will collaboratively decide how to put excessive trackers back to sleep in Case (1). It is up to the application level protocols to handle how to select an optimal set of trackers for a task and what optimization criterion should be used. However, these are beyond the scope of this project.

A tripwire has three operational states which corresponds to three machine states, namely: monitoring state (TM), listening state (TL) and execution state (TE). In a TM state, the tripwire radio and MCU are active periodically and its sensor is always on. In a TL state, the tripwire's MCU and its sensor, including the Analog-to-Digital Convertor (ADC), are turned off while its radio is kept on and only the receiver is actually used. In a TE state, the tripwire's MCU and radio are kept on while its sensor and ADC are turned off. TE is a transit state. Table 4.3 shows component states in different tripwire states. In Tab. 4.3, the symbol $\sim$denotes

Table 4.4: State-Beacon Table

	T-WAKEUP	T-QUENCH	T-ALARM
TM	FORWARD	×	×
TE	×	FORWARD	×
TL	T-QUENCH	×	FORWARD

a component that is turned on periodically.

Once an alarm is confirmed by a false alarm detection scheme, this tripwire issues a wake-up beacon called *T-WAKEUP* and it goes to a TL state. A multi-hop cueing algorithm is used to wake up trackers. The tripwires switch states in the proposed protocol as follows, and the combinations of beacons and states other than the following cases are simply discarded.

1. In TM state, on receiving a T-WAKEUP beacon or having detected an event (alarm), this tripwire sends or forwards a T-WAKEUP beacon and goes into TE state.

2. In TE state, on receiving a T-QUENCH beacon, it forwards this beacon and switches to TL state; on receiving a T-WAKEUP beacon, the tripwire forwards it.

3. In TL state, on receiving a T-ALARM beacon, it switches to TM state.

4. In TL state, on receiving a T-WAKEUP beacon, it replies by a T-QUENCH beacon.

5. On timeout in TE state, a tripwire goes to TM state.

6. On timeout in TL state, a tripwire goes to TM state.

Notice that the timeout value of a TE state is shorter than that of a TL state. A new try can be initiated when no T-QUENCH beacon is received in the TE timeout period. The TL state timeout is to restore a tripwire to a TM state when a T-ALARM cannot reach certain tripwires. Item 4 above is used to put into a TL state these tripwires which newly detected a target in the same monitoring region. Note that only tripwires in a TE state respond to T-QUENCH beacons.

T-QUENCH is used to put tripwires into a TL state when an event is detected. T-ALARM is used to turn on tripwires for monitoring when an event is out of the detection region of a tracker. To deal with the intrinsic unreliability of tripwires, the low quality of its radio link and possible beacon collisions, the timeout mechanism is used to restore tripwire states when a target is out of the detection region or to initiate a new try to wake up trackers if a target is still in a detection range. Table 4.4 summarizes the tripwire responses to these three beacons where × represents no reponse to a beacon at that state. Figure 4.7 summarizes the state transitions of a tripwire.

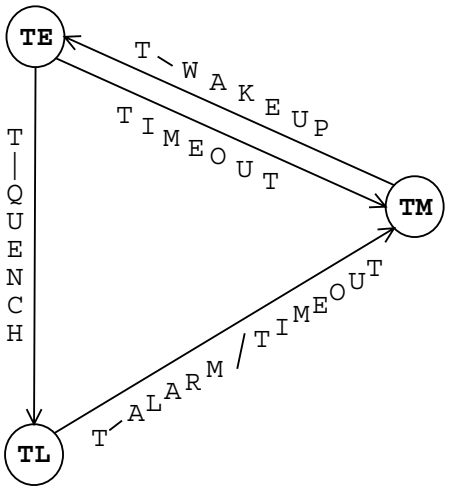


Figure 4.7: State Diagram of Tripwire

5 Results and Discussion

Microcontrollers (μC) are commonly found in sensor nodes [11,7] and they are limited in resources. The PASTA [11] sensor node uses several microcontrollers for various functionalities including function of a tripwire. CYGNAL C8051 F020 has only 4352 bytes of on-chip memory and up to 64K off-chip RAM, and the access to off-chip memory is 7 times slower than on-chip memory. It supports internal and external oscillators with a programmable clock rate. In our experimental setup (see Fig. 5.1 for the experiment setting), we used an internal oscillator with system clock rate set at 16 MHz. Two 32-bit integer memory was required during the energy ratio comparison; one for the current ratio, which is temporary and overwritten each iteration, the other is used to store the largest ratio. The memory requirement was kept low so that off-chip memory could be totally eliminated. With this, an on-chip oscillator could be used and it allowed the sensor node to be in deep-sleep mode while the tripwire was running. The whole detection could be done within 10 ms. The extra processing energy cost was kept around 10 mJ per detection.

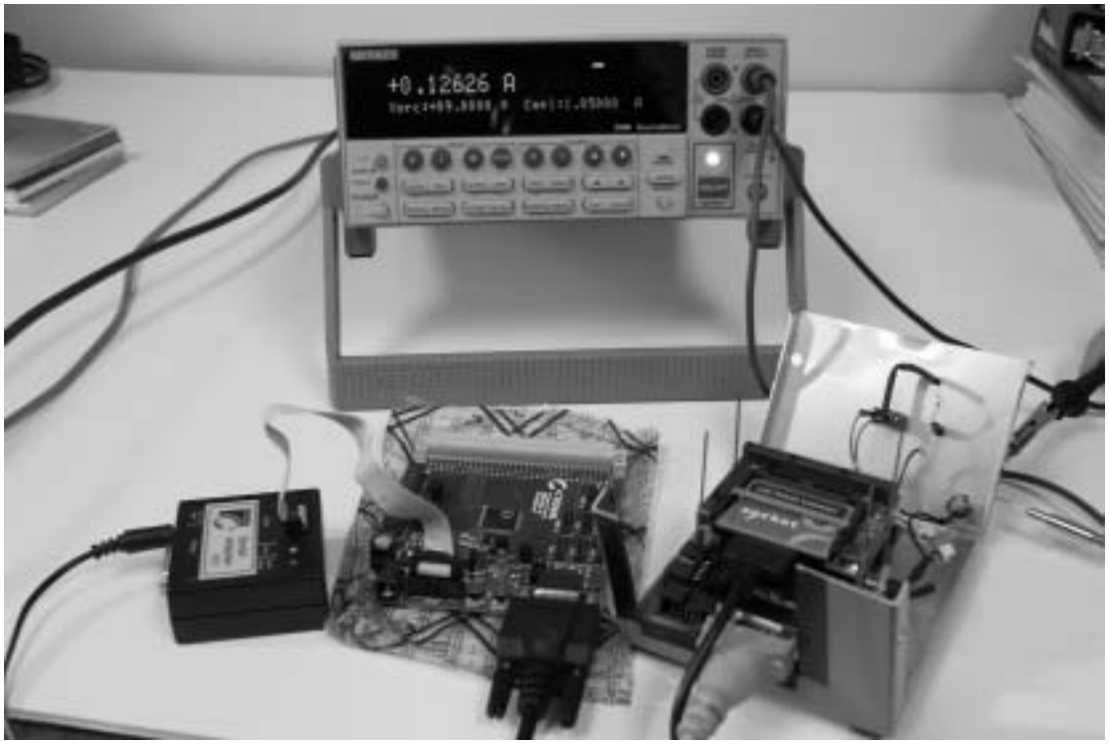


Figure 5.1: Experimental Setup with MCU Test Board and PASTA Node

This μC was integrated in to the PASTA node [11]. We were using the AP HILL data set (Figure 5.2 shows that experiment configuration) from field experiments to verify the distributed detection algorithms as well as our field calibration scheme. These experiments tested our implementation of the proposed detection algorithm and field calibration of the detection scheme, as well as the tripwire cueing protocol.

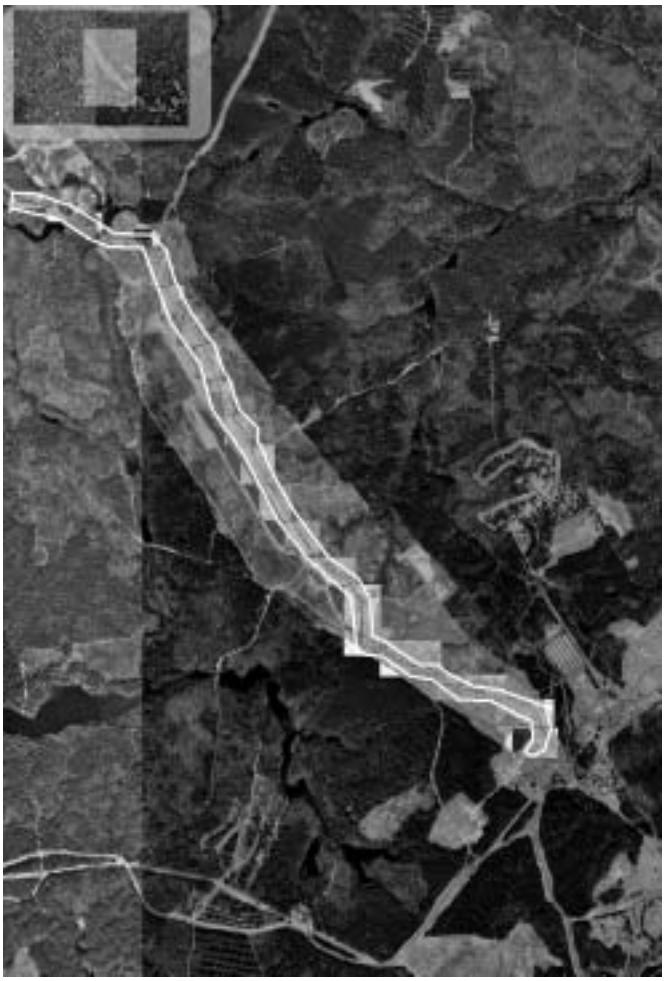


Figure 5.2: Network Configuration in APHILL Field Experiment

In this study of coding/compression performance, we used ESPIHT and SPIHT (their LINUX implementations are presented in [57]). *In these experiments, we only measured power dissipation due to an RF component of the transceiver. Other circuitry power dissipation of the radio was excluded.* In simulations using OPNET, bit error thresholds of convolutional codes with different puncturing patterns were obtained using the estimation method of the previous section. The default Error Correction Code (ECC) model in the OPNET radio pipeline was replaced by our ECC procedure based on RCPC code with corresponding ECC bit error thresholds. In order to output energy dissipation data on each frame from OPNET, we inserted a new pipeline stage procedure in the radio simulation pipeline. Therefore, radio standby energy was not included when radios were not active. Input data to the OPNET simulator was output from ESPIHT or SPIHT, and there were three data chunks ,while each chunk corresponded to one of $\mathcal{U}_h, \mathcal{U}_m, \mathcal{U}_l$. A new process module was developed to read an input file and feed the source size information to a radio transmitter. The first OPNET simulation had one mobile receiver and one stationary transmitter. There were three segments on the path trajectory of the mobile node. Simulation parameters are shown in Table

Table 5.1: Simulation Parameter Settings

Channel Parameter	Parameter Value
network dimension	1km x 1km x 10 m
simulation duration	148 seconds
modulation scheme	bpsk
radio rx sensitivity	-111 dBm
data rate	19.2 kbps
base frequency	900 mHz
bandwidth	7.2 mHz
maximum Tx power	10 dBm
number of RCPC codes	10
total number of power levels	10
Tx/Rx antenna gain	0 dBi
radio closure	ray-tracing
propagation model	free space
noise	accumulating

5.1. The simulation duration was determined by the data set of length 148 seconds. Although we present the simulation results based on one data set, we did test two other data sets (SensIT SITEX00 and ARL APG). The results were similar to what is presented here.

In these experiments, convolutional code with rate $\frac{2}{3}$ and constraint length of 6 was used for both $\mathcal{U}_h$ packets and $\mathcal{U}_m$ packets, and RCPC code with rate $\frac{3}{4}$ was used for $\mathcal{U}_l$ packets. The left plot of Fig. 5.3 shows the energy dissipation using jointly coded ESPIHT, and jointly coded SPIHT vs. the baseline. Compared to baseline energy dissipation, average energy savings were about 80% and 65% by jointly coded ESPIHT and jointly coded SPIHT, respectively. The right plot of Fig. 5.3 shows the energy savings using jointly coded ESPIHT and standard ESPIHT, while the standard ESPIHT is uniformly coded by either $\frac{1}{2}$, $\frac{2}{3}$, or $\frac{3}{4}$ RCPC codes. Average energy savings due to power aware coding was at least 20%. Large variation on energy savings was due to signal amplitude fluctuation among frames. In our implementation, we selected a Coiflet basis with 3 vanishing moments. However, other types of bases could be used.

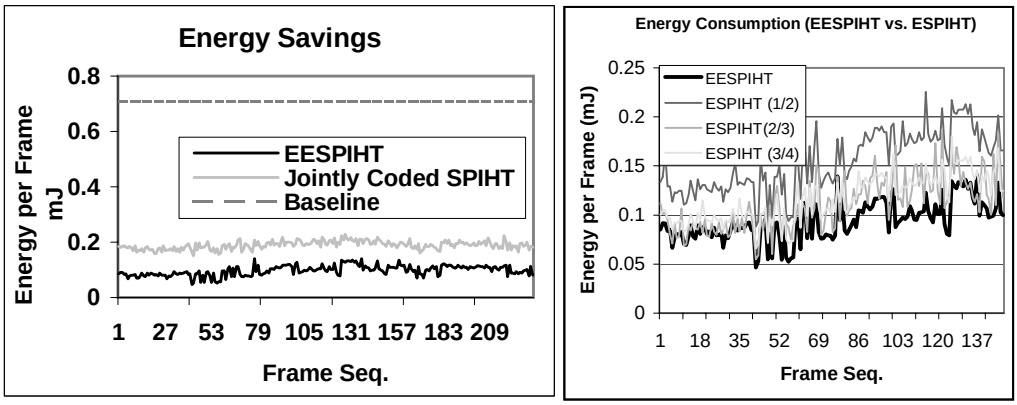


Figure 5.3: Energy Dissipation Comparisons

The left plot of Fig. 5.4 shows SNR loss of jointly coded ESPIHT as compared to ESPIHT. In this plot, SNR differences among frames are shown at two bitrates: 4 iterations and 10 iterations. The 4-iteration case gives about 25 dB SNR gain and 10-iteration case gives about 50 dB SNR gain on average among these frames. (The corresponding SNR gain of ESPIHT used in this comparison is shown in the right plot of Fig. 5.4.) The i-th iteration corresponds to a tree traversal of MSB with $k - \ i - 1$ bits, where k is the number of bits of the largest coefficient in the binary tree (it is 13 for the data set used in the experimental study). Worse case, jointly coded ESPIHT of the 4-iteration case has 0.58 dB more SNR loss than that of ESPIHT over all frames. Jointly coded ESPIHT of the 10-iteration case has 2.4 dB more SNR loss than that of ESPIHT over all frames . (This loss is within the bound as that shown in Theorem 1.)

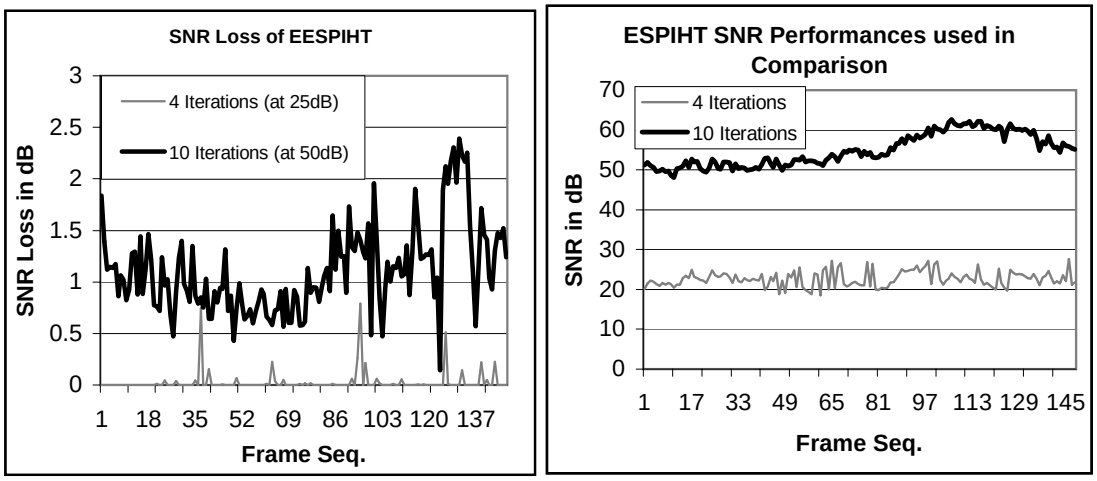


Figure 5.4: SNR Comparisons

The SNR loss of the SPIHT vs. jointly coded SPIHT, with the same experimental configuration as those in the left plot of Fig. 5.4, has also been obtained. The worse case SNR loss of jointly coded SPIHT is 0.34 dB and 1.9 dB for the cases of 4-iteration and 10-iteration, respectively.

For the next set of simulations, we used OPNET to simulate the network and verify how jointly coded ESPIHT based design can save energy using rate-adaptable power allocation. Refer to [59] for the radio power profile used in these simulations. The simulation is based on pairs of nodes with different distances. The parameters are also shown in Table 5.1. Assume that the distances are all inside the propagation limit and also assume that every node has 2Kx8 bits of information to send in the baseline every second. The network consists of a number of uniform nodes randomly distributed in a field of size 1km x 1km, while five of them are mobile nodes. Fig. 5.5 shows the normalized powers in jointly coded ESPIHT compared to the best fixed power case, and a scheme using distance with a given a SNR constraint with a varying number of nodes. From Fig. 5.5, when the fixed or distance/SNR based power allocation is used, the powers increase as the number of nodes increase, while the power is almost constant when more nodes are added to the network under jointly coded ESPIHT. This is simply due to the lower power and less bits under jointly coded ESPIHT, which significantly reduces the interference.

The energy dissipation comparison with the 10-node case for a simulation duration of 60 seconds is

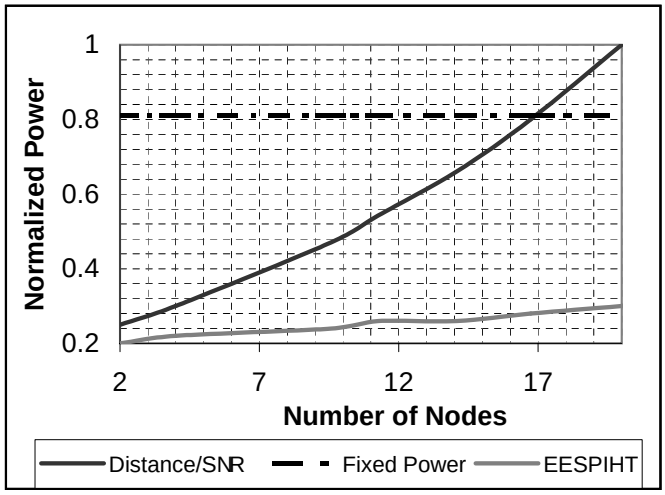


Figure 5.5: Power Dissipation Comparison

Table 5.2: Energy Dissipation Comparison

	Ave. Power	E/Bit	Tot. Eng.
jointly coded ESPIHT	2.3 mW	0.12 μJ	370 mJ
Distance (SNR)	3.4 mW	0.18 μJ	1.53 J
Fixed Power	3.9 mW	0.21 μJ	1.95 J

shown in Table 5.2. From Table 5.2, it can be seen that on average, jointly coded ESPIHT saves about 20 mJ per second network-wide. Note that the whole network (radio RF) without using jointly coded ESPIHT dissipates at least 25 mJ per second (excluding energy consumed by individual nodes) under the distance-SNR based allocation scheme. Jointly coded ESPIHT has the least energy per bit cost among these three schemes.

In what follows, we shall compare the performances of three correlating schemes, namely, our proposed estimation and tracking scheme denoted by LP/DKF, the scheme used in [20] denoted by LP/SD and the least squared estimation (using cross correlation coefficients) denoted by LSE. For the comparison, we take into account four aspects: (1) tracking accuracy; (2) required communication; (3) processing overhead of the algorithms; (4) total energy cost which combines both processing and communication costs.

The experimental setup is shown in Table 5.3. In this experiment, communication between two nodes is on the average of one packet every five seconds, and node interference can be totally eliminated via proper scheduling. Therefore, for simplicity, the communication energy cost estimate does not consider the affects of the network density and communication interference. Our experiment is directly based on the C8051 microcontroller with its integrated development environment. It does not need any operating system support. The program is directly loaded to a designated memory area and it starts to execute once the node is turned on.

Table 5.4 shows comparisons with an experimental run of 235 seconds from one of a few different

Table 5.3: Experiment Setting

Processing unit [6]	CYGNAL C8051 F020
On-Chip Memory	4096 + 256 Bytes
Clock Rate	16 MHz
Radio type [8]	Chipcon CC1000
Communication energy cost	3.2 μ J per bit

Table 5.4: Performance and Cost Comparisons (Per Frame)

	Mem. (Bytes)	Error (dB)	Comm. (bits)	Time (ms)	Energy (μJ)
LP/SD	128	-10.4	0.9 K	120	470
LP/DKF	128	-11.9	0.3 K	230	330
LSE	144	-11.5	3.2 K	150	830
LSE/DKF	144	-11.8	0.8 K	250	520

data sets. In Table 5.4, the memory is based on the worst-case RAM requirement. In the cases of LP/SD and LP/DKF, the worst-case RAM requirements are dominated by linear prediction, and those of LSE and LSE/DKF are dominated by linear estimation. The error in Table 5.4 is taken as the average estimate error in RMS divided by the average signal RMS (as the signal power), and it is computed in logarithm form (dB). The error in RMS is computed using the innovations with reference to real coefficients over 235 seconds. The low estimation error of LSE, compared to LP/SD, is not due to its algorithmic superiority, but the high volume of raw data transmitted. The processing overhead is measured in the average worst-case time over all frames. Table 5.4 shows that the proposed approach, when combined with SD or LSE, can reduce communication significantly without loss of accuracy.

Figure 5.6 shows the average prediction error in RMS and the signal power of the sender, where the estimation and tracking update is done in 1/5 uniform form. The analysis step size has a length of 6 frames, with one frame length of samples estimated by SD and 5 frames length of samples updated using tracking; i.e. roughly 17% of the samples need to be sent. The other approaches use different analysis step sizes with varying number of tracking frames so that a close estimation error can be obtained for comparison purposes. Figure 5.7 shows the corresponding gradients as the input controls to the proposed tracking algorithm, and these vectors show some slight temporal correlation.

The per-frame energy cost of the compared schemes is shown in Table 5.4. The proposed scheme has a lower energy consumption, which includes both communication energy cost and processing energy cost.

Figure 5.8 shows the worst case estimation error vs. different analysis step sizes for different data sets of two modalities with an LSE based baseline. Two different data sets are used in this comparison. One is the far-field sitex00 acoustic and seismic data, and the other is near-field ARL acoustic data. There is clearly a trade-off between the frequency of analysis and fidelity. (Low residual power means high fidelity.) In Fig. 5.8, the tracking update interval is determined based on signal energy variation in the adaptive method, and a relatively short interval is used when the signal fluctuates. A longer interval is used when the signal energy stablizes around a level. The performance of the near-field acoustic data set has

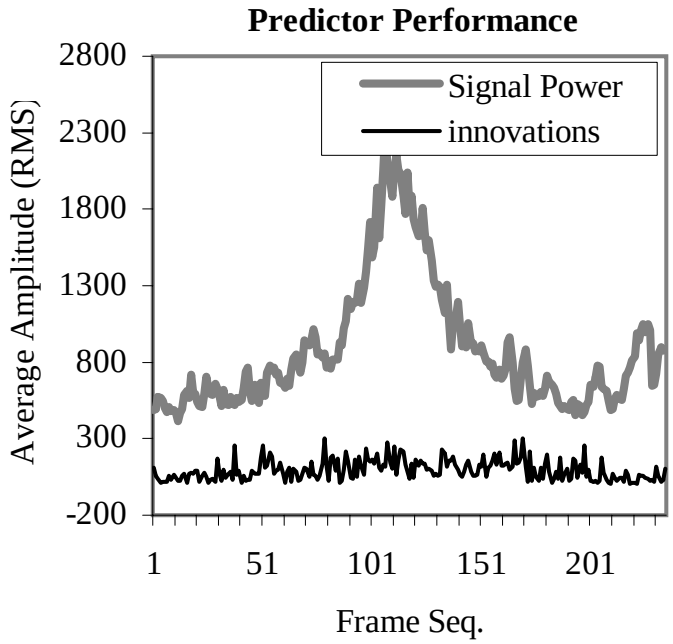


Figure 5.6: Tracking Performance

Table 5.5: Power Profiles of Tracker and Tripwire

mW	CPU	RADIO	MCU/RADIO	SENSOR
IDLE	185	126	18	n/a
ACTIVE	635	171	30	0.9
SLEEP	67	38	0.1	n/a

the best performance. Without this analysis, it would have been almost impossible to exploit the spatial correlation of sensor readings in a far-field for any good use. In this energy/power comparison study for a two-tier heterogeneous sensor network with detection, a PASTA [11] tracker and PASTA tripwires are used. Their power profiles are shown in Table 5.5, where power consumption data of the PXA255 CPU and the PASTA radio are itemized since they are not active at the same time. Tripwires are dissected into three parts - MCU, tripwire beacon radio, and ADC, where MCU and the radio are not always active. ADC on a tripwire or a tracker is active as long as this tracker or tripwire is on. The application used in this experiment is a time-domain Line-Of-Bearing (LOB) beamforming [48]. Energy per LOB computation including transmission is 245 mJ, and the duty-cycle energy per detection is 1.5 mJ. Tripwires are turned on periodically, where a tripwire radio is on during the first 100 ms of every second and off during the remaining period of a second. However, tripwire synchronization cost is omitted since it is required very infrequently. Energy dissipation comparison is shown in Fig. 5.9. Figure 5.10 shows the power consumptions of a tripwire and tracker. From Fig. 5.10, with cueing protocol and detection at 80% false alarm rate and 1% false positive rate, the duty-cycle power consumption can be brought down within a 10 mW regime. We shall further quantify energy savings of the proposed protocol.

One scenario studied in this section is shown in Fig. 5.11 where two targets are present in a field.

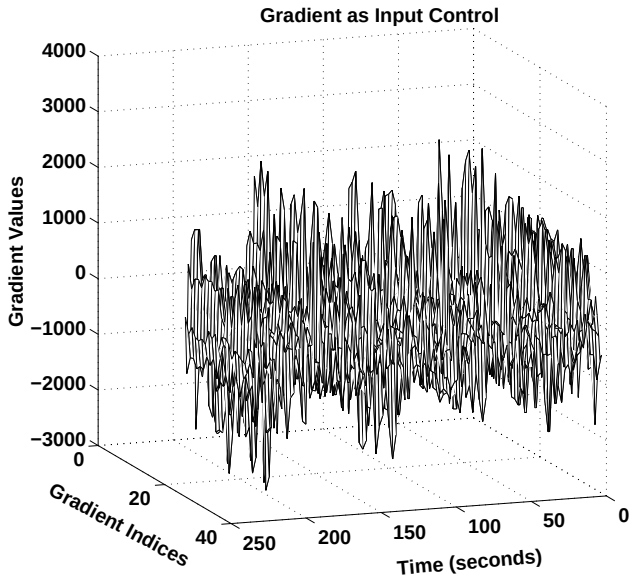


Figure 5.7: Kalman Filter Input Controls

Positions of trackers and trajectories of targets are from field measurements of target GPS ground truth data. Simulation parameters are defined in Table 5.6. The left plot of Fig. 5.12 shows the power consumption of all tripwires and the total power consumption of both tripwires and trackers for a period of 90 seconds. The corresponding energy dissipations are shown in the right plot of Fig. 5.12, where the energy ordinate is measured and plotted in logarithmic scale. From these plots, it can be seen that overall tripwire energy dissipation is comparably negligible, and the savings in trackers under the proposed scheme is significant as to be shown next.

Assumptions of these three schemes to be compared are defined in Table 5.7, where "Tr" is for tracker; "T/W" is for tripwire; "L/H" for long haul data radio on tracker and "Snr" is for the sensors on tracker. "W(S)" is for wake-up by tracker detection, and "W(T)" is for wake-up by tripwire detection. In the duty-cycle scheme, the tracker's sensors have to be always kept on for monitoring events and so does its radio for possible wake-up commands from other trackers. To compute the energy dissipation, a timer is used to check the power of every sensor (tracker and tripwires) every millisecond and add the joules to the total energy. This should be a very close approximation to the real energy dissipation.

The left plot of Fig. 5.13 shows the power consumption for a 90-second simulation with one target. There are three curves corresponding to these three schemes. Power consumption by MHC is significantly less than those of the baseline scheme and the duty-cycle scheme. The right plot of Fig. 5.13 shows the energy dissipation for this 90-second simulation. The energy savings of MHC is more than 50% when compared to the duty-cycle scheme. MHC saves energy because it keeps trackers awake only for the period of task tracking.

The left plot of Fig. 5.14 shows the power consumption for a 90-second simulation with two simultaneous targets. The power consumption of MHC is still significantly smaller than those of the baseline

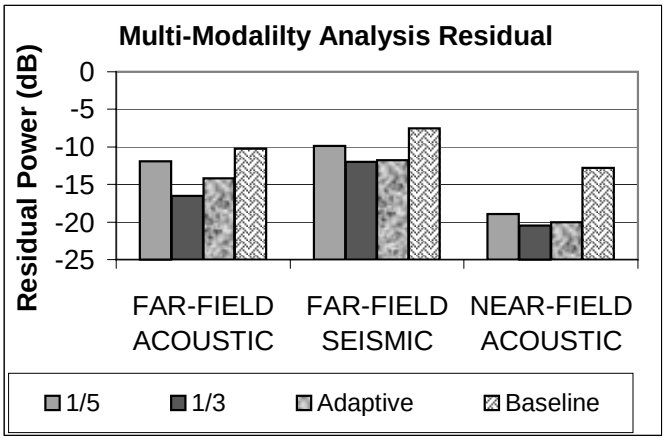


Figure 5.8: Fidelity Effects of Analysis Steps

Table 5.6: Field Simulation Parameters

Number of Tripwires	25
Number of Trackers	18
Field Dimension	$8012\text{x}4015\text{m}^2$
Tripwire Sensor Radius	10 m
Tripwire Radio Radius	10 m
Tracker Sensor Radius	15 m
Tracker Radio Radius	100 m
Tracker Radio Bitrate	1 kbps
TL Timeout	1 second
Fusion Timeout	100 milliseconds
Predication Timeout	50 milliseconds
Beacon Length	8 bits
Wake-up Channel Bit Error Rate	0.01
Beacon Error Probability	2×10^{-3}
Tracker Duty Cycle Rate	50%

scheme and the duty-cycle scheme. Energy dissipation for this comparison is shown in the right plot of Fig. 5.14. When compared to the single target case, the savings are reduced somewhat. In general, when many targets are in a field, the savings may not continue since most of the trackers may be awake most of the time. However, in practical applications, events happen infrequently. Therefore, the proposed scheme gives significant energy savings.

Since different applications require different granularities on sensor data accuracy and processing results, e.g. detection or tracking, tracker density may be different to meet this requirement. In general, it is expected that MHC can obtain a better energy efficiency with a higher density of trackers while the average number of tripwires per tracker is kept constant and the assumption of connectivity also holds. Figure 5.15 shows the average field energy dissipation measured in terms of one fixed measurement area with a varying number of trackers. The curves in Fig. 5.15 conform to the above claim on tracker density when the curve for 27 trackers has the least average energy dissipation.

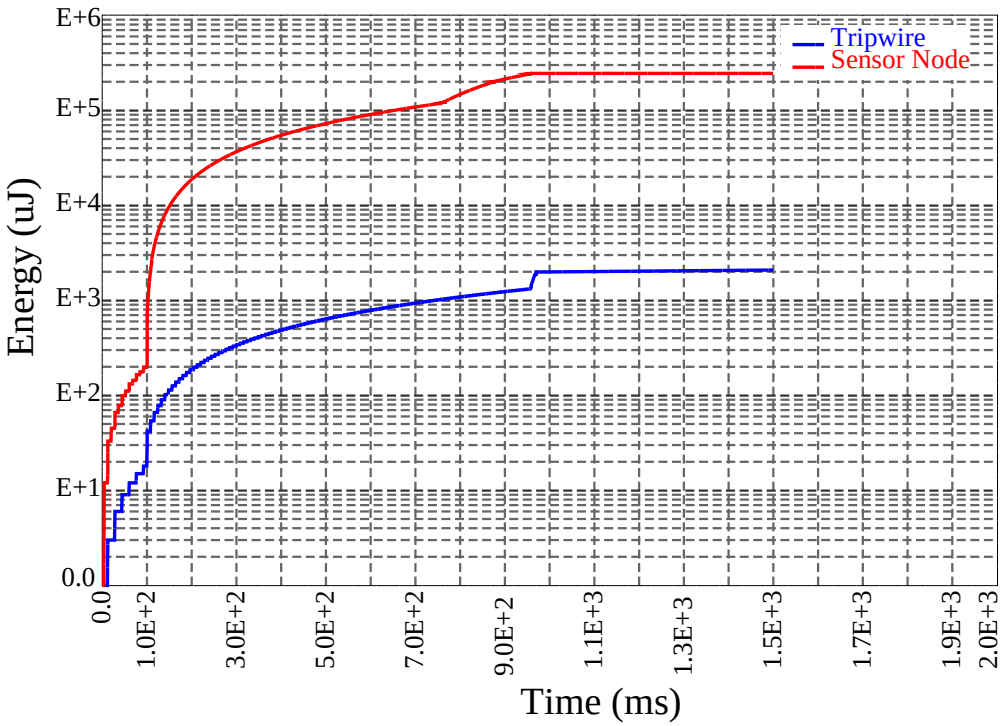


Figure 5.9: Energy Comparison of Detection and Tracking

Table 5.7: Key Scheme Assumptions

	Tr	Snr	L/H	T/W
Baseline	ON	ON	ON	n/a
Duty/C	W(S)	ON	∼	n/a
MHC	W(T)	W(T)	W(T)	ON

Notification latency may be a concern when the MHC scheme is used instead of direct detection as that in the baseline scheme. Another simulation on cueing has been conducted using an NS-2 simulator [1]. Parameters used in the simulation are shown in Table 5.8. Ground two-ray module is selected for the short-path loss modeling of radio communication since tripwires' radios are close to the ground and ground effects are non-negligible.

Figure 5.16 shows the average alarm notification for 100 random events to four trackers with different tripwire densities and bit error rates. In this set of simulations, tripwires are optimally deployed for the 25-tripwire case while tripwires are randomly deployed in the other cases. As more tripwires are deployed, robustness increases. However, in general, latency increases due to collision and beacon error. Figure 5.17 shows the maximum latency of notification to these four trackers. Due to the beacon design with error resilience, the effects of channel BER on notification latency is rather limited by noticing that the curves are rather flat along the BER axis, and the maximal latency can be controlled under 200 ms. This latency can be further controlled based on application requirements by adjusting the TE timer, increasing the minimum distance of codes and/or decreasing the tripwire-per-tracker ratio.

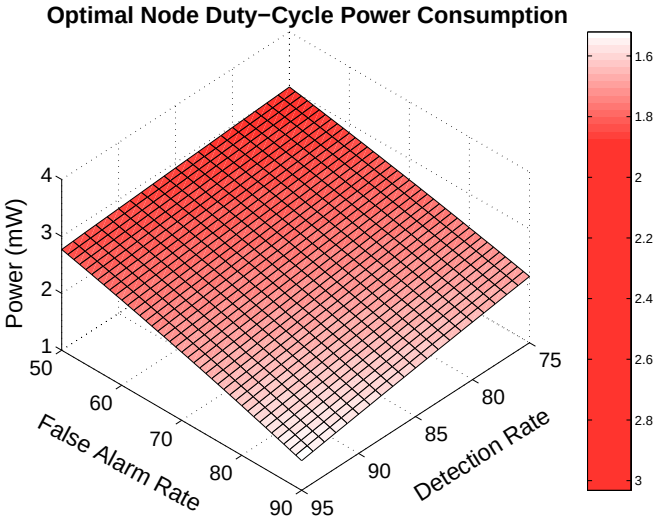


Figure 5.10: Duty-Cycle Power Consumption of Detection

Table 5.8: Cueing Simulation Parameters

Number of Trackers	4
Field Dimension	1000x1000m^2
Radio Radius	200 m
Radio Pathloss Model	Two-Ray Ground
Radio Bitrate	1 kbps
Beacon Delay	10 ms
Beacon Length	8 bits
Wake-up Channel BER	0.01
Beacon Code	$(8, 4, 4)$ block code

We have conducted a set of field tests using acoustic signals in evaluating our detection scheme. One configuration in these tests uses three tripwires, and results from two of these tripwires are shown. Figure 5.18 shows signal energy plots from two of these tripwires (left plot for tripwire 2 and right plot for tripwire 3). Decision statistics are shown in the corresponding two plots in Fig. 5.19. Note that the absolute values are used in detection, not the actual mean value since distance is used in the detection decision. Final detection results from these two tripwires are shown in Fig. 5.20, where value 0 is for no alarm (i.e. failed energy alarm test), 1 is for false alarm and 2 is for event. From both plots, the detection rate is more than 90% and false positive rate is less than 1%.

Figure 5.11: A Two-Target Scenario

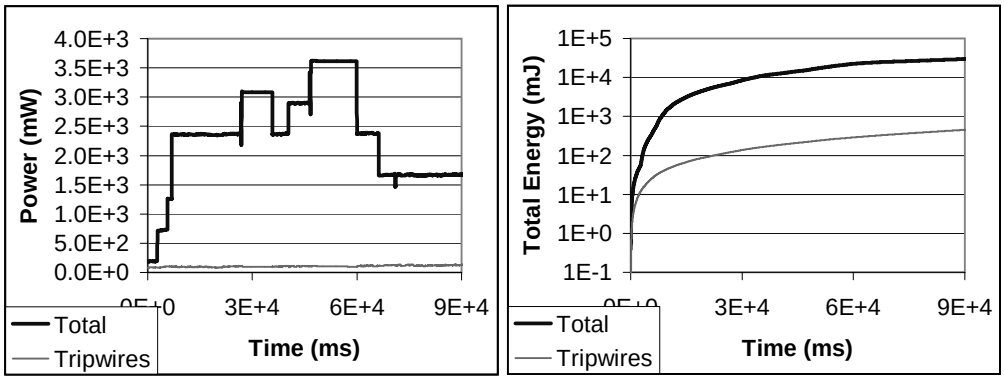


Figure 5.12: Field Power and Energy Comparison

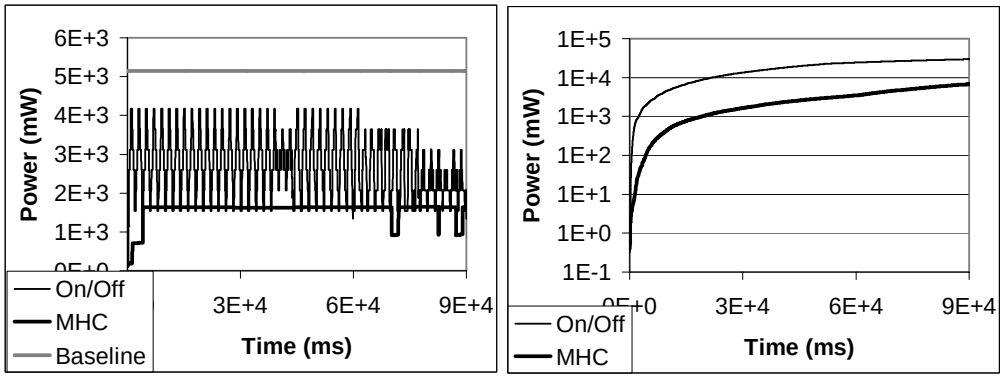


Figure 5.13: Power and Energy Comparison (1 target case)

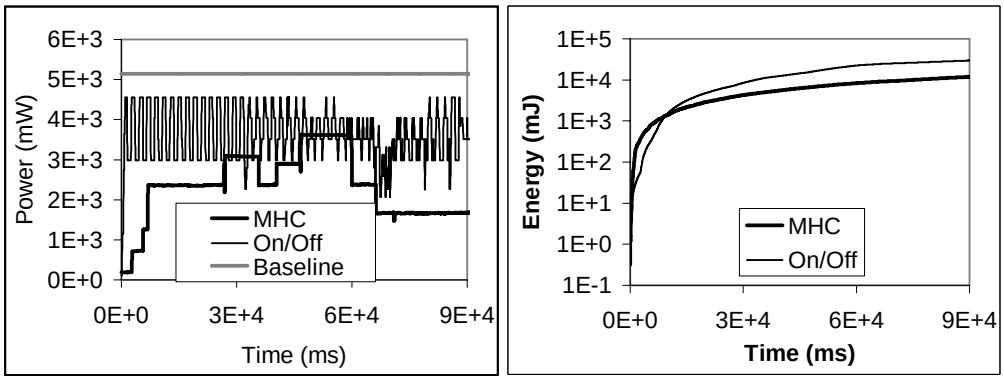


Figure 5.14: Power and Energy Comparison (2 target case)

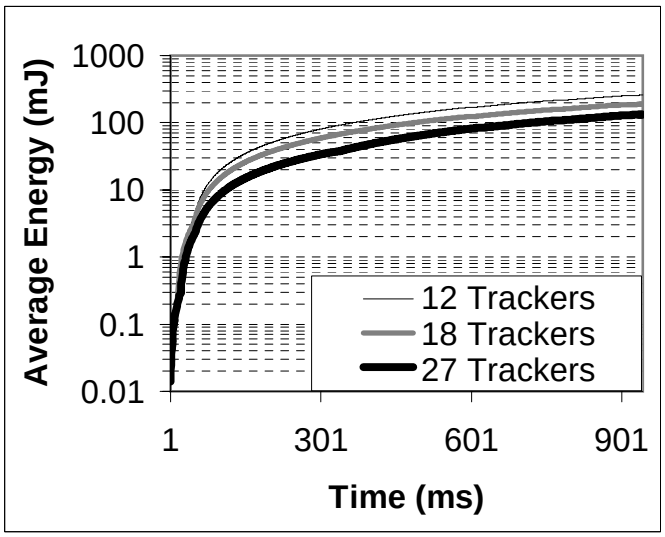


Figure 5.15: Average Energy Dissipation with Varying Number of Trackers

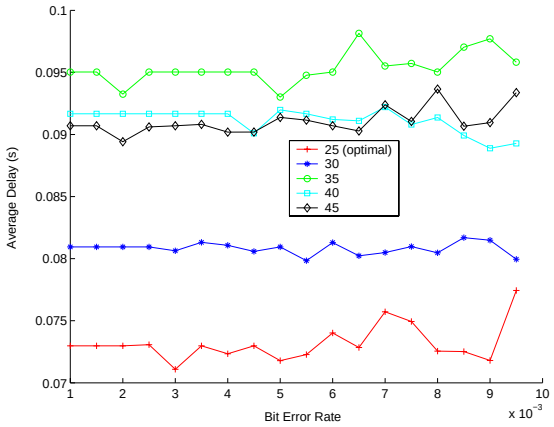


Figure 5.16: Alarm Notification Average Latency

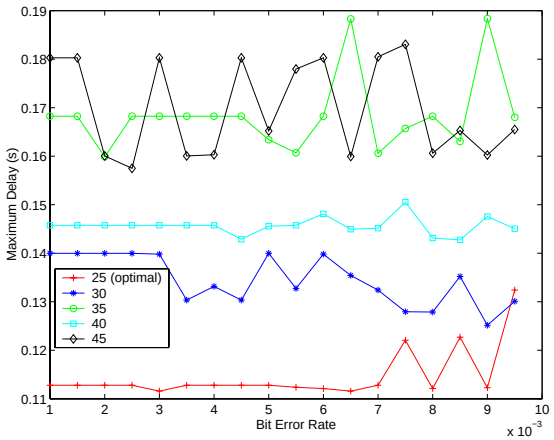


Figure 5.17: Alarm Notification Maximum Latency

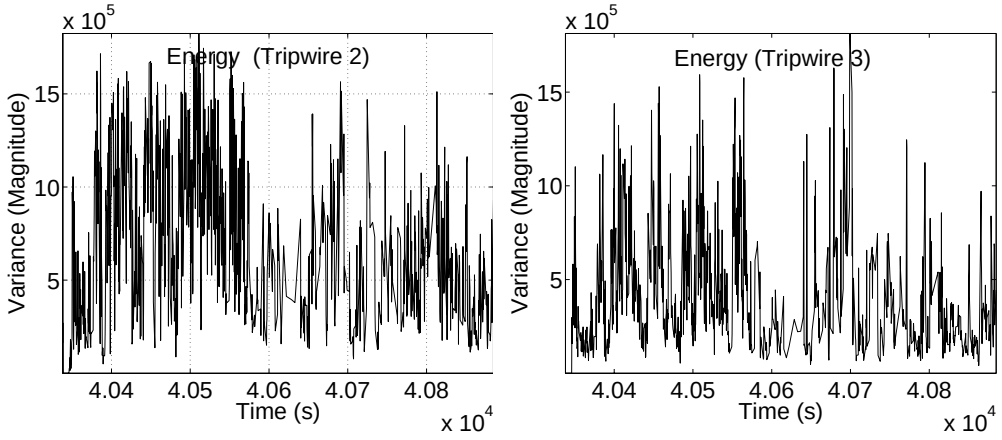


Figure 5.18: Energy in Sampled Variance Form

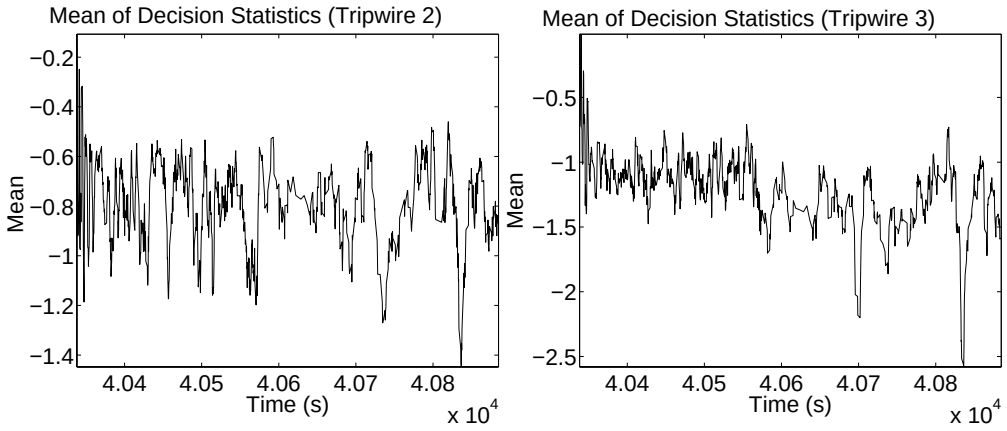


Figure 5.19: Decision Statistics in $\mathcal{L}^1$ Norm

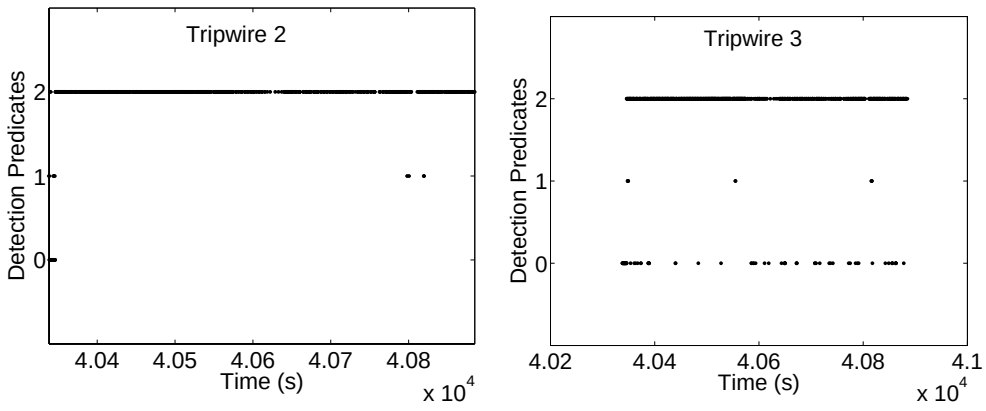


Figure 5.20: Detection Results

6 Conclusions

Wireless sensor networks are very useful for many applications including large scale habitat monitoring, disaster relief, etc. However, sensors are seriously constrained by resources and energy. Promising research on power awareness and energy efficiency span from computation energy efficiency, communication energy efficiency and power/energy management. In this research, we focused on signal processing algorithmic aspects of energy efficiency in wireless sensor networks. Significant energy savings can be obtained by devising new power aware algorithms, analysis and experiments. In fact, communications energy cost is the dominating factor in many applications, and this research exploits new characteristics in wireless sensor networks for energy efficiency.

In this project, a novel power aware coding scheme that exploits spatio-temporal correlation in sensor readings is presented. This scheme combines optimal power allocation with ARQ to select proper coding parameters. The lower bound on energy savings and the upper bound on SNR loss of the scheme are obtained. Experimental and simulation results show that more than 60% energy savings can be obtained, as compared to the uncoded case, and more than 35% energy savings, as compared to temporal coding or spatial coding schemes.

We developed a scheme to track correlations in wireless microsensor networks. It uses a linear prediction model to find the initial correlation coefficients and then uses a discrete Kalman filter model to track the correlations. We have applied this algorithm to source coding and eliminated the calibration of sensor readings. Experiments show that this algorithm gives accurate tracking results of the correlations with manageable processing overhead and low communication cost.

We also developed a novel multi-hop tripwire cueing and detection scheme for a two-tier wireless sensor network to conserve energy. It solved both the false alarm problem and the exposed alarm problem using an iterative detection algorithm and a cueing protocol. This scheme is simple and only primitive communication is required for tripwires. Therefore, it fits well into a two-tier wireless sensor network consisting of inexpensive tripwires and more complex signal processing sensor nodes. Simulation and field experimental results show that the proposed scheme saves a significant amount of energy when compared to other types of network configurations and schemes.

References

[1] The Network Simulator - ns-2, "ns-2 simulator-[online]," *Available: http://www.isi.edu/nsnam/ns*.

[2] IEEE Standard 802.15.4/D18 -2003 "Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (LR-WPANs) [online]," Available: http://www.ieee802.org/15/pub/TG4.html.

[3] Chipcon AS, "CC2420 Radio Model [online]," Available: http://www.chipcon.com.

[4] U.S. Federal Communications Commission, "FCC Part 15 [online]," Available: http://www.fcc.gov.

[5] IEEE, "The IEEE 802.16 WirelessMAN standard [online]," *Available: http://wirelessMAN.org*, 2000.

[6] Silicon Laboratories, "C8051 F02x Developer Notes [online]," *Available: http://www.silabs.com/products/microcontroller*.

[7] Crossbow Technology Inc., "Mica Motes [online]," *Available: http://www.xbow.com*.

[8] CHIPCON AS., "SmartRF cc1000 [online]," *Available: http://www.chipcon.com*.

[9] Sensoria Co., "Sensoria sensor node [online]," *Available: http://www.sensoria.com*.

[10] INTEL PXA255, "Intel PXA255 Procesor Electrical, Mechanical, and Thermal Specification[online]," *Available: http://www.intel.com/design/pca/applicationsprocessors/manuals/278780.htm*.

[11] University of Southern California - Information Systems Institue, "PASTA Microsensor Nodei[online]," *Available: http://pasta.east.isi.edu*, 2003.

[12] A. Aaron and B. Girod, "Compression with side information using Turbo codes, " *Proc. IEEE DCC*, 252–261, Apr. 2002.

[13] H. Abrach, S. Bhatti, J. Carlson, H. Dai, J. Rose, A. Sheth, B. Shucker, J. Deng, and R. Han, "MANTIS: System Support For MultimodAl NeTw orks of In-situ Sensors," *2nd ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)* 2003, pp. 50-59.

[14] N. Bambos, "Toward Power-Sensitive Network Architectures in Wireless Communications: Concepts, Issues, and Design Aspects," *IEEE Personal Communications*, June, 1998.

[15] R. P. Brent, F. G. Gustavson and D. Y. Y. Yun, "Fast solution of Toeplitz systems of equations and computation of Padé approximants", *Journal of Algorithms*, vol. 1, 259-295, 1980.

[16] A. E. Brouwer, and T. Verhoeff, "An Updated Table of Minimum-Distance Bounds for Binary Linear Code, " *IEEE Transactions on Information Theory*, Vol. 39, No.2, pp. 662-677, March 1993.

[17] A. R. Calderbank, I. Daubechies, W. Sweldens, and D. Yeo, "Wavelet Transforms That Map Integers to Integers", *Applied and Computational Harmonic Analysis*, vol. 5, no. 3, pp. 332– 369, July 1998.

[18] A. R. Calderbank and N. Seshadri, "Multilevel codes for unequal error protection," *IEEE Transactions on Information Theory*, vol. IT-39, pp. 1234–1248, July 1993.

[19] J. C. Chen, R. E. Hudson and K. Yao, "Source Localization and Beamforming," *IEEE Signal Processing Magazine*, March, 2002.

[20] J. Chou, D. Petrovic and K. Ramchandran, "A Distributed and Adaptive Signal Processing Approach to Reducing Energy Consumption in Sensor Networks," *Proc. INFOCOM*, April 2003, San Francisco, CA, USA.

[21] J. Chou, S. S. Pradhan and K. Ramchanran, "Turbo and trellis codebooks for source coding with side information," *Proc. of Data Compression Conference*, Snowbird, March 2003.

[22] R. Cristescu, B. Beferull-Lozano and M. Vetterli, "On Network Correlated Data Gathering," *Proc. of INFOCOM*, 2004.

[23] I. Daubechies, "Ten lectures on wavelets," *SIAM*, 1992.

[24] J. Durbin, "The fitting of time-series models, " *Review of the International Institute of Statistics*, vol. 28, pp. 233-243, 1960.

[25] T. J. Flynn, and R. M. Gray. "Encoding of Correlated Observations", *IEEE Transactions on Information Theory*, vol. IT-33, no. 6, pp. 773-787, Nov. 1987.

[26] D. Ganesan, D. Estrin and J. Heidemann. "DIMENSIONS: Why do We Need a New Data Handling Architecture for Sensor Networks?", *First Works hop on Hot Topics in Networks (HotNets-1)*, Princeton, New Jersey, Oct. 28-29, 2002.

[27] J. Garcia-Frias and Y. Zhao, "Compression of correlated binary sources using Turbo codes," *IEEE Comm. Letters*, 5:417–419, Oct. 2001.

[28] D. Ganesan, B. Greenstein, D. Perelyubskiy, D. Estrin and J. Heidemann. "An Evaluation of Multiresolution Storage for Sensor Networks", *Proceedings of ACM SenSys '03*, Nov. 5-7, 2003, Los Angeles, CA.

[29] A. Goldsmith and M. Effros, "Joint design of fixed-rate source codes and multiresolution channel codes," *IEEE Transactions on Communications*, vol. 46, pp. 1301–1312, Oct. 1998.

[30] T. J. Flynn and R. M. Gray. "Encoding of Correlated Observations", *IEEE Transactions on Information Theory*, vol. IT-33, pp. 773-787, Nov. 1987.

[31] J. V. Greunen and J. Rabaey, "Lightweight Time Synchronization for Sensor Networks," *Proc. Second ACM International Workshop on Wireless Sensor Networks and Applications (WSNA '03)*, Sep. 2003, San Diego, CA, USA.

[32] J. Hagenauer, "Rate-compatible punctured convolutional codes (RCPC codes) and their applications," *IEEE Transactions on Communications*, vol. 36, pp. 389–400, Apr. 1988.

[33] R.E. Kalman, "A New Approach to Linear Filtering and Prediction Problems, " *Transactions of the ASME-Journal of Basic Engineering*, 82 (series D): 35-45, 1960.

[34] L. E. Kinsler, A. R. Frey, A. B. Coppens, and J. V. Sanders, "Fundamentals of Acoustics" (Chapter 8), John Wiley & Sons, Inc, 1982.

[35] P. V. Kumar and H, -F. Lu, "Lecture Notes on Error-Correcting Codes [Online]", Available: http://ceng.usc.edu/∼vijayk", 2002.

[36] R. E. Learned and A. S. Wilsky, "A Wavelet Packet Approach to Transient Signal Classification, ", *Applied and Computational Harmonic Analysis*, vol. 2, pp. 265-278, 1995.

[37] S. Lindsey, C. S. Raghavendra and K. Sivalingam, "Data Gathering Algorithms in Sensor Networks Using Energy Metrics, ", *IEEE Transactions on Parallel and Distributed Systems*, Sept. 2002.

[38] Z. Liu, S. Cheng, A. D. Liveris, Z. Xiong, "Slepian-Wolf Coded Nested Quantization (SWC-NQ) for Wyner-Ziv Coding: Performance Analysis and Code Design," *Proc. of IEEE Data Compression Conference*, March 2004.

[39] A. D. Liveris, Z. Xiong, and C. N. Georghiades, "Distributed compression of binary sources with side information at the decoder using LDPC codes," *IEEE Communication Letters*, 6:440-442, Oct. 2002.

[40] A. D. Liveris, Z. Xiong, and C. N. Georghiades, "Joint source-channel coding of binary sources with side information at the decoder using IRA codes," *Proc. of the IEEE International Workshop on Multimedia Signal Processing, St. Thomas, Virgin Islands, USA*, pp. 53–56, Dec. 2002.

[41] D. Li, K. D. Wong, Y. H. Hu, and A. M. Sayeed, "Detection, classification, and tracking of targets," *IEEE Signal Processing Magazine*, 19(2), pp. 17–29, Jan. 2002.

[42] J. Makhoul, "Linear Prediction: A Tutorial Review," *Proc. of the IEEE*, vol. 63, April 1975.

[43] A. J. Martin, M. Nyström, K. Papadantonakis, P. I. Penzes, P. Prakash, C. G. Wong, J. Chang, K. S. Ko, B. Lee, E. Ou, J. Pugh, E.-V. Talvala, J. T. Tong, A. Tura, "The Lutonium: A Sub-Nanojoule Asynchronous 8051 Microcontroller," *Proc. of 9th IEEE International Symposium on Asynchronous Systems & Circuits*, May 2003.

[44] S.S. Pradhan, and K. Ramchandran. "Distributed Source Coding Using Syndromes (DISCUS)". *IEEE Transactions on Information Theory*, vol. 49, no. 3, March 2003.

[45] J. Rabaey, J. Ammer, T. Karalar, S. Li, B. Otis, M. Sheets and T. Tuan, "PicoRadios for Wireless Sensor Networks: The Next Challenge in Ultra-Low-Power Design", *Proceedings of the International Solid-State Circuits Conference*, San Francisco, CA, February 3-7, 2002.

[46] T. S. Rappaport, *Wireless Communications: Principles and Pactice*, Prentice Hall PTR, 1996.

[47] L. K. Rasmussen and S. B. Wicker, "The performance of Type-I trellis coded Hybrid-ARQ protocols over AWGN and slowly fading channels," *IEEE Transactions on Information Theory*, vol. 40, pp. 418–428, Mar. 1994.

[48] R. A. Riley, B. Schott, J. Czarnaski and Sohil Thakkar, "Power-Aware Acoustic Processing", *Lecture Notes in Computer Science*, vol. 2634, Springer-Verlag Heidelberg, 2003, pp. 566 - 581.

[49] M. Rouanne and D. J. Costello Jr., "An algorithm for computing the distance spectrum of trellis codes," *IEEE Journal on Selected Areas in Communications*, vol. 7, pp. 929–940, Aug. 1989.

[50] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec using set partitioning in hierarchical trees," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, pp. 243–250, June 1996.

[51] D. Schonberg, S. S. Pradhan, K. Ramchandran, "LDPC codes can approach the Slepian-Wolf bound for general binary sources," *40th Annual Allerton Confrence*, Champaign, IL, Oct. 2002.

[52] E. Shih, S. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan, "Physical layer driven algorithm and protocol design for energy-efficient wireless sensor networks," *Proc. of the Fifth Annual International Conference on Mobile Computing and Networks (MOBICOM'99), Rome, Italy*, pp. 272–286, July 1999.

[53] D. Slepian, and J. K. Wolf. "Noiseless Coding of Correlated Information Sources", *IEEE Transactions on Information Theory*, vol. IT-19, no. 4, pp. 471-480, July 1973.

[54] T. Spyropoulos, C. S. Raghavendra and V. K. Prasanna, "A Distributed Algorithm for Waking-up In Heterogeneous Sensor Networks", *Proceedings of IPSN 03*, April 2003.

[55] S. Pattem, B. Krishnamachari and R. Govindan, "The Impact of Spatial Correlation on Routing with Compression in Wireless Sensor Networks", *Proc. of IPSN 04*, 2004.

[56] D. Swanson, "Signal Processing for Intelligent Sensor Systems," *Marcel Dekker*, May 2000.

[57] C. Tang and C. S. Raghavendra, "Compression Techniques for Wireless Sensor Networks", Wireless Sensor Network (Chapter 10), Kluwer Academic Publishers, 2004.

[58] C. Tang and C. S. Raghavendra, "A rate-adaptive power control scheme for wireless microsensor networks," *Proc. of International Conference on Mobile and Wireless Communications Networks (MWCN'03)*, Oct. 2003.

[59] C. Tang and C. S. Raghavendra, "Power aware wireless sensor networks using tripwire detection and cueing," *Proc. of Hawaii Intl. Conf. on System Sciences* (to appear), Jan. 2005.

[60] C. Tang and C. S. Raghavendra, "Energy efficient adaptation of multicast protocols in power controlled wireless Ad Hoc networks," *Mobile Networks and Applications*, 9:311– 317, Aug. 2004.

[61] C. Tang, C. S. Raghavendra, and V. K. Prasanna, "An energy efficient adaptive distributed source coding scheme in wireless sensor networks," *Proc. of International Conference on Communications (ICC'03), Anchorage, AK, USA*, pp. 732– 737, May 2003.

[62] C. Tang and C. S. Raghavendra, "Providing Power Awareness to Wireless Microsensor Networks via Tripwires", *IEEE 6th Circuits and Systems Symposium on Emerging Technologies: Frontiers of Mobile and Wireless Communication*, May 2004.

[63] A. J. Viterbi, "Convolutional codes and their performance in communication systems," *IEEE Transactions on Communications Technology*, vol. COM-19, pp. 751–772, Oct. 1971.

[64] G. Welch and G. Bishop, "An Introduction to the Kalman Filter" TR 95-041 (revised version), Department of Computer Science, University of North Carolina at Chapel Hill, March, 2002.

[65] M. V. Wicherhauser, "INRIA Lectures on Wavelet Packet Algorithms", http://www.math.wustl.edu/∼ victor/papers/lwpa.pdf, Nov. 1991.

[66] A. D. Wyner and J. Ziv, "The Rate-Distortion Function for Source Coding with Side-Information at the Decoder", *IEEE Transactions on Information Theory*, vol. IT-22, pp. 1-10, Jan. 1976.

[67] A. D. Liveris, Z. Xiong, and C. N. Georghiades, "Distributed compression of binary sources with side information at the decoder using LDPC codes," *IEEE Communication Letters*, 6:440-442, Oct. 2002.

[68] Y. Xu, J. Heidemann, and D. Estrin. "Geography-informed Energy Conservation for Ad Hoc Routing," *Proc. of ACM/IEEE International Conference on Mobile Computing and Networking, Rome, Italy*, pp. 70-84, July, 2001.

[69] H. Yamamoto and K. Itoh, "Viterbi decoding algorithm for convolutional codes with repeat request," *IEEE Transactions on Information Theory*, vol. IT-26, pp. 540–547, Sept. 1980.

[70] W. Ye, J. Heidemann and D. Estrin, "Medium Access Control with Coordinated, Adaptive Sleeping for Wireless Sensor Networks", University of Southern California/ISI Technic al Report ISI-TR-567 ("also available: http://www.isi.edu/∼johnh/PAPERS/ye03a.pdf"), 2003

[71] R. Zamir, S. Shamai (Shitz), and U. Erez, "Nested linear/lattice codes for structured multiterminal binning," *IEEE Trans. on Inform. Theory*, IT-48:1250–1276, Jun. 2002.

[72] J. Zhao, R. Govindan, "Understanding Packet Delivery Performance In Dense Wireless Sensor Networks," *Proc. of the ACM Sensys*, November 2003.

List of Acronyms

COTS	Commercial Off The Shelf
LOB	Line Of Bearing
ESPIHT	Embedded Set-Partitioning Iteratively Hierarchical Tree
DKF	Discrete Kalman Filter
FFT	Fast Fourier Transform
DISCUS	Distributed Source Coding Using Syndrome
LDPC	Low Density Parity-check Code
SNR	Signal Noise Ratio
LP	Linear Prediction
ROC	Receiver Operating Characteristics
CFAR	Constant False Alarm Rate
DFAD	Distributed False Alarm Detection
EEADSC	Energy Efficient Adaptive Distributed Source Coding
ADC	Analog-to-Digital Convertor
TCQ	Trellis Coded Quantization
SPIHT	Set-Partitioning Iteratively Hierarchical Tree
ARQ	Automatic Repeat Request
RCPCC	Rate-Compatible Punctured Convolutional Code
RMS	Root-Mean Square
FAD	False Alarm Detection
FAP	False Alarm Problem
MCU	Microcontroller
MHC	Multi-Hop Cueing