LEVEL BASED SAMPLING TECHNIQUES FOR ENERGY CONSERVATION IN LARGE SCALE WIRELESS SENSOR NETWORKS

by

Hadi Alasti

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Electrical Engineering

Charlotte

2009

Approved by:

Dr. Asis Nasipuri

Dr. Ivan Howitt

Dr. Jiang (Linda) Xie

Dr. Teresa Dahlberg

Dr. Kent Curran

ABSTRACT

HADI ALASTI. Level based sampling techniques for energy conservation in large scale wireless sensor networks (Under direction of DR. ASIS NASIPURI)

As the size and node density of wireless sensor networks (*WSN*) increase, the energy conservation problem becomes more critical and the conventional methods become inadequate. This dissertation addresses two different problems in large scale *WSNs* where all sensors are involved in monitoring, but the traditional practice of periodic transmissions of observations from all sensors would drain excessive amount of energy.

In the first problem, monitoring of the spatial distribution of a two dimensional correlated signal is considered using a large scale *WSN*. It is assumed that sensor observations are heavily affected by noise. We present an approach that is based on detecting contour lines of the signal distribution to estimate the spatial distribution of the signal without involving all sensors in the network. Energy efficient algorithms are proposed for detecting and tracking the temporal variation of the contours. Optimal contour levels that minimize the estimation error and a practical approach for selection of contour levels are explored. Performance of the proposed algorithm is explored with different types of contour levels and detection parameters.

In the second problem, a *WSN* is considered that performs health monitoring of equipment from a power substation. The monitoring applications require transmissions of sensor observations from all sensor nodes on a regular basis to the base station, which is very costly in terms of communication cost. To address this problem, an efficient sampling technique using level-crossings (*LCS*) is proposed. This technique saves communication cost by suppressing transmissions of data samples that do not convey

much information. The performance and cost of *LCS* for several different level-selection schemes are investigated. The number of required levels and the maximum sampling period for practical implementation of *LCS* are studied. Finally, in an experimental implementation of *LCS* with *MICAz* mote, the performance and cost of *LCS* for temperature sensing with uniform, logarithmic and a combined version of uniform and logarithmically spaced levels are compared with that using periodic sampling.

DEDICATION

To my wife *Homa* and my daughter *Mahsa*

For their patience and supports

To my Mother *Tahereh* and to the memory of my Father *Aliasghar*

For their most honest guidance, understanding and compassion

ACKNOWLEDGEMENTS

I would like to express my appreciation to the people who helped me during my PhD.

First of all, I would like to express my deep and sincere gratitude to my advisor, Dr. Asis Nasipuri. His positive suggestions support, helps and personal guidance provided a good basis for my research during my PhD. Throughout the dissertation-writing period, he provided positive encouragement, sound advice and plenty of good ideas.

I would like to appreciate my PhD committee members Dr. Ivan Howitt, Dr. Linda Xie, Dr. Teresa Dahlberg and Dr. Kent Curran for accepting being in my PhD committee, for their detail review and for their superb advice during my defense.

I would like to express my appreciation to Dr. Kasra Daneshvar for his constructive encouragements, sound advice and approving my admission request for studying PhD. I also would like to thank Dr. Susan Sell for tuition bursary during my PhD. This work was partially supported by the Electric Power Research Institute (EPRI).

I want to appreciate Dr. Seyed Mehdi Miri and his family for their excellent helps when my daughter Mahsa was born. I also want to appreciate Dr. Mohammad Kazemi for recommending me for an on-campus summer job in 2008.

I want to appreciate the very kind staffs of Electrical Engineering department and also the ISSO dean, Marian Bean and the ISSO staffs for their friendly helps.

Last but not least, I would like to express my deepest gratitude to my wife Homa and my daughter Mahsa for their patience and spiritual supports, and to my brother Mehdi for his helpful guidance and sharing his experience with me. I would also like to express my appreciation to my mother and also to my relatives Farideh, Farkhondeh, Majid, Hosnieh, Reza, Zhiela and Hossein who helped me during my PhD.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABREVIATIONS

WSN	Wireless Sensor Network
SN	Sensor Node
MSE	Mean Square of Error
RMSE	Root Mean Square of Error
ONLi	Opportunistic Neighbor Listening
QV	Quantized Value
GC	Gradient Compass
ED	Estimation of the displacement
ATU	Adaptive Time Update
NATU	Normalized Adaptive Time Update
LCS	Level Crossing Sampling
LCSH	Level Crossing Sample and Hold
A/D	Analog to Digital Converter

CHAPTER 1: INTRODUCTION

A Wireless Sensor Network (*WSN*) is an emerging category of wireless networks that consists of spatially distributed wireless devices equipped with sensors to monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, pollutants, etc. in various locations [1]. Wireless sensor networking is an emerging field that is poised to revolutionize various automated monitoring and tracking operations. A wide range of applications of *WSN* have been reported in numerous domains. Examples include: precision agriculture [2], monitoring battle-filed [3], structural monitoring of buildings [4], transportation [5], wild-life monitoring [6,7,8], management of disaster-zones [9], monitoring the industrial environments such as power plants and power substations [10], refineries [11], manufacturing factories [12,13], healthcare [17,68,69], weather monitoring [14], etc.

A *WSN* (Figure 1.1) generally includes a number of wireless sensor nodes (*SN*) and

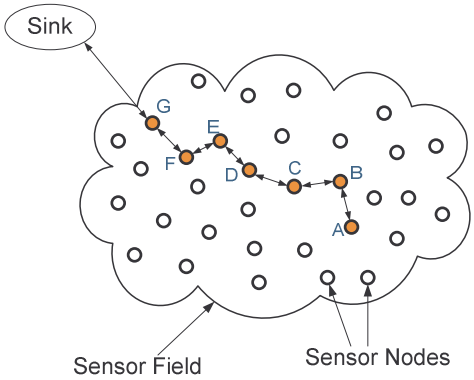


Figure 1.1: Wireless sensor network with single base station

one or more base-station(s) or sink(s) that are normally connected to a server or database through a gateway. *WSNs* optionally may also have cluster heads. Each *SN* is a miniature computer which usually has interfaces and components with extremely basic functionalities. *SN*s usually have a processing unit with limited computational power and small memory, sensing components that can be on a mountable sensor board, a communication transceiver (radio), and a power source which is usually in the form of a battery (sometimes accompanied with solar cells) [1]. Although the capabilities of individual *SN*s are limited, the idea is to utilize the combined processing and sensing capabilities of a large number of such *SN*s using appropriate distributed processing and networking protocols to develop a powerful distributed sensing system. The base stations are components of a *WSN* with computational, energy and communication resources, which act as gateway(s) between sensor field and the end users.

Although the field of wireless sensor networks poses a variety of interesting research challenges that range across many disciplines, energy conservation is an aspect that must be addressed by all solutions. This is because of the fact that the utility of the *WSN* is completely dependent on the life of the energy resources (usually battery) at the sensor nodes. As discussed in various literatures, the energy consuming operations in *WSN* are usually categorized in three major groups: communication, computation, and sensing. A comparative study shows that for a generation of Berkeley *SN*, the ratio of the required energy for single bit communication over the required energy for processing of a bit ranges from 1000 to 10000 [15]. This huge ratio clearly shows that to have a *WSN* with longer life-time, the successful protocols, signal processing algorithms and network planning schemes should shift the network's operating mode from communication-

dominant to computation-dominant mode. For instance, a considerable part of the network's energy is wasted on inefficient multi-hopping and packet collisions in the network. This energy loss may be reduced by signal processing schemes like localized in-network information compression [16], and collaborative signal processing [15].

Another major challenge that is critical for designing protocols and algorithms for *WSN*'s is scalability. Network scalability is the adaptability of the network's protocols and algorithms to the variation of the node density for maintaining a defined network's quality. As also defined in [17], a network is scalable if the quality of service available to each node does not depend on the network size.

In many cases the challenges related energy consumption and network size are inter-twinted in the sense that one arises due to the other. Hence, special solutions are required that involve development of theoretical design principles as well as applications of innovative practical approaches to address these problems. These solutions depend largely on the application.

In this research we investigate solutions for achieving energy efficiency in large-scale *WSN* in two different problem areas. The first problem involves applications of large scale *WSN* for monitoring spatial distributions of signals. The approach that we present to solve this problem is to remove spatial redundancy to conserve energy. This is achieved by developing special mechanisms that require only small subset of sensor nodes to transmit for enabling spatial and temporal monitoring of a signal.

The second problem is experimental. Here, we present solutions that are applied to a large-scale wireless sensor network for monitoring equipment health status from power substations over long period of time. The key challenge here is that periodically obtained

data from all nodes can make the data traffic prohibitively high, and hence, special techniques are required to limit data transmissions, thereby conserving battery, without sacrificing monitoring performance. The performance of the proposed solution for this problem is shown using experimental results.

Throughout the dissertation the signals are assumed stationary and in some cases the probability density function of the signal is assumed known for optimization. The "communications cost" term is used with the meaning of "the number of transmissions".

This dissertation is organized as follows. Following this introduction, we present a literature survey on related work in chapter 2. In chapter 3, we present the problem of monitoring the spatial distribution of signals using a large-scale *WSN* and the proposed approach to achieve energy efficiency. We present analysis and performance results which motivate the optimization of parameters and development of specific design strategies for using the contour-based spatial monitoring approach. These design aspects and performance results are presented in chapter 4. In chapter 5 we present solutions for tracking temporal variations of spatial distributions using this approach. The experimental part of this research is presented in chapter 6. We describe the experimental context, the problem statement, and the solutions implemented to address the problems. Conclusions are presented in chapter 7.

CHAPTER 2: BACKGROUND REVIEW

Energy conservation in *WSN* is a challenging problem [1]. Sensor selection, i.e. efficiently selecting the sensors for sensing is a smart solution that partially resolves this problem. Sampling of the signal at the intersection of the appropriately defined reference levels is an active research area for decreasing the power consumption in mobile technology based equipment [19, 20].

*WSN*s have great potentials for environmental monitoring, such as detection of chemical gas diffusion, contaminant spreading and so on. These phenomena often span over a large geographical areas instead of a specific location. Due to their large size and time-varying nature, they cannot be adequately monitored or tracked using traditional techniques with few isolated sensor nodes. For this, dense large scale *WSN* is normally proposed. A distributed data fusion approach is often adopted in such a system so that the remote base station can observe the monitored phenomena periodically based on the extracted information without the large amount of raw data.

Monitoring the spatial distribution of a signal by detection of a set of its contours using dense, large scale *WSN* in the absence of observation noise was discussed in [18]. For that, edge nodes, sensors whose observations are in a given δ margin of the closest contour levels, send their related contour level to the sink. To monitor a correlated spatial distribution periodically, detection of the contours, as proposed in [18] is not efficient.

In many cases, sensor observations are highly affected by noise. For such cases, spatiotemporal monitoring of signal distribution from its contours with the proposed approach in [18] must be modified because of the following reasons:

- Because of the presence of noise, the contour levels cannot be taken very closely, and then it is not possible to model the spatial distribution accurately with a set of contour levels. Accordingly, a principal question that should be answered is: where must the contour levels be defined to reduce the modeling error for better monitoring?
- The noise intensity affects the quality of detection and tracking of the contour tracking. For this, appropriate distributed algorithms can reduce of the effect of noise. It is worthy to mention that any missed detection may reduce the performance of the signal reconstruction at sink, and any false detection exhausts the in-network energy for sending to the sink.

Although to the best of our knowledge, no research has been reported on solutions of spatiotemporal monitoring of noisy signals using contour detection, several papers present related work. In this chapter the related literatures on the following groups are reviewed:

- Spatial contour detection algorithms using wireless sensor networks
- Tracking the temporal variation of contours using wireless sensor networks
- Level-crossing sampling schemes for finding the appropriate contour levels

2.1 Survey on Spatial Contour Detection

Using edge node instead of edge line to delineate the borders of a region was firstly proposed in [21]. They discussed three methods for detection of the edge nodes for binary

edges that divide a region into interior and exterior parts (see Figure 2.1). These methods were: statistical approach, filter-based approach and classifier approach. These approaches are briefly reviewed here. In these approaches, edge nodes are detected in a two step decision: a) local decision which is based on sensor node's measurement; and b) fusion decision which determines if the probable edge node is selected as edge node.

1- Statistical approach:

This approach does not assume the location of the nodes and uses a relatively simple algorithm. In comparison to filter-based and classifier approaches, statistical approach has poorer performance.

2- Filter-based approach:

In this approach the geographical information of the nodes is assumed and classical image processing filtering is applied to detect the edge nodes. Unlike the pixels of images, the sensors are not typically located over grids then normalization coefficients were defined to rectify the effect of random distribution of the nodes.

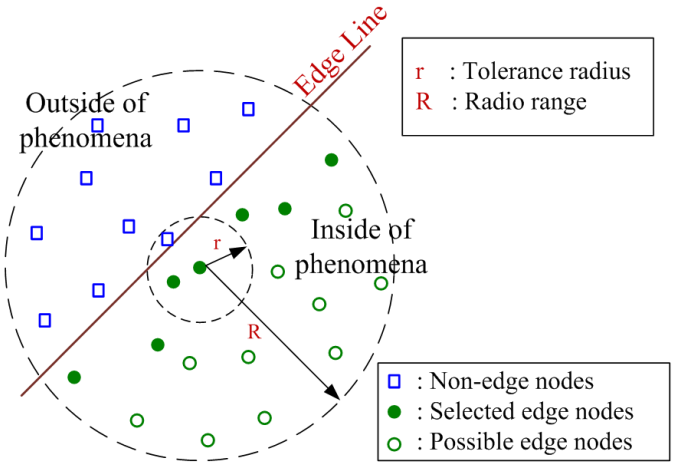


Figure 2.1: Defining the edge nodes based on the inside and outside regions in [21].

3- Classifier approach:

This approach which also assumes the geographical information of the nodes, is similar to techniques applied for pattern recognitio. Based on the local decision of the sensors, the radio range is divided into two sets. A linear classifier is adopted to detect the edge nodes. Unlike statistical and filter-based approach, classifier approach does not use detection threshold for detection of the edge nodes. This approach is sensitive to the level of interfering noise.

After [21], many edge, boundary and contour detection algorithms were proposed that can be categorized in one of the three mentioned types. Next we review the related work in statistical and filter-based categories. Besides these three types, a few of the other schemes, such as geometrical and graph theory based boundary detection schemes are also reviewed in this section.

2.1.1 Algorithms with Statistical Approach

Liao, et al. [22] proposed a one-and-two-level approach based on local and global maximum likelihood ratio test to enhance the statistical approach proposed by [21] for edge detection. They re-defined the edge based approach on assuming the detected edge nodes at both sides of the border of the event within tolerance range *r*, to relieve the shortcoming of the statistical approach in [21]. This approach is difficult to implement when the signal power and noise information are unknown.

To overcome the difficulties of the proposed approach in [22], a composite hypothesis approach was proposed in [23]. This approach consists of a two steps detection algorithm to find firstly, if the real edge crosses the radio range and secondly, if

the real edge crosses the tolerance edge. This algorithm works properly when the radio range (R) is much larger than the tolerance range (r). In sensor network radio range is limited by transmission power and the tolerance range is limited by node density.

Liao, et al. [24] enhanced the detection of the edge region in statistical approach using Neyman-Pearson (*NP*) criteria. Based on *NP* criteria they proposed an idea that addressed to the threshold selection problem. They also compared the performance of their approach with classifier-based approach under the assumption of location error and showed that their statistical approach performs better than classifier approach of [21].

Liao, et al. [25-27] proposed contour line detection instead of edge detection to not only recognize the region of a certain phenomena, but also extract other information such as signal amplitude and source location. The proposed method in [25-27] is based on regional clustering and having communication between cluster-heads (see Figure 2.2). It can track changes, but is not a localized approach and some sensor nodes need to have some more capabilities to be a cluster-head.

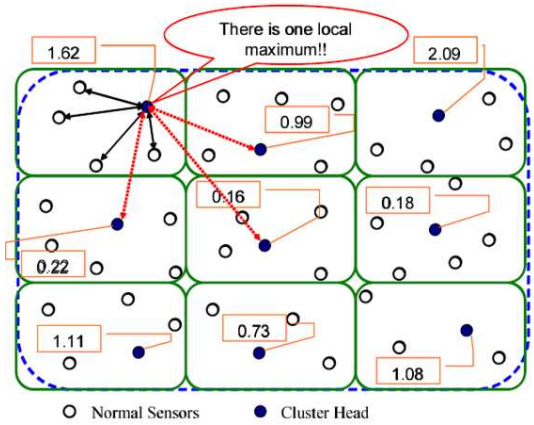


Figure 2.2: Illustration of the coarse localization of the local extreme points in [25].

The effect of data quantization error, sensing noise and communication channel imperfections was considered in evaluation of contour line detection in wireless sensor network in [29]. They showed that the performance of algorithm by changing the performance measures from hard decision, like false and missed detections to the soft decision measures under variation of parameters like bit-error-rate (*BER*), number of quantization bits, node density and probing range, was different from what expected, occasionally. For example they observed that there is a radio range limit that optimized the quality of contour detection.

The quality of the transmitted data through the channel is affected by channel condition. The data-loss in the radio channel causes excessive power consumption and depletes the battery power of a sensor node. As a solution to the problem in [29], an adaptive data fusion scheme is proposed to improve the energy efficiency which is shown in Figure 2.3.

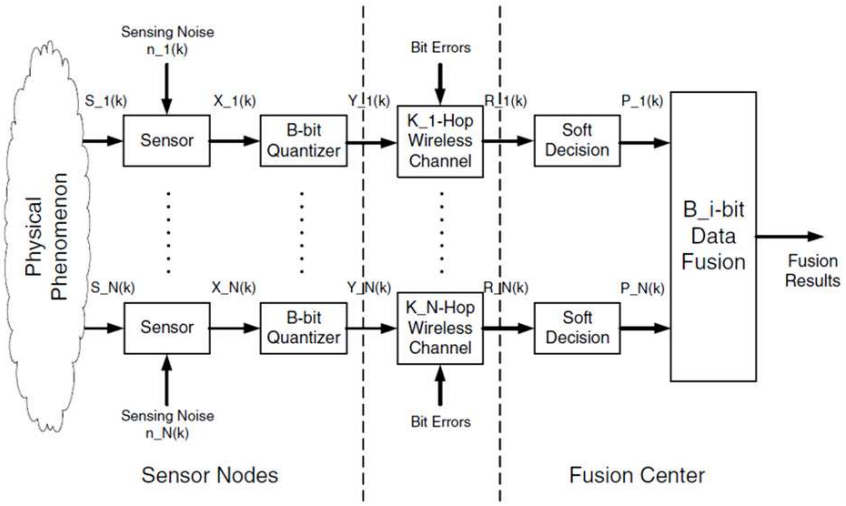


Figure 2.3: The system model in adaptive date fusion

2.1.2 Filter Based Approaches

By applying *Laplacian* and local variance operators, a different filter-based approach from [21] was proposed to detect the boundary of a physical process in large scale sensor network in [36]. They studied the problem under when each sensor knows the coordinates of its immediate neighborhood and evaluated their proposed edge detection based on computer simulations. They assumed that the sensor observation is along with noise. In the paper, the immediate radio neighborhood was divided into two inner and outer circular regions (see Figure 2.4). This division was used in calculation of the mean value which was used in the definition of the local variance operator. The advantage of their introduced operators over the filter-based approach in [21] was that it was orientation independent. They showed that there is a tradeoff between the radio range coverage and the node density.

A robust filter-based contour detection algorithm in wireless sensor network was proposed in [53]. We used a multi-level quantization along with collaborative detection for detection of the contour line of a two-dimensional Gaussian distribution when the sensor observations are heavily affected by noise. The detailed review of this work is

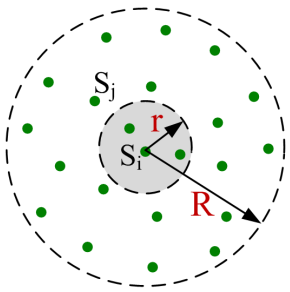


Figure 2.4: Partitioning the area of the radio range into inner and outer regions for calculation of the local mean in [36].

presented in chapter 3.

2.1.3 Geometrical and Graph Theory Based Algorithms

Jiang, et al. [28] proposed a cluster-based R-tree scheme for detection and tracking the phenomenon region in wireless sensor networks. R-tree is a dynamic index structure for spatial searching. They assumed that all of the nodes are over the grid points and any additional node is treated as backup node (see figure 2.5). The backup node assumed for higher reliability and fault tolerance of the sensor network. The algorithm includes three principal parts: a) boundary detection for detection of the chain-of-vertex of the event, b) merging to unify the regions with contiguous chain of vertex, c) removing the unnecessary chain of vertex for simplification. They showed that their algorithm was accurate and efficient in memory, communication and energy costs. They assumed that the sensor observations were noise free.

For tracking, the variation of the center centre of the boundary in the average meaning over the elapsed time was evaluated. They showed that in the absence of noise it is able to track the direction of the variations.

Isoline has been introduced in [30]. Isoline in essence is a neighbor-to-neighbor

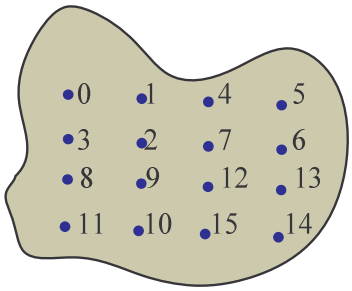


Figure 2.5.a: The sensors network over the grids

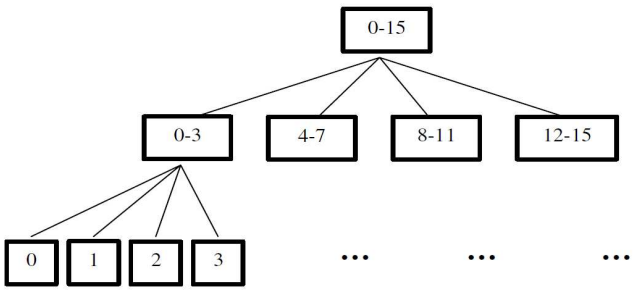


Figure 2.5.b: The R-tree illustration of the area

protocol to optimally collect data by reduction of redundant transmissions. Initially, nodes which detect Isoline level in their immediate neighborhood report to the neighborhood and this information will be updated periodically and maintained by the neighborhood. In addition to the periodic update, significant sensed changes also lead to local reports in neighborhood. Finally the best node (closest value to Isoline level and perhaps closest distance) sends its values (or the isoline code) to the data collection sink. In order to reduce the cost, to send the data to the sink, first the data is collected to the vertices of the polygon of sink's data-distribution-tree and then the stored data in vertices will be sent to the sink. To evaluate the proposed protocol, *CSMA MAC* protocol in ns2 simulator was used for a grid sensor network. In [30] the effect of the network size and also the noisy measurement on the result was ignored.

To have some more information about the sub-contours of a contour level, Iso-Map has been proposed in [31]. Iso-Map is similar to Isoline with this difference that to have the connections of the sub-contours, the local gradient of signal along with sensed value

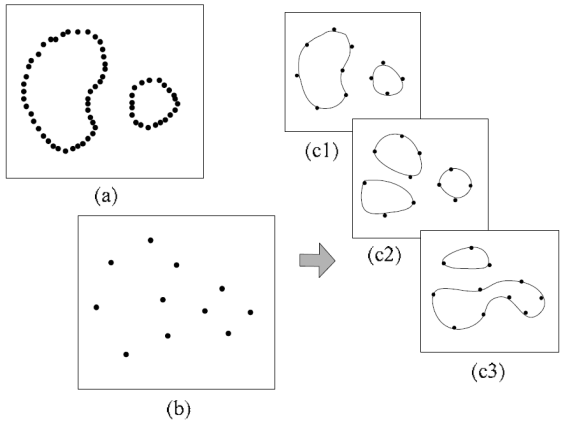


Figure 2.6: Contour mapping from isoline nodes. (a) Dense deployment of sensor nodes leads to the isolines; (b) Sparse deployment of sensor nodes provides ambiguous information; (c1) – (c3) three possible contour maps of (b).

and sensor location was be sent to the sink. The inclusion of the gradient help reduce the ambiguity of the collected information in construction of the contours. Figure 2.6 illustrates this ambiguity information. To calculate the local gradient of the signal, the signal in the immediate neighborhood of the node has been estimated by the gradient of regression and for simplicity the linear regression has been assumed. Besides considerations of isoline information, the traffic of network and the required overhead were analyzed and compared with other approaches.

A data-driven distributed algorithm for searching a collection of sensors representing the iso-contours of a spatial signal distribution within a specified signal range was proposed and discussed in [34]. They developed a gradient-based routing algorithm from query messages of any query nodes that was also used for discovery of the contour-tree. The querying was localized in the immediate neighborhood of each node for finding the iso-contours. They also considered the presence of noisy data reading, but the effect of noise intensity on quality of the proposed algorithm was not discussed.

In [32,33] the iso-contour lines of a signal distribution was approximated with the closest k-vertex polygons in sensor field. Distributed ADAPTIVE-GROUP-MERGE (*AGM*) algorithm was proposed in [32] for approximating iso-contours with polygons that guarantees the approximation error with a constant factor. They showed that the performance of *AGM* is comparable to the performance of centralized optimal scheme. In [33] they proposed an algorithm that guaranteed approximation error within a small constant error of $\Theta(k/\log m)$ with Hausdroff error in k-vertex approximation where m is the number of iso-contours. In the proposed algorithms in [32, 33] the effect of observation noise was ignored.

The problem of approximating a family of iso-contours with topologically equivalent polygons in wireless sensor field using a simple and efficient algorithm was discussed in [35]. They evaluated their algorithm for real and synthetic data using simulation. The effect of uncertainty in sensor readings and tracking the variations of the contours were not taken into account.

2.2 Review of Contour Tracking Literatures

Unlike target tracking problem in wireless sensor network, which has been studied comprehensively in numerous literatures, contour tracking was studied just in a few of the recently published references [37,38,39]. Although different in approach, they were similar at two points: *i*) they repair the broken parts of the contours, and *ii*) periodically update the variations of the contours.

An energy-efficient algorithm for continuous boundary tracking from noisy observation using range wireless sensors was discussed in [37]. For spatial boundary estimation a regression-line based approach was proposed and for tracking the temporal variation of the boundary, a Kalman-filter based approach was used. The range sensing scenario and the estimation of the confident range after estimating the regression line are

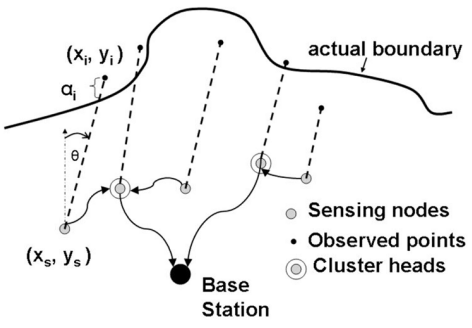


Figure 2.7.a: Range sensing scenario

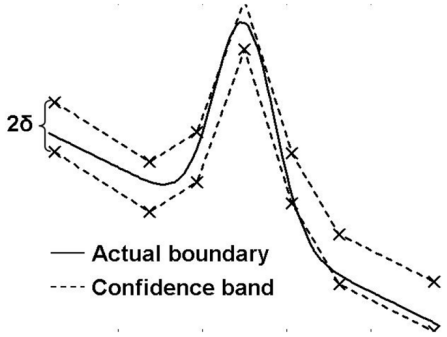


Figure 2.7.b: Estimation of the confident range

illustrated in Figures 2.7.a and 2.7.b, respectively. The range sensors were assumed to be able to align their antenna to sense the signal at an angle for locating a point about the boundary. It was also assumed that all of the sensors were located in interior side of the boundary. Boundary was estimated with piece-wise linear model of the detected point-wise boundary. The performance measures were communication overhead and accuracy of measurement. To make sure that the algorithm can detect the boundary under any conditions, a mixed combination of Kalman filter and spatial detection which was called dynamic boundary tracking (*DBTR*) was proposed and evaluated. The Evaluation results proved that *DBTR* is more consistent and accurate than spatial estimation. The performance of the proposed algorithm for spatiotemporal estimation of the boundary was approved using *TOSSIM* simulation.

A light-weight distributed algorithm for repairing the broken lines of a contour in order to track its variations using binary sensors, instead of periodically detecting the contours was proposed and discussed in [38]. As stated, the proposed algorithm was not proper for the case where the sensor observation is affected by strong noise (like salt and pepper noise in image processing). The proposed algorithm searches for an efficient and appropriate connecting path, considering topology maintenance. The principal

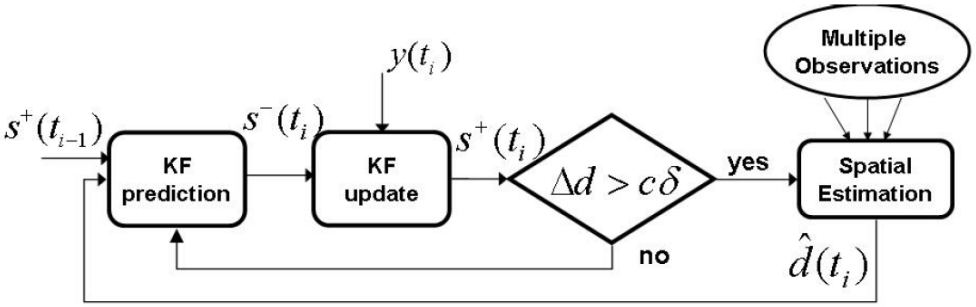


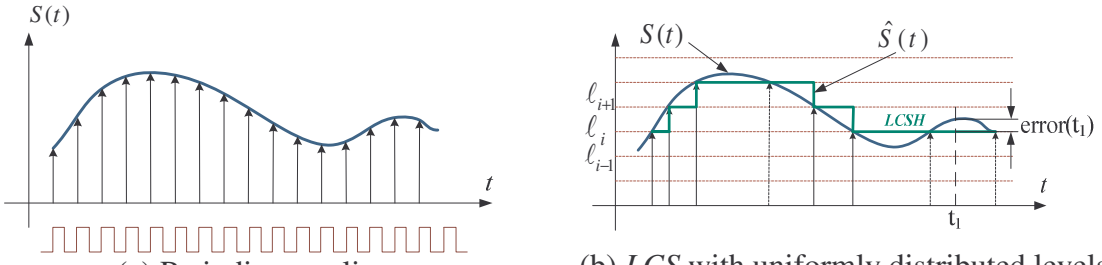
Figure 2.8: Combination of spatial with Kalman filter in DBTR in [37].

contribution of the work is that it proposed the minimum path to repair the broken contour.

Distributed Event Contour Tracking (*DECT*) was proposed in [39]. *DECT* used multi-sink along with traffic-load balancing for increasing the network life, and periodic contour detection for tracking the boundaries contour(s). For these *DECT* should perform three operations: 1) network topology establishment close to hexagonal base, 2) detection of the contours of the area, and 3) partitioning the area related to the segments of the contours and determining their corresponding sink. The number of contour segments is equal to the number of sinks. In development of *DECT* the presence of noise in sensor observation was ignored.

2.3 Level-Crossing Sampling (*LCS*)

LCS is a subclass of non-uniform sampling based on sampling at the crossing points of a set of levels. Figures 2.9.a and 2.9.b visually compare the periodic sampling and *LCS*. *LCS* has been studied under several different names including event-based sampling, magnitude driven sampling, deadbands, and send-on-delta [40]. Although the subject has been investigated for many years, its potential received renewed attention in recent years was due to applications where signals with variable characteristics are



(a) Periodic sampling

(b) *LCS* with uniformly distributed levels.

Figure 2.9: Illustration of level crossing sampling (*LCS*) with zero-order spline reconstruction, i.e. level crossing sampling and hold (*LCSH*)

present, such as in voice-over IP (*VOIP*) and sensor systems. The published work in the recent years on *LCS* can be categorized in two applications: *i*) performance improvement issues in *LCS* [19, 20, 41-46, 49], and *ii*) information theoretic related issues [47,48].

The performance of *LCS A/D* converter is highly affected by reference level placement. Guan et. al. in [44] showed that it is possible to sequentially and adaptively implement these levels. Based on this, they proposed an adaptive *LCS A/D* converter which sequentially updates the reference sampling levels to properly decide where and when to sample. They analytically proved that as the length of the signal's sequence increases, their adaptive algorithm's performance approaches to the best possibility.

The speed of sampling of the signal for proper *LCS* depends of the bandwidth of the signal. When the signal's bandwidth changes with time or is unknown, the sampling speed should be adaptively found. Using short-time Fourier transform (*STFT*) with constant time-frequency resolution is very common in analysis of the time-varying signals. Qaisar et. al. in [49] proposed a computationally reduced adapted *STFT* algorithm for level-crossing sampling. In their algorithm the sampling frequency resolution and the window of *STFT* is adapted based on the characteristics of the signal. As the sampling rate is adapted, the processing power of the *LCS* based *A/D* has been significantly reduced. In another related work of these authors, to filter the non-uniformly spaced samples of the *LCS* output, and adaptive rate finite impulse response (*FIR*) filter was proposed in [20] and its computational complexity was reduced. In the proposed scheme, linear as well as non-linear techniques were used. The proposed adaptive algorithm was proposed for application in power limited mobile systems for applications where the signal's amplitude remains constant for a long period of time.

The use of oversampled *A/D* along with low resolution quantization for reconstruction of a specific class of non-bandlimited signals was studied in [46]. They showed that the studied sampling scheme outperforms the periodic sampling. They also studied the sampling of finite rate innovation signals using *LCS* [41]. They proposed an algorithm for perfect reconstruction of this category of signals after *LCS*.

The processing of non-stationary signals after *LCS* was studied in [19]. The difficulty of signal reconstruction due to non-uniformly spaced samples and the time-varying statistical properties of the signal were reviewed. The shortcomings of *STFT* which is the appearance of spurious components and the drawbacks of wavelet transform which are low spectral resolution at high frequencies and low temporal resolution at low frequencies were reviewed. Clock-less signal dependent transforms were proposed to improve the reconstruction performance. Because the spectral characteristics of the non-stationary signals vary with time, the signal dependent transform should be adapted locally.

Signal reconstruction after *LCS* using cardinal spline was discussed in [42] for *LCS* samples of speech signal. It was shown that in many of the cases the applied reconstruction approach works properly, but not always. Using the non-uniformly spaced reference levels was mentioned as a tentative solution to this problem.

An *LCS* based *A/D* was introduced in [43], where issues like speed, resolution and hardware complexity are investigated. For evaluation of the quality of the sampling, the reconstruction root mean square of the error was calculated after zero and first order polynomial interpolation of the non-uniformly spaced samples and then uniform sampling. In addition to polynomial interpolation, decimation was used for increasing the

overall resolution of the converter. Based on simulation, a modest decimation factor was suggested based on the amount of generated error outside of the signal's bandwidth.

Application of *LCS* for efficiently sampling the bursty signals was studied in an information theoretical framework in [47]. They showed that while *LCS* has less total sampling rate can convey the same amount of information as periodic sampling. The idea was proposed for data communication and compression. They proposed to use the probability density function of the signal in designing the *LCS A/D* reference levels for optimal *LCS*.

A systematic framework for reference level sampling of the random signals at quantized time was proposed and discussed in [48]. The main concentration of the work was on data compression and a structure for the non-uniform predictive encoder and decoder was proposed. The proposed algorithm was applied on image compression and the performance of the result was numerically discussed.

2.4 Summary

The chapter reviews the related literatures of sampling the one and two-dimensional signals at levels. For two-dimensional, the detection techniques of the contour line of a correlated spatial distribution in wireless sensor field is reviewed. The techniques are reviewed based on the applied approaches that are statistical, filter-based, classification, graph theory based, and geometrical approaches. Next the related contour tracking approaches are reviewed. Unlike the spatial contour detection approaches that are numerous, the temporal approaches are just a few.

The literatures of sampling the one-dimensional signal at levels or so-called level-crossing sampling (*LCS*) are reviewed then. The focus of these literatures is mainly in two areas of improvement issues in *LCS*, and information theoretic issues in *LCS*.

CHAPTER 3: CONTOUR DETECTION FOR SPATIAL DISTRIBUTION MONITORING IN LARGE SCALE WIRELESS SENSOR NETWORKS

3.1 Introduction

A large number of environmental monitoring applications require that the spatial distribution of a signal, such as temperature, soil humidity, etc., is to be monitored over a wide area. Examples include the monitoring of spatial density of air pollutants in large cities, monitoring of surface temperature in deserts and polar icecaps, monitoring of density of contaminants in water resources, monitoring of the density of poisonous gas density in chemical attacks, determining the soil moisture over an agricultural area, and so on. The number of sensors involved in such monitoring applications may be very large, and consequently, traditional methods such as periodic sampling from all sensors can lead to a prohibitively heavy traffic and energy consumption. However, since the signal distribution in such applications is expected to be spatially correlated, special techniques that utilize this property can be used to reduce the number of transmissions over periodic sampling. This chapter presents a scheme that uses contour detection for efficiently monitoring such spatially correlated signals over large areas.

Application of wireless sensor network for monitoring the correlated spatial distributions has been studied in a few researches. The following three examples are among them.

Fundamental sensor network design issues for statistical inference about correlated random fields in two-dimensional setting was discussed in [55,56]. To quantify the amount of information from sensor network, *Kullback-Leibler* information (*KLI*) and mutual information (*MI*) measures were studied in spectral domain. The asymptotic information rate for general d-dimensional stationary Gaussian field and in particular, for two-dimensional the asymptotic behaviour of ad hoc lattice sensor network deployed over correlated random field were characterized. Under specific assumptions, they obtained fundamental scaling laws for total information, energy efficiency, coverage, node density and variation of consumed energy. These results might be used as guidelines for sensor network design.

A model-based approach for monitoring of a spatial distribution over time using sensor network in the absence of observation noise was proposed in [18]. In their approach, the spatial distribution was modelled with its contours. The equally spaced contour lines were detected periodically by the sensor nodes. They studied the cost of their proposed scheme using computer simulations with ns2. They also proposed using time-series for estimating the contour's position at the intermediate instances between the periodic contour detections.

Reconstruction of a correlated bandlimited random field using several analytical techniques from randomly positioned sensor nodes in the presence and absence of observation noise was studied in [57-61]. They studied the field reconstruction error versus the node density and the bandwidth of the field. In their research they described the spatial distribution with a limited number of harmonics. In the presence of noise, the reconstruction performance of matched filter (*MF*), linear minimum mean square of error

(*LMMSE*) and zero-forcing (*ZF*) were compared when the signal-to-noise ratio changed. The result of their work can be used as guideline when the reconstruction quality, signal to noise ratio, node density and the bandwidth should be traded off.

Use of contour lines or isolines is a common technique used in various disciplines for depicting variations of physical quantities over two-dimensional space. Examples include geographical science (for depicting elevations, water depth, magnetic fields, etc.), meteorology (for depicting temperature, barometric pressure, precipitation, etc.), geology, oceanography, etc. Here we address the use of contour lines as a method for depicting spatial distributions with the objective of minimizing the number of sensors that need to be involved in the spatial monitoring process. For this, the first step is to efficiently and accurately detect a set of contours that can be used to depict the spatial distribution of the signal. Related detection schemes based on sensor observations have already been reviewed in chapter 2 of this dissertation. In these schemes, after applying the contour detection algorithm, the detected sensors send their observations to the sink or their related cluster head. When the sensor observations are heavily affected by noise, a large extra number of sensors may be detected as contour nodes which would waste energy and increase the error in estimating the signal distribution. We propose distributed collaborative processing as an approach for reducing erroneous detections caused by noise. In particular, our proposed scheme of distributed contour detection reduces part of unnecessary communications throughout the network and saves the in-network energy.

3. 2 Problem Statement

3.2.1 Application Scenario and Objective

We consider a given correlated random signal distribution over the extent of an area as is shown in Figure 3.1.a. This signal may be noisy due to unpredictable environmental factors and calibration errors that may have a significant effect on the weak signal being detected, as shown in Figure 3.1.b. This signal is sampled by a large scale, dense, uniformly distributed *WSN* as is shown in Figure 3.1.b. By using these samples the spatial distribution of the signal may be monitored at the sink. The objective of this chapter is to present a mechanism to perform this task that is energy efficient and effectively reduces the effect of noise. In particular, our objective is to design mechanisms to avoid the need for sampling the signal at all sensors in the *WSN* that may be deployed over the whole region but *use a subset of the sensors* to estimate the desired signal distribution to reduce the number of transmissions.

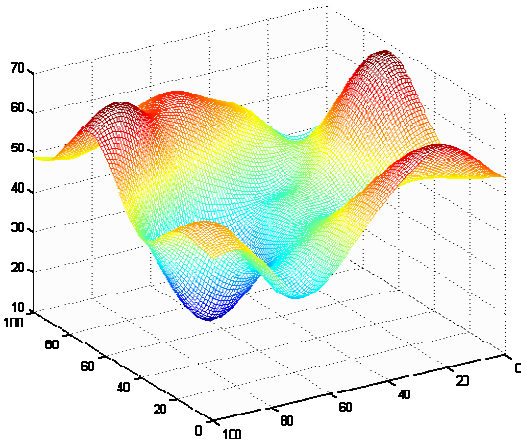


Figure 3.1.a: Correlated spatial distribution.

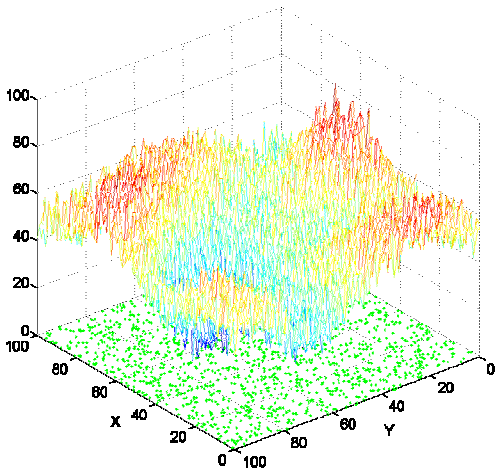


Figure 3.1.b: Corrupted spatial distribution is sampled by a large scale dense wireless sensor network.

3.2.2 Assumptions

We make the following assumptions on the signal, noise and sensors:

- The range of the signal strength of the correlated spatial distribution is known.
- Additive noise has Gaussian distribution with finite power with zero mean.
- A large scale, dense, uniformly distributed static *WSN* is assumed throughout the extent of correlated spatial distribution.
- Wireless sensors have limited processing power and memory.
- All sensors are aware of their locations and each sensor knows its immediate neighborhood in a single-hop radio range.
- To send the processed samples to the sink(s), multi-hopping is required.
- All of the sensors have omni-directional antenna.

3.3 Design Procedure for Collaborative Contour Detection

The proposed scheme for efficient monitoring of the spatial distribution uses detection of a set of contours of the signal distribution and then applying a two-dimensional (*2D*) reconstruction method. Contour detection is the first step in modelling the spatial distribution. The quality of the signal monitoring, i.e. the error between the reconstructed signal and the actual spatial distribution, is related to the quality of detected contour levels. We propose a distributed collaborative processing algorithm for accurately detecting contours from noisy sensor observations under the stated assumptions. The performance of the algorithm for detection of contour lines is evaluated using computer simulations.

3.3.1 Contour Detection

We assume a scenario where a large number of wireless sensor nodes are randomly distributed in a given area of interest. Each sensor can obtain periodic observations of the signal in the sensor field, such as the temperature distribution over the given area.

A contour in the sensor field is defined as the line of separation between regions that are above and below a signal level S_0 . For the sake of estimating errors in estimation, we use a tolerance margin Δ to determine the thickness of the contour level (see figure 3.2.a). which specifies the region near the contour line such that a sensor node located in the region is termed as a *contour node*. In boundary detection, a tolerance planar distance r similar to [21] is used to determine the boundary thickness. This is illustrated in Figure 3.2.b.

3.3.2 Collaborative Processing Scheme for Spatial Contour Detection

The sensor nodes perform local and collaborative processing of their observed signal

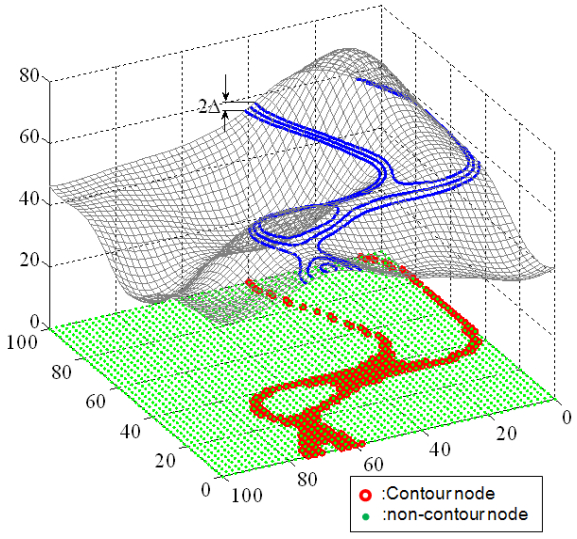


Figure 3.2.a: Illustration of a contour in a sensor field and contour nodes near it.

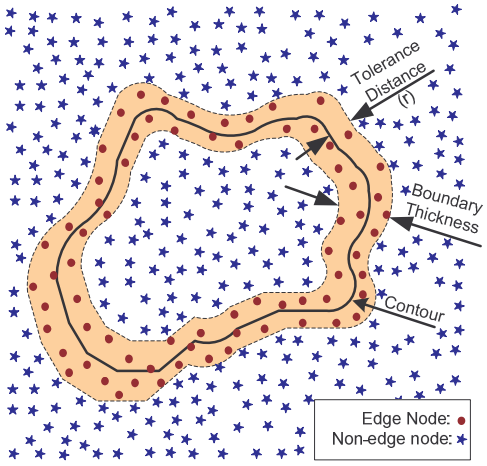


Figure 3.2.b: Illustration of a boundary in a sensor field.

samples to determine if they are contour node nodes. The followings are the key challenges for performing this task:

- Firstly, true distribution of the signal across the contour line may not have a sharp change, which makes the traditional contour detection algorithms ineffective. In this sense, our contour detection problem is different from contour detection.
- Secondly, quantization noise and other noise sources introduce additional scope of errors.
- Thirdly, unlike pixels of an image, the sensing nodes are usually distributed randomly. This demands some modifications on classical contour detection process.

We propose a collaborative processing scheme to address the challenges. In addition, we aim to keep the number of transmissions for collaboration to a minimum.

With these objectives of reduction of contour detection error and costs, a two-stage process is proposed for the detection of contour nodes near a contour in the sensor field [53]. In the first stage, all nodes periodically use their local observations to decide if they are probable contour nodes. In the second stage, nodes that test positive in the first stage

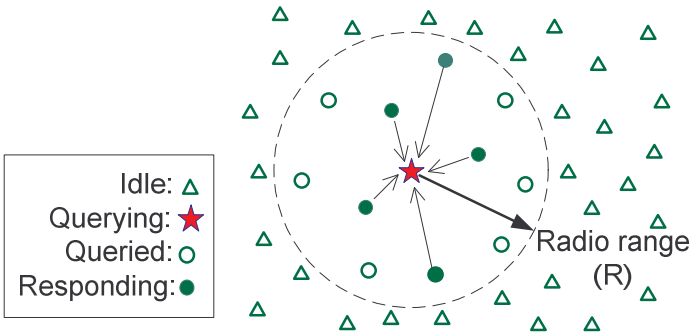


Figure 3.3: Illustration of the collaborative processing scheme

transmit query messages to their neighbours and process the returned information using a filter-based algorithm to confirm their decisions (see Figure 3-3). Details of the process are described in the following:

Stage 1: Determination of candidate contour nodes

- Initially each node determines if it is a probable contour node by comparing its shifted observation sample $F(s)$ within the two ranges of quantizer, *-MAX* and *MAX* (see Figure 3.4):

$$if \mid F(s) - S_0 \mid < MAX \Rightarrow probable \;\; edge \;\; sensor$$

Stage 2: Collaborative processing for confirmation of contour nodes

- A probable contour node broadcasts a *query packet* to obtain the values of observations from its neighbors. Nodes that receive the query packets respond by sending their observation samples.
- After receiving the neighborhood observations, the probable edge sensor node maps them to a quantized value *(QV)* using a multi-level quantizer as depicted in

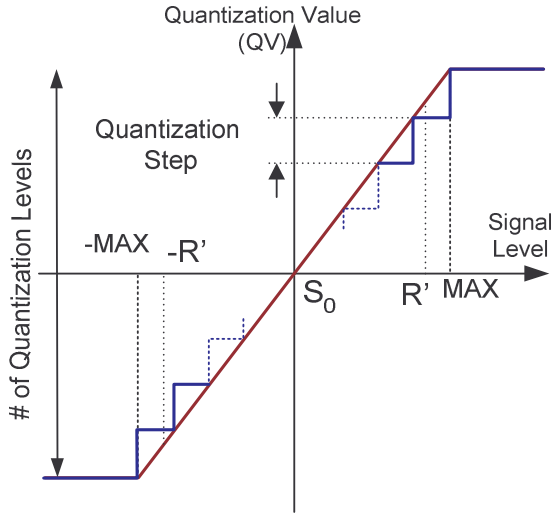


Figure 3.4: Illustration of multi-level quantization of the sensor signals.

Figure 3.4. Thus the probable edge sensor S_0 obtains a set of quantized observations $QV_S, S \in N(S_0)$, where $N(S)$ represents the neighborhood of node *S*.

- The sensor node applies a spatial filtering technique to the quantized neighbourhood samples to decide if it is an edge or contour node. This spatial filtering method results in a test that involves the generation of a decision variable $DV(S_0)$ and applying a threshold test:

$$\begin{aligned} &if\ DV(S_0) < \gamma_0 \quad \Rightarrow \quad edge\ sensor \\ &else \qquad\quad \Rightarrow \quad not\ edge\ sensor \end{aligned} \tag{3-1}$$

Details of the function of $DV(S_0)$ are described next.

3.3.3 Spatial Filtering

Spatial filtering has been widely used in image processing for edge detection applications. This is commonly accomplished by performing a spatial differentiation of the image field followed by a thresholding operation to determine points of steep amplitude change. Horizontal and vertical spatial derivatives are defined as [54]:

$$G_x(x,y) = \frac{\partial F(x,y)}{\partial x} \ , \ G_y(x,y) = \frac{\partial F(x,y)}{\partial y} \tag{3-2}$$

where $F(x,y)$ is the value of the signal at the point (x,y). With this, the gradient at the point (x,y) can be written as:

$$\nabla F(x,y) = \left|G_x(x,y)\right| + \left|G_y(x,y)\right| \tag{3-3}$$

As mentioned earlier, there are three major limitations for applying this edge detection method in sensor networks. We now describe some adaptations to the spatial filtering approach to address the three limitations for applying edge detection to our problem.

The issue of non-uniform sample points as well as that of unequal (and random) number of nodes that are expected to lie on either sides (above and below, right and left) of the querying node were solved by adding a weighting function for calculation of the gradient [21]. With this, we construct a weighted *Prewitt difference filter* at node S_0 where gradient vector components are defined as follows [53]:

$$\begin{aligned} G_{x\_diff}(s_0) &= \sum_{\forall s \in N(s_0)} W_x(x_s, y_s)[QV_{MAX} H_x(x_s, y_s) - QV_s] \\ G_{y\_diff}(s_0) &= \sum_{\forall s \in N(s_0)} W_y(x_s, y_s)[QV_{MAX} H_y(x_s, y_s) - QV_s] \end{aligned} \tag{3-4}$$

Here $W_x(.)$ and $W_y(.)$ are the weighting functions as introduced in [21] and are calculated as follows:

$$W_x(x_s, y_s) = \begin{cases} \dfrac{1}{n_{right}} & x_s > x_{s_0} \\ \dfrac{1}{n_{left}} & x_s < x_{s_0} \end{cases} \quad , \quad W_y(x_s, y_s) = \begin{cases} \dfrac{1}{n_{up}} & y_s > y_{s_0} \\ \dfrac{1}{n_{down}} & y_s < y_{s_0} \end{cases} \tag{3-5}$$

Here n_{left}, n_{right}, n_{up}, and n_{down} are the number of nodes in the neighborhood of the querying node s_0 to the left (i.e. $x_s < x_{s_0}$), right (i.e. $x_s > x_{s_0}$), above (i.e. $y_s > y_{s_0}$), and below (i.e. $y_s < y_{s_0}$) the node s_0, respectively. $H_x(.)$ and $H_y(.)$ are additive *virtual jumps* in the *x* and *y* directions, respectively, that create artificial sharp edges and improve the quality of contour detection [53]:

$$\begin{aligned} H_x(x, y) &= -1 \text{ if } x < x_0 \text{ , } 1 \text{ if } x < x_0 \\ H_y(x, y) &= -1 \text{ if } y < y_0 \text{ , } 1 \text{ if } y < y_0 \end{aligned} \tag{3-6}$$

and QV_s is the quantized value of the neighboring node. A scaling factor QV_{MAX} is required with $H_x(.)$ and $H_x(.)$ in equation (3-3) to take into account the multi-level quantization of the sensor values received from the neighboring nodes.

- With these, the decision variable for the test for detecting contour nodes is described as:

$$DV(s_0) = \left|G_{x\_diff}(s_0)\right| + \left|G_{y\_diff}(s_0)\right| \quad (3\text{-}7)$$

- The querying node processes all replies to obtain a decision variable $DV(s_0)$. It then decides if it is an contour node or not by performing a threshold test as follows. This decision condition is the same as (3-1).

$$\begin{aligned} &if\ DV(s) < \gamma_0 \quad \Rightarrow \quad edge\ sensor \\ &else \quad \Rightarrow \quad not\ edge\ sensor \end{aligned}$$

3.3.4 Considerations for Reducing Communication Cost

To improve the efficiency of the proposed contour detection algorithm in [53] we present two schemes to decrease the number of transmissions:

Scheme-1: Decreasing the number the possible contour nodes

To reduce the number of query packets and replies, the number of probable contour nodes is reduced at the cost of higher probability of missed detection. This is implemented by introducing a new parameter R', where $R' \le MAX$ (see Figure 3.3). This way, only those nodes for which $|F(s) - S_0| < R'$ are considered to be probable contour nodes. This scheme does not change step-size (accuracy) of quantization.

Scheme-2: Opportunistic neighbor listening (ONLi)

According to this scheme, each node responds to a query packet only once, assuming that neighboring nodes that already received their QV_s have saved it for future use for *Prewitt difference filtering*. Consequently, for each query packet, only those nodes respond that have not already sent their QV_s. This eliminates multiple transmissions of the same information from nodes in response to multiple query packets.

3.4 Performance Evaluation of Spatial Contour Detection with Simulations

In this section, we present the performance evaluation of the proposed collaborative contour detection scheme from computer simulation.

3.4.1 Performance Evaluation Measures

For the evaluation of the contour detection performance the following measures are defined:

i) Missed detection rate: This measure factor demonstrates the number of missed contour nodes which are in the *r* distance of the contour to the total actual contour nodes in the *r* distance of the contour, according to (3-8):

$$p_m = \frac{N_{Actual} - N_{Detected}}{N_{Actual}} \tag{3-8}$$

where N_{Actual} and $N_{Detected}$ are the number of actual and detected contour nodes, respectively.

ii) The total number of false detection of contour nodes: The total number of detected contour nodes that are out of *r* distance of the contour is the number of false detection of contour nodes.

iii) For the evaluation of the algorithm's cost, the average number of data packet transmissions for contour detection per node is calculated.

3.4.2 Performance Evaluation Results

In this section, we present results obtained from computer simulations to illustrate the performance of the proposed contour detection scheme. We assume a network of 2601 sensor nodes that are randomly and uniformly distributed on a 100x100 m^2 area. Each sensing node is equipped with an omni-directional antenna having a transmission range of 7 m. The tolerance radius is assumed to be 1 m. It is assumed that the data is gathered

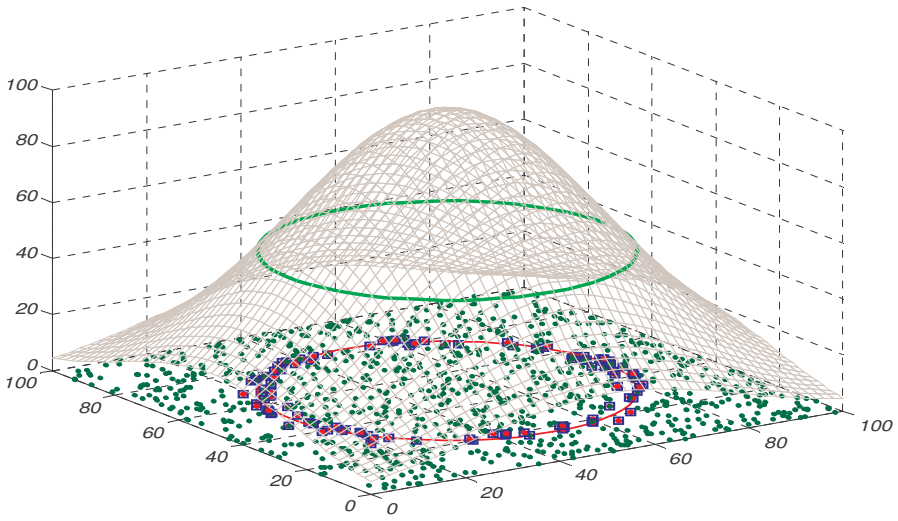


Figure 3.5: A snapshot of the outcome of contour detection using the proposed scheme depicting true contour nodes in blue squares and detected contour nodes in large red dots

in a single hop neighborhood of each node. We also assume that each node knows its coordinates and broadcasts its position to its neighbors.

For our simulations, we assume that the spatial signal distribution is a Gaussian in the sensor field, centered at (50,50) with a peak value of 100. Although a contour at a specified signal level with this signal distribution is a closed circle, it is worth mentioning that the proposed algorithm works equally well for non-closed contours as well. The noise in the sensor observations is assumed be Gaussian and independent from one sensor to another.

Figure 3.5 depicts the spatial distribution along with a snapshot of detected contour nodes from one of the simulation runs that were performed to detect a contour at signal level 50. The true contour nodes are marked by blue squares and the sensors that are detected by the proposed algorithm are marked in red. The rest of the nodes are marked by green dots.

We first evaluate the effect of Q (the number of quantization levels) and the value of *MAX* on the error performance of the proposed contour detection scheme by plotting the probability of missed detection against the decision threshold γ_0 for different Q and *MAX* values. The results, shown in Figure 3.6, indicate that multi-level quantization results in significant improvement in performance. The probability of missed detection drops noticeably when the number of quantization levels is increased from 2 to 8, however the relative improvement is less pronounced when it is increased to 16. The performance also improves with a higher value of *MAX*. Hence, we use Q=8 and *MAX*=10 for most of our other simulations. Note that our proposed scheme with binary quantization becomes similar to that presented in [21] when applied to contour detection. Hence, the results using multi-level quantization in Figure 3.6 also indicates the

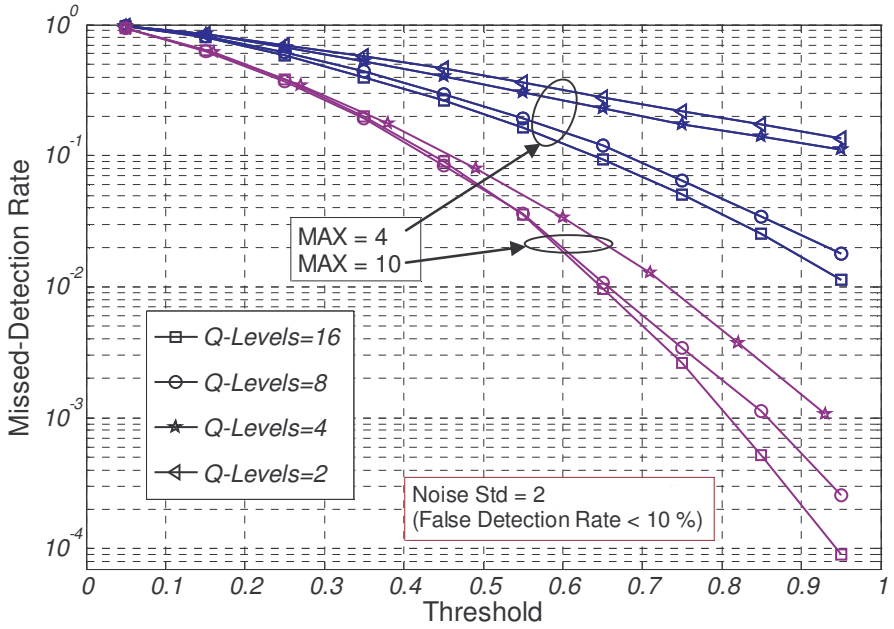


Figure 3.6: Missed detection rate for different number of quantization levels and *MAX* values.

comparative performance improvement obtained using the proposed scheme and that presented in [21], which is most related to this work.

We evaluate the effect of noise on the proposed contour detection scheme by determining the variation of the probability of missed detection and the number of falsely detected nodes at a contour level of 50 under different noise levels (Figure 3.7).

These results show that a higher amount of noise increase the probability of missed-detections but has negligible effect on the number of false detections. Note that there are detection errors even when there is no noise. This is explained from the fact that depending on the slope of the signal distribution at the contour threshold S_0, the set of nodes within the artificial edge created by our quantization process may not be exactly the same as those considered to be true contour nodes. The reason is that while the first set includes only those nodes whose signal values are within a certain range of the

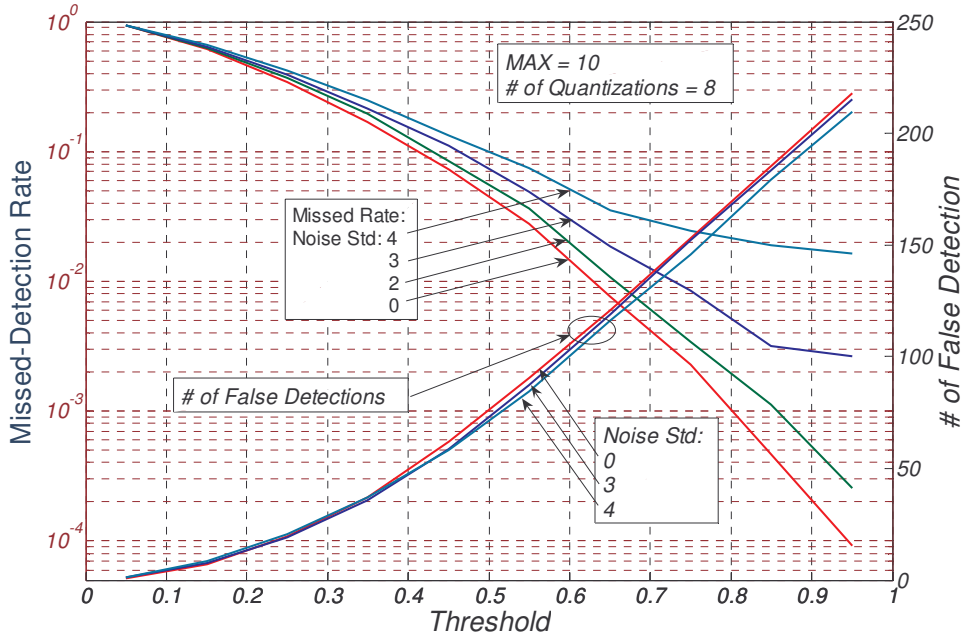


Figure 3.7: Missed detection rate and the number of falsely detected contour nodes at a contour level of 50 with varying thresholds γ_0.

contour threshold S_0, true contour nodes are defined by the tolerance distance *r*. Despite these apparent inconsistencies of detection errors with respect to noise, we still consider probability of missed detection and the number of false alarms to be indicative of the error performance of the proposed contour detection scheme. For instance, if the locations of the detected contour nodes are used to predict the location of the contour, higher missed detections and false detections both would contribute to a higher amount of error in the prediction. In that respect, our experiments indicate that false detections affect the prediction of a contour location more than missed detections. This is observed from the fact that average distance from detected contour nodes for *MAX*=10 and *Q*=8 is found to be 0.63 when the threshold $\gamma_0 = 0.25$, where the probability of missed detection p_m = 0.367 and the number of false detections N_F = 18.44. However, the average distance increases to 1.44 at $\gamma_0 = 0.85$, where the probability of missed detection p_m = 0.001 and number of false detections N_F =185.4.

The above findings provide the main motivation for reducing the number of probable contour nodes using the proposed *Scheme-1* described in section 3.3.3. We note that while a smaller value of R' will reduce the number of transmissions, it can also increase the probability of missed detection.

To evaluate this effect, we obtain the average distances of the detected contour nodes from the true contour as well as the communication cost (determined by the number of packet transmissions) as obtained for specific set of parameters, as shown in Table 3.1. The results show that although a small value of R' generates a high level of missed detections, the mean distance error is still low for $R' = 0.2$ *MAX*. On the other hand, this value of R' reduces the number of transmissions by a factor of 5.

Table 3.1. Error distance and communication cost vs. *R'*
MAX = 10, Number of Quantization level = 8, Threshold = 0.55, Noise Std = 2

$\frac{R'}{MAX}$	p_m	N_F	Distance Error		Comm. Cost
			Mean	Standard Deviation	
0.2	0.413	25.15	0.745	0.528	1.13
0.4	0.117	56.86	0.86	0.565	2.28
0.6	0.045	79.26	0.969	0.613	3.39
0.8	0.0367	85.74	0.95	0.572	4.54
1.0	0.0357	88.5	1.01	0.71	5.66

Finally, we evaluate the savings in communication cost obtained by using the proposed *ONLi* scheme. Table 3.2 shows the average number of transmissions in the network, normalized to the total number of nodes in the network that were required for contour detection with and without using the *ONLi* scheme.

The results indicate that avoiding multiple transmissions in response to query packets can reduce the number of transmissions to about 12% of that without using *ONLi*.

3.5 Conclusion

We proposed to detect and use contour lines in wireless sensor networks for monitoring spatial distributions of signals. A collaborative processing scheme is presented for detecting contours of the signal distribution of the sensor field. The proposed scheme uses a multi-level quantizer for emulating an edge in the signal distribution in the sensor field and then applies spatial filtering. Appropriate design

Table 3.2. Average number of transmissions
(Noise Std = 2, Threshold: 0.55, *R' = MAX*)

MAX	With *ONLi*	Without *ONLi*
10	0.73	5.68
7	0.56	3.9

considerations are presented to apply a spatial *Prewitt* filter to distributed data processing in sensor networks. The proposed scheme reduces error in contour detection caused by noise in signal observations and incurs a low cost of communication. Overall, this scheme can vastly reduce the number of transmissions that would be required to estimate the spatial distribution of the signal over a large area using a wireless sensor network by using contour detection. The filter-based approach presented in this chapter can also be used for tracking the temporal variations of signal distributions with small number of transmissions.

Motivated by these aspects, we explore design features of contour-based Spatio-temporal monitoring in the next chapters.

- In the next chapter we present and discuss spatial monitoring where spatial design considerations and performance issues on using contour-based monitoring are presented.
- As a continuation of contour detection in static case, in chapter 5 of this dissertation, an efficient algorithm for tracking the direction and deformation of contours using localized computations and collaborative processing in wireless sensor networks is presented, evaluated and discussed.

CHAPTER 4: PERFORMANCE ISSUES OF SPATIAL MONITORING USING CONTOUR DETECTION

4.1 Introduction

Modeling of a spatial distribution from its contours (see Figure 4.1) is widely used in various applications such as geographical information systems (*GIS*), medical imaging, visualization, etc. For simplicity or perhaps because of the lack of interest, in these applications the contour levels are usually assumed uniformly spaced. However, a number of performance measures can be optimized by the selection of appropriate (non-uniform) levels for the contour lines that are used to describe a spatial distribution.

In wireless sensor networks a critical performance issue is the energy consumption, which majorly depends on the communication cost in the network. Since the detected edge nodes representing contours need to be converged to the base station for the purpose

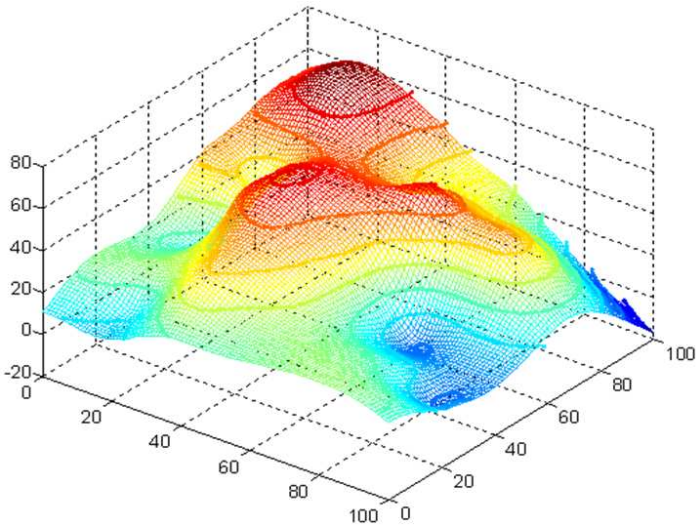


Figure 4.1: Modeling a correlated spatial distribution with its contours

of reconstructing the spatial signal distribution, the communication cost for the proposed contour-based spatial monitoring technique largely depends on the number of levels used. Hence for the given number of levels, it is worthwhile to investigate how they can be optimized, i.e. where the contours are obtained, to minimize the error in reconstructing the spatial signal distribution from the contours. This is the main goal of this chapter.

4.2 Problem Statement and Objective

The problem of finding the M non-uniformly spaced contour levels $\{\ell_i\}_{i=1}^{M}$ for monitoring a spatial distribution can be solved by using one of the following two approaches:

- The M contour levels $\{\ell_i\}_{i=1}^{M}$ are to be found so that for a given acceptable contour detection cost, the reconstruction error is minimized.
- The M contour levels $\{\ell_i\}_{i=1}^{M}$ are to be found so that for a given reconstruction error, the total number of contour nodes is minimized.

For the first approach, the celebrated *Lloyd-Max* approach's closed form solution can be used [51]. This needs the probability density function (*pdf*) of the signal strength of spatial distribution.

The second approach also requires the knowledge of the *pdf* of the signal distribution to minimize the total number of the nodes involved in contour detection. If this *pdf* is assumed known and equal to $f_S(s)$, equation (4-1) shows the total number of the actual contour nodes in Δ margin of the M contour levels $\{\ell_i\}_{i=1}^{M}$ in the sensor field with total N_0 sensors in the field.

$$N = N_0 \sum_{i=1}^{M} \int_{\ell_i - \Delta}^{\ell_i + \Delta} f_S(x)\, dx \approx \sum_{i=1}^{M} 2\Delta\, N_0\, f_S(\ell_i) \tag{4-1}$$

Figure 4.2 illustrates these marginal nodes for a single contour. Besides the *pdf*, this approach also may need local information of the spatial distribution. As observed, the solutions of the approaches require the knowledge of the *pdf* of the signal which is a complex assumption. We consider the first approach as it needs less information. Furthermore, we introduce a heuristic non-uniform level selection scheme. The proposed scheme is suboptimal, but performs better than uniformly selected results, as observed from our analysis.

To summarize, our objective is to find M non-uniformly spaced contour levels $\{\ell_i\}_{i=1}^{M}$, to minimize the reconstruction error of the spatial distribution with small number of transmissions (cost). In addition, efficiently applying this solution to monitoring the spatial distribution of signals using multiple iso-contours imposes additional design issues, which are the main focus of this chapter. These design issues are related to

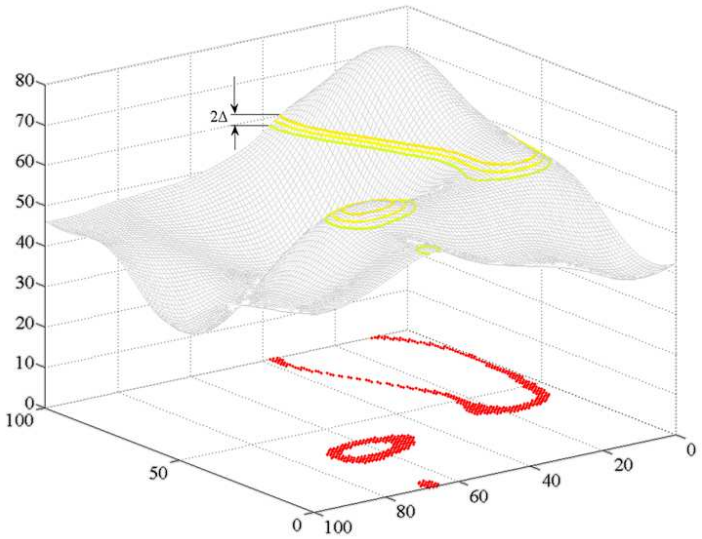


Figure 4.2: Illustration of the marginal contour nodes in Δ margin of a given contour

selection of appropriate parameters in the wireless sensors that are related to the contour detection scheme presented in chapter 3, with the objective of applying contour detection to the selected non-uniformly spaced levels to minimize reconstruction error and monitoring cost (number of transmissions). In particular, we consider the design of the following parameters under the above objectives:

- Radio range or collaboration range
- Contour detection threshold
- Quantization levels
- Collaboration margin

These are to be selected under various levels of noise and cost (i.e. the number of transmissions) considerations to minimize the monitoring error.

4.3 Non-uniformly Spaced Contour Levels for Spatial Monitoring

Finding the *M* non-uniformly spaced contour levels to minimize the reconstruction error of a spatial distribution for monitoring require the knowledge of the characteristics of the signal. The probability density function (*pdf*) is among these characteristics which are not easy to achieve practically. In this section, the requirement of finding the optimal *M* contour levels is reviewed and then a heuristic, practical approach in finding the *M* contour levels is proposed and discussed. Throughout this chapter we assumed that the signal strength of spatial distribution is stationary.

4.3.1 Optimization of Contour Levels under when the Signal's *pdf* is Known

Lloyd and Max separately showed that assuming *M* levels, there is an optimal set of levels which minimize the flat-top reconstruction error of the signal distribution

(quantization distortion) [51]. These optimal levels for a spatial distribution with *pdf* $f_X(x)$ in the known range (L_0, L_{M+1}), is obtained by solving the set of *M* equations (4-2).

$$L_i = \frac{\int_{y_i}^{y_{i+1}} x f_X(x)dx}{\int_{y_i}^{y_{i+1}} f_X(x)dx}, \quad i = 1,2,...,M \qquad y_i = \frac{L_i + L_{i-1}}{2}, \; i = 1,2,...,M+1 \tag{4-2}$$

We use these optimum levels as a benchmark for evaluating the performance of our proposed sub-optimal solution for level selection that does not require knowledge of the *pdf* of the signal.

4.3.2 A Heuristic Practical Solution for Selecting Non-uniform Contour Levels

As *a-priori* knowledge of signal's *pdf* may not be available, we propose a practical scheme that is based on the following basic principles:

1. In general, the most frequently seen levels of the signal are the best sampling (sensing) levels. At these levels the signal is sensed more frequently. This implies that the samples of the signal which are obtained at these levels provide higher information about the signal. In other words, the statistical modes of the *pdf* are the best candidates of the sampling levels.
2. As the spatial distribution is continuous in its range, by getting far from the statistical mode, the probability of having the levels becomes gradually less. Accordingly, the spacing between the levels with less frequency of happening should be more than the ones with higher frequencies.

Based on these heuristic principles, we consider non-uniformly spaced contour levels where the levels are chosen using a scheme similar to a logarithmic (*μ-law*) expander for speech coding. Equation (4-3) defines the logarithmic levels for the case where the

statistical-mode is in the middle of the range and the statistics of the signal is symmetric about the mode. In (4-3), $\{U_i\}_{i=1}^{M}$ are the uniformly spaced levels in the signal range (L_0, L_{M+1}), and $\{\mu_i\}_{i=1}^{M}$ are the *μ-law* based levels in the same range.

$$U_i = \frac{2(\frac{L_{M+1}-L_0}{M} i + L_0 - \frac{L_{M+1}+L_0}{2})}{L_{M+1}-L_0} \qquad (4\text{-}3)$$

$$\mu_i = \left[\frac{\mathrm{sgn}(U_i)}{\mu} \left[(1+U_i)^{\mu} - 1 \right] \right] \frac{L_{M+1}-L_0}{2} + \frac{L_{M+1}+L_0}{2} \quad , \quad i = 1,2,...,M$$

If the statistical mode of the signal is not in the middle of the signal range, scaling should be applied before applying *μ-law*.

4.4 Error of Modeling the Spatial Distribution Using Contour Detection

The error between the reconstructed signal from a set of contours and the original signal depends on the accuracy of the contour detection scheme, the number of contours, and the selected set of contour levels. However, a critical aspect for error measurement is the method applied for reconstructing the spatial distribution from its detected contours. A large volume of work has been reported on reconstruction algorithms of two-dimensional signals which vary in complexity and reconstruction performance. Among these researches, two and multi-dimensional field reconstruction from randomly distributed sensor observations in the absence and presence of noise were discussed using several analytical techniques such as matched-filter, linear minimum mean square of error, and zero-forcing in [57-61].

Before we present the design considerations of the contour detection schemes for minimizing the reconstruction error and communication cost, we select an appropriate reconstruction method that is tractable and yet accurately captures the important parameters that affect the error in signal reconstruction from a detected set of contours.

In the previous chapter, we presented the performance of the proposed collaborative contour detection scheme using false detections and missed detections of the contour nodes. Although these measures are appropriate for evaluating the effectiveness of the contour detection algorithm, they do not directly determine the reconstruction error of the two-dimensional reconstruction method. It is clear that a larger number of false detections increase the contours' thickness and higher missed detections might lead to loss of information about parts of the contours. However, it is hard to quantitatively relate false and missed detections to the error in signal reconstruction. Here our objectives are to evaluate the performance of different level-selection policies and the optimization of the parameters of the contour detection algorithm (as mentioned din chapter 4.2) for spatial monitoring, i.e. how well the estimated (or reconstructed) signal distribution matches the actual spatial distribution of the signal. To achieve these ends, we apply a first order linear spline method for reconstructing the two-dimensional signal from its contours, and use the root mean-square of error (*RMSE*) to evaluate the performance of the proposed contour based spatial estimation method.

Although other reconstruction tools can provide smaller errors, we show that the proposed simple reconstruction tool captures the effect of both false and missed detections. This is illustrated in Figure 4.3.a, where a one-dimensional (linear) sensor network is assumed. In Figure 4.3.a the sensors are over the d axis and the goal is to detect the sensors for which the observation samples are at the or near the specified levels $\{\ell_i\}_{i=1}^{5}$. In this fictitious scenario, we assume a detection process that is similar to the discussed contour detection process is applied. By applying the point detection algorithm, a few sensors might be detected corresponding to each level, or no sensors are detected

for a few levels due to missed-detection. In this figure, the detected sensors are shown with dark cross-marks over the d axis. The value of each level is assigned to its corresponding detected sensors. These values are shown with light gray cross-marks at the same location of the sensor. The reconstructed (one-dimensional) signal using the proposed linear spline method is obtained by connecting the light gray cross-marks using

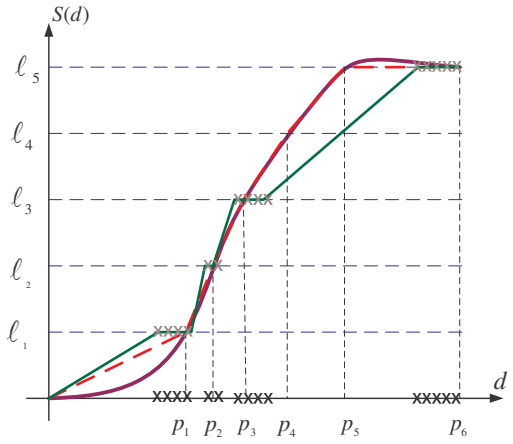


Figure 4.3.a: Illustration of the flat-top linear reconstruction scheme. It is observed that false detections (extra crosses) and missed detections (levels) both increase the reconstruction error when using this scheme.

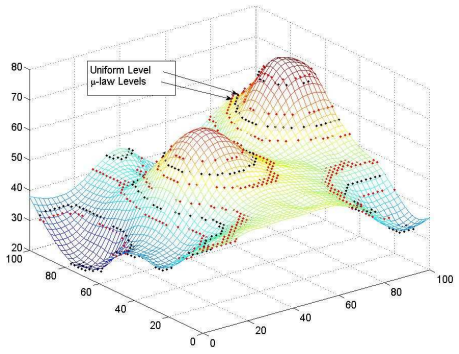


Figure 4.3.b: A two-dimensional correlated spatial distribution with 6 uniform and non-uniformly spaced levels.

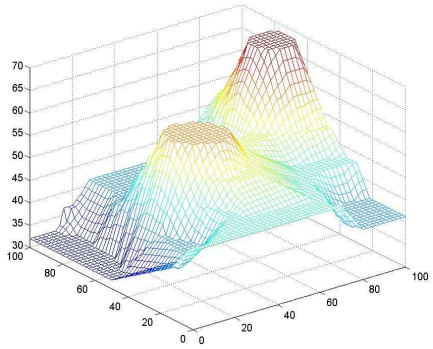


Figure 4.3.c: The reconstruction of the two-dimensional correlated spatial distribution with the introduced method from μ-law based levels.

a piece-wise linear graph.

Note that the example in Figure 4.3.a includes missed detections at locations p_4 and p_5, and several false detections at all other level-crossings locations. It is clear that the reconstructed signal using the linear spline model suffers from additional error from both these events. Missed and false detections cause the linear approximation to derivate further from the curve, thereby increasing the *RMSE*. Hence we are satisfied that the first order spline can be appropriate in capturing missed and false detections of the contour sensors as well as the effect of other parameters that may introduce errors in spatial monitoring using the proposed contour detection method.

Illustration of the reconstruction of a two-dimensional signal distribution using this method is shown in Figure 4.3.b and c. Figure 4.3.b shows a correlated spatial distribution with two kinds of contour level spacing and Figure 4.3.c shows the corresponding reconstructed signal distribution using the proposed first-order spline from the detected levels at non-uniformly spaced contour levels based on μ-law.

4.5 Performance Evaluation and of Optimization of Parameters for Spatial Monitoring

The performance evaluation results of the proposed contour detection monitoring scheme with the first-order spline is presented and discussed in this section. In this evaluation, computer based Monte-Carlo simulations using are used.

4.5.1 Generation of a Correlated Spatial Distribution for Simulations

So far many approaches have been proposed for spatial distribution modeling. Most of these models were proposed for spatial distribution of animals. Some recent categories of spatial distribution modeling approaches were reviewed in [67]. To model the correlated spatial signal distribution, two approaches have been potentially used in the

previous researches. These approaches are *Markovian* model [62,63], and *Diffusion process* model [64-66].

Using a rigorous mathematical procedure, Markovian model was described in [62,63] and a simple practical method for extraction of the model parameters from real traces was proposed. The remote sensing data was used for real traces data. The efficient way of generation of the synthetic traces on a given topology using the model parameters was discussed. The model was verified by statistically comparing synthetic and real data. The performance of the model was compared with the algorithms whose behavior depends on the degree of spatial correlation in data for real and synthetic data. The comparison showed that the proposed Makovian model is more accurate and general than the used jointly Gaussian model.

In diffusion model, the effect of the distributed sources, which are randomly located in the area, reduces inversely proportional to a power of distance to the source. Various exponent values were reported in different researches. The final spatial distribution is the result of superposition of the effect of sources throughout the distribution.

We use the diffusion model for performance evaluations of the spatial distribution monitoring using contours because of its simplicity. Accordingly we consider randomly positioned point sources with random amplitudes where the effect of each source is modeled as a Gaussian distribution. The superposition of the distributed sources produces the spatially correlated distribution. For the sake of evaluating the effect of *pdf* of the spatial distribution, two groups of correlated spatial distributions with Gaussian-like and Laplace-like *pdf* (see Figure 4.4) are assumed each with 150 different sample spatial distributions with the same *pdf*. To have a unique statistical characteristic for all sample

spatial distributions, a number of principal sources and a larger number of minor sources are assumed, each with constant standard deviation. All of the simulations were repeated for these two saved groups of correlated spatial distributions.

For the Gaussian-like *pdf*, 60 principal Gaussian sources with standard deviation 13, and uniformly distributed, random amplitudes in the range (-50,50) along with 500 minor Gaussian sources with standard deviation 5 and random amplitude in the range (-1,1) are used. For the Laplace-like *pdf*, 20 principal Gaussian sources with standard deviation 13, and uniformly distributed, random amplitudes in the range (-50,50) along with 200 minor Gaussian sources with standard deviation 2 and random amplitude in the range (-1,1) are used. Finally, the spatial distributions are scaled to locate in the range interval (-50, 50) and then are shifted by 50.

4.5.2 Simulation Assumptions

The same assumptions that were made in chapter 3.2.2 for single contour detection are made here for performance evaluation of spatial distribution using multi-contour detection. The only difference is that in the present chapter the nodes are located over a

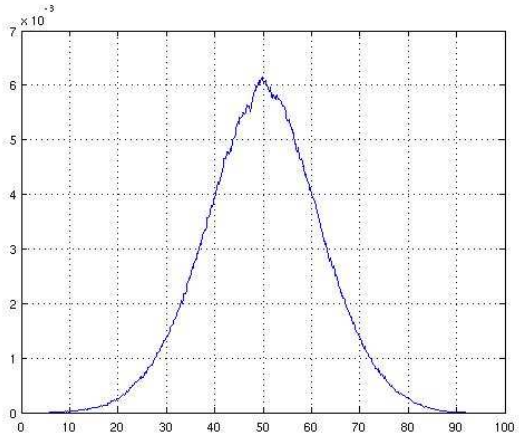


Figure 4.4.a: Gaussian-like *pdf*

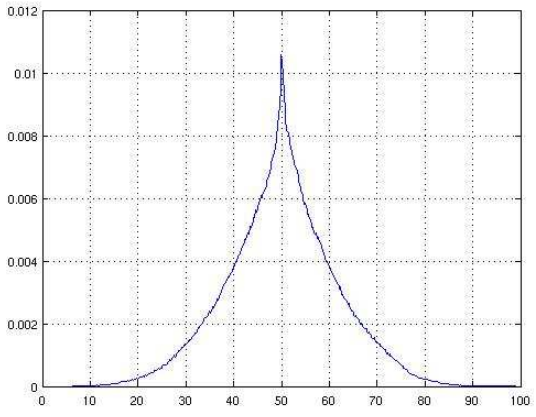


Figure 4.4.b: Laplace-like *pdf*

grid with grid size of 2. This assumption is for removing the interpolation error effect.

4.5.3 Simulation Results

We now present the performance of the proposed spatial monitoring scheme using collaborative contour detection. We evaluate the performance of the scheme using the proposed μ-law based non-uniform level selection method and compare it with that obtained using uniform levels as well as optimal levels (*Lloyd-Max* solution).

To evaluate the effectiveness of collaborative processing, we compare the performance results of the proposed scheme with one in which contour detection is done non-collaboratively. In non-collaborative contour detection, we assume that have sensor observations within $\pm\delta_{nc}$ of the contour levels to be contour nodes. Hence, non-collaborative contour detection does not need any filtering or communications for the detection of contour nodes.

The simulations were repeated for spatial distributions with the Gaussian-like and Laplace-like *pdf*. Since performance results with both *pdf*s were almost the same, just the

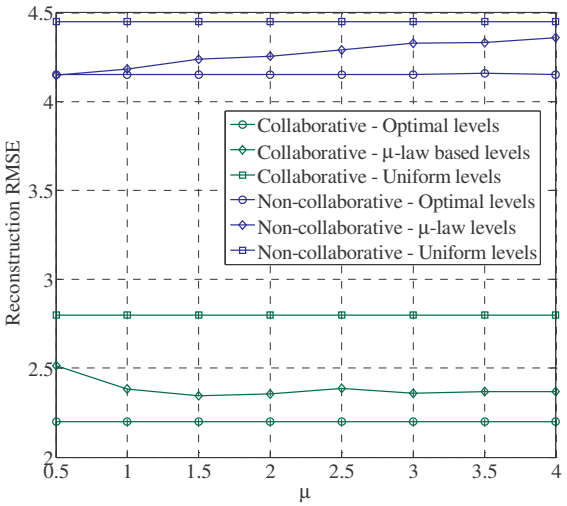


Figure 4.5.a: Variation of monitoring performance in μ-law based scheme versus μ.

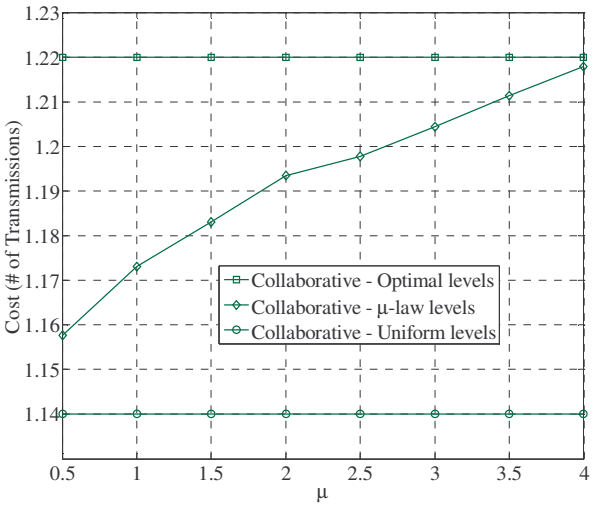


Figure 4.5.b: Variation of monitoring cost in μ-law based scheme versus μ (L = 7).

simulation results related to the spatial distribution with Gaussian-like *pdf* are presented here.

4.5.3.1 Optimal μ

In our study of the performance and cost of spatial monitoring with contour detection with μ-law based level selection scheme, the μ value was chosen equal to 3. The performance and cost of μ-law based approach changes with μ. These variations happen because μ is a key parameter in definition of the spacing between levels. Figure 4.5.a and 4.5.b show the variation of monitoring *RMSE* and cost versus μ. As observed from these figures, by increasing the μ value, the monitoring performance with collaborative detection and non-collaborative detection are improved and degraded, respectively. The detection cost of collaborative scheme also increases with μ according to Figure 4.11.b.

4.5.3.2 Optimal Margin δ_{nc} in Non-collaborative Contour Detection

In non-collaborative contour detection, δ_{nc} margin is selected based on having a few nodes in the margin of the contours. Increasing this margin causes higher final

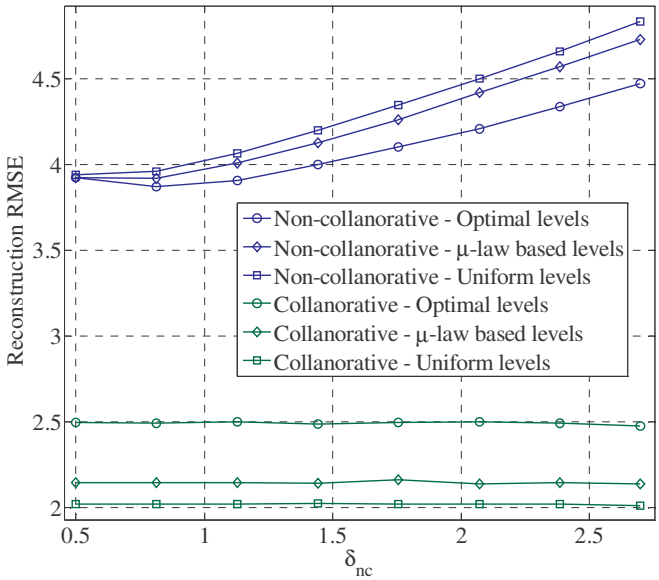


Figure 4.6: Study of the effect of variation of δ_{nc} on the performance of monitoring with non-collaborative contour detection (L = 8)

reconstruction error, and limitation on the number of contour levels in non-collaborative scheme. Decreasing the δ_{nc} margin also leads to missing the contour nodes, specifically when the observation noise is strong.

Figure 4.6 depicts the effect of the variation of non-collaborative margin δ_{nc} on the performance of spatial monitoring with contours. As this figure shows, there is an optimal δ_{nc} which optimizes the monitoring *RMSE*.

4.5.3.3 Effect of the Radio Range (Collaboration Range)

Collaboration range is the radio range (*R*) of the wireless sensors. This parameter is assumed to be the same for all sensor nodes (*SN*). From the signal processing angle, *R* is equivalent to the window size of the *Prewitt* difference filter that is used for spatial filtering. The window size is one of the parameters which have to be adjusted for proper performance

Figure 4.7.a illustrates the variation of the reconstruction *RMSE* with respect to the radio range *R,* from which we make the following observations:

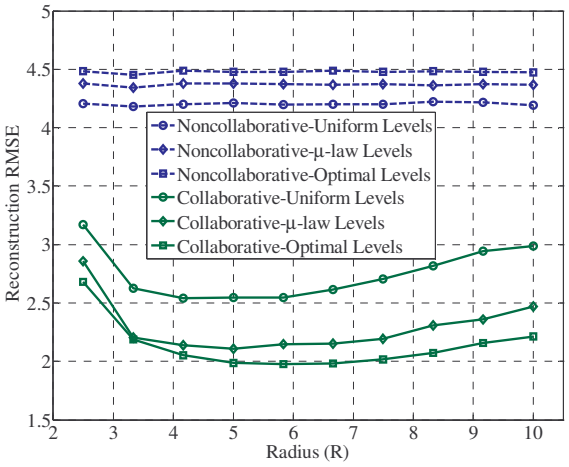


Figure 4.7.a: Variation of the reconstruction *RMSE* versus Threshold

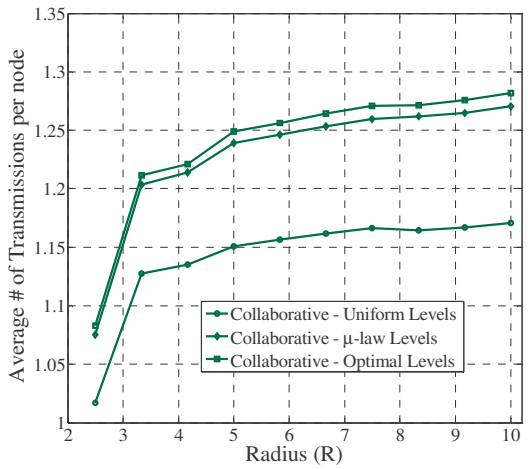


Figure 4.7.b: Variation of the Average Cost versus Threshold

Simulation Parameters: $\sigma = 5$, $\delta = 2.7$, $\mu=3$, $R = 5$, $\delta_{nc} = 2.0$, $L = 8$, $Q=3$

- There is a significant benefit of using collaboration for contour detection. This important is particularly heavy for noisy conditions such as that shown in Figure 4.7.
- The performance of the proposed with μ-law based contour level selection is better than uniformly spaced levels and is close to the case where the contour levels are optimally spaced.
- There is an optimal range of R for which the reconstruction *RMSE* is lowest. If R is made too small or too large, the *RMSE* increases.

The average contour detection cost is illustrated in figure 4.7.b. This cost is calculated by the average number of packet transmissions per node for an edge detection operation. This figure shows that:

- As the collaborative range (R) increases, the cost of the collaborative detection also increases.
- Monitoring the spatial distribution using μ-law based contour levels has more average detection cost in comparison to that with uniformly spaced contour levels and a bit less than with the optimally spaced contour levels.

4.5.3.4 Effect of the Contour Detection Threshold

In chapter 3, it was shown that the performance of the detected contours is a function of detection threshold. If the detection threshold is very small, lots of contour nodes will be missed, and if it is too large, the contour thickness will be too big which will increase the *RMSE*. Intuitively, an optimal detection threshold is expected to minimize the *RMSE*.

Figure 4.8.a presents the reconstruction error with varying detection threshold (γ_0). For comparison, we also plot the results using the non-collaborative scheme. We make the following observations from this figure:

- There is an optimal detection threshold for collaborative contour detection. Interestingly, this optimal threshold is the same or approximately the same for all level selection schemes.
- The *RMSE* using the proposed μ-law base levels model is much better than that using uniformly spaced levels. It is almost as good as that of optimally spaced levels.
- When the detection threshold is inappropriate, monitoring with non-uniformly spaced contour levels can be poorer than that with uniformly spaced levels.

Figure 4.8.b compares the average detection cost of monitoring the spatial distribution using collaborative contour detection for the three kinds of the defined levels: uniform, μ-law, and optimal. This figure shows that:

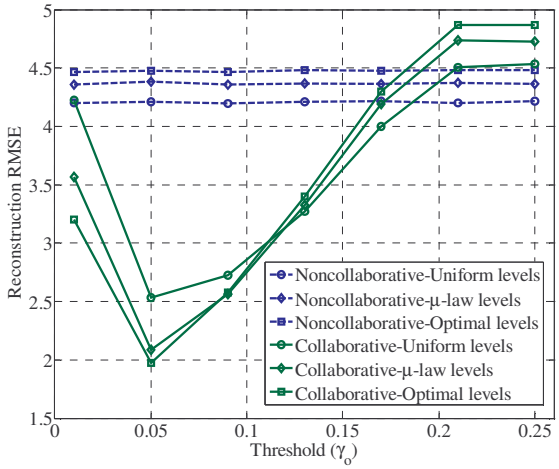


Figure 4.8.a: Variation of the reconstruction *RMSE* versus Threshold

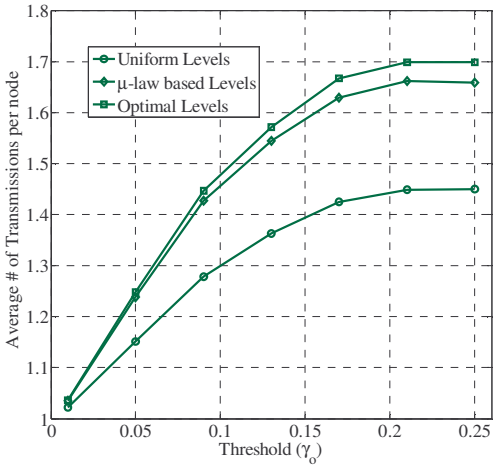


Figure 4.8.b: Variation of the average number of transmissions vs. Threshold

Simulation Parameters: $\sigma = 5$, $\delta = 2.7$, $\mu = 3$, $R = 5$, $\delta_{nc} = 2.0$, $L = 8$, $Q = 3$

- The cost of monitoring using μ-law based contour levels is more than that with uniformly spaced contour levels and a bit less than the cost of optimally spaced levels.
- As the detection threshold increases, the cost increases and after a specific limit, the average cost saturates.

4.5.3.5 Effect of Quantization Levels

Quantization as a technique to reduce the effect of additive noise and simplification of the signal model may improve the performance of the signal extraction in the proposed technique. Performance evaluation of the contour detection for small amount of additive noise approves the performance improvement due to increase of number of quantization levels. By increasing the quantization step over noise intensity, the performance of the proposed approach is expected to become poorer.

Figure 4.9.a compares the reconstruction *RMSE* using the three kinds of level selection schemes with varying number of quantization levels. These results have been obtained at

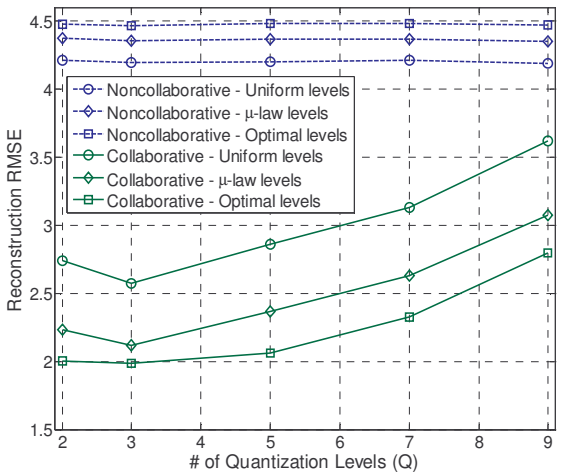


Figure 4.9.a: Variation of reconstruction *RMSE* versus # of quantization levels

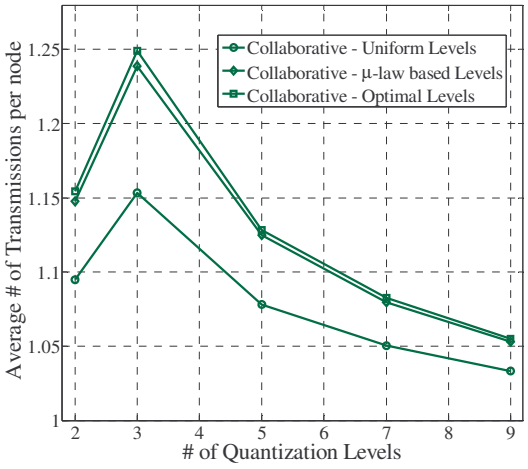


Figure 4.9.b: Variation of the average number of transmissions vs. # of quantization levels

Simulation Parameters: $\gamma = 0.05$, $\delta = 2.7$, $\sigma = 5$, $\mu = 3$, $R = 5$, $\delta_{nc} = 2.0$, $L = 8$

a low *SNR* (noise σ=5). This figure shows that:

- The monitoring performance in terms of the reconstruction *RMSE* is best when the number of quantization levels in multi-level quantization contour detection algorithm is 3. It is interesting to note that a higher number of quantization levels do not improve the performance.
- Monitoring performance of spatial monitoring with μ-law based contour levels is better than that with uniform contour levels and poorer than that with optimally spaced contour levels.

Figure 4.9.b compares the average contour detection cost for modeling the spatial distribution with any of the three kinds of the defined levels. This figure shows that:

- By increasing the number of quantization levels the cost decreases. This can be justified by the increase of missed-detection rate caused by noise. This cost reduction is due to the fact that after detection of the contour nodes, each detected contour node notifies their neighborhood of the final status of the detected contour. Then the increase of the missed detection rate results in the decrease of the cost.
- The cost of monitoring with μ-law based levels is very close to optimally spaced levels and much more than uniform scheme.

4.5.3.6 Variation of Collaboration Margin

In the proposed contour detection approach, the probable contour sensors send query packets to their neighborhood for collaboration and final decision on being contour nodes. Performance evaluation for the circular contour in chapter 3 showed that by

increasing the collaboration margin (*R'*) the quality of the detected contour is affected (less missed detections and more false detections).

For detection of a number of contours for modeling the spatial distribution, increasing the collaboration margin is limited to the separation between the levels. By increasing the collaboration margin the margins of the two neighboring levels can get merged, which may cause confusion of the detected contour nodes. Also, increasing the collaboration margin increases the communication cost. Then it is expected that the quality of signal distribution modeling using multiple contours is limited by the level definition scheme and also the maximum affordable cost.

Figure 4.10.a compares the performance of monitoring with collaborative detection with the three level definition schemes with varying collaboration margins. This figure shows that increasing the number of participating nodes in collaboration improves the performance of monitoring, i.e. reduces the reconstruction *RMSE*. This is due to the reduction of the missed detection rate.

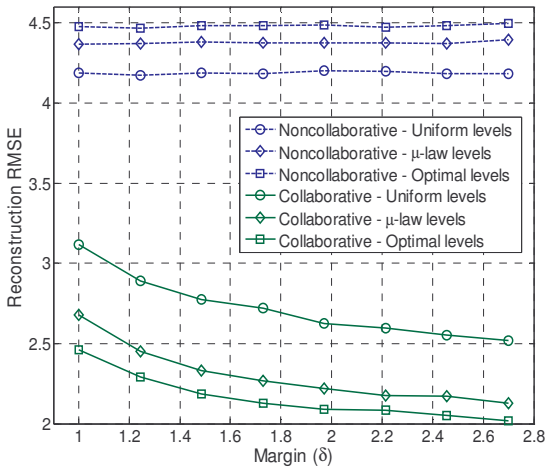


Figure 4.10.a: Variation of reconstruction *RMSE* versus collaboration margin

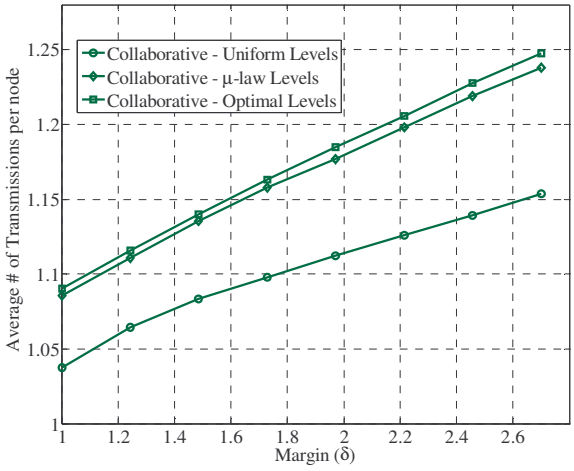


Figure 4.10.b: Variation of the average # of transmissions vs. collaboration margin

Simulation Parameters: γ=0.05, δ= 2.7,σ = 5, μ=3, R = 5, δ_{nc}= 2.0, Q=3

Figure 4.10.b compares the contour detection cost. This figure shows that increasing the collaboration margin increases the cost of the collaborative contour detection algorithm almost linearly. The order of the cost of the schemes is also the same as that it was with variation of the previously discussed parameters like R, Q, δ, γ.

4.4.3.7 Effect of the Number of Contour Levels

The monitoring performance of spatial distribution is expected to improve as the number of contour levels increases. This is observed in Figure 4.11.a, which compares the variation of the reconstruction *RMSE* versus the number of contour levels for the three schemes of definition of contour levels in collaborative detection, and non-collaborative detection. These results indicate that with increasing the number of contour levels, the monitoring performance of the collaborative scheme is improved. This degrades the performance of non-collaborative scheme slightly especially when non-uniformly spaced contour levels are used. This can be justified by considering the effect of noise. Increasing the number of levels in non-uniformly spaced level scheme means the level spacing becomes smaller. Since non-collaborative uses no filtering scheme, the

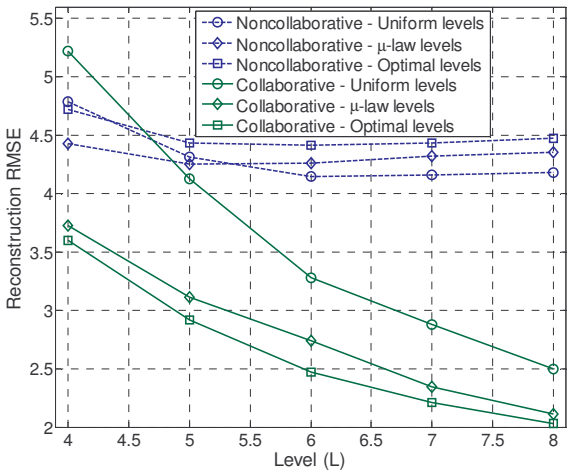


Figure 4.11.a: Variation of reconstruction *RMSE* versus # of levels

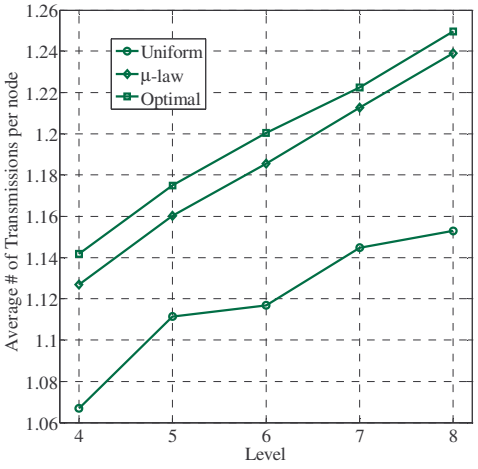


Figure 4.11.b: Variation of the average # of transmissions vs. # of levels

Simulation Parameters: γ=0.05, δ= 2.7,σ = 5, μ=3, R = 5, δ_{nc}= 2.0, Q=3

effect of noise causes selection of the contour nodes of the neighboring level by mistake.

Figure 4.11.b compares the average cost of detection of contour levels with collaboration. This figure shows that the monitoring cost with contours using μ-law based scheme is between monitoring cost with optimally spaced scheme and monitoring cost with uniformly spaced scheme.

A key design issue is that under a monitoring performance constraint (monitoring *RMSE*), which of the level definition schemes is the most cost efficient scheme?

Figure 4.11.c answers this question. This figure depicts the number of levels as well as the cost for different *RMSE* values for different levels selection schemes. It is observed that the desired monitoring *RMSE* is smaller than 2.55, uniformly spaced contour levels is not a proper solution and among μ-law and optimal solution, μ-law which has smaller cost is beneficial. If the maximum acceptable *RMSE* is more than 2.55, then uniformly spaced levels with smaller cost is the desired solution.

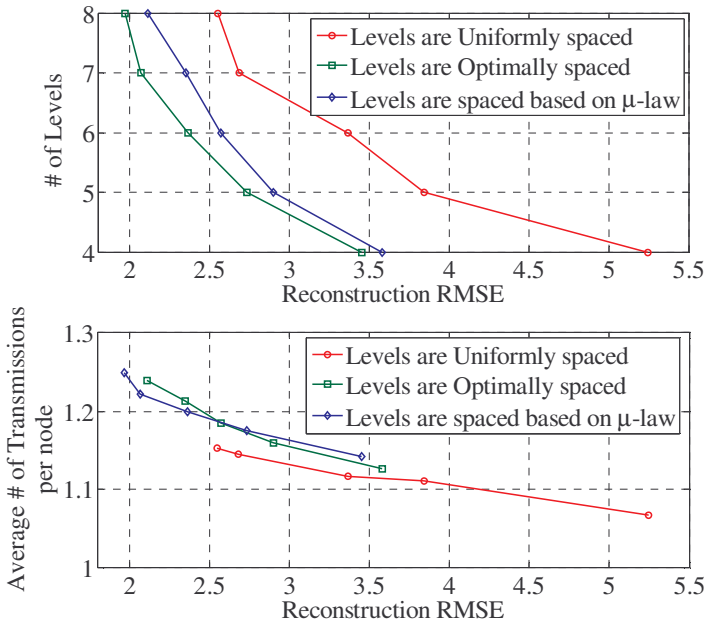


Figure 4.11.c: Study of the number of required levels and the number o transmissions vs. the monitoring performance (*RMSE*)

4.5.3.8 Effect of the Observation Noise Power

When the noise power increases, the performance of the modeling is expected to become poorer. Collaborative detection helps to improve the robustness of the spatial monitoring to the observation noise. Figure 4.12.a compares the monitoring performance of collaborative detection scheme with non-collaborative scheme under different values of σ, the standard deviation of noise. This figure has the following results:

- In comparison to non-collaborative scheme, spatial monitoring with the collaborative scheme is less affected by noise.
- The order of monitoring performance with contours using collaborative contour detection is the same that it was before: μ-law scheme has better performance than that with uniformly spaced scheme and poorer than that with optimum scheme.
- Monitoring the spatial distribution using non-collaborative contour detection with non-uniformly spaced contour levels has poorer performance than that with uniformly spaced contour levels.

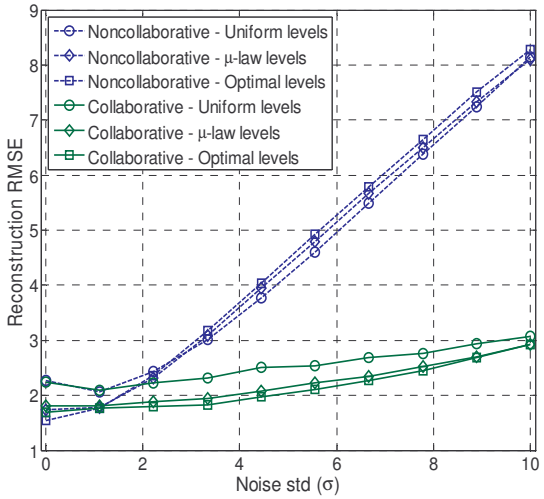


Figure 4.12.a: Variation of reconstruction *RMSE* vs. noise std (σ)

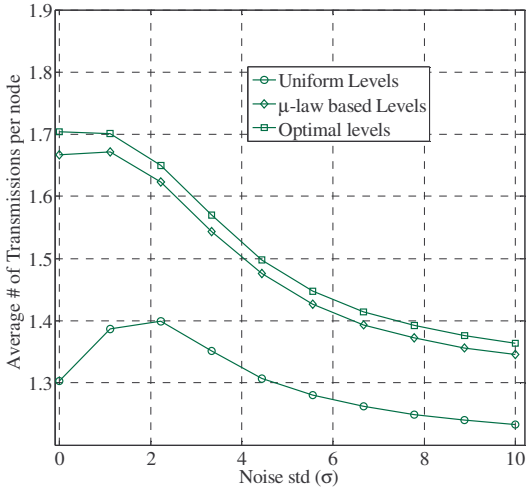


Figure 4.12.b: Variation of the average # of transmissions vs. noise std (σ)

Simulation Parameters: γ=0.05, δ= 2.7, μ=3, δ_{nc}= 2.0, L = 8, R = 5,Q = 3

Figure 4.12.b compares the average cost of monitoring with collaborative contour detection for the three levels definition schemes. As observed from this figure, the communication cost decreases with increasing values of the noise power. This can be justified by the same reasoning as described for Figure 4.12.b, that is, the missed-detection rate increases with increasing noise power, consequently, the total number of communications after detection is decreased.

When the noise intensity is small, the number of false detections is small, and the cost due to this type of nodes is also small. The increasing rate of false detections causes a peak in monitoring cost with contours using uniformly spaced contours. This peak shows that in monitoring using uniformly spaced contour levels, false detection rate dominates the missed detection rate in heavily noisy signal conditions.

4.9 Conclusion

The monitoring of a random correlated spatial distribution using a large-scale wireless sensor network by detecting its contours at different levels is addressed. The collaborative signal processing scheme introduced in chapter 3 is applied for extraction of the contours. It is assumed that sensor observations are heavily affected by noise. The proposed collaborative processing algorithms avoid the communication cost that would be incurred by obtaining periodic samples from all sensors in the network. To evaluate the effect of contour detection errors, a simple piece-wise linear approach for error calculation is proposed.

Performance issues of the proposed algorithm for estimating spatial distributions based on the extracted contours under various detection parameters is explored. It is shown that the spatial distribution is monitored more accurately when collaborative

detection with proper detection parameters is employed. The performance of monitoring using the tractable heuristic contour level definition approach based on μ-law is compared with the performance of monitoring using uniformly spaced and optimally spaced (*Lloyd-Max*) contour levels. It is shown that the μ-law based approach is a suboptimal solution.

CHAPTER 5: CONTOUR TRACKING ALGORITHM FOR TEMPORAL MONITORING OF SPATIAL DISTRIBUTION IN WIRELESS SENSOR NETWORK

5.1 Introduction

Spatial monitoring of correlated distributions with contours was discussed in chapter 4. To monitor the spatial distribution over time, the contour detection algorithm may be repeated over time. However, periodically repeating the contour detection algorithms in all parts of the old contour to repair the broken parts can be unnecessarily costly. In many situations the spatial distribution of a signal undergoes only partial variations of a contour as depicted in Figure 5.1. For example, if the application involves the monitoring the surface temperature of a geographical region, the slow changes of the distribution from one sample instant to another may include small changes of contours. This motivates the development of contour tracking algorithm to monitor the temporal variation of a signal

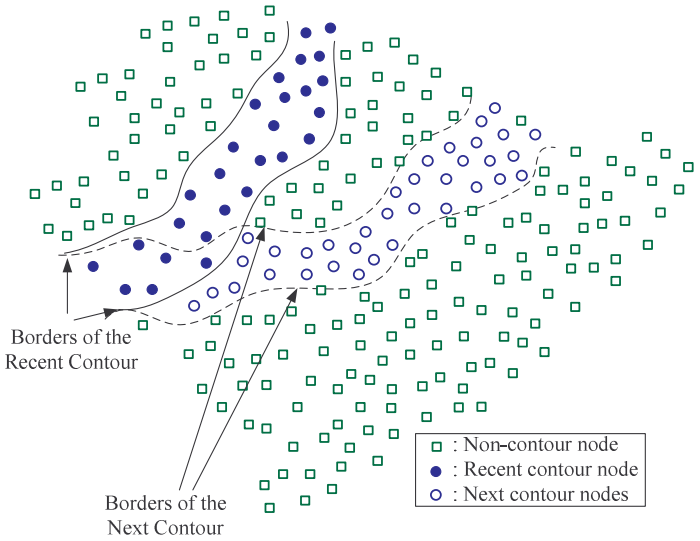


Figure 5.1: Temporal variation of the contour of a spatial distribution

distribution to determine incremental changes of existing contour without repeating the spatial monitoring algorithm.

Recently a few algorithms have been proposed for contour tracking using static wireless sensor networks [37-39]. In a general view, these algorithms, which have been reviewed in chapter 2 of this dissertation, have the following features:

- All of these algorithms are based on periodically repeating a procedure
- Most of them concentrate on repairing the broken parts of the contour
- Most of them ignore the effect of observation noise

Usually in wireless sensor network, the sensor nodes are tiny and cheap computers with limited computational capabilities that have been equipped with a few sensors and radio. Often it is not desirable to use of sophisticated wireless sensors with sensing capabilities like radar as proposed for contour tracking in [37].

Contour detection in static wireless sensor network that was discussed in chapter 3 is a baseline approach for modeling a correlated spatial distribution with contours. Modeling correlated signal distributions with contours is an approach that was discussed in chapter 4 for spatial monitoring application in sensor field.

Development of an algorithm to efficiently track the variations of the broken pieces existing contours of a correlated spatial distribution is the goal of this chapter.

5.2 Problem Statement

Temporal variation of spatial distributions results in change of its contours which are needed to be detected over time. Periodically repeating the contour detection algorithms for tracking the temporal variations of the contours depletes the in-network energy. These variations might happen in limited parts of a few contours in a short time interval.

Meaningful correlation between the relevant pieces of the same contour over time heuristically supports that *a-priori* information from the past contour nodes can be used to efficiently update the broken contour pieces. This idea has been already used by employing *Kalman* filtering scheme in [37] for contour tracking using range sensors.

In this chapter an efficient distributed collaborative contour tracking algorithm for application in dense wireless sensor network is proposed and its performance is evaluated using computer simulations. Efficient space and mechanisms that work over time are added to the algorithm to adaptively and efficiently track the changes.

The performance of the tracking algorithm with the space, time and space-time cost reduction schemes are evaluated for different speeds. This evaluation includes the following scenarios:

- The contour is tracked when the spatial distribution is *shifted* with constant speed
- The contour is tracked when part of the spatial distribution is varied

5.3 Collaborative Processing for Contour Tracking in Wireless Sensor Network

5.3.1 Basic Contour Tracking Protocol

The basic contour tracking algorithm is based on a step-by-step tracking of the contours at periodic instants. In the course of time as the spatial distribution changes, its contours are needed to be updated. We assume that at periodic intervals of time the most recent contour nodes, which are at the variation region, apply the contour detection test. Those contour nodes that fail the test by broadcasting a query to their immediate neighborhood notify their neighboring nodes to apply the contour detection test. Then the notified neighbors apply the contour detection test to check if they are contour nodes in

Figure 5.2: The basic contour tracking protocol

```
// To check if the contour has changed from its last state
* Contour node(i) applies the detection algorithm
      {
       Send_query();    // to the immediate neighborhood
       receive_query_replies();
       map_to_the_quantizer();
       r = prewitt_filtering_output();
       comparing_with_threshold(r,γ );
     }
// In case the past contour node fail in the test
       Request_the_neighborhood_to_query();
// neighboring node perform the process as marked (*)
       Apply_contour_detection_process();
       Send settlement packet to the neighborhood();
```

the new time instant. This procedure is repeated periodically in the network. This basic protocol is summarized in Figure 5.2.

This protocol may be somewhat more efficient than periodically repeating the contour detection procedure when the rate of variation of the contours is low, but it can still be fairly costly. Firstly, many of the notified nodes that apply the test and fail, could have been avoided by space filtering schemes. Secondly, the protocol repeats the procedure periodically, but the period of repetition is hard to determine. Hence, to make sure that the protocol tracks the pace of the variations of the spatial distribution, the network need to extravagantly spend energy. Hence, a filtering approach that uses the most appropriate times is required to reduce the number of transmissions.

To adapt this protocol to the requirements of wireless sensor network, two cost reduction schemes are proposed and evaluated using simulations. These schemes are: space oriented cost reduction schemes, and time oriented cost reduction schemes. In the next subsections of this chapter, these efficient schemes are described.

5.3.2 Space-Oriented Cost Efficient Schemes

To avoid excessive expenditure of in-network energy, a major portion of the notified sensors must stop applying the contour detection test. In this section we show that by applying a proper space-oriented filtering, a subset of the neighboring nodes can be notified as the next possible contour nodes.

5.3.2.1 The Gradient Compass (*GC*) Mechanism

GC algorithm is essentially a derivative from the spatial contour detection algorithm. For better description of this algorithm, we revisit equation (3-4) and rewrite it according to (5-1). In this equation, the horizontal and vertical components of the difference gradient vector are decomposed into two parts, each of which are related to the nodes in the two lateral semi-planes of right and left, or up and down of the probing node.

$$\begin{aligned} G_{x\_diff}(s_0) &= \underbrace{\left\{ \sum_{\forall s \in N(s_0): x > x_{s_0}} \frac{1}{n_{right}} [QV_{Max} - QV_s] \right\}}_{DV_{right}} + \underbrace{\left\{ \sum_{\forall s \in N(s_0): x < x_{s_0}} \frac{1}{n_{left}} [-QV_{Max} - QV_s] \right\}}_{DV_{left}} \\ G_{y\_diff}(s_0) &= \underbrace{\left\{ \sum_{\forall s \in N(s_0): y > y_{s_0}} \frac{1}{n_{up}} [QV_{Max} - QV_s] \right\}}_{DV_{up}} + \underbrace{\left\{ \sum_{\forall s \in N(s_0): y < y_{s_0}} \frac{1}{n_{down}} [-QV_{Max} - QV_s] \right\}}_{DV_{down}} \end{aligned} \tag{5-1}$$

The spatial gradient components in equation (5-1) can be rewritten according to (5-2).

$$\begin{aligned} G_{x\_diff}(s_0) &= DV_{right} + DV_{left} \\ G_{y\_diff}(s_0) &= DV_{up} + DV_{down} \end{aligned} \tag{5-2}$$

Note that the DV_{right} and DV_{up} terms in (5-1) are always positive, and the DV_{left} and DV_{down} are always negative. Accordingly to our contour detection algorithm, a node is accepted as a contour node when $|G_{x\_diff}(s_0)| + |G_{y\_diff}(s_0)|$ and consequently $|G_{x\_diff}(s_0)|$, $|G_{y\_diff}(s_0)|$ are not more than the given detection threshold γ_0.

When a contour changes shape, a given past contour node may not be a contour node anymore, and the variations amount of DV_{right}, DV_{up}, DV_{left} and DV_{down} show the direction of the movement of the contour. This information, although useful, but is likely to be inaccurate and designing a space-filtering based on it may not be very effective. For a better result, we re-write the *DV* values to a new form and continue the rest of the discussion based on it for a better result.

$$DV_{right} = QV_{Max} - \frac{1}{n_{right}} \sum_{i=1}^{n_{right}} \iint_{\forall s \in N(s_0): x > x_{s_0}} f(x,y)\delta(x - x_i, y - y_i)dxdy$$
$$DV_{left} = -QV_{Max} - \frac{1}{n_{left}} \sum_{i=1}^{n_{left}} \iint_{\forall s \in N(s_0): x < x_{s_0}} f(x,y)\delta(x - x_i, y - y_i)dxdy \quad (5\text{-}3)$$

In (5-3), $f(x,y)$ is the QV_s value of the node S and $\delta(x - x_i, y - y_i)$ is the two-dimensional *Dirac delta* function. Using (5-3), the horizontal component of the gradient in (5-1) can be re-written as follows:

$$G_{x\_diff}(s_0) = -\frac{1}{n_{right}} \sum_{i=1}^{n_{right}} \iint_{\forall s \in N(s_0): x > x_{s_0}} f(x,y)\delta(x - x_i, y - y_i) - \frac{1}{n_{left}} \sum_{i=1}^{n_{left}} \iint_{\forall s \in N(s_0): x < x_{s_0}} f(x,y)\delta(x - x_i, y - y_i) \quad (5\text{-}4)$$

The integration interval is the domain of calculation of the gradient which is illustrated in Figure 5.3. If the sensor nodes are distributed like a net and the areas of the

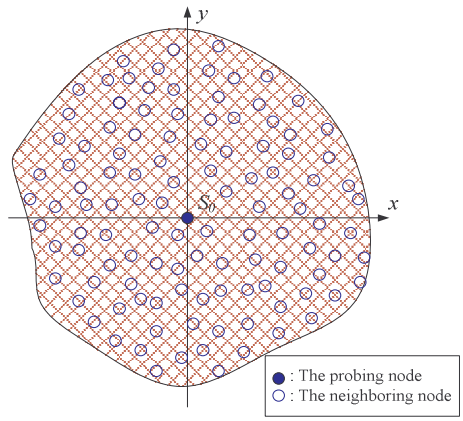


Figure 5.3: Domain of gradient calculation in contour detection

gradient domain in the four quadrants are also assumed to be the same, then n_{right} and n_{left} should be the same and equal to an assumed n. Then (5-4) changes to (5-5).

$$G_{x\_diff}(s_0) = \frac{-1}{n}\sum_{i=1}^{n}\left[\iint_{\forall s\in N(s_0):x>x_{s_0}} f(x,y)\delta(x-x_i, y-y_i)dxdy + \iint_{\forall s\in N(s_0):x<x_{s_0}} f(x,y)\delta(x-x_i, y-y_i)dxdy\right] \quad (5\text{-}5)$$

By windowing the gradient domain with an embedded rectangular window centered at the probing node S_0 (see Figure 5.4), the integration range is given based on x_0 and y_0, which is half of the size of the sides of the rectangular window. With these assumptions, (5-5) is simplified to (5-6). In this equation, the probing node's position is assumed to be the origin. In equation (5-6), we also include the time information.

$$G_{x\_diff}(t_0, s_0) = \frac{-1}{n}\sum_{i=1}^{n}\int_{-y_0}^{y_0}\int_{-x_0}^{x_0} f(x,y)\delta(x-x_i, y-y_i)dxdy \quad (5\text{-}6)$$

In the study of the tracking problem, let us first consider the case where the spatial distribution shifts to the right for ℓ_0. With this shift, $f(x,y)$ (or QV_s value) in (5-6) shifts with the same amount. Equation (5-7) shows the horizontal gradient component after shift.

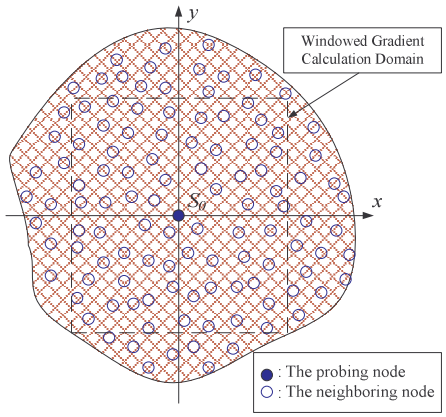


Figure 5.4: Windowed domain of gradient calculation in contour detection

$$G_{x\_diff}(t_1, s_0) = \frac{-1}{n} \sum_{i=1}^{n} \int_{-y_0}^{y_0} \int_{-x_0}^{x_0} f(x - \ell_0, y) \delta(x - x_i, y - y_i) dxdy \tag{5-7}$$

After simple manipulations, we obtain the following from equation (5-7).

$$G_{x\_diff}(t_1, s_0) = \frac{-1}{n} \sum_{i=1}^{n} \int_{-y_0}^{y_0} \int_{-x_0-\ell_0}^{x_0-\ell_0} f(x, y) \delta(x + \ell_0 - x_i, y - y_i) dxdy \tag{5-8}$$

By subtracting (5-6) from (5-8), the variation of the horizontal component of the gradient is achieved.

$$\Delta G_{x\_diff}(\Delta t, s_0) = G_{x\_diff}(t_1, s_0) - G_{x\_diff}(t_0, s_0) =$$
$$\frac{-1}{n} \sum_{i=1}^{n} \int_{-y_0}^{y_0} \int_{-x_0-\ell_0}^{x_0-\ell_0} f(x, y) \delta(x + \ell_0 - x_i, y - y_i) dxdy + \frac{1}{n} \sum_{i=1}^{n} \int_{-y_0}^{y_0} \int_{-x_0}^{x_0} f(x, y) \delta(x - x_i, y - y_i) dxdy \tag{5-9}$$

Because of the net distribution assumption of the nodes, the *Dirac delta* function is dropped from (5-9). Then, the variation amount of the horizontal component of the gradient after shift, with a few simple manipulations changes to (5-10).

$$\Delta G_{x\_diff}(\Delta t, s_0) = \frac{-1}{A} \int_{-y_0}^{y_0} \int_{-x_0-\ell_0}^{-x_0} f(x, y) dxdy + \frac{1}{A} \int_{-y_0}^{y_0} \int_{x_0-\ell_0}^{x_0} f(x, y) dxdy \tag{5-10}$$

The A value in the denominator of (5-10) is the half area of the rectangular window. The value of (5-10) for small values of the displacement amount ℓ_0 and when $f(x,y)$ is monotonic, is proportinal to the amount of displacement ℓ_0. The monotonic assumption is usually a valid assumption, provided the radio range is selected properly and the node density is large enough.

With the same reasoning, the variation amount of the vertical component of the gradient after shift is obtained as is shown in (5-11).

$$\Delta G_{y\_diff}(\Delta t, s_0) = \frac{-1}{A} \int_{-x_0}^{x_0} \int_{-y_0-\ell_0}^{-y_0} f(x, y) dxdy + \frac{1}{A} \int_{-x_0}^{x_0} \int_{y_0-\ell_0}^{y_0} f(x, y) dxdy \tag{5-11}$$

For simplicity, in the rest of this discussion we assume that rectangular window of gradient calculation is a square. This discussion results the following theorem.

Gradient Compass Theorem: In a dense wireless sensor network, the variation amount of the horizontal component of the gradient for a given node S_0 as defined in (5-6), when the spatial distribution shifts for ℓ_0, is proportional to the displacement amount when ℓ_0 is not large, provided that the gradient is windowed with an embedded sqaure, centered at the position of the S_0. A similar result is valid for the variation amount of the vertical component of the gradient.

As the nodes are uniformly distributed over the area, and the procedure of calculation of the vertical and horizontal components of the gradient are the same, it is expected that for the local area of a few sensors we have the following immediate result:

Corollary: When the displacement amount of the spatial distribution is not large, the variation amount of the horizontal and vertical components of the gradient are proportional to the displacement amount in their direction. In (5-12), Δx_0 and Δy_0 are the displacement amount in x and y direction, respectively.

$$\frac{\Delta G_{x\_diff}(\Delta t, s_0)}{\Delta G_{y\_diff}(\Delta t, s_0)} = \frac{\Delta x_0}{\Delta y_0} \tag{5-12}$$

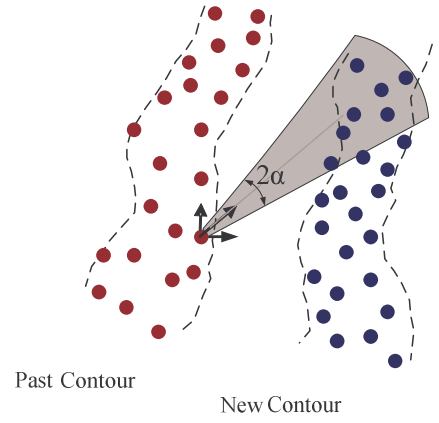


Figure 5.5: Prompting the neighboring nodes with noisy gradient in an angle interval in Gradient Compass (*GC*)

Accordingly, based on the localized information, each node can find the direction of the displacement of the spatial distribution in its local area. If the *x* and *y* components of the gradient are calculated based on the same scale, then the gradient vector's element works like a compass. In other words, $\overrightarrow{\Delta G}(s,t)=\begin{bmatrix}\Delta G_{x\_diff}(s,t)\\ \Delta G_{y\_diff}(s,t)\end{bmatrix}$ is the direction of movement of the contour, and consequently the most recent contour node should notify its neighboring nodes which are in the direction of the local variation of its gradient. This is illustrated in Figure 5.5.

Because of the noisy sensor observations, the QV_s values in (5-1) are noisy which affects the gradient values specifically for small contour variations. Hence in our contour tracking algorithm, we consider that the next possible contour nodes are determined selecting those nodes that lie within an angle α from the gradient-variation vector as illustrated in Figure 5.5 (the grey zone).

When the correlated spatial distribution experiences slow deformation instead of shift, the similar result is expected, and it is expected that by increasing the angle α, the effect of the deformation in the efficient algorithm is included.

To have a rough estimate of the amount of angle α, let us consider the case where the node density is ρ_n , the radio range is *R*, and the missed detection rate is p_m. Then the total number of nodes that are notified and are not missed in contour tracking process should be more than 1, i.e.

$$\frac{2\alpha}{2\pi}\pi R^2 \rho_{\mathrm{n}} (1 - p_m) > 1 \quad \Rightarrow \quad \alpha > \frac{1}{R^2 \rho_{\mathrm{n}} (1 - p_m)} \tag{5-13}$$

Equation (5-13) is an over-estimation of the angle α because most of the next possible contour nodes are notified by some of the recent contour nodes.

5.3.2.2 Estimating the Displacement (*ED*)

We can further reduce the communication cost (the number of transmissions) of the space-oriented contour tracking algorithm with *GC* still can be reduced by limiting the number of the possible next contour nodes. Because the nodes are randomly distributed and the exact statistical characteristics of the spatial distribution and its variation are not known, preparation of a reasonably exact statistical approach is not easily possible. Luckily, there are a few statistical inequalities like *Tchebycheff* and *Markov* inequalities which can be applied here.

Tchebycheff inequality: For any random variable like *X*, and for any assumed constants *a*, *ε*, and positive constant *n*, the following inequality is valid [52].

$$\Pr ob\{|X-a|>\varepsilon\} \le \frac{1}{\varepsilon^n} E\{|X-a|^n\} \tag{5-14}$$

For our specific application, this inequality is modified according to (5-15).

$$\Pr ob\{|X-\overline{X}|>k\sigma\} \le \frac{1}{\kappa^2} \tag{5-15}$$

Equation (5-15) states that the probability of having any contour nodes in the $\kappa\sigma$ distance from the average values $\overline{\Delta X_m}$ and $\overline{\Delta Y_m}$ is less than $\frac{1}{\kappa^2}$, where σ, $\overline{\Delta X_m}$ and $\overline{\Delta Y_m}$ are the standard deviation, the horizontal mean displacement and the vertical mean displacement of the contour nodes in the previous contour update, respectively.

For example by taking κ equal to 4, with probability of 0.9376 all of the next possible contour nodes are included in the range of $\overline{\Delta X_m} - 4\sigma_x < x < \overline{\Delta X_m} + 4\sigma_x$ and $\overline{\Delta Y_m} - 4\sigma_y < y < \overline{\Delta Y_m} + 4\sigma_y$ of the previous contour node.

The performance of the discussed contour tracking algorithm immediately depends on the performance of the previous step, and the selected set of parameters such as speed of update, the acceptance angle α, and the *Tchebycheff* κ value. In other words, all of the history of the contour tracking process is hidden in the most recent step. If the temporal variation of the spatial distribution is assumed stationary, the contour tracking can be modeled with a stationary *Markov* process. Accordingly, the statistical moments of the process at most recent step can be used in contour renewal process. With Ergodic assumption, *ED* uses the mean and variance of the most recent contour update to notify a selected group of the nodes in the next contour update to reduce the cost.

ED should be used along with *GC*, otherwise it tries a lot of irrelevant nodes. Firstly,

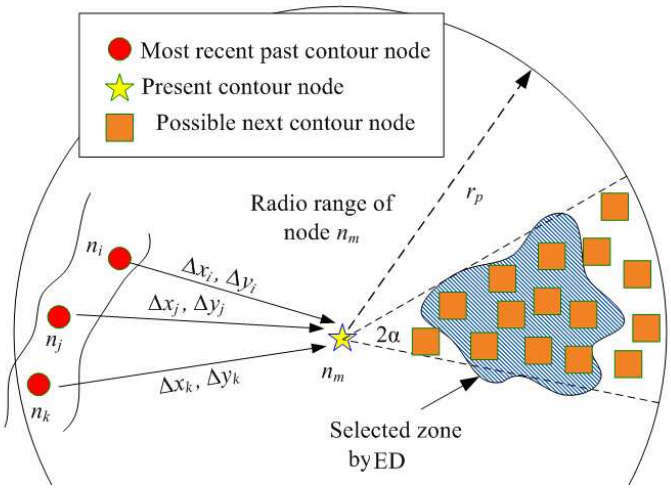


Figure 5.6: Illustration of *ED* for cost efficient contour detection

because of the erroneous calculation of the statistical moments of the past contour update. And secondly, because of the gradient compass theorem. Accordingly, wherever *ED* is used, *GC* is also included. Then cost reduction mechanism in *ED* is based on filtering a few of the possible next contour nodes which have been already selected as possible next contour nodes in *GC*. Figure 5.6 illustrates how *ED* reduces the cost of the algorithm by selectively reducing the number of the possible contour nodes.

5.3.3 Time-Oriented Cost Efficient Schemes

Space-oriented cost efficient schemes reduce the contour tracking cost by filtering out a large portion of the possible next contour nodes, but they do not give any information about the period of contour update. The rate of variation of the spatial distribution is not known to the sensors network. Therefore to track the changes, it is reasonable to use a high contour update rate, which excessively increases the cost (number of transmissions).

To solve this problem, in this section two time-oriented cost efficient schemes are proposed: adaptive time update (*ATU*), and normalized adaptive time update (*NATU*).

5.3.3.1 Adaptive Time Update (*ATU*)

The main concern is that the spatial distribution varies over time but its rate of variation is not already known. Periodically repeating the contour tracking algorithm for keeping the track of the contour is costly. To efficiently track the contours of a spatial distribution, the sensors should smartly estimate and adapt to the local trend of variation of the signal. *ATU* is a smart distributed scheme which lets the nodes decide when the contour update must happen. The following principle is the basis for *ATU*:

If the periodical sensing does not show tangible local variation of the signal, the contour tracking updates must be postponed. Accordingly, the contour nodes watch

the variation of the signal by in-place sensing and without performing any communications.

Because of the noisy sensor observations, the comparison between the consecutive samples is not useful to find out the variation trend of the contours. To alleviate the effect of noise, in-place averaging is proposed. Immediately after the contour update, each contour node computes a moving average on its own *m* periodically sensed observations and records this value η_0 to use it as reference level for comparison (see Figure 5.7). The period of in-place sensing depends on how fast the distribution changes. Then the contour nodes cyclically replace the periodical observations, apply moving average, and compare the result with the reference value η_0. At the time instant t_1, in which the difference between the moving average filter's output and the reference value η_0 becomes more than an assumed ξ, the relevant contour is locally updated.

$$| S(t_1) - S(t_0) | = | S(t_1) - \eta_0 | \geq \xi \qquad (5\text{-}16)$$

In (5-16), $S(t_1)$ is the in-place filtered sensor observation at time instant t_1 and $S(t_0)$ is the reference sensor observation at time instant t_0, close to when the node was selected as contour node.

5.3.3.2 Normalized Adaptive Time Update (*NATU*)

The performance of *ATU* highly depends on ξ that should be adjusted based on the

$$\eta_0 = \frac{1}{m} \sum_{i=1}^{m} v_i(t_0 + m\ \Delta t)$$

v_1	v_2	v_3	.	.	.	.	.	.	v_m

Figure 5.7: Frame of in-place moving average in *ATU*

local slope of the spatial distribution. To explain the problem, let us consider the case where after the time delay τ, the moving average filter's output value goes outside of $\pm\xi$ of the reference value η_0. If after time delay τ, the related contour has moved out of the radio range of the past contour node S_0, then the basic protocol is not able to track the contour and at least for that part of the contour, contour detection algorithm should be repeated. In other words the tracking algorithm has failed. Let us call it *tracking failure*. Figure 5.8, illustrates the tracking failure.

Normalization of the ξ to the average local variation of the spatial distribution in the transmission distance of the contour node can solves this issue. For this reason, the sensor nodes at the time instant of applying the contour detection test need to calculate the average variation of the signal in the scope of their radio range in the direction of the previous contour displacement. Figure 5.9 shows the formation of the semi-neighborhoods in the direction of the previous contour displacement for calculation of ξ. Equation (5-17) conservatively defines a value of ξ for *NATU*. ξ should be adaptively

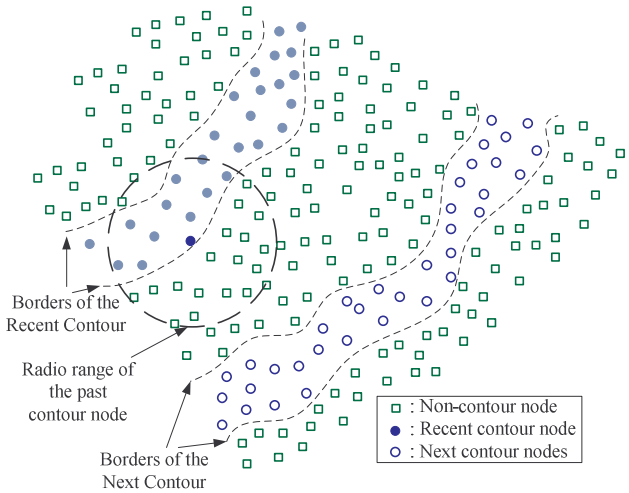


Figure 5.8: Tracking failure issue in *ATU*

selected based on the direction of the move. If the most recent direction of the move is chosen as an approximation for the present direction of the move, the half-planes in which the displacement direction is perpendicular to their separation line are the most appropriate ones. This is illustrated in Figure 5.9.

$$\xi = \left| \sum_{\forall s \in Semi-neighborhood\ \#1} \frac{1}{n_1} S_i(t_0) - \sum_{\forall s \in Semi-neighborhood\ \#2} \frac{1}{n_2} S_i(t_0) \right| \qquad (5\text{-}17)$$

In equation (5-17) the $S_i(t_0)$ values are the sensor observations of the neighboring sensors at time instant t_0, and n_1 and n_2 are the number of the nodes in each of the shown semi-neighborhoods in Figure 5.9.

5.3.4 Hybrid Space and Time Oriented Cost Reduction Schemes

In previous sections, the space-oriented and time-oriented cost reduction schemes were introduced. Basically, simultaneous application of these schemes for higher cost

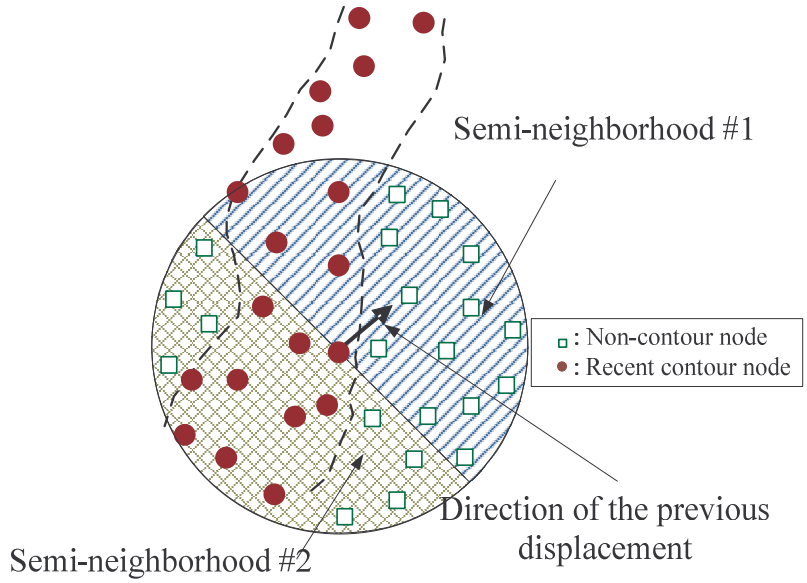


Figure 5.9: Formation of semi-neighborhood for calculation of ξ in *NATU*

reduction has no conflict. The only limiting issue is the performance of the combined scheme that sometimes is affected by the simultaneous imperfections of the schemes, like additional missed-detection rate. In section 5.4, performance of some of the cost efficient contour tracking schemes is compared.

5.4 Performance Evaluation

In this section, the performance of the tracking protocol under specific assumptions and application scenarios is evaluated.

5.4.1 Assumptions

We make the same assumptions on the signal, noise and sensors as we made in section 3.2.2. The only difference is that the spatial distribution is varied over time.

5.4.2 Evaluation Measures

The following performance measures are assumed for evaluating the contour tracking problem:

- *Target graph*: this graph shows the variation of tracking cost ratio versus the missed-detection rate. This graph specifies the suitability of the algorithm. When the target graph shows that the solution is proper, the next results also are given. The effect of the false detections is implicitly included in the average cost.
- Missed-detection rate and the number of false detections versus detection threshold
- Variation of average cost ratio of tracking versus the speed of movement
- Variation of the average cost ratio of tracking versus detection threshold
- Variation of the missed detection versus the speed of movement
- Variation of the number of false detections versus the speed of movement

To compare the cost of the discussed contour tracking algorithms with each other and with spatial contour detection algorithm, the average cost ratio is used in evaluation of the contour tracking algorithms, which is defined as follows:

$$\overline{C}_T = \frac{C_T}{C_S \times N_{rept}} \tag{5-18}$$

In (5-18), C_T and C_S are the total cost of tracking and spatial contour detection algorithms, respectively and N_{rept} is the total number of repetitions.

5.4.3 Performance Evaluation Results

In this section the performance of the introduced space, time and space-time contour tracking algorithms for the following scenarios are evaluated.

Scenario 1: The whole contour moves with the a constant speed

One application scenario of the contour tracking is the case where all part of the contour changes. For simplicity, the case where the whole spatial distribution is shifted with the same speed is studied. A two-dimensional Gaussian is assumed as the spatial distribution according to Figure 3.5 and one contour of it at level 50 is tracked using static wireless sensor network.

Tracking the temporal variations of a contour when the speed of variation is not known is very risky. *ATU* and *NATU* algorithms allow tracking the variations of spatial distribution when the speed of variation is unknown. Applicability of *ATU* and *NATU* for stop-and-move scenario is tested using simulation. For this scenario a shifting two-dimensional Gaussian is also assumed and one contour of it is tracked.

Scenario 2: Part of the contour is varied

Variation of a part of an assumed spatial distribution is more probable than that the case where all of the distribution changes. Accordingly, tracking the temporal variation of a piece of a contour is studied in this scenario.

For this scenario, a spatial distribution made up of the superposition of a large spatial distribution and few small, two-dimensional Gaussians in a corner zone is assumed. By shifting the small Gaussians, part of the contours of the spatial distribution changes. With this analytical model, contour tracking with randomly distributed sensing nodes is possible.

The basic objective of applying the contour tracking algorithm is to reduce the tracking cost. The desired case is when the tracking cost is less than %15 of the spatial contour detection. Then in our evaluations, when this condition along with performance is fulfilled, the cost efficient contour tracking scheme is discussed in detail.

5.4.3.1 The whole contour is shifted with the same speed

Case 1: No efficient technique is used

Basically the first graph which is studied for each of the tracking cases in the target graph. This graph clearly illustrates the suitability of the contour tracker from two points: performance of tracking, and the tracking cost (the relative number of transmissions).

Generally, when the missed detection rate is small (for example 10 to 20 percent), and the average cost ratio is also small (for example %15) the tracking approach has an acceptable performance and is cost efficient.

Figure 5.10 illustrates the target graph of contour tracking for different speed of shift using the basic tracking protocol when none of the introduced efficient schemes are used. This figure shows that the tracking cost is always high and in some cases it cannot properly track the circular contour of the two-dimensional Gaussian distribution (for example when $V_x = V_y = 5$), specifically when the speed of move is high.

Accordingly using the contour tracking protocol without the efficient schemes is very costly and is not recommended.

Case 2: Gradient Compass (GC) is used

In space-oriented efficient contour tracking schemes like *GC* and *ED*, the speed of variation of the spatial distribution is not basically known. These schemes just filter out the relevant cooperating parts to reduce the communication cost (the number of transmissions). As a basic assumption we assume that in space-oriented schemes we assume that the tracking speed is the same as the speed of variation of the spatial

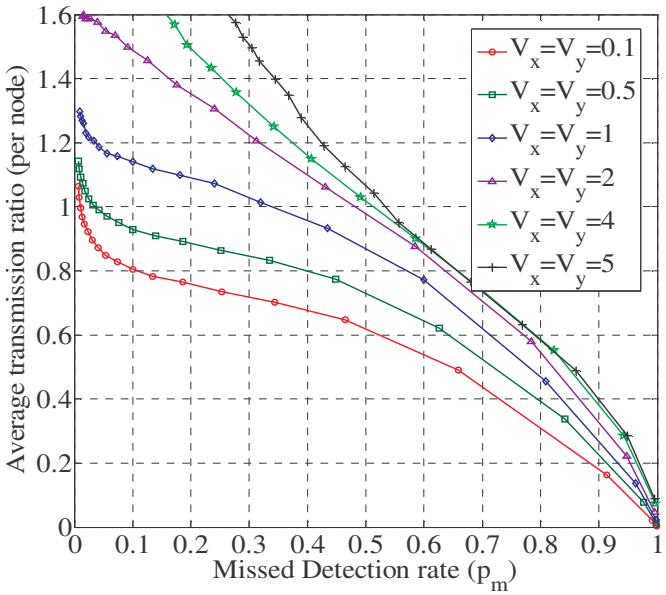


Figure 5.10: Target graph of "contour tracking without cost efficient schemes"

distribution.

The performance of *GC* is affected by the speed of the spatial distribution, noise, and the value of the acceptance angle of *GC*. Figure 5.11.a compares the effect of speed of shift on target graph of *GC* when the acceptance angle is 20 degrees and the standard deviation of the additive noise is 4. Comparison of this figure with Figure 5.10 shows that with this acceptance angle, *GC* reduces the number of transmissions. Nevertheless, it is unable to properly track the variations at high speeds yet ($V_x = V_y = 4$ and 5).

Acceptance angle highly affects the tracking performance of *GC*. For instance small acceptance angle (for example 5 degrees) increases the missed-detection rate and big acceptance angle (for example 50 degrees here) highly increases the average cost ratio of tracking. The effect of acceptance angle (α) is shown in Figure 5.11.b.

Case 3: Estimating the Displacement (ED)

ED by filtering a subset of the selected nodes of *GC* reduces the communication cost

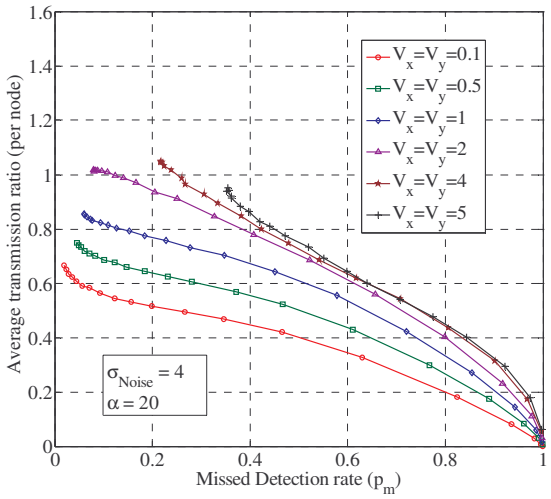


Figure 5.11.a: Target graph of *GC* for different speed of shift of two-dimensional Gaussian distribution

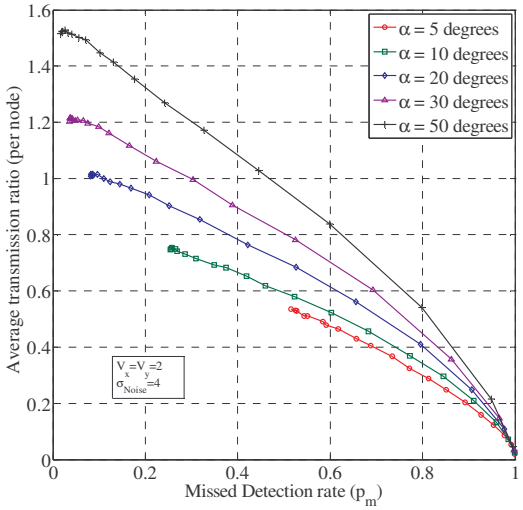


Figure 5.11.b: Target graph of *GC* for different acceptance angles of algorithm

of the algorithm. Then it is expected that the missed detection rate of *ED* be higher than the missed detection rate of *GC*. Figure 5.12.a shows the effect of the value of κ on missed detection rate and cost of *ED*. When κ is small (for example κ = 0.5), the performance of contour tracking is very poor and unacceptable. When this factor is chosen big enough (for example κ=3 or 4) the performance is in the acceptable range. As κ increases to bigger values (for example κ=10), the cost highly increases.

The effect of acceptance angle (α) on *ED* is shown in Figure 5.12.b. As this figure shows the effect of α on *ED* is similar to the effect of this parameter on *GC*, i.e. at small angles the missed detections is high and at large angles the cost becomes too high.

Case 4: Adaptive Time Update (ATU)

Unlike the introduced space-oriented schemes, *ATU* which is a time oriented approach is able to track the variation speed of the spatial distribution in a distributed way. In this part of the chapter we will see that *ATU* is a successful approach which has proper performance and affordable cost.

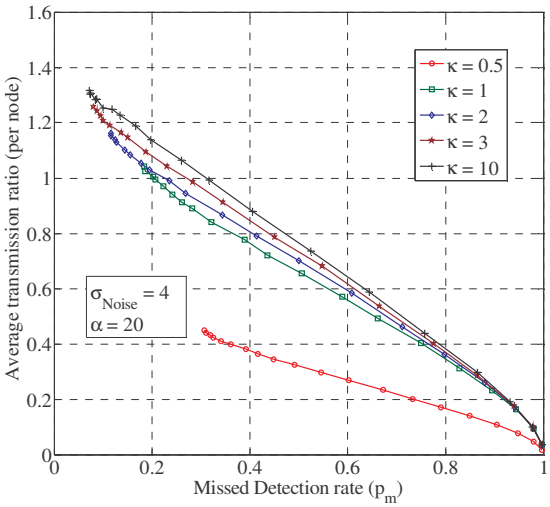


Figure 5.12.a: Target graph of *ED* for different κ factors

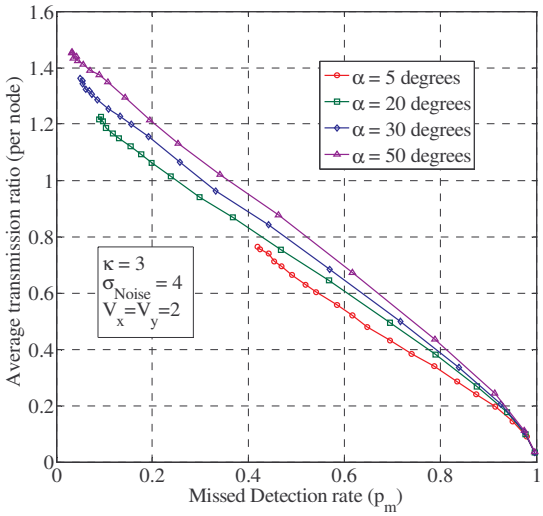


Figure 5.12.b: Target graph of *ED* for different acceptance angles

The target graph of *ATU* when ξ=8 and the standard deviation of the observation noise is σ=4 is shown in Figure 5.13.a. Comparison of this graph with the target graph of *ED* and *GC* clearly shows that *ATU* is more successful than the space-oriented approaches, specifically at low speeds. For this graph, the ξ value has been chosen properly. When the ξ value becomes smaller, the average tracking cost increases due to the increase of the number of contour updates. The cost is also slightly varies versus the detection threshold. These two recent facts are shown in Figure 5.13.b.

One of the other issues in contour tracking is variation of the cost ratio with speed. As the speed of move increases, the average cost ratio of the contour tracking in expected to increase. Figure 5.13.c shows this trend for some detection thresholds. Because *ATU* adaptively captures the variations of the spatial distribution over time, the average cost ratio is not as high as the space-oriented schemes. This figure also shows that the average cost ratio is not very sensitive to the detection threshold γ.

Figures 5.13.d and 5.13.e depict the variation of missed detection and the number of false detections versus speed at some detection thresholds, respectively. Based on these two figures, the detection threshold should be chosen properly to have small number of false detections and missed detections too. At γ = 0.6, it looks working properly.

Figure 5.13.f simultaneously shows the variation of missed detection rate and the number of false detections versus the detection threshold. The increasing and decreasing attribute of these two measures support the results of Figures 5.13.d and 5.13.e that speed has fairly small effect on the performance of *ATU*.

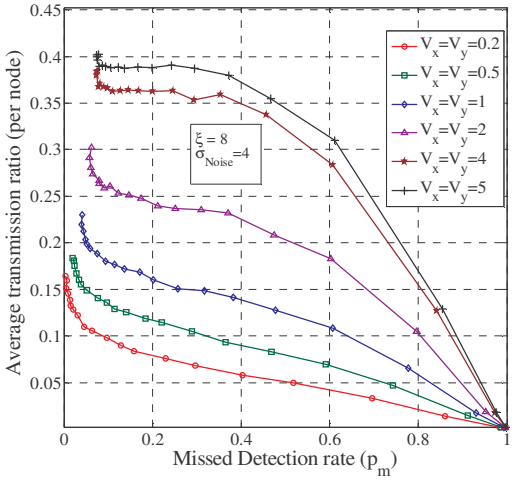


Figure 5.13.a: Target graph of *ATU* for various speeds of shift

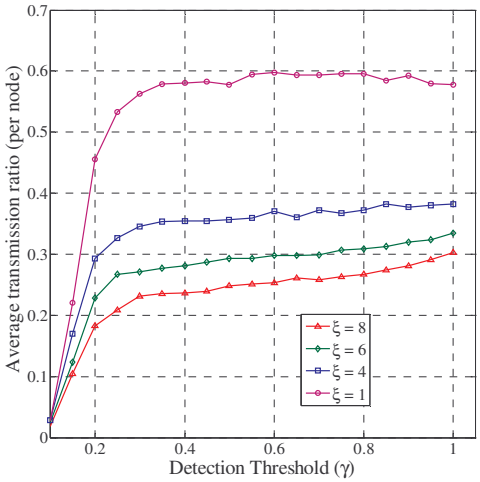


Figure 5.13.b: Variation of average cost ratio vs. detection threshold (V_x=V_y=2)

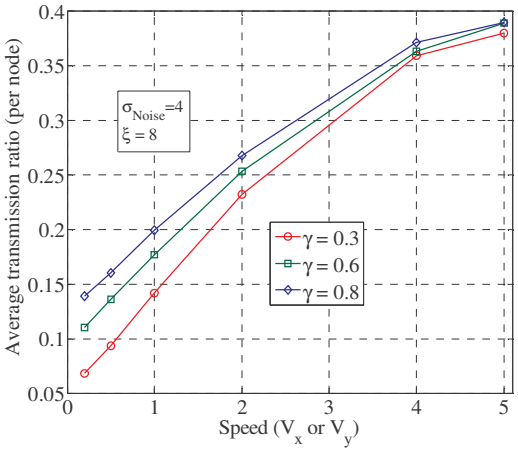


Figure 5.13.c: Variation of the average cost ratio vs. speed for some thresholds

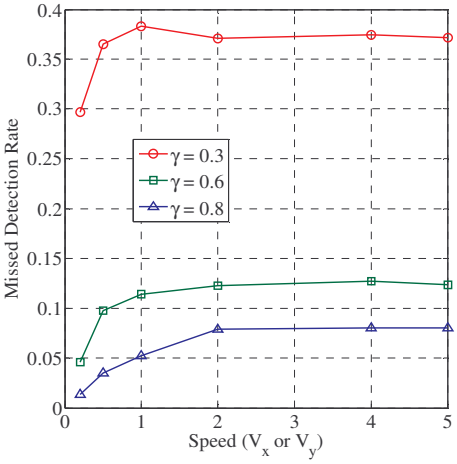


Figure 5.13.d: Variation of missed detection rate vs. speed for some threshold

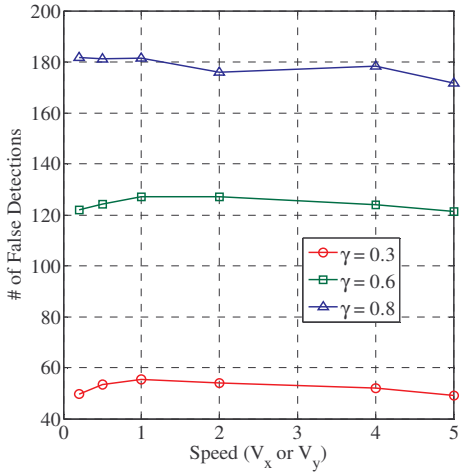


Figure 5.13.e: Variation of # of false detections vs. speed for some threshold

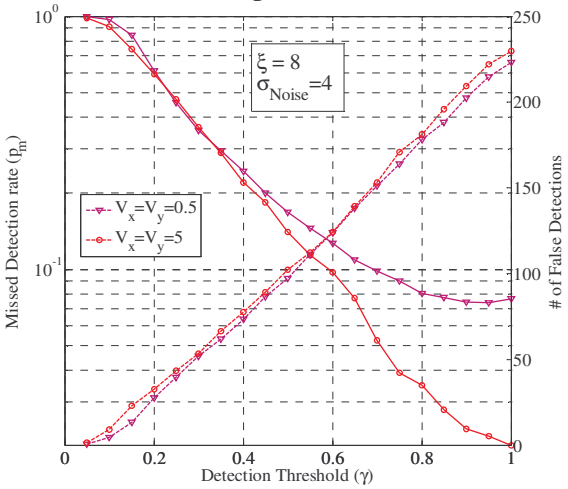


Figure 5.13.f: Simultaneous performance curves vs. threshold at two speeds

Case 5: *Normalized Adaptive Time Update (NATU)*

Using *NATU* instead of *ATU* has two major advantages: tracking with higher performance, and reducing the excessive costs due to adaptive time update. It is expected that at low speeds its dominant effect is reduction of the cost (advantage over *ATU* and space oriented approaches) and at high speeds its dominant effect is to avoid performance degradation (advantage over space-oriented approaches). Figure 5.14 compares *ATU* and *NATU* with their target graph. Although the ξ value was chosen skillfully for *ATU*, but *NATU* performs better.

One interesting application of *NATU* is for the case where the speed of variation of the spatial distribution changes. As a simple example stop-and-move and constant speed that in general is interpreted to variable speed of variation.

Case 6: Comparison between the schemes

Applicability of the introduced efficient contour tracking schemes has been evaluated based on the target graph or the other graphs. A comparison between the cost of the

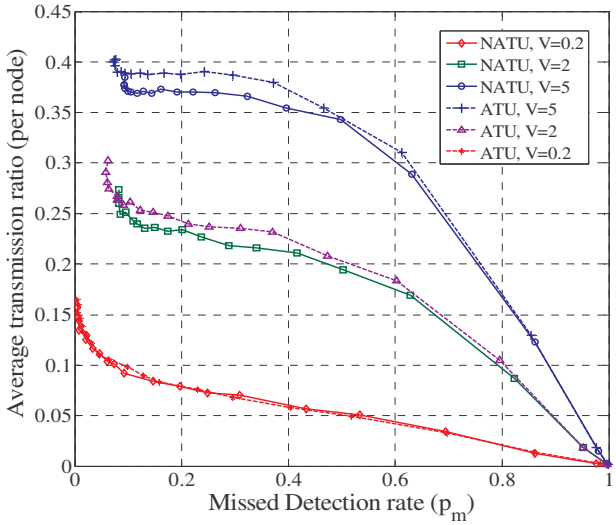


Figure 5.14: Target graph of *NATU* along with *ATU* with ξ=8 (σ_N=4)

tracking schemes can help understand the effectiveness of each of them. Figure 5.15 compares the average cost ratio of schemes versus the speed of move. According to this graph, *GC* alone can reduce the cost of the tracking scheme, but it is not sufficient. *NATU* by itself reduced the cost of the contour tracking and maintain the performance of the performance of the tracking. When *GC* is combined with *NATU*, it reduces a few percent of the cost of *NATU*.

One major drawback with *GC* is that if the acceptance angle is chosen small, it degrades the performance of the tracking.

5.4.3.2 Part of the contour is deformed over time

Applicability of the *NATU* as a time-oriented scheme for tracking a shifting contour was tested. Clearly the performance of the contour tracking algorithm does not change when it is applied to the case where just part of the contour changes. Then in evaluation,

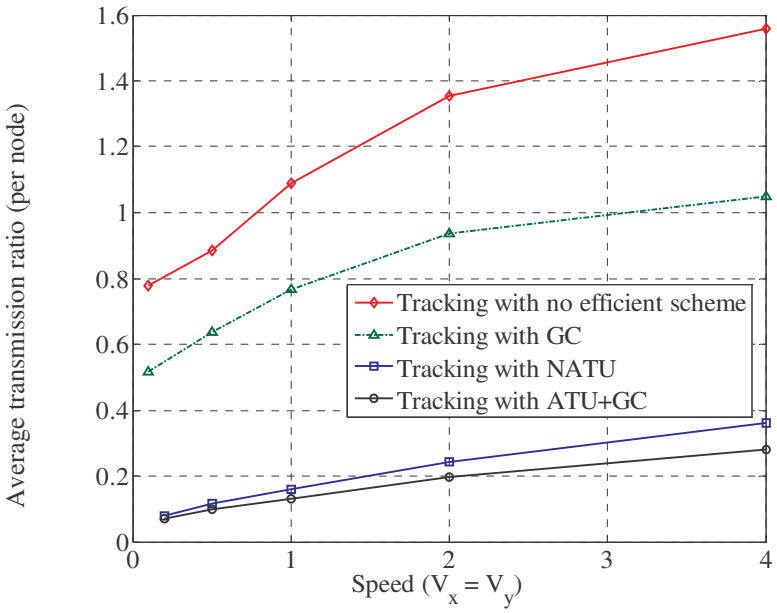


Figure 5.15: A comparison between the variation and range of average cost ratio versus speed for different contour tracking schemes.

it looks to be sufficient to inspect the cost of the tracking algorithm. Figure 5.16 shows the snapshot of the originally detected contour along with the final tracked contour nodes. Because of the uncertainty of selecting the acceptance angle in *GC*, in tracking the part of the contour just *NATU* was used.

For evaluation of this scenario also we used the target graph. This graph is shown in Figure 5.17. In this graph the moving portion moves with the speed of $V_x=V_y=2$. A comparison between this graph and the target graph in Figure 5.13 results that the cost of contour tracking in the same speed has been reduced. It also shows that *NATU* in tracking of the moving portion of this contour is very insensitive to the noise.

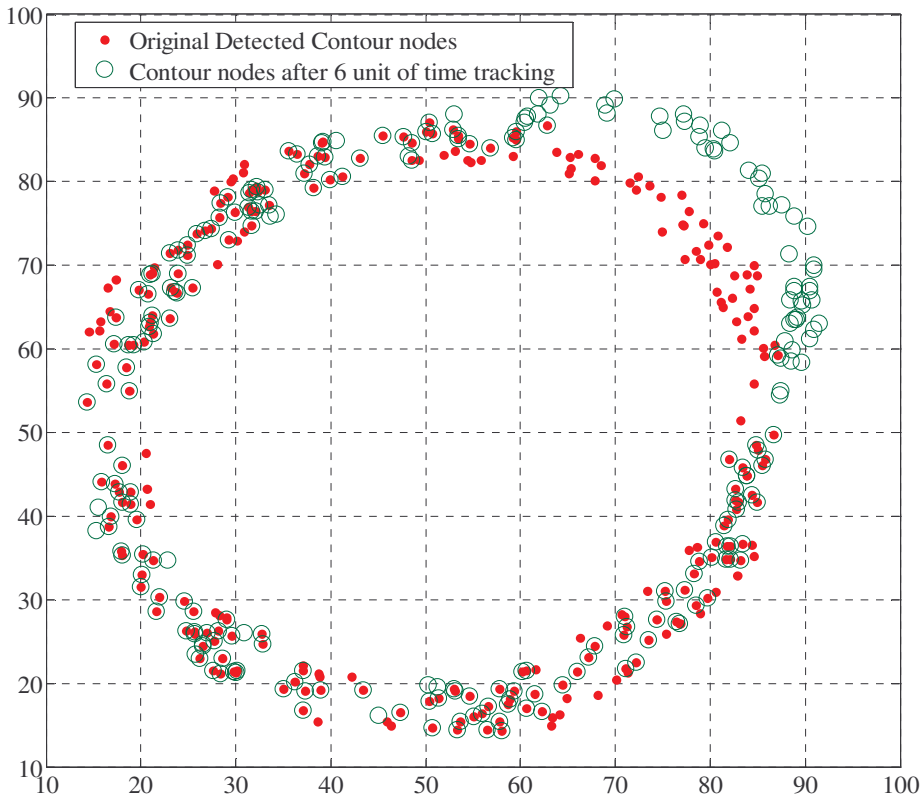


Figure 5.16: Tracking the temporal variation of a piece of contour with *NATU*

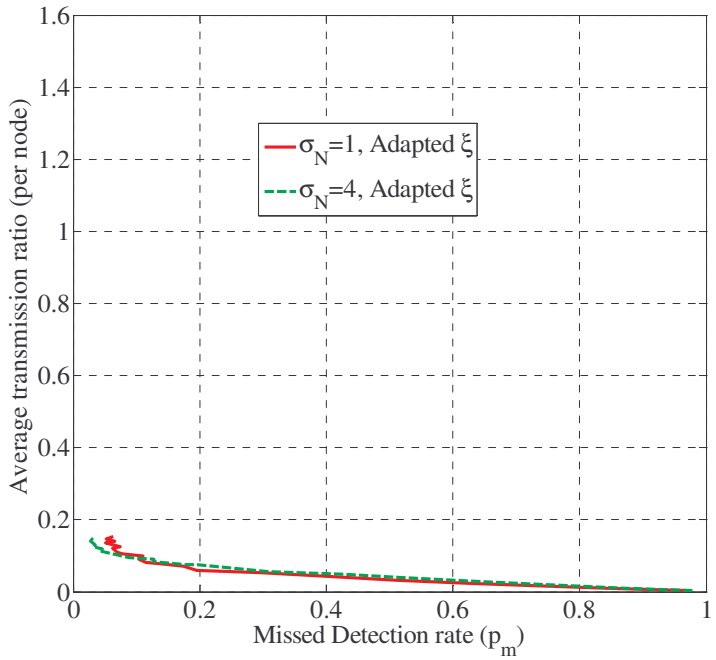


Figure 5.17: The target graph for tracking a portion of contour using *NATU*

5.5 Overview

In chapter 4, we observed that collaborative detection can be used to reduce the error in detecting the contours of random spatial distributions from noisy observations. However, the viability (in term of the cost) of monitoring the spatial distribution of a stationary spatial distribution is an important issue. The average number of transmissions (which determines the communication cost) that depends on several parameters such as the total perimeter of the contours at all the levels, the radio range, etc. It has been observed that for the introduced spatial detection, the average cost of spatial monitoring with contour detection is relatively large and periodically repeating it, is very costly (1.1~1.6 transmissions for each of the nodes in the network).

However, the proposed contour tracking protocol and space and time oriented cost reduction algorithms are found to be successful in reducing of the tracking cost ratio (the average number of transmissions per node for tracking, over the average number of

transmissions for spatial detection) to about 0.15. Hence, the proposed solution is found to be viable for spatiotemporal monitoring of correlated signals over large regions such as those considered in this work.

5.6 Conclusion

A contour tracking algorithm along with four cost reduction scheme is proposed and discussed. The performance of the tracking algorithm in the presence of noise under when the contour or part of it varies is evaluated using simulations. The simulation results that the time-oriented cost reduction schemes are more reliable and more efficient, though the space-oriented schemes also can be combined with them to reduce part of the cost as an auxiliary schemes.

CHAPTER 6: DESIGN AND IMPLEMENTAION OF LEVEL BASED SAMPLING SCHEMES FOR LARGE SCALE WIRELESS SENSOR NETWORK

6.1 Introduction

The conventional approach of periodic sampling of analog signals is usually motivated by the need for perfect reconstruction of the signal from its samples, i.e. by sampling the signal at the *Nyquist* rate or higher. However, periodic sampling may not be appropriate for many applications of wireless sensor networks, where reduction of communication cost is a critical requirement. In such applications, non-uniform sampling provides an excellent solution to conserve communication cost by scarifying some accuracy of signal reconstruction. The basic idea is to suppress or slow down transmissions when the samples do not carry much information, e.g. signal has not changed much. This is particularly important for temporally sparsed (bursty) and variable bandwidth signals. For such cases, periodic sampling usually results in redundant samples that should be eliminated before transmission or storage to maintain efficiency. Non-uniform sampling with sampling intervals that vary according to the short-term bandwidth of the signal is an effective solution for achieving efficiency in such applications. One of the main challenges in non-uniform sampling is the choice of sampling instants that should be defined based on the short-term characteristics of the random signal. Level crossing sampling (*LCS*), which is a subclass of non-uniform

sampling, has been proposed to resolve this issue by sampling the signal when the signal crosses a set of predefined levels [40, 46, 47].

In this chapter we present the design and implementation of *LCS* based sampling in a real-life wireless sensor network. Various design issues of *LCS* are presented such as considerations for determining the number of levels, level-selection, and appropriate sampling periods that are needed for achieving higher energy efficiency without lossof useful information from the *WSN*.

6.2 Application Scenario and Motivation

Our motivation is drawn from the requirements of an experimental project called ParadiseNet, which is a large scale *WSN* in a power substation in KY. The wireless sensors are used for equipment monitoring in the substation. This work is part of an EPRI (Electric Power Research Institute) sponsored research project that is being conducted at UNC Charlotte since Sept 2006. Till date, above 120 wireless sensor nodes have been deployed at the test location in TVA's Paradise substation to obtain various monitored signals from power equipment and transmit to a location in Charlotte. A key design issue of the *WSN* deployed in Paradise is to achieve adequate performance when the network is

Figure 6.1: Location of the nodes in ParadiseNet

scaled up to150 nodes. It has been determined that periodic data transmissions with sampling interval of 15 minutes from 150 nodes would deplete the batteries in each node (assumed twp AA disposable batteries) in about 6 months. Hence adequate energy saving considerations are required to achieve battery life greater than one year with 150 nodes. Increasing the sampling interval with periodic sampling is not a solution as that will not catch temperature variations when they occur. So our objective is to develop a level based sampling scheme that transmits samples based on variations of the signal sensed.

Figure 6.1 shows the locations of the 74 deployed nodes in ParadiseNet. ParadiseNet includes sensors for obtaining ambient and metal surface temperatures, vibration and sound intensities, and others. These sensors monitor ambient environmental conditions as well as key signals from substation requirement like oil-filled and gas filled circuit-breakers, transformers, and compressors (Figure 6.2). While the exact characterization of the monitored signals is unknown beforehand, it is reasonable that equipment faults can be detected from anomalies in vitas signs such as surface temperatures, vibrations and sound. Hence our applications have been designed to perform periodic sampling of these

Figure 6.2.a: An oil-cooled transformer in TVA's Paradise substation, showing two wireless sensors marked by yellow circles

Figure 6.2.b: Oil-filled circuit breakers in TVA's Paradise substation. The wireless sensor nodes that sense the surface temperatures are marked by red circles

signals. However, as explained in the previous section, much of the samples are redundant, and hence we apply *LCS* to these monitoring applications.

6.3 Problem Statement

LCS involves challenging design issues such as selection of the number of levels and their corresponding values. These factors are critical for determining the accuracy of reconstructing the signals from its samples as well as the average communication cost. In addition, a key constraint is the maximum number of levels that is physically possible to use in the hardware-constrained low-cost wireless sensors. Hence, we focus on the analysis of *LCS* with non-uniform levels in order to meet the requirements of signal reconstruction error, average sampling rate, and maximum number of levels. We first obtain the optimum set of levels that minimize the mean square error (*MSE*) between the reconstructed signal and the original signal, which are obtained from knowledge of the probability density function (*pdf*) of the bandlimited stationary signal. We show that, for a specified number of levels, the *pdf*–aware *LCS* with optimum levels results in less reconstruction *MSE* than that with uniform *LCS*. We then propose a μ-law based non-uniform *LCS*, that is similar in concept to the μ-law based level selection scheme presented in Chapter 4. The proposed μ-law based *LCS* can be designed under some basic assumptions on the characteristics of the signal.

The concept of *LCS* in comparison to periodic sampling is illustrated in Figure 6.3. In *LCS*, the signal is sampled at the points at which the signal crosses some predefined levels $\{\ell_i\}_{i=1}^{M}$. Consequently, sampling instances in *LCS* are no longer uniform, but are determined by the characteristics of the signal itself. Also, unlike periodic sampling there is no concrete mechanism for perfect reconstruction of the original signal for *LCS* using

non-iterative methods. A discussion of reconstruction methods for *LCS* is given in [43]. In this chapter, we assume a piece-wise constant or the zero-order-spline reconstruction of the signal [50]. This is equivalent to a level-crossing sample and hold (*LCSH*) operation from the level crossing samples of the signal (see Figure 6.4.b). This simple method, although not as efficient as reconstruction methods that use higher order splines, allows us to get tractable solution for the *MSE*.

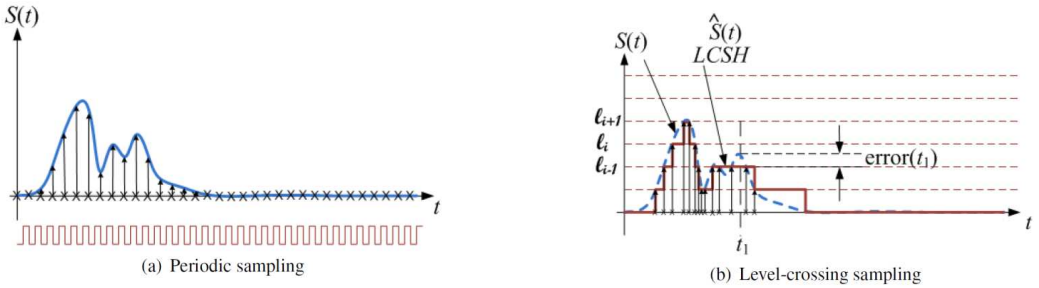
The reconstruction error signal between the *LCSH* and original signals is defined by:

$$\varepsilon_S(t) = S(t) - \hat{S}(t) \tag{6-1}$$

where $S(t)$ is the original signal and $\hat{S}(t)$ is *LCSH* with level-crossing set $\{\ell_i\}_{i=1}^{M}$ at *t*. Because of the stochastic behavior of the signal and its samples, the error power changes randomly with time. The *MSE* is then given by:

$$E[\varepsilon_S^2(t)] = \int_{-\infty}^{\infty} \varepsilon_S^2(t) f_\varepsilon(\varepsilon;t) d\varepsilon \tag{6-2}$$

where $f_\varepsilon(\varepsilon;t)$ is the *pdf* of the error signal at time instance *t*. Our objective is to minimize the *MSE* by better tracking of the dynamics of the signal, which can be obtained by appropriate selection of the levels. Note that the average number of samples obtained from *LCS* also depends on the chosen sampling levels, which we address later.



(a) Periodic sampling (b) Level-crossing sampling

Figure 6.3: Sampling of temporally sparsed signals: (a) Periodic sampling; (b) Level-crossing-sampling (*LCS*) and zero-order spline reconstruction, i.e. level-crossing-sampling and hold (*LCSH*).

6.4 Error Analysis

The reconstruction *MSE* in (6-2) can be rewritten as:

$$E[\varepsilon_S^2(t)] = \sum_{i=1}^{M+1} E[\varepsilon_{S,i}^2(t)]\; p_i \tag{6-3}$$

where $E[\varepsilon_{S,i}^2(t)]$ is the *MSE* between level ℓ_i and ℓ_{i-1}, and p_i is the probability of that the signal resides inside level-interval ℓ_i and ℓ_{i-1}, $i=1,2,...,M$.

$$p_i = \Pr(\ell_{i-1} < S < \ell_i) = \int_{\ell_{i-1}}^{\ell_i} f_S(s)\, ds \tag{6-4}$$

The marginal *pdf* of the signal between each two neighboring levels is:

$$f_{S,i}(s) = \frac{f_S(s)}{\int_{\ell_{i-1}}^{\ell_i} f_S(s)\, ds}\;,\; \ell_{i-1} < s < \ell_i \;,\; i = 0,1,...,M+1 \tag{6-5}$$

Note that at upward crossing of the signal at level ℓ_i (see Figure 6.4), *LCSH* takes value ℓ_i until the signal crosses ℓ_{i+1} and the error will be $\varepsilon_{S,i}(t) = S(t) - \ell_i$. When the signal crosses downwards at the level ℓ_{i+1} and resides between levels ℓ_i and ℓ_{i+1}, the error will be $\varepsilon_{S,i}(t) = S(t) - \ell_{i+1}$. With this idea we write:

$$\begin{aligned} f_{\varepsilon,i}(\varepsilon) = {} & f_{\varepsilon,i}(\varepsilon \mid \text{up-crossing}).\Pr(\text{up-crossing}) + \\ & f_{\varepsilon,i}(\varepsilon \mid \text{down-crossing}).\Pr(\text{down-crossing}) \end{aligned} \tag{6-6}$$

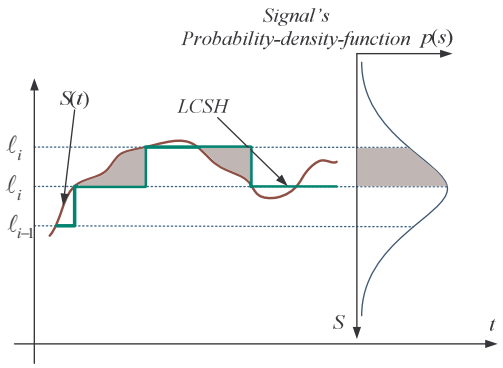


Figure 6.4: Illustration of error between signal and *LCSH*

Hence (6-3) can be written as:

$$E[\varepsilon_S^2(t)] = E[\varepsilon_{S,1}^2(t)]\,p_1 + E[\varepsilon_{S,M+1}^2(t)]\,p_{M+1} + \sum_{i=2}^{M}\frac{1}{2}\,p_i\left\{\int_{\ell_{i-1}}^{\ell_i}(\ell_{i-1}-s)^2 f_{S,i}(s) + (\ell_i - s)^2 f_{S,i}(s)\,ds\right\} \tag{6-7}$$

Simplifying (6-7) leads to:

$$E[\varepsilon_S^2(t)] = \int_{\ell_0}^{\ell_1}(\ell_1 - s)^2 f_S(s)\,ds + \int_{\ell_M}^{\ell_{M+1}}(\ell_M - s)^2 f_S(s)\,ds + \sum_{i=2}^{M}\frac{1}{2}\left\{\int_{\ell_{i-1}}^{\ell_i}(\ell_{i-1}-s)^2 f_S(s) + (\ell_i - s)^2 f_S(s)\,ds\right\} \tag{6-8}$$

which gives the tracking *MSE* for an arbitrary set of levels.

6.4.1 Optimizing Levels for Minimizing Tracking *MSE*

We now obtain the solution to the optimum set of levels which leads to the minimal tracking *MSE* by setting

$$\frac{\partial E[\varepsilon_S^2(t)]}{\partial \ell_i} = 0, \quad 1 \le i \le M \tag{6-9}$$

This results in *M* simultaneous non-linear integral equations, as follows:

$$\ell_1 = \frac{\int_{\ell_0}^{\ell_1} x\,f_S(x)\,dx + \int_{\ell_0}^{\ell_2} x\,f_S(x)\,dx + \frac{1}{2}f_S(\ell_1)(\ell_2 - \ell_1)^2}{\int_{\ell_0}^{\ell_1} f_S(x)\,dx + \int_{\ell_0}^{\ell_2} f_S(x)\,dx}$$

$$\ell_i = \frac{\int_{\ell_{i-1}}^{\ell_{i+1}} x\,f_S(x)\,dx}{\int_{\ell_{i-1}}^{\ell_{i+1}} f_S(x)\,dx}, \quad 1 < i < M \tag{6-10}$$

$$\ell_M = \frac{\int_{\ell_M}^{\ell_{M+1}} x\,f_S(x)\,dx + \int_{\ell_{M-1}}^{\ell_{M+1}} x\,f_S(x)\,dx - \frac{1}{2}f_S(\ell_M)(\ell_M - \ell_{M-1})^2}{\int_{\ell_M}^{\ell_{M+1}} f_S(x)\,dx + \int_{\ell_{M-1}}^{\ell_{M+1}} f_S(x)\,dx}$$

Note that (6-10) indicates that the optimal set of levels is such that each level is located at the ensemble mean value of the signal between its two immediate neighboring levels.

These equations can be solved numerically, for instance by using an iterative approach. For the detail of (6-10) see Appendix A.

6.4.2 Proposed μ-law Based Approach

It is obvious that derivation of the optimum levels as described above requires knowledge of the *pdf* of the signal. In the absence of such information, we propose a sub-optimal *LCS* scheme where the non-uniform sampling levels are chosen under the assumption that the signal *pdf* is unimodal and symmetric, with the peak of the *pdf* lying at the mean value. This is a valid assumption for most realistic signals, and allows us to define non-uniform sampling levels that are more concentrated near the mean of the signal distribution, and hence, expected to capture the dynamics of the signal more effectively than uniform *LCS*. We propose to use a μ-law based levels $\{\mu_i\}_{i=1}^{M}$ as defined below, which are based on standard μ-law expansion formula used for non-uniform quantization in *PCM* [51].

$$U_i = \frac{2(\frac{\ell_{M+1}-\ell_0}{M} i + \ell_0 - \frac{\ell_{M+1}+\ell_0}{2})}{\ell_{M+1}-\ell_0}$$
$$\mu_i = \left[\frac{\mathrm{sgn}(U_i)}{\mu}\left[(1+U_i)^{\mu}-1\right]\right]\frac{\ell_{M+1}-\ell_0}{2} + \frac{\ell_{M+1}+\ell_0}{2}, \quad i = 1,2,...,M \tag{6-11}$$

In this equation U_i is the *i*th level uniform level and ℓ_0 and ℓ_{M+1} are the lower and upper expected range of the signal.

6.5 Level-Crossing Sampling Design Problem

According to the general non-uniform sampling theorem, the required sample volume for unique representation of a bandlimited signal should not be less than *Nyquist* rate

[50]. Then assuming *M* sampling levels, the average *LCS* rate R_{LCS} should be more than twice of the bandwidth of the signal.

$$R_{LCS} = \sum_{i=1}^{M} \int_{-\infty}^{\infty} |x'| f(x' \mid x = \ell_i)\, d \mid x' \mid > 2\, BW \tag{6-12}$$

For stationary bandlimited Gaussian signal this formula is simplified to (6-13) (after the Rice's contributions).

$$R_{LCS} = \sum_{i=1}^{M} \frac{1}{\pi} \sqrt{\frac{-R''(0)}{R(0)}}\ \exp(-\frac{\ell_i^2}{2R^2(0)}) > 2\, BW \tag{6-13}$$

By using the definition of *rms* bandwidth according to (6-14) [52], and then the definition of autocorrelation function of the signal $R_X(t)$ according to (6-15), (6-14) changes to (6-16).

$$BW_{rms} = \sqrt{\frac{\int_{-\infty}^{\infty} f^2 S_X(f) df}{\int_{-\infty}^{\infty} S_X(f) df}} \tag{6-14}$$

$$R_X(t) = \int_{-\infty}^{\infty} S_X(f) \exp(j2\pi ft)\, df \tag{6-15}$$

$$BW_{rms} = \sqrt{\frac{\frac{-1}{4\pi^2} R''(0)}{R(0)}} = \frac{1}{2\pi} \sqrt{\frac{-R''(0)}{R(0)}} \tag{6-16}$$

Accordingly, the average level crossing rate is connected to the BW_{rms} as follows:

$$R_{LCS} = 2BW_{rms} \sum_{i=1}^{M}\ \exp(-\frac{\ell_i^2}{2\sigma^2}) \tag{6-17}$$

To have a proper sampling, R_{LCS} should be greater than average *Nyquist* sampling rate,

$$BW_{rms}\sum_{i=1}^{M}\ \exp(-\frac{\ell_i^2}{2\sigma^2}) > BW \tag{6-18}$$

Then normalized average level-crossing rate should satisfy in (6-19):

$$\overline{R}_{LCS} = \sum_{i=1}^{M}\ \exp(-\frac{\ell_i^2}{2\sigma^2}) > \frac{BW}{BW_{rms}} \tag{6-19}$$

For simplicity, by approximating the power spectral density of the signal with flat-top spectra according to (6-20), the relationship between BW_{rms} and the signal's bandwidth is simplified to $BW_{rms} = \frac{1}{\sqrt{3}} BW$.

$$S(f) = \begin{cases} L & |f| \le BW \\ 0 & \text{otherwise} \end{cases} \tag{6-20}$$

Accordingly, the average *LCS* rate in terms of the signal's bandwidth and the selected set of levels is according to (6-21):

$$R_{LCS} = \frac{2}{\sqrt{3}} BW \sum_{i=1}^{M}\ \exp(-\frac{\ell_i^2}{2\sigma^2}) \tag{6-21}$$

Then the levels $\{\ell_i\}_{i=1}^{M}$ should be chosen so that normalized level crossing sampling rate as defined in (6-22), becomes larger than the specified limit in (6-22).

$$\overline{R}_{LCS} = \frac{R_{LCS}}{BW} = \sum_{i=1}^{M}\ \exp(-\frac{\ell_i^2}{2\sigma^2}) > \sqrt{3} \tag{6-22}$$

6.5.1 *LCS* Design Examples

The problem is finding a set of sampling levels $\{\ell_i\}_{i=1}^{M}$ which satisfy in (6-22). For this the followings are assumed:

- The signal is bandlimited, stationary Gaussian $N(0,1)$
- The dynamic range of the signal is assumed known

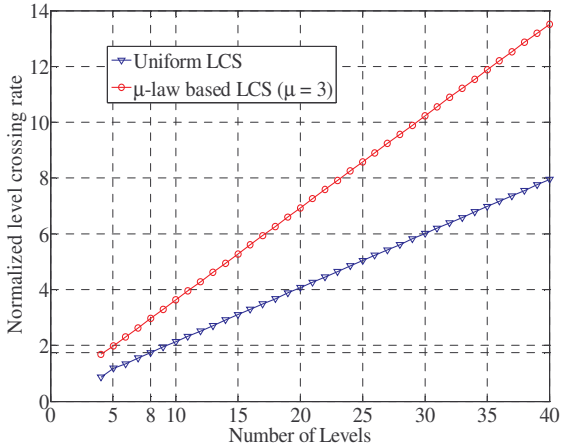


Figure 6.5: Normalized level crossing rate for uniform and μ-law *LCS*

- The power spectral density function of the signal is assumed flat-top

• Case 1: *LCS* using uniformly spaced levels:

In the signal dynamic range interval $(\frac{-\Delta}{2}, \frac{\Delta}{2})$, the M uniformly spaced levels U_i

$U_i = -\frac{\Delta}{2} + (i-1)\frac{\Delta}{M-1}$, $i = 1,2,...,M$, should be chosen so that satisfies in (6-22).

For this, M should be found so that:

$$\overline{R}_{LCS} = \sum_{i=1}^{M} \exp\left(-\frac{(-\frac{\Delta}{2} + (i-1)\frac{\Delta}{M-1})^2}{2}\right) > \sqrt{3} \tag{6-23}$$

• Case 2: *LCS* using μ-law based levels:

For μ-law based *LCS*, levels are defined according to (6-11) and for that, the minimum acceptable number of sampling levels M, which satisfies the proper sampling criteria should be chosen so that:

$$\overline{R}_{LCS} = \sum_{i=1}^{M} \exp\left(-\frac{\left\{\left[\frac{\mathrm{sgn}(U_i)}{\mu}\left[(1+U_i)^{\mu}-1\right]\right]\frac{\Delta}{2}\right\}^2}{2}\right) > \sqrt{3} \tag{6-24}$$

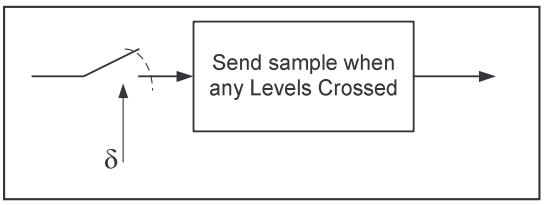


Figure 6.6: Schematic of the *LCS* when sampler probes the signal periodically

As result, we conclude that the number of sampling levels for proper *LCS* sampling using uniform and μ-law based *LCS* should be at least than 8 and 5, respectively (Figure 6.5).

6.5.2. Minimum Sampling Period for Proper *LCS* using Periodic Sampler

For applications where sensors have already been designed for periodic sampling, adjusting sampling period for *LCS* and discarding the irrelevant samples so that the sampler in consecutive sampling instances does not miss any sampling levels is important (see Figure 6.7). In the rest of this section periodic sampling rate issue for implementation of *LCS* is discussed. Selection of the sampling period is a function of the instantaneous slope of the signal.

As signal changes randomly with time, we need to use the statistical characteristics of the signal. If the minimum spacing between the two neighboring sampling levels is $d_{\min}$ and the sampling period is δ (see Figure 6.7), the maximum sampling slope should

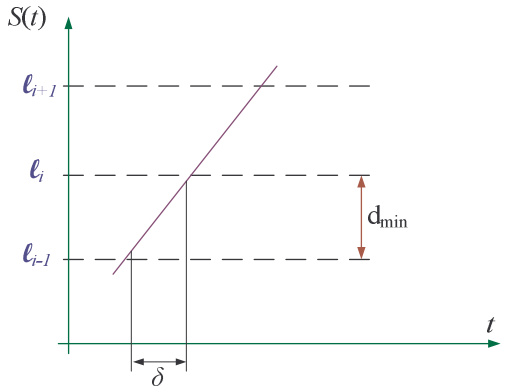


Figure 6.7: Schematic of the *LCS* when sampler probes the signal periodically.

satisfy in (6-25).

$$|S_{Max}| < \frac{d_{\min}}{\delta} \tag{6-25}$$

With Gaussian assumption, the instantaneous slope will be Gaussian too. With probability of 99.95 percent, the slope of the Gaussian signal is in the range of $(-4\sigma_{x'}, 4\sigma_{x'})$. With this approximation and by using (6-25), δ is bounded to (6-26).

$$\delta < \frac{d_{\min}}{4\sigma_{x'}} \tag{6-26}$$

For simplification again let us assume that the signal has a bandlimited flat-top power spectral density like $S(f)$ in (6-20). Then there is the following relationship between the standard deviation of the signal and its slope:

$$\sigma_{x'}^2 = \frac{(2\pi\, BW)^2}{3}\sigma_x^2 \tag{6-27}$$

By using (6-26) and (6-27), the minimum sampling period δ should not be more than the following limit.

$$\delta < \frac{d_{\min}}{4\sigma_{x'}} = \frac{\sqrt{3}}{8\pi}\frac{d_{\min}}{BW}\frac{1}{\sigma_x} \tag{6-28}$$

6.6 Numerical and Experimental Results

In this section, we first evaluate the performance of the *LCS* based sampling and compare the reconstruction error with *LCSH* and cardinal spline for the three *LCS* schemes using simulations and numerical analysis. The performance evaluation also includes the trade-off issue between the reconstruction performance and the sampling cost of *LCS*. Later, we present experimental results of implementation of *LCS* based sampling using *MICAz* wireless sensor platforms. Experimental results were first

obtained at home from which we get comparative performance results of different *LCS* schemes. Following these experiments, several *LCS* based sampling schemes were implemented in ParadiseNet. These nodes were deployed in September 11, 2009, and only a few days worth of data were obtained at the time of writing this dissertation. Some of these data are presented at the end of this chapter.

6.6.1 Numerical Results

To evaluate the performance of the *LCS* schemes, we use computer simulations to determine their reconstruction *MSE* and average sampling rate obtained for a temporally sparsed (bursty) Gaussian signal, using three level-definition schemes: uniformly spaced levels, the proposed μ-law based levels, and the optimal set of levels as obtained from solving the equations in (6-10). For these simulations, we use a bursty Gaussian signal that is generated by adding 50 sinusoids with random amplitudes and frequencies in the range of the assumed bandwidth. We use an activity factor of 0.1, which implies that the signal has no variations for 90% of the time. The amplitude of the random signal is normalized, and its average power σ_x^2 is obtained numerically in the simulations.

6.6.1.1 Optimal μ

We first investigate the effect of μ on the reconstruction *MSE* obtained with the proposed μ-law based *LCS* scheme. In Figure 6.8 we see the variation of the reconstruction *MSE* (with *LCSH*) using the μ-law based *LCS* versus μ, for different numbers of levels *M* for the synthetic Gaussian signal. It is observed that the lowest reconstruction *MSE* is obtained for $\mu = 3.5$ for all *M*, which is an interesting result. On all subsequent performance evaluations we use this value in μ-law based *LCS*.

6.6.1.2 Reconstruction Error

Reconstruction error and sampling rate (cost) are two constraints in *LCS* design problem. Selection of the levels based on (6-22) implicitly implies that the samples will uniquely represent the signal. However, sampling below the Nyquist rate is also desirable when the cost is a critical constraint of the problem and the reconstruction error will not be high. Study of the reconstruction error in terms of the number of levels for different level definition schemes can be a good approach to study the effect of under and over *Nyquist* sampling on reconstruction error. In section 6.5, after a few theoretical assumptions, the relationship between the minimum sampling rate and the set of levels was inspected.

The reconstruction method is one of the major research concentrations in *LCS*. So far, adapted window short time Fourier transform (STFT), and cardinal spline have been proposed [42, 43].

In this section the effect of three level definition schemes on the reconstruction error

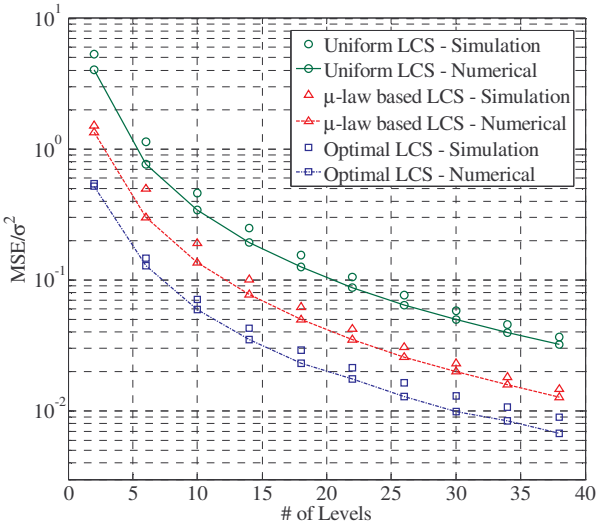


Figure 6.9: Study of the reconstruction *MSE LCS* using *LCSH* reconstructor for the three level-definition schemes.

of the bandlimited Gaussian signal is inspected. Figure 6.9, illustrates the reconstruction *MSE* of *LCS* using *LCSH* reconstructor for these level definition schemes: uniformly spaced levels, μ-law based levels, and optimal levels based on (6-9) using numerical analysis and simulations. The reason of using *LCSH* is that firstly, the optimal levels are designed based on minimizing the error when this reconstructor is used and secondly, it is also good to compare the reconstruction error of the other level definition schemes with the reconstruction error of the optimal scheme. This figure includes the reconstruction error from numerical solution on the side of computer simulation result. As this figure shows, the reconstruction error of μ-law based *LCS* for any specific number of levels is between optimal and uniform *LCS*.

We next evaluate the reconstruction *MSE*s of the three level selection schemes for the bandlimited Gaussian signal described above. For each scheme, the reconstruction is performed by passing the samples through cardinal spline with adapted tension. The corresponding *MSE*s obtained for different values of *M* using computer simulations are

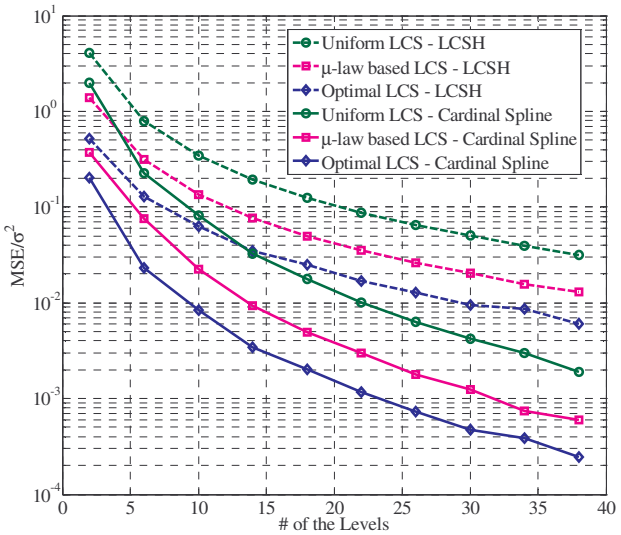


Figure 6.10: Comparison between the reconstruction error of *LCSH* and cardinal spline with adapted tension for the three schemes: uniform, μ-law based and optimal *LCS*.

shown in Figure 6.10. The results show that the reconstruction *MSE* obtained using the proposed μ-law based levels is very close the optimum reconstruction *MSE*, particularly at higher values of *M*. For any desired *MSE*, the μ-law based *LCS* requires much fewer numbers of levels than *LCS* with uniform levels. For instance, to get reconstruction error equal to $MSE=0.3 \times 10^{-2}$, the number of sampling levels required with uniform levels is 34, whereas it can be achieved with only 19 levels and 15 levels using μ-law based and optimum levels, respectively.

The comparison of the required number of levels required with the different level selection schemes for various *MSE* values is depicted on the left side of Figure 6.11. Note that fewer numbers of levels do not necessarily imply a lower average sampling rate, as depicted by the sampling efficiencies of the three *LCS* schemes for different *MSE* values that are plotted on the right side of Figure 6.11. In this figure, the sampling efficiency (γ)

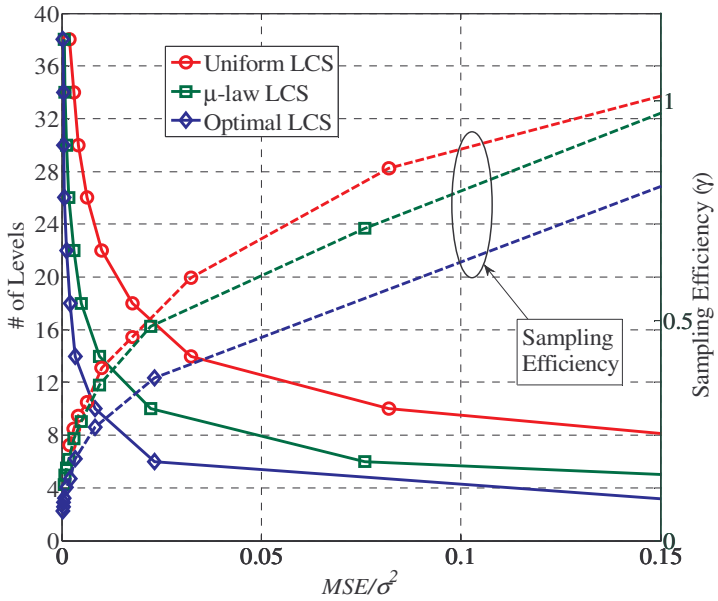


Figure 6.11: Required number of levels and the sampling efficiency for different *MSE* in performance comparison of different *LCS* schemes after C*ardinal spline* filtering.

is $\gamma = \frac{R_{Nyquist}}{R_{LCS}}$ where $R_{Nyquist}$ and R_{LCS} are the *Nyquist* and *LCS* sampling rate, respectively. In fact, Figure 6.11 indicates that selection of uniform levels is most efficient in terms of the average sampling rate. However, this is achieved by using a larger number of levels. So, it may be concluded that when the number of sampling levels can be high enough, uniform *LCS* which leads to the highest sampling efficiency is the most appropriate level definition scheme.

6.6.2 Experimental Validation and Evaluation of *LCS*

To experimentally investigate the efficiency of *LCS* and compare it with periodic sampling, three *LCS* schemes were implemented with *MICAz* wireless sensor platform from *Crossbow Technology Inc.*

6.6.2.1 Wireless Sensor Platform

The *MICAz* (see Figure 6.12 is an Atmel based 8-bit ATMega128L wireless platform that contains a 2.4GHz IEEE 802.15.4 compliant RF transceiver and 4KB of RAM. The main specifications of the *MICAz* mote are summarized in Table 6.1.

Figure 6.12: *MICAz* wireless sensor platform.

Table 6.1: Hardware specifications of *Crossbow MICAz* (Courtesy of *Crossbow Inc.*)

Processor Performance	**MPR2400CA (ATmega128L)**
Clock	0~8 MHz
Program Flash Memory	128K byte
Measurement (serial) Flash	512K byte
Configuration EEPROM	4K byte
Current Draw	8mA (Active mode)
	< 15 μA (Sleep mode)
RF Transceiver	
Transmission Data Rate	250 kbps
Frequency Band	2400 MHz to 2483.5 MHz
Outdoor Range	75m~100m
Indoor Range	10m~30m
Current Draw	19.7mA (Receive mode)
	17.4mA (Transmit 0 dB)
Electromechanical	
Battery	2x AA Batteries
External Power	2.7V ~3.3V

The *MICAz* motes run on TinyOS, which is open source operating system specifically designed for embedded sensor network devices. It features various component libraries that include sensor drivers along with network protocols. Since TinyOS is open source, it is very easy to extend the base functionality of the pre-distributed drivers to suit the unique needs of the developer.

6.6.2.2 Sensing Board

MDA320 data acquisition board (*DAQ*) that is mounted on *MICAz* along with a

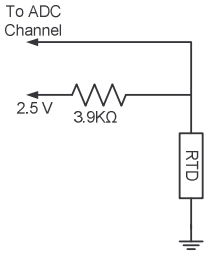


Figure 6.13.a: Voltage divider RTD board.

Figure 6.13.b: MDA320CA foreside.

simple resistive temperature device (*RTD*) board with 2 *RTD* was used for temperature sensing (see Figures 6.13.a and 6.13.b). The sensed voltage was launched to one of the analog to digital inputs of *MDA320 DAQ*.

MDA320CA is designed as a general measurement platform for the *MICAz* and *MICA2*. It has eight single ended analog-to-digital input channels, three differential analog channels, eight digital input channels, and one counter channel.

For this test, 2 analog input channels of *MDA320 DAQ* were used. Figure 6.14.a shows the connection configuration of the *RTD* board, *DAQ* and the *MICAz* mote.
All of these three parts were inserted in the box which is illustrated in Figure 6.15.b. The box had two magnets to connect it the metallic structures. The *RTD*s were devised inside the metallic washers that were connected to the magnets of the box.

To appropriately sense and sample the temperature, it was oversampled at 1 Min interval sampling periods. Whenever the samples cross any of the levels, the new sample value was sent to the base station, provided the previous crossed level and the recent one are not the same.

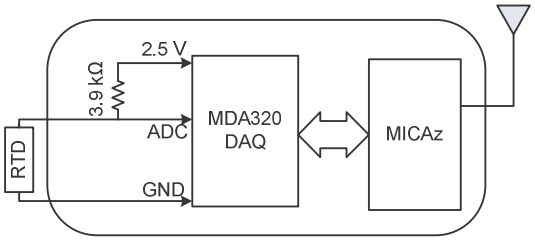


Figure 6.14.a: Connection configuration of *MICAz* mote, *MDA320 DAQ* and the *RTD* board for temperature sensing. This figure shows the connection to only single channel.

Figure 6.14.b: The box and its magnets. All parts of the temperature sensing are enclosed inside the magnet. (The photo shows one of the temperature sensing nodes that were deployed in Paradise substation, KY).

6.6.2.3 Test Scenario

To experimentally study the effect of different level definition schemes on the sampling efficiency of *LCS*, three types of levels for sensing temperature in the range of 20 to 60 degree Celsius were defined. Three wireless sensor nodes were programmed for sampling based on *LCS* using each of these three types of levels. These three nodes sensed the outside temperature of an AC compressor fan. To compare the performance and the cost of the *LCS* schemes to periodic sampling, a fourth node with the same parts was programmed to sense the temperature periodically with period 1 minute.

The sampled data was sent to the base-station which had a *MICAz* mote as base-station, a *MIB520* as gateway which was connected to a computer through a USB cable, and, xlisten, the open source data logger, and a computer. The schematic of this small network is shown in Figure 6.15. The network used XMESH protocol over IEEE 802.15.4 physical and data link layer protocol. The three sets of levels were defined based on the following three schemes:

A- *Uniform LCS*: Based on the analysis in section 6.5, for sampling a normalized Gaussian temperature, at least 8 uniformly spaced levels are required. This result is valid for normalized Gaussian that its dynamic range is approximately 10. For sampling the temperature with Gaussian statistics in the

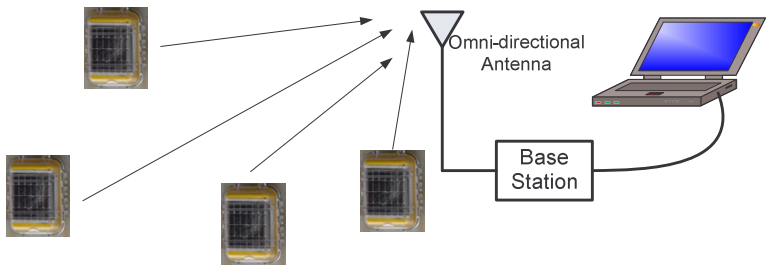


Figure 6.15: A schematic of the deployed wireless sensor network

range of 40 degrees with the same sampling precision, the number of levels should be 4 times more, which means 32 levels.

B- *μ-law based LCS*: In section 6.5, we showed that for uniquely representing the normalized Gaussian signal based on its samples from *LCS*, the number of sampling levels in μ-law scheme should not be less than 5. For sampling in the selected temperature range (20 ^{o}C to 60 ^{o}C) with the same justification of part A, the number of levels should not be less than 20.

C- *Mixed Mode Sensing LCS*: In behavioral study and monitoring of a phenomenon, the usual variation range of the quantities is not very interesting and it is normally preferred to focus on the critical ranges. With this assumption, to study the effect of high temperatures with good accuracy, the variation range of temperature was divided into two parts: 20 ^{o}C to 40 ^{o}C as the normal variation range, and 40 ^{o}C to 60 ^{o}C as the critical sampling range. The normal variation range was sampled with uniformly spaced *LCS* with coarse levels, and the critical sensing range was sampled with oversampled μ-law based *LCS*. The number of levels in the uniform *LCS* and μ-law based *LCS* were chosen 4 and 12, respectively.

The levels which have been used based on these scenarios are tabulated in Appendix B.

6.6.2.4 Measurement Results

The plot of the sampled temperature and the number of the received packets using the introduced schemes in 6.6.2.3 are shown in Figures 6.16 and 6.17, respectively. These figures show how *LCS* trades off between the performance and cost. The summary of the results from these figures are:

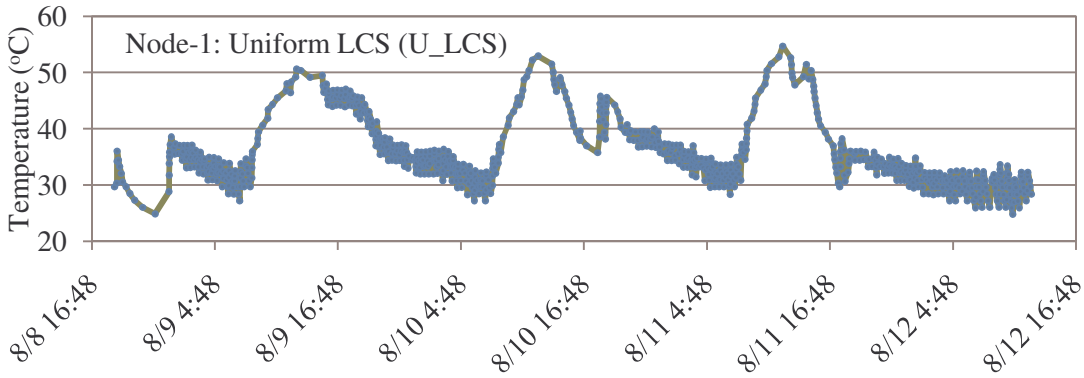


Figure 6.16.a: Sampled temperature with uniform *LCS* in an experiment

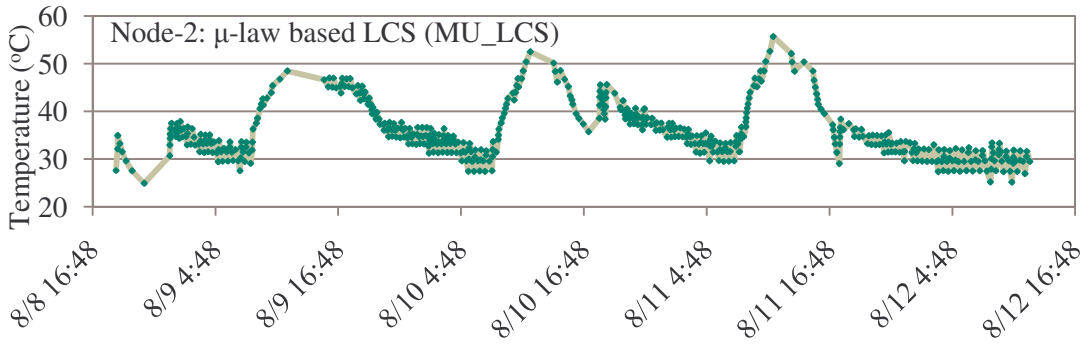


Figure 6.16.b: Sampled temperature with µ-law based *LCS* in an experiment

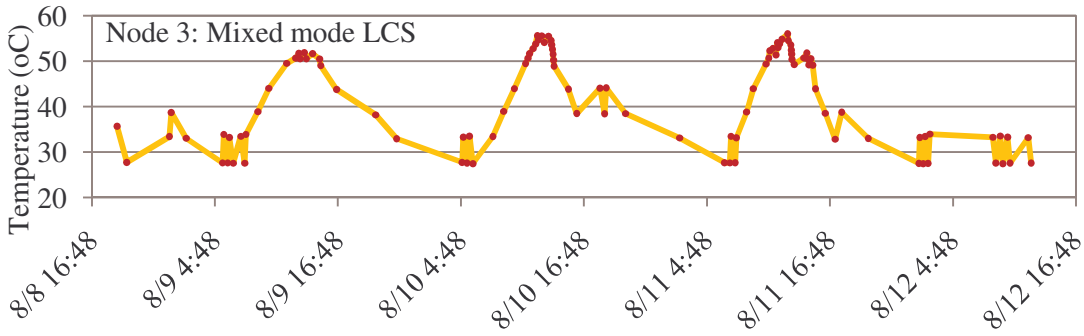


Figure 6.16.c: Sampled temperature with mixed mode *LCS* in an experiment

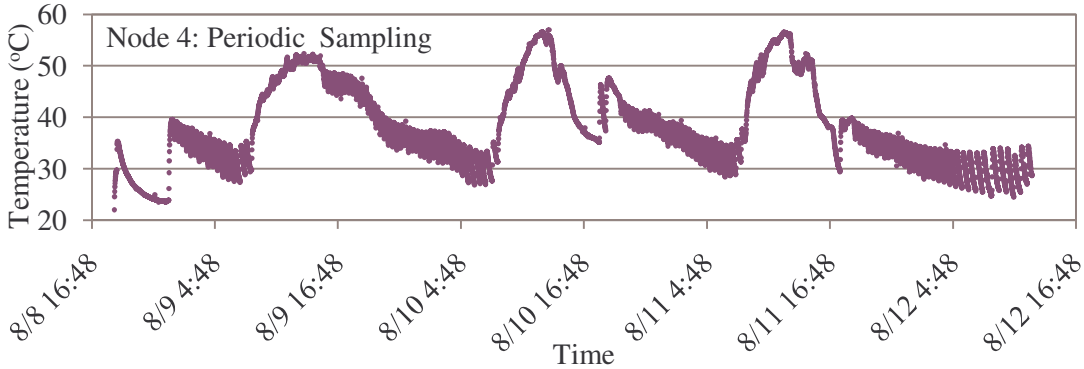


Figure 6.16.d: Periodically sampled temperature with in an experiment

- When the bandwidth of the signal is not known exactly, but the amplitude range is known, *LCS* based sampling help to reduce the sampling cost. This result is very helpful for sampling of roughly known signals using wireless sensor network.
- The reconstruction performance of *LCS* using μ-law based levels leads to higher reconstruction performance in comparison to uniform and the mixed mode *LCS*, provided the number of levels is chosen appropriately.

To compare the sampling performance of the introduced schemes, cardinal spline with adapted tension is used. Periodically oversampled signal is used as the reference and the sampling error is found after that interpolation of the spline curve is fitted. Accordingly, the error at time t_i is calculated based on the (6-29). In (6-29) ΔT is the period of the periodic samples S_P.

$$e(t_i) = | S_{LCS}(t_i) - S_P(t_n) | \quad , \qquad n = \arg \min |t_i - n\Delta T| \tag{6-29}$$

Table (6-2) shows the reconstruction mean square of error of *LCS* based sampling. Based on these results, μ-law based *LCS* and uniform *LCS* have almost the same performance and the mixed mode sensing has the highest error. For this comparison, the

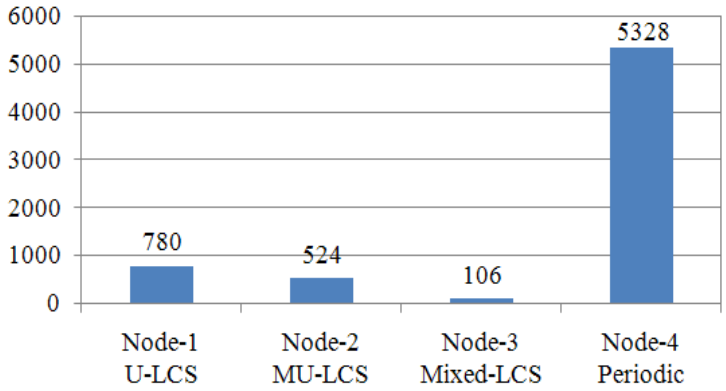


Figure 6.17: The number of the received packets in different scenarios

RMSE is calculated from (6-30).

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} e^2(t_i)} \tag{6-30}$$

Mixed mode *LCS* sampling helps in significantly reducing the communication cost. In spite of its very low cost, it needs very accurate knowledge of the signal's statistics to properly define the levels. As usually these statistics is not available, unlike the defined levels for this experiment, the uniformly spaced levels are recommended for the critical sensing range.

6.6.3 Results from ParadiseNet

LCS based sampling is used in a set of the deployed nodes in ParadiseNet. Figures 6.18.a and 6.18.b illustrate the plot of the sense temperature at two samples of these nodes. One of these nodes is with periodic sampling and the other one is with *LCS* with uniformly spaced levels as level spacing of 2 degree Celsius. These nodes were deployed at two different locations and sense the ambient temperature. These plots show that the nodes with *LCS* based sampling are able to characterize the signal with smaller cost.

For this example, the node with *LCS* has 57% less cost. However, this efficiency improvement is not comparable with the expected amount from the analysis and the experimental work. This can be justified with the parallel *LCS* based sampling for

Table 6-2: Comparison between the reconstruction error of the three *LCS* based sampling scenarios: Uniform *LCS*, μ-law based *LCS* and Mixed mode *LCS* after cardinal spline

	Uniform *LCS*	μ-law based *LCS*	Mixed mode *LCS*
RMSE	6.10	7.70	19.30

temperature and vibration. In other words, a set of the received packets are due to *LCS* based temperature sensing and the other group are due to *LCS* based vibration sensing.

6.7 Battery Consumption and Lifetime Calculations

We performed calculations of energy usage at the wireless sensor nodes under various network sizes and network characteristics. These calculations were validated by laboratory experiments. The basic approach for performing the analytical calculations of average energy consumption in a node is measurement of current consumption of the mote under various activity periods or events (such as packet transmit, receive, sense,

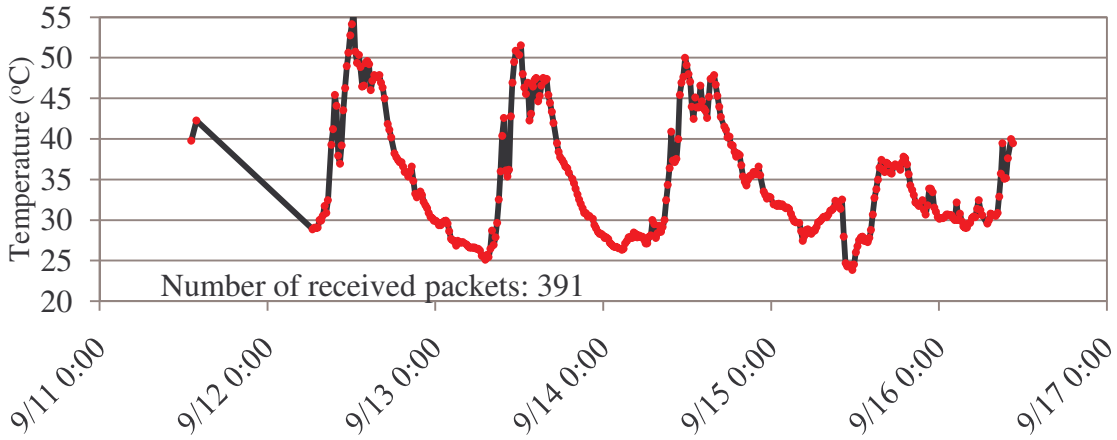


Figure 6.18.a: Plot of the sensed temperature at node number 43 (MTS310) from the periodically sampled data – The total number of the received packets is 391.

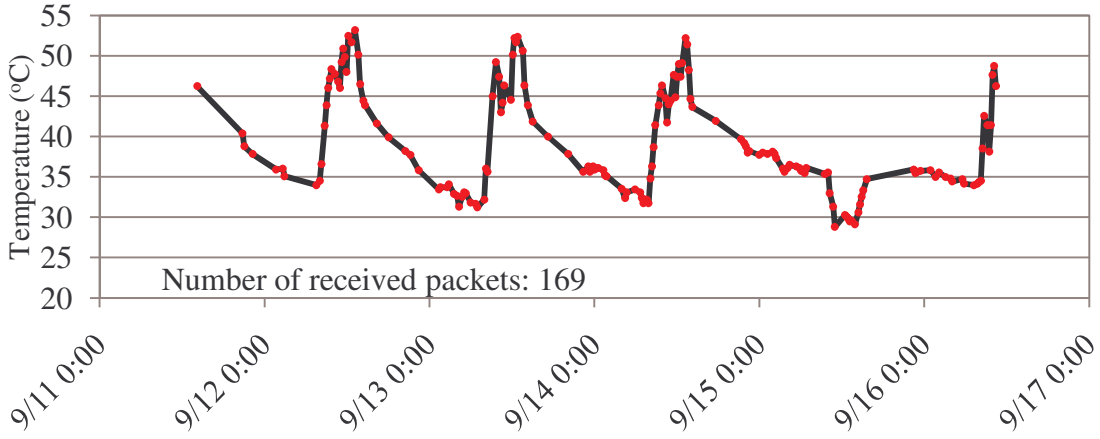


Figure 6.18.b: Plot of the sensed temperature at node number 68 (MTS310) from the periodically sampled data – The total number of the received packets is 169.

etc.) and then determining the occurrences of each of these events at the node under the assumed network characteristics. The current drawn by a *MICAz* mote in a typical temperature sensing node (with 6 *RTD* sensors interface to the mote using an *MDA300* board) under various events are shown in Figure 6-19. These events, the corresponding current consumptions and durations are show in Table 6-3.

Using these data, we can estimate the current consumption in the *MICAz* mote to be as follows [70]:

$$I = I_{R_t}\frac{T_{R_t}}{T_{RUI}} + I_{D_t}\frac{T_{D_t}}{T_D} + N(I_{R_r}\frac{T_{R_r}}{T_{RUI}} + I_{D_r}\frac{T_{D_r}}{T_D}) + I_S\frac{T_S}{T_{D_t}} + 8I_P T_P \qquad (6\text{-}31)$$

where I_x and T_x represent the current drawn and the duration of an event *x*, respectively, as described in Table 6-3, and I_{RUI} and T_D represent the route update interval (*RUI*) and data transmit intervals used, respectively.

With these assumptions the estimated battery life of the working wireless sensors in ParadiseNet, for the three *LCS* based sampling schemes in section 6.6.2.4 is depicted in

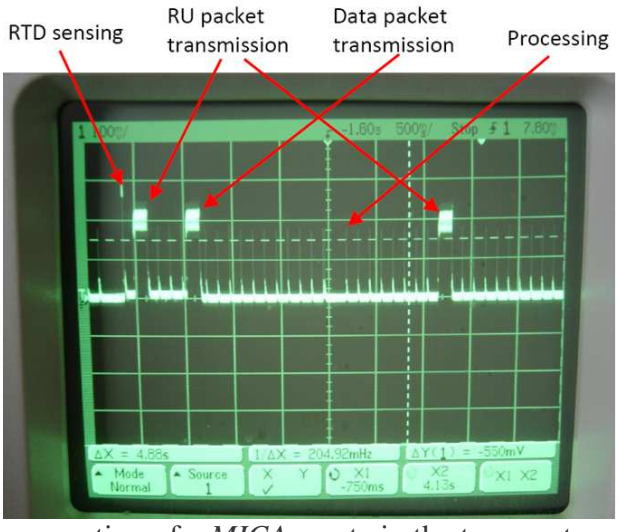


Figure 6.19: Current consumption of a *MICAz* mote in the temperature sensing (from [70]).

Figure 6-20. In this graph the route-update is 2 hours and the sampling period is 15 Min.

Table 6-3: *MICAz* mote in the temperature sensing application

Event	Current (mA)	Duration (ms)
Route update transmit R_t	27	180
Route update receive R_r	20	180
Data transmit D_t	27	185
Data receive D_r	20	185
Sensing S	10	10
Processing P	8	3

The assumed number of the nodes in the network is 150 nodes and it is assumed that the nodes are using a 5000 mAH battery. This graph also compares the battery lifetime of the wireless nodes with *LCS* based sampling and periodic sampling when the sampling period for all of the nodes is 15 minutes.

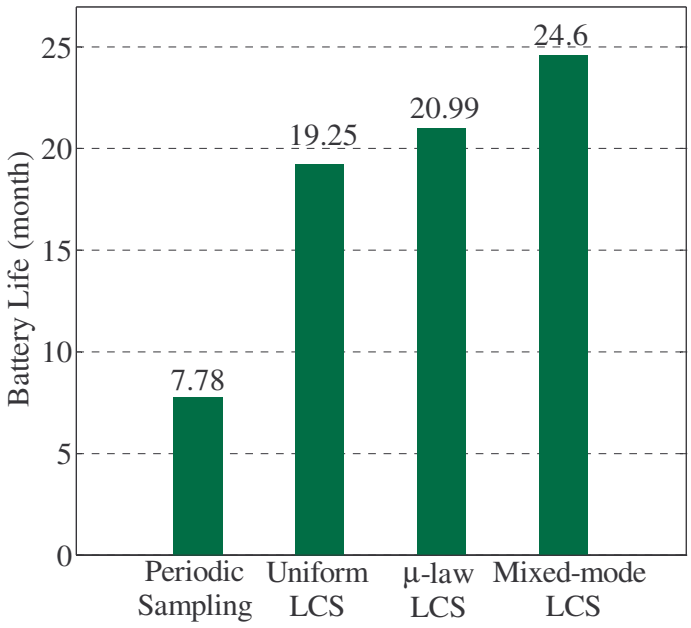


Figure 6.20: Estimation of the battery-life for the three *LCS* based sampling schemes.

6.8 Overview

In periodic sampling of the bandlimited signals, many of the consecutive samples are very similar and sometimes the signal remains unchanged over a period of time. These samples can be interpreted as redundant. For this, transmission of all of the periodic samples from all of the sensors in a large scale wireless sensor network is wasteful. Level based sampling in time is proposed for energy conservation in real-life application of large scale wireless sensor networks to increase the network lifetime by avoiding the transmission of redundant samples.

6.9 Conclusion

LCS for application in wireless sensor networks is proposed and discussed. The optimum set of levels to have the minimum reconstruction error is investigated. μ-law based *LCS* is proposed that unlike optimal *LCS* which needs the *pdf* of the signal, needs a few of the signal's statistics. The reconstruction error of the *LCS* based samples of correlated normalized Gaussian signal with sample and hold reconstructor is achieved numerically and using computer simulations for optimal, μ-law based, and uniform *LCS*. The minimum required number of levels for uniquely representing the *LCS* based sampled signal is investigated. It is shown that under conditions, the minimum required number of levels is independent from the spectral characteristics of the signal.

The minimum required periodic sampling rate for *LCS* based sampling is obtained on the characteristics of the signal. It is shown that this rate is proportional to the bandwidth, the power of the signal and the minimum spacing between the levels.

The reconstruction error and the cost of *LCS* based sampling with uniform, μ-law based and a mixture of these two schemes is experimentally tested and compared with

periodic sampling. It is shown that the performance of μ-law based is poorer than that uniform *LCS* after cardinal spline, but uniform *LCS* has slightly higher cost. This result does not match with the numerically achieved result that can be since of the difference between the statistics of the signal and the assumptions such as poor estimation of the statistical mode of the signal's distribution. Mixed mode *LCS* is applied by simultaneously using uniform and μ-law based *LCS* on two different ranges. The experimental results showed that the mixed mode *LCS* has much smaller cost in comparison to the uniform and μ-law based *LCS* schemes, but it has higher reconstruction error.

Comparison between the battery-life of the *MICAz* nodes with *LCS* based sampling and periodic sampling in the network shows that *LCS* is able to increase the network lifetime.

CHAPTER 7: CONCLUSION AND FUTURE WORK

7.1 Conclusion

7.1.1 Spatiotemporal Monitoring of Signal Distribution with Contours

Monitoring of spatial distributions using *WSN* is an active area of research. Modeling a spatial distribution with contours is an approach that has been proven to be prospective to economize the activities in *WSN*. So far, only limited work has been reported on contour detection and contour tracking algorithms for use in wireless sensor networks. Most of these algorithms assume that the sensor observations are noise-free and the tracking protocol is repeated, periodically. The research presented in this dissertation addresses these short comings.

Towards this objective, a collaborative distributed processing scheme is presented for detecting contours of the signal distribution in a localized, static sensor field. The proposed scheme uses a multi-level quantizer for emulating an edge in the signal distribution in the sensor field and then applies spatial filtering. The algorithm is particularly beneficial when the sensor observations are noise, which is common in sensor observations in the real world. The proposed scheme reduces error in contour detection caused by noise in signal observations and has a relatively low cost of communication.

The monitoring of a random, correlated spatial signal distribution in a large-scale wireless sensor network using a novel collaborative contour detection scheme is

explored. To evaluate the performance of monitoring, a simple piece-wise linear model is used in error calculation. To model the spatial distribution with contour levels, the optimum quantization levels of the spatial distribution are first determined from by applying the *Lloyd-Max* algorithm. These optimum results are compared with those obtained using uniformly spaced contour levels and a proposed heuristic non-uniform level selection schemes that is borrowed from μ-law based companding. Performance evaluations of spatial monitoring of random signal distribution with contours using these three level selection schemes show that monitoring with optimal levels has minimum error and maximum cost, and uniformly spaced levels has highest error and minimum cost. It is shown that when the desired reconstruction error is small, monitoring of a spatial distribution with the proposed non-uniformly spaced contour levels leads to minimal cost provided that the number of contour levels is chosen appropriately. The optimum detection parameters such as radio range, detection threshold, and the number of quantization levels that minimize the error are derived by using computer simulations. Besides these, the effect of noise intensity, collaboration margin, and the number of contour levels for the three types of the levels are investigated.

To avoid periodically repeating the spatial contour detection, a contour tracking protocol is proposed to monitor the temporal variations of the spatial distribution of the signal. To reduce the communication costs of the tracking protocol, space oriented and time oriented filtering schemes are proposed. The novel space oriented filtering uses *a-priori* information of the existing contour or edge nodes to select a subset of the sensors to try the contour detection test. Two new approaches, namely, the gradient compass (GC) and estimation of the displacement (ED) are proposed for space oriented filtering.

Space oriented filtering, although is able to improve the cost efficiency, is not able to estimate the most appropriate time for contour updates. Hence, it needs to be repeated periodically. Unlike space oriented filtering, time oriented filtering is able to estimate the most appropriate contour update times, and applies the collaborative contour tracking procedure only in the broken segment of the contour. An adaptive time update (ATU) and normalized adaptive time update (NATU) are also proposed to perform time oriented, efficient contour tracking.

For performance evaluation, two scenarios are considered. First, a contour line of a Gaussian distribution is tracked. Later, a situation is considered where the algorithm tracks the contour lines of the superposition of multiple Gaussian signals where just a portion of the superposed distribution varies. Computer simulations show that the space and time oriented filtering schemes is able to considerably reduce the contour tracking cost.

7.1.2 Experimental Application of Level Crossing Sampling (*LCS*) in *WSN*

LCS for experimental applications in *WSN* is proposed. Several issues like the effect of the selected set of sampling levels, the minimum number of required sampling levels, and the maximum sampling period that guarantees required performance bounds using *LCS* are explored for correlated random signal with Gaussian distribution. The reconstruction error of the *LCS* based samples of correlated normalized Gaussian signal with sample and hold reconstructor is achieved numerically and using computer simulations for optimal, μ-law based, and uniform *LCS*. The reconstruction is repeated with adapted tension cardinal spline for the three types of the sampling levels. The simulation results show that, similar to the spatial monitoring of distribution, the optimal

levels that leads the minimum reconstruction error has the highest cost but low reconstruction error, and the uniformly spaced levels has the highest error and the minimum cost.

LCS is implemented over *MICAz* wireless platform for experimental applications. Three different types of levels namely uniformly spaced levels, μ-law based levels and a combined version of these two types are tested. Experimental results from implementation of *LCS* show that in comparison to periodic sampling, *LCS* may highly reduce the sampling cost, when the range of the signal is known. These results show that if the statistical mode of the signal is not accurately known in the definition of the μ-law based levels, it may have poorer performance than uniformly spaced levels. The collected data from the recently deployed wireless sensor nodes with *LCS* in an experimental WSN deployed in a power substation in Kentucky (ParadiseNet) also prove that this method of sampling is cost efficient for application in *WSN*.

7.2 Future Work

7.2.1 Spatiotemporal Monitoring of Signal Distribution with Contours

The following are proposed for the future work in spatiotemporal monitoring of signal distribution with contours:

- Cost evaluation of the proposed algorithms along with the other issues in the network like fault in sensors, collision, and network layer costs (for an assumed MAC protocol).
- Performance evaluation of the proposed filter-based contour detection and tracking algorithm when channel imperfections are considered.

- Distributed and adaptive selection of the detection threshold from the spatial distribution characteristics.
- Adaptive selection of the effective radio range from the characteristics of the signal and based on the node density.

7.2.2 Experimental Application of *LCS* in *WSN*

The collected data from the deployed nodes with *LCS* based sampling in ParadiseNet may be used to prepare an energy consumption model for the network. Besides this model, the performance of the sampled quantity in the long term is needed to be studied when nodes with periodic sampling regime, relay the packet of the *LCS* based nodes.

REFERENCES

I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Journal of Computer Networks*, Elsevier, pp. 393-422, Dec 2001.

J. McCulloch, P. McCarthy, S. M. Guru, W. Peng, D. Hugo, and A. Terhorst, "Wireless sensor network deployment for water use efficiency in irrigation," *in the Proceedings of the workshop on Real-world wireless sensor networks* (REALWSN'08), pp. 46-50, 2008.

N. Alsharabi, R. F. Li, Z. Fan, and M. Ghurab, "Wireless sensor networks of battlefields hotspot: Challenges and solutions," *in the proceedings of Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks and Workshops, 2008. WiOPT 2008. 6th International Symposium on*, pp. 192-196, April 2008.

N. Xu, S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin, "A wireless sensor network For structural monitoring," *in the proceedings of the 2nd international conference on Embedded networked sensor systems* (sensys'04), pp. 13-24, 2004.

A. Senart, M. Karpinski, M. Wieckowski and V. Cahill, "Using Sensor Networks for Pedestrian Detection," *in the proceedings of Consumer Communications and Networking Conference, CCNC'08, 2008.*

P. Juang, H. Oki, Y.Wang, M.Martonosi, L. S. Peh, and D. Rubenstein. "Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet," In Proceedings of ASPLOS-X, pages 96–107, 2002.

A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson. "Wireless sensor networks for habitat monitoring," *In Proceeding of WSNA*, pages 88–97, 2002.

R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler "An analysis of a large scale habitat monitoring application," *In Proceedings of SenSys*, pages 214–226, 2004.

G. Werner-Allen, K. Lorincz, M. Welsh, O. Marcillo, J. Johnson, M. Ruiz, and J. Lees "Deploying a wireless sensor network on an active volcano," *IEEE Internet Computing*, 10(2):18–25, 2006.

Asis Nasipuri, Robert Cox, Hadi Alasti, Luke Van der Zel, Bienvenido Rodriguez, Ralph McKosky, Joseph A. Graziano, "Wireless sensor network for substation monitoring: design and deployment," *in the Proceedings of sensys'08*, pages 365-366, 2008.

Xiaojuan Chao; Dargie, W.; Guan Lin, "Energy Model for H2S Monitoring Wireless Sensor Network," *in the Proceedings of Computational Science and Engineering, 2008. CSE '08. 11th IEEE International Conference on*, pages 402-409, July 2008.

L. Krishnamurthy, R. Adler, P. Buonadonna, J. Chhabra, M. Flanigan, N. Kushalnagar, L. Nachman, M. Yarvis, "Design and deployment of industrial sensor networks: experiences from a semiconductor plant and the north sea," *in the Proceedings of snsys'05,* pages 64-75, 2005.

James W. Fonda, Maciej J. Zawodniok, S. Jagannathan, Al Salour, Donald Miller Jr, "Missouri S&T Mote-Based Demonstration of Energy Monitoring Solution for Network Enabled Manufacturing Using Wireless Sensor Networks (WSN)," *in the Proceedings of IPSN'08*, Pages 559-560, 2008.

H. B. Lim, K. V. Ling, W. Wang, Y. Yao, M. Iqbal, B. Li, X. Yin, T. Sharma, "The national weather sensor grid," *in the Proceedings of the 5th international conference on Embedded networked sensor systems* (SenSys'07), 2007.

F. Zhao, L. Guibas, "Wireless Sensor Networks: an information processing approach," *Morgan Kaufmann publisher*, 2004.

A. Swami, Q. Zhao, Y.-W. Hong, L. Tong, "Wireless Sensor Networks: Signal Processing and Communications," *Wiley*, 2007.

D. Vassis, P. Belsis, C. Skourlas, G. Pantziou, "A pervasive architectural framework for providing remote medical treatment," *in the Proceedings of Proceedings of the 2nd International Conference on PErvsive Technologies Related to Assistive Environments* (PETRA'08), 2008.

J. Lian, L. Chen, K. Naik, Y. Liu, and G. B. Agnew, "Gradient Boundary Detection for Time Series Snapshot Construction in Sensor Networks," *IEEE TRANS. on Parallel*

M. Greitans, "Processing of Non-stationary Signals Using Level-Crossing Sampling", in *Proceedings of International Conference on Signal Processing and Multimedia Applications*, pp 170-177, 2006.

S.M. Qaisar, L. Fesquet, M. Renaudin, " Adaptive Rate Filtering for a Signal Driven Sampling Scheme", in *Proceedings of IEEE International Conference on Acoustic, Speech and Signal processing*, vol. 3, pp. 1465-1468, 2007.

K. K. Chintalapudi and R. Govindan, "Localized edge detection in sensor fields," in *Proceedings of IEEE international Workshop on Sensor Network Protocols and Applications*, pp. 59-70, 2003.

Pie-Kai Liao, Min-Kuan Chang and C.-C. Jay Kuo, "Distributed Edge Sensor Detection with One-And Two-Level Detections," in *Proceedings IEEE ICASSP*, pp. 297-330, 2004.

Pei-Kai Liao, Min-kuan Chang and C.-C. Jay Kuo, "Distributed edge detection with composite hypothesis test in wireless sensor networks," in *Proceedings of IEEE Globecom* 2004.

Pie-Kai Liao, Min-Kuan Chang and C.-C. Jay Kuo, “Statistical Edge Detection with Distributed Sensor under the Neyman-Pearson (NP) Optimality,” in *Proceedings IEEE VTC*, 2006.

Pie-Kai Liao, Min-Kuan Chang and C.-C. Jay Kuo, “Contour line extraction in a multi-modal field with sensor networks," *Proceedings IEEE Global Telecommunication Conference (Globecom)*, 2005.

Pei-Kai Liao, Min-Kuan Chang and C.-C. Jay Kuo, "A distributed approach to contour line extraction using sensor networks," in *Proc. IEEE VTC*, 2005.

Pei-Kai Liao, Min-Kuan Chang and C.-C. Jay Kuo, "Contour line extraction with wireless sensor networks," in *Proc. IEEE International Conference on Communications (ICC)*, 2005.

C. Jiang, G. Dong, B. Wang, “Detection and Tracking of Region-Based Evolving Targets in Sensor Networks,” in *Proceedings IEEE ICCCN*, 2005.

Pei-Kai Liao, Min-Kuan Chang and C.-C. Jay Kuo, “A Cross-Layer Approach to Contour Nodes Inference with Data Fusion in Wireless Sensor Networks,” *in the Proceedings of Wireless Communications and Networking Conference (IEEE WCNC’07)*¸ pages 2773-2777, 2007.

I. Solis, K. Obraczka, “Isolines: energy-efficient mapping in sensor networks,” in the *Proceedings of the 10th IEEE Symposium on of Computers and Communications, ISCC’05*, pages 379-385, 2005.

Y. Liu; M. Li, “Iso-Map: Energy-Efficient Contour Mapping in Wireless Sensor Networks,” in the *Proceedings of Distributed Computing Systems, ICDCS '07. 27th International Conference on*, pages 36-42, 2007.

C. Buragohain, S. Gandhi, J. Hershberger, and S. Suri, “Contour Approximation in Sensor Networks,” DCOSS '06, June 18-20, 2006.

S. Gandhi, J. Hershberger, and S. Suri, “Approximate Iso-contours and Spatial Summaries in Sensor Networks,” IPSN'07, April 25-27, 2007.

R. Sarkar, X. Zhu, J. Gao, L.J. Guibas, J.S.B. Mitchell, “Iso-Contour Queries and Gradient Descent with Guaranteed Delivery in Sensor Networks,” in *Proceedings of IEEE INFOCOM’08*, 2008.

S. Gandhi, J. Hershberger, S. Suri, “Approximate Isocontours and Spatial Summaries for Sensor Networks,” in *Proceedings of IPSN’07*, 2007.

K. Seo, and S.Y. Lee, “Edge Detection Using Orientation-Independent Operators in Large-Scale Sensor Networks,” in *Proceedings of Future generation communication and networking (FGCN’07)*, pp. 124-129, 2007.

S. Duttagupta, K. Ramamritham, P. Kulkarni, and K. M. Moudgalya, "Tracking Dynamic Boundary Fronts usig Range Sensors," in *Proceedings of European Wireless Sensor Network (EWSN'08)*, 2008.

X. Zhu, R. Sarkar, J. Gao, J.S.B. Mitchell, "Light-Weight Contour Tracking in Wireless Sensor Networks," in *Proceedings of INFOCOM'08*, 2008.

J.R. Jiang, J.W. Wu, G.S. Du, "Energy-Efficient and Traffic-Dispersive Event Contour Tracking in Multi-sink Wireless Sensor Networks," in *Proceedings of IEEE APSCC'08*, pp. 1120 – 26, 2008.

M. Miskovicz, "Efficiency of Level-Crossing Sampling for Bandlimited Gaussian Random Process", in *Proc. IEEE Intl. Workshop on Factory Communication Systems*, pp. 137-142, June 2006.

K. Guan, A.C. Singer, "Opportunistic Sampling by Level-Crossing," in *Proceedings of IEEE international conference on acoustic, speech and signal processing (ICASSP'07),* vol 3, pp: 1513-1516, 2007.

M. Greitans, R. Shavelis, "Speech sampling by level-crossing and its reconstruction using spline-based filtering", in *Proceedings of EURASIP Conference on Speech and Image Processing, Multimedia Communications and Services*, pp. 292-295, 2007.

N. Sayiner, H.V.Sorensen, T.R. Viswanathan, "A level-crossing sampling scheme for A/D conversion", *IEEE Transactions of Circuits and Systems*, vol. 43, pp. 335-339, 1996.

K.M.Guan, S. S. Kozat, and A.C. Singer, "Adaptive Reference Levelss in a Level-Crossing Analog-to-Digital Converter," *EURAIP Journal of Advances in Signal Processing*, Volume 2008,

K. M. Guan, A. C. Singer, "Opportunistic Sampling of Bursty Signals by Level-Crossing – an Information Theoretical Approach", in *Proceeding of Conference on Information Science and Systems*, pp.701-707, March 2007.

K. Guan, A.C. Singer, "A Level-Crossing Sampling Scheme for Non-Bandlimited Signals", in *Proceedings of International Conference on Acoustic, Speech and Signal processing*, vol. 3, pp. 381-83, 2006.

K. M. Guan, A. C. Singer, "Opportunistic Sampling of Bursty Signals by Level-Crossing – an Information Theoretical Approach", in *Proceeding of Conference on Information Science and Systems*, pp.701-707, March 2007.

J.W. Mark , T.D. Todd, "A Nonuniform Sampling Approach to data Compression", *IEEE Transactions on Communications*, vol.29, pp 24-32, January 1982.

S. M. Qaisar, L. Fesquet, M. Renaudin, "An Adaptive Resolution Computationally Efficient Short-Time Fourier Transform," *EURAIP Research letters in signal processing*.

F. Marvasti, Nonuniform Sampling. Kluwer Academic, 2001.

K. Sayood, Introduction to Data Compression. Morgan Kaufmann, 2000.

W. A. Gardner, Introduction to Random Processes: with applications to signals and systems. Macmillan, 1986.

H. Alasti, AW.A. Armstrong, A. Nasipuri, "Performance of a Robust Filter-based Approach for Contour Detection in Wireless Sensor Networks," in *Proceedings of ICCCN'07*, pages 159-164, 2007.

"Digital Image Processing, 3rd edition" by William K. Pratt, John Wiley & Sons, Inc., 2001.

Y. Sung, H.V. Poor, H. Yu, "How Much Information Can One Get From a Wireless Ad Hoc Sensor Network Over a Correlated Random Field?," *IEEE Trans. on Information Theory*, Vol. 55, No. 6, June 2009.

____, "Optimal node density for two-dimensional sensor arrays," in *Proceedings of the 5th IEEE Sensor Array and Multichannel Signal Processing Workshop, SAM'08,* pages: 271-274, 2008.

A. Nordio, C.F. Chiasserini, E. Viterbo, "Performance of Linear Field Reconstruction Techniques With Noise and Uncertain Sensor Locations," in IEEE Trans. on Acoustic Speech and Signal Processing, Vol. 56, Issue, 8, Part 1, pages: 3535-3547, Aug. 2008.

____, "Quality of Field Reconstruction in Sensor Networks," in *Proceedings of IEEE conference on computer communications INFOCOM'07*, pages: 2406 – 2410, 2007.

____, "Reconstruction of Multidimensional Signals from Irregular Noisy Samples," in IEEE Trans. on Signal Processing, Vol. 56, Issue 9, page: 4274 – 4285, 2008.

____, "The Impact of Quasi-equally Spaced Sensor Layouts on Field Reconstruction," in *Proceedings of International symposium on Information Processing in Sensor Networks (IPSN'07)*, pages: 274-282, 2007.

____, "On quasi-equally spaced sampling in wireless sensor networks," in *Proceedings of the 19th international symposium on personal indoor and Mobile Radio Communications (PIMRC'08)*, pages: 1-5, 2008.

A. Jindal, K. Psounis, "Modeling spatially-correlated data of sensor networks with irregular topologies," in *Proceedings of IEEE International Conference on Sensor and Ad hoc Communications and Networks (SECON'05),* Santa Clara, CA, September 2005.

____, "Modeling Spatially Correlated Data in Sensor Networks," *ACM Transactions on Sensor Networks*, November 2006.

S. Pattem, B. Krishnmachari, and R. Govindan, "The impact of spatial correlation on routing with compression in wireless sensor networks," in *Symposium on Information Processing in Sensor Networks (IPSN)*, Apr. 2004.

J. Faruque and A. Helmy, "Rugged: Routing on fingerprint gradients in sensor networks," in *IEEE International Conference on Pervasive Services (ICPS'2004)*, July 2004.

M. C. Vuran and I. F. Akyildiz, "Spatial correlation-based collaborative medium access control in wireless sensor networks," in *IEEE/ACM Trans. on Networking*, Vol. 14, No. 2, 2006.

Z. Ma and W. Krings, "Spatial Distribution Patterns, Power Law, and the Agent-based Directed Diffusion Sensor Networks," in the proceedings of the sixth international IEEE conference on pervasive computing and communications (Percom'08), pages: 596-601, 2008.

B.Quach, M.Balakrishnan, D.Benhaddou, X. Yuan, "Implementation of integrated wireless health monitoring network", *in the Proceedings of Proceedings of the 1st ACM international workshop on Medical-grade wireless networks* (WiMD'09), 2009.

E. Becker, V. Metsis, R. Arora, J. Vinjumur, Y. Xu, F. Makedon, "SmartDrawer: RFID-based smart medicine drawer for assistive environments," *in the Proceedings of Proceedings of the 2nd International Conference on PErvsive Technologies Related to Assistive Environments* (PETRA'09), 2009.

"Substation-Wide Monitoring Through Applications of Networked Wireless Sensor Devices", Phase-II: Scalability and Sustainability Studies", EPRI report, 2008.

APPENDIX A: OPTIMAL LEVELS FOR LEVEL CROSSING SAMPLING

The optimal set of levels is the particular solution of equation (6-9) that minimize the equation (6-8). Equation (6-8) is repeated here in (A-1). For this, we take partial derivative with respect to each of the levels and then put it equal to zero as is shown in (A-2) to find its optimal roots.

$$E[\varepsilon_S^2(t)] = \int_{\ell_0}^{\ell_1} (\ell_1 - s)^2 f_S(s)\, ds + \int_{\ell_M}^{\ell_{M+1}} (\ell_M - s)^2 f_S(s)\, ds + \sum_{i=2}^{M} \frac{1}{2} \left\{ \int_{\ell_{i-1}}^{\ell_i} (\ell_{i-1} - s)^2 f_S(s) + (\ell_i - s)^2 f_S(s)\, ds \right\} \tag{A-1}$$

$$\frac{\partial E[\varepsilon_S^2(t)]}{\partial \ell_i} = 0, \quad 1 \le i \le M \tag{A-2}$$

Equation (A-1) is re-written as follows:

$$E[\varepsilon_S^2(t)] = \int_{\ell_0}^{\ell_1} (\ell_1^2 + s^2 - 2s\ell_1) f_S(s)\, ds + \int_{\ell_M}^{\ell_{M+1}} (\ell_M^2 + s^2 - 2s\ell_M) f_S(s)\, ds + \frac{1}{2} \sum_{i=1}^{M-1} \left[2 \int_{\ell_i}^{\ell_{i+1}} s^2 f_S(s)\, ds + (\ell_i^2 + \ell_{i+1}^2) \int_{\ell_i}^{\ell_{i+1}} f_S(s)\, ds - (\ell_i + \ell_{i+1}) \int_{\ell_i}^{\ell_{i+1}} s f_S(s)\, ds \right] \tag{A-3}$$

When the level is not the first and the last level, the partial derivative of the (A-1) with respect to the level is as follows:

$$\left. \frac{\partial E[\varepsilon_S^2(t)]}{\partial \ell_i} \right|_{i \neq 1, M} = \ell_i \int_{\ell_{i-1}}^{\ell_{i+1}} f_S(s)\, ds - \int_{\ell_{i-1}}^{\ell_{i+1}} s\, f_S(s)\, ds + \frac{1}{2} \ell_{i-1}^2 f_S(\ell_i) - \ell_i \ell_{i-1} f_S(\ell_i) - \frac{1}{2} \ell_{i+1}^2 f_S(\ell_i) + \ell_{i+1} \ell_i f_S(\ell_i) \tag{A-4}$$

$$\ell_i = \frac{\int_{\ell_{i-1}}^{\ell_{i+1}} s\, f_S(s)\, ds + \frac{1}{2} f_S(\ell_i) \left[(\ell_{i+1} - \ell_{i-1})(\ell_{i+1} + \ell_{i-1} - 2\ell_i) \right]}{\int_{\ell_{i-1}}^{\ell_{i+1}} f_S(s)\, ds}, \quad i = 2,3,\ldots,M \tag{A-5}$$

Equation (A-5) shows the level with respect to itself and the other neighboring levels. In this equation then the second term of the numerator becomes too small because of

$(\ell_{i+1} + \ell_{i-1} - 2\ell_i)$, then this term is ignored in comparison to $\int_{\ell_{i-1}}^{\ell_{i+1}} s\, f_S(s)\, ds$. Hence, (A-5) is simplified to (A-6), which is the *Lloyd-Max* equation.

$$\ell_i = \frac{\int_{\ell_{i-1}}^{\ell_{i+1}} s\, f_S(s)\, ds}{\int_{\ell_{i-1}}^{\ell_{i+1}} f_S(s)\, ds} \quad , i = 2,3,\ldots,M \tag{A-6}$$

We repeate the same process for the first level as follows:

$$\frac{\partial E[\varepsilon_S^2(t)]}{\partial \ell_1} = \ell_1 \int_{\ell_0}^{\ell_1} f_S(s)\, ds + \frac{1}{2}\ell_1^2\, f_S(\ell_1) - \int_{\ell_0}^{\ell_1} \mathrm{s}\, f_S(s)\, ds - \frac{1}{2}\ell_1^2\, f_S(\ell_1) \tag{A-7}$$

$$+ \ell_1 \int_{\ell_0}^{\ell_2} f_S(s)\, ds - \int_{\ell_0}^{\ell_2} \mathrm{s}\, f_S(s)\, ds - \frac{1}{2}\ell_2^2\, f_S(\ell_1) + \ell_1\ell_2\, f_S(\ell_1)$$

$$\frac{\partial E[\varepsilon_S^2(t)]}{\partial \ell_1} = 0 \;\Rightarrow$$

$$\ell_1 = \frac{\int_{\ell_0}^{\ell_1} s\, f_S(s)\, ds + \int_{\ell_0}^{\ell_2} s\, f_S(s)\, ds + \frac{1}{2}(\ell_1 - \ell_2)^2 f_S(\ell_1)}{\int_{\ell_0}^{\ell_1} f_S(s)\, ds + \int_{\ell_0}^{\ell_2} f_S(s)\, ds} \tag{A-8}$$

And for the last level, we repeat the same process:

$$\frac{\partial E[\varepsilon_S^2(t)]}{\partial \ell_M} = \ell_M \int_{\ell_{M-1}}^{\ell_{M+1}} f_S(s)\, ds + \frac{1}{2}\ell_1^2\, f_S(\ell_1) - \int_{\ell_{M-1}}^{\ell_{M+1}} \mathrm{s}\, f_S(s)\, ds + \frac{1}{2}\ell_{M-1}^2\, f_S(\ell_M) \tag{A-9}$$

$$+ \ell_M \int_{\ell_M}^{\ell_{M+1}} f_S(s)\, ds - \frac{1}{2}\ell_M^2\, f_S(\ell_M) - \int_{\ell_M}^{\ell_{M+1}} \mathrm{s}\, f_S(s)\, ds + \frac{1}{2}\ell_M^2\, f_S(\ell_M)$$

$$\frac{\partial E[\varepsilon_S^2(t)]}{\partial \ell_M} = 0 \;\Rightarrow$$

$$\ell_M = \frac{\int_{\ell_M}^{\ell_{M+1}} s\, f_S(s)\, ds + \int_{\ell_{M-1}}^{\ell_{M+1}} s\, f_S(s)\, ds - \frac{1}{2}(\ell_M - \ell_{M-1})^2 f_S(\ell_M)}{\int_{\ell_M}^{\ell_{M+1}} f_S(s)\, ds + \int_{\ell_{M-1}}^{\ell_{M+1}} f_S(s)\, ds} \tag{A-10}$$

The optimal levels are the roots of the M simultaneous integral equations of (A-6), (A-8) and (A-10).

APPENDIX B: SELECTED EXPERIMENTAL SAMPLING LEVELS

#	Uniform LCS		μ-law based LCS		Mixed mode LCS
1	21.2		25.21		24.83
2	2.42		27.55		29.65
3	23.6		29.65		34.48
4	24.9		31.55		39.3
5	26.1		33.25		44.13
6	27.3		34.79		45.22
7	28.5		36.17		46.23
8	29.7		37.41		47.17
9	30.9		38.53		48.06
10	32.1		39.54		48.87
11	33.3		40.46		49.64
12	34.6		41.47		50.36
13	35.8		42.59		51.13
14	37		43.83		51.95
15	38.2		45.21		52.83
16	39.4		46.75		53.77
17	40.6		48.45		
18	41.8		50.35		
19	43		52.45		
20	44.2		54.79		
21	45.5				
22	46.7				
23	47.9				
24	49.1				
25	50.3				
26	51.5				
27	52.7				
28	53.9				
29	55.2				
30	56.4				
31	57.6				
32	58.8				