Performance of a Robust Filter-based Approach for Contour Detection in Wireless Sensor Networks

Hadi Alasti, William A. Armstrong, Jr., and Asis Nasipuri
Department of Electrical and Computer Engineering
The University of North Carolina at Charlotte, Charlotte NC
halast@uncc.edu, warstr2@bellsouth.net, anasipur@uncc.edu

***Abstract*—A robust filter-based approach is proposed for wireless sensor networks for detecting contours of a signal distribution over a 2-dimensional region. The motivation for contour detection is derived from applications where the spatial distribution of a signal (such as temperature, soil moisture level, etc.) is to be determined over a large region with minimum communication cost. The proposed scheme applies multi-level quantization to the sensor signal values to artificially create an edge and then applies spatial filtering for edge detection. The spatial filter is localized and is based on an adaptation of the Prewitt filter used in image processing. Appropriate mechanisms are introduced that minimizes the cost for communication required for collaboration. Simulation results are presented to show the error performance of the proposed contour detection scheme and the associated communication cost (single-hop communications with immediate neighborhood in average) in the network.**

Keywords- wireless sensor networks; collaborative processing; contour detection; Prewitt filter.

I. INTRODUCTION

A key design objective in wireless sensor networks is to derive benefits from the *collective* processing power of a large number of energy and hardware constrained wireless sensor nodes that are distributed over a region of interest. The type of information required depends on the application, which includes obtaining periodic signal levels such as in environmental monitoring, generating alarms based on specific signal conditions, tracking a mobile target within an area of interest, and more. However, in almost all cases, the combination of information from a group of sensors can make the system more robust and accurate. The main challenge is to design distributed collaborative information processing schemes under the limited processing, energy and communication capabilities of the small sensor nodes.

This paper addresses the design of collaborative processing algorithms for detecting *contours* of a signal distribution in a sensor field. Contour detection is useful in a large number of environmental monitoring applications. A typical application scenario is where spatial variations of a signal distribution need to be detected over a large region using an array of sensors. To minimize the communication cost for obtaining periodic signal samples from all sensors in the region, the spatial distribution can be estimated from a set of contours corresponding to specific signal levels [3]. Our approach for contour detection is to use the contour level as a threshold to introduce an artificial edge that demarcates two different regions within the sensor field (on either sides of the contour) and then apply a filter-based edge detection algorithm. The algorithm uses collaborative processing among a group of randomly distributed sensor nodes in the region of interest. Appropriate measures are incorporated to ascertain that the absence of a "true edge", i.e. a region characterized by a sharp variation of the signal level, does not introduce much error in estimating the contour. Primary design considerations are minimization of error of estimated contour and the communication cost incurred for collaborative processing.

This paper is organized as follows. In section II, we review existing work on edge detection in sensor networks that are related to this work. In section III, we describe the distributed contour detection problem targeted in this paper. In section IV, the filter-based approach for edge detection and issues on robustness and communication cost are described. Performance results of the proposed scheme obtained from computer simulations are presented in section V. In section VI, we present our conclusions and describe some future work on edge tracking.

II. RELATED WORK

Edge detection has been widely researched for image processing, where signal levels are available at regularly located sample points, i.e. pixels, and the algorithm assumes the availability of information from all data points for processing [8]. Consequently, adaptations of efficient edge detection algorithms from image processing for applications in sensor networks have been explored in literature. The primary goal of these algorithms is to detect *edge nodes*, i.e. nodes that are located within a certain tolerance distance from the ideal edge of the signal that is to be determined.

Chintalapudi, et al. [1] proposed three general approaches for localized edge detection: a) a statistical approach, b) a filter-based approach and c) a classifier-based approach. While the statistical approach does not require location information at the sensor nodes, the other two can only function when all nodes are aware of their geographic locations. The authors

showed that although the statistical approach is more robust in the presence of noise, it has higher errors and is more difficult to apply in practice due to difficulties in selecting a proper threshold. Liao, et al. [2] proposed an approach that is based on one and two level decisions based on the local and global maximum likelihood ratio to enhance the statistical approach proposed by [1] for edge detection. Their work presented in [4] enhanced the detection of the edge region in statistical approach using *Neyman-Pearson* (NP) criteria. Based on NP criteria they proposed an idea that addresses to solve the threshold selection problem. They also compared the performance of their approach with classifier-based approach under the assumption of location error and show that their statistical approach performs better than classifier approach. The works in [5-7] propose the usage of 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 [5-7] is based on regional clustering and having communication between cluster-heads. It can track changes, but is not a localized approach and some sensor nodes require additional capabilities to be a cluster-head.

In this work, we generalize the notion of the binary edge described in [1] to a contour line that is described with multiple levels and show that by this assumption the probabilities of missed and false detections of edge nodes are reduced. We evaluate the error in estimating a contour by measuring the distance of the detected edge nodes from the true contour. In addition, we determine the expected savings in communication cost of the proposed system. Although our approach has been evaluated using both the statistical and filter-based algorithms, in this paper we present results that are obtained primarily using the filter-based approach due to its superior performance. A comparison of the performances of the two approaches is also shown.

III. Problem Statement and Approach used for Contour Detection in Sensor Networks

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 a given area. A contour in the sensor field is defined as the line of demarcation between regions that are above and below a threshold S_0. For the sake of estimating errors in estimation, we use a tolerance distance r to determine the contour thickness, which is the region near a contour such that a sensor node located in the region is termed as an edge node. This is illustrated in Figure 1. There can be two types of decision errors. When a node decides that it is an edge node when it is not within a distance r from the true contour, we have a false detection. On the other hand, when a node that is actually an edge node (i.e. located within a distance r from the true contour) fails to be detected, we have a missed detection.

The sensor nodes perform local and collaborative processing of their observed signal samples to determine if they are edge nodes. Errors in detecting edge nodes can occur due to

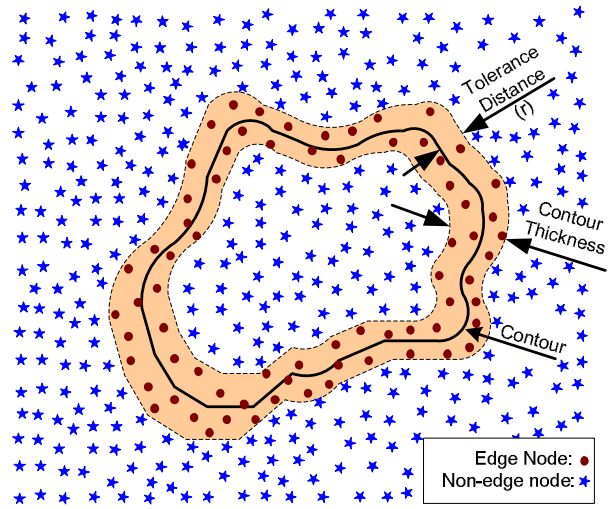

Figure 1: Illustration of a contour in a sensor field and edge sensors near it.

two reasons. First, although the actual edge or boundary demarcates two regions having different signal properties, the true distribution of the signal across the edge may not have a sharp change. In this sense, a contour detection problem is different from edge detection. Secondly, quantization noise and other environmental noise sources introduce additional scope of detection errors.

The primary goal of collaborative processing is to reduce the effect of these errors. In addition, we aim to keep the communication cost for collaboration to a minimum. With these objectives, we propose a two-stage process for the detection of edge sensors near a contour in the sensor field. In the first stage, all nodes periodically use their local observations to decide if they are probable edge nodes (which can be obtained by checking if its local observation lies between two given thresholds). In the second stage, nodes that test positive in the first stage transmit query messages to their neighbors and process the returned information using a filter-based algorithm to confirm their decisions (see Figure 2).

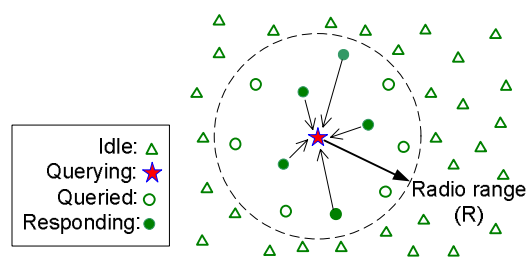

Figure 2: Illustration of the collaborative processing scheme.

IV. Proposed Contour Detection Method using Spatial Filtering

The proposed contour detection algorithm is described as follows:

- Initially all nodes map their observed samples to a quantized value (QV) using a multi-level quantizer as depicted in Figure 3. The quantizer produces output values that are integers in the range $(-MAX, MAX)$ centered around

the contour threshold S_0. The quantization step size is $\tau = 2.MAX/L$, where L is the number of quantization levels.

- A node determines if it is a probable edge sensor or not by comparing its observation sample $F(s)$ to two thresholds as follows:

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

A probable edge sensor broadcasts a *query packet* to obtain quantized values of observations from its neighbors.

- When any node receives a query packet, it replies by sending its own quantized observation value.
- The querying node processes all replies to obtain a decision variable $DV(s)$. It then decides if it is an edge sensor or not by performing a threshold test as follows:

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

We note that the multi-level quantization of observation samples as described in Figure 3 introduces an artificial *edge* having values *MAX* and $-MAX$ at either sides of the contour threshold S_0. Although the edge is expected to be granular, depending on the number of quantization levels *L,* we can apply an edge detection algorithm to the quantized samples at the sensor nodes for detecting edge sensors. We show later that using a value of $L > 2$ as opposed to binary quantization actually reduces the error in contour detection.

The edge detection problem in sensor networks is similar to that in image processing except for the following factors [1]. One factor is that as opposed to pixels that are located on a uniform grid, wireless sensor nodes are usually located in a random fashion. Hence, appropriate measures must be taken to account for the non-uniform locations of the sampled data while applying filtering in sensor networks. A second factor is that sensor nodes are typically required to perform on-site and collaborative processing of data. Hence, there is some cost for communication that is associated with obtaining data from different nodes in the network, which is not an issue in image processing.

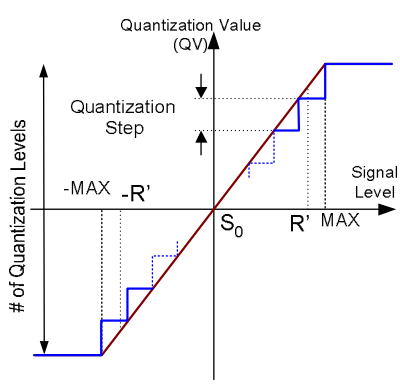

Figure 3: Illustration of multi-level quantization of the sensor signals. In this figure, MAX=QV_{Max} x number of quantization levels.

A number of techniques may be applied to obtain the decision variable for performing the threshold test at the probable edge sensor. In the statistical approach, $DV(s)$ is obtained as the absolute value of the average of all quantized observation values received from the neighbors of the querying node [1,3]. Here we describe an edge detection scheme that is based on an adaptation of the Prewitt filter for edge detection in image processing.

A. Spatial Filtering using Prewitt Filter

Edge detection in image processing is commonly accomplished by performing a spatial differentiation of the image field followed by a threshold operation to determine points of steep amplitude change. Horizontal and vertical spatial derivatives are defined as:

$$d_x = \frac{\partial F(x,y)}{\partial x}\ ,\ d_y = \frac{\partial F(x,y)}{\partial y} \tag{1}$$

where $F(x,y)$ is the value of the signal at the point (x,y). The gradient magnitude of $F(x,y)$ is then

$$|\nabla F(x,y)| = \sqrt{d_x^2 + d_y^2} \tag{2}$$

An edge is then judged to be present if the gradient exceeds a given threshold. To reduce computational load, the gradient at point (x,y) can be simplified to:

$$\nabla F(x,y) = |d_x| + |d_y| \tag{3}$$

In digital image processing pixels replace coordinates, and an edge detection algorithm performs spatial processing on an image to create a new image with pronounced changes in spatial amplitudes of the original image. The processed image $G(j,k)$ is usually described as the combination of two gradient components: the *row gradient* $G_R(j, k)$, and the *column gradient* $G_C(j, k)$ as follows

$$G(j,k) = |G_R(j,k)| + |G_C(j,k)| \tag{4}$$

where j and k are horizontal and vertical indices of a pixel. The simplest method of discrete gradient generation is to form the running difference of pixels along rows and columns of the image. In that case, the row gradient is defined as:

$$G_R(j,k) = F(j, k) - F(j, k-1) \tag{5}$$

and the column gradient is:

$$G_C(j,k) = F(j, k) - F(j+1, k) \tag{6}$$

where $F(j,k)$ represents the original image. Alternatively, a differential filter may be applied to generate the gradient vectors from the original image. For example, in *Prewitt* filtering, the row and column gradient vectors are obtained as:

$$\begin{aligned} G_R(j,k) &= F(j,k) \otimes H_R(j,k) \\ G_C(j,k) &= F(j,k) \otimes H_C(j,k) \end{aligned} \tag{7}$$

where $H_R(j,k)$, $H_C(j,k)$ for a filter of size 3 (i.e. operates on three adjacent pixels to produce one value) are:

$$H_R(j,k) = \frac{1}{3}\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad H_C(j,k) = \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \tag{8}$$

We introduced several modifications to the basic Prewitt filter described above, to suit the requirements for detecting edge sensors from their quantized values according to the proposed scheme. One issue is the possibility of detecting multiple

edges caused by multi-level quantization. We solve this problem by introducing the notion of *Prewitt difference filtering* where differences instead of multiplications are used in the filtering operation. Essentially, this causes the quantized signals obtained from different nodes to be subtracted from an (spatially) odd function at the probing node. In this way while the past gradient remains the same for the real edge nodes, the gradient of the non-edge nodes increases. A candidate odd function is the *Signum* function described as $sig(x) = 1$ if $x \geq 0$, and $sig(x) = -1$ otherwise. Accordingly, we adopt $H_x(.)$ and $H_y(.)$ to be *scaled Signum* functions, with their maximum value being the maximum value of QV_s or QV_{Max}, received from the neighboring nodes.

The problem of random node locations can be resolved by utilizing the concept of a continuous Prewitt filter [3]:

$$H_x(x,y) = -1 \text{ if } x < x_0 \text{, } 1 \text{ if } x > x_0$$
$$H_y(x,y) = -1 \text{ if } y < y_0 \text{, } 1 \text{ if } y > y_0 . \quad (9)$$

Here, we use a *weighted* continuous Prewitt filter, to account for the random number of nodes on different sides (above and below, right and left) of any node where the processing is to be performed. This is implemented using weighting functions $W_x(.)$ and $W_y(.)$ that are calculated as follows:

$$W_x(x_s, y_s) = \begin{cases} \dfrac{1}{n_{right}} & x_s > x_{s_0} \\ \dfrac{1}{n_{left}} & x_s < x_{s_0} \end{cases}$$
$$W_y(x_s, y_s) = \begin{cases} \dfrac{1}{n_{up}} & y_s > y_{s_0} \\ \dfrac{1}{n_{down}} & y_s < y_{s_0} \end{cases} \quad (10)$$

where 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. With these, the decision variable for the test for detecting edge sensors is described as

$$DV(s_0) = |G_{x\_diff}(s)| + |G_{y\_diff}(s)|$$

where

$$G_{x\_diff}(s) = \sum_{\forall s \in N(s_0)} W_x(x_s, y_s)[H_x(x_s, y_s) - QV_s]$$
$$G_{y\_diff}(s) = \sum_{\forall s \in N(s_0)} W_y(x_s, y_s)[H_y(x_s, y_s) - QV_s] \quad (11)$$

B. Considerations for Reducing Communication Cost

We now present two schemes that are proposed to decrease the communication cost for the proposed collaborative edge detection algorithm:

- *Scheme-1:Decreasing the number the possible edge nodes*: This is implemented by introducing a threshold parameter R', $R' \leq MAX$, such that only those nodes for which $|F(s) - S_0| < R'$ are considered to be probable edge nodes, although quantization is still performed according to the multi-level scheme described in Figure 3. This reduces the number of query packets and replies generated in the network without affecting the step-size (accuracy) of quantization. However, this reduction in communication cost will be achieved at the possible cost of higher probability of missed detection.
- *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 its QV_s (sent in response to an earlier query packet) 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.

V. PERFORMANCE EVALUATION

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 uniformly distributed on a 100x100 m2 area. Each sensor node is equipped with an omni-directional antenna having a transmission range of 7 m. The tolerance radius that is used to define true edge sensors is assumed to be 1 m. It is assumed that the data is gathered 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 signal distribution has a Gaussian distribution 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 to have a Gaussian distribution.

Figure 4 depicts the signal distribution along with a snapshot of detected edge sensors from one of our simulation runs that were performed to detect a contour at signal level 50. The true edge sensors 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.

In Figure 5, we compare the performances of the statistical and Prewitt filter based approaches for localized edge detection. Here, the variations of the false detection and missed detection rates are plotted using 8 quantization levels and two different noise levels. The results show that the Prewitt filter based approach generates lower detection errors in comparison to the statistical approach. Henceforth, we present all results obtained from the Prewitt filter based approach only.

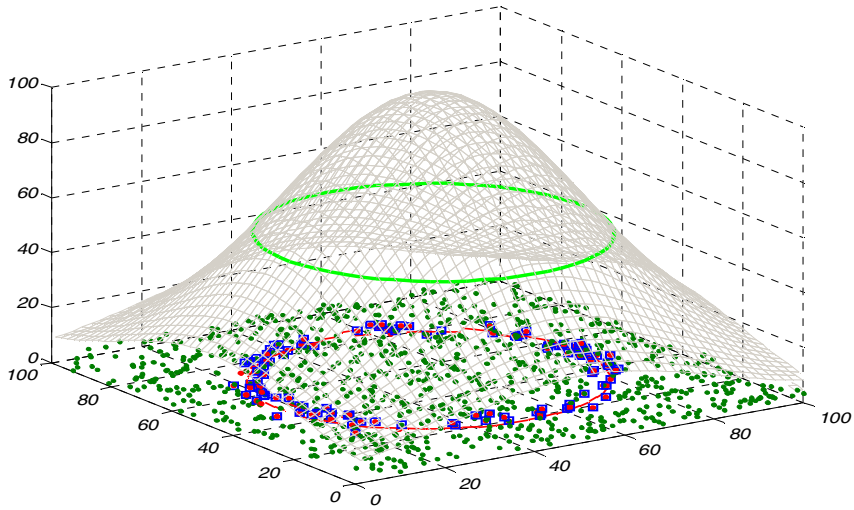

Figure 4: A snapshot of the outcome of edge detection using the proposed scheme depicting true edge sensors in blue and detected edge sensors in red.

We next evaluate the effect of L (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 *L* and *MAX* values. The results, shown in Figure 6, indicate that multi-level quantization results in significant improvement in performance. The probability of missed detection drops noticably 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 *L*=8 and *MAX*=10 for most of our other simulations. Note that our proposed scheme with binary quantization becomes similar to that presented in [1] when applied to contour detection. Hence, the results using multi-level quantization in Figure 6 also indicates the comparative performance improvement obtained using the proposed scheme and that presented in [1], 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 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 edge sensors. The reason is that while the first set includes only those nodes whose signal values are within a certain range of the contour threshold S_0, true edge sensors 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 edge sensors are used to predict the location of the contour, higher missed

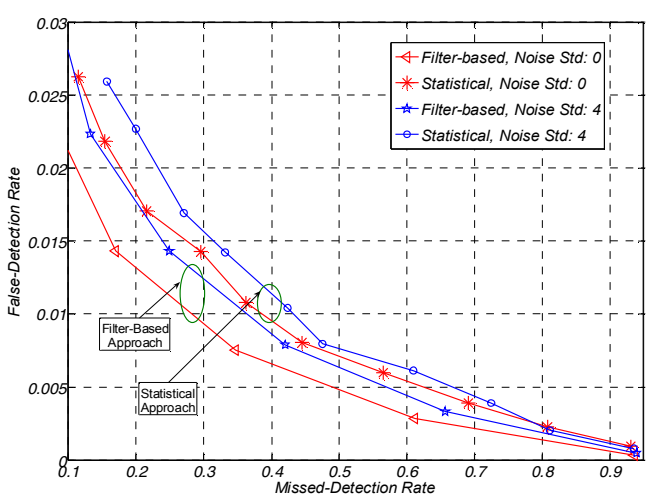

Figure 5: Comparison of detection performance using the statistical and Prewitt filter based approaches..

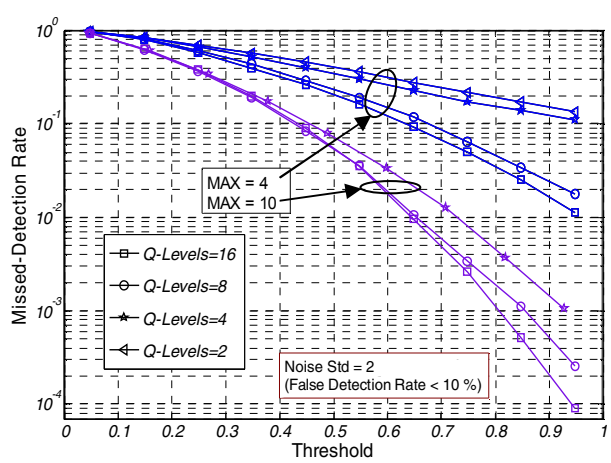

Figure 6: Probability of missed detection for different number of quantization levels and MAX values.

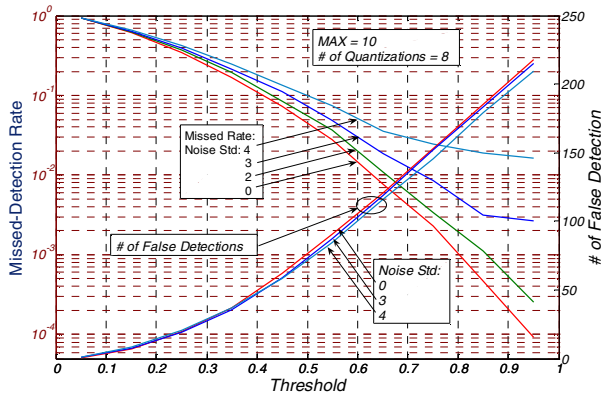

Figure 7: Probabilities of missed detection and the number of falsely detected edge nodes at a contour level of 50 with varying thresholds γ_0.

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 edge sensors for MAX=10 and L=8 is found to be 0.63 when the threshold $\gamma_0 = 0.25$, where the probability of missed detection = 0.367 and the number of false detections = 18.44. However, the average distance increases to 1.44 at $\gamma_0 = 0.85$, where the probability of missed detection=0.001 and number of false detections=185.4.

The above findings provide the main motivation for reducing the number of probable edge sensors using the proposed Scheme-1 described in section 4.2. We note that while a smaller value of R' will reduce the communication cost, it can also increase the probability of missed detection. To evaluate this effect, we obtain the average distances of the detected edge 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 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 communication cost by a factor of 5.

Finally, we evaluate the savings in communication cost obtained by using the proposed ONLi scheme. Table-II 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 communication cost to about 12% of that without using ONLi.

TABLE I: Error distance and communication cost vs. R'
MAX = 10, Number of Quantization level = 8, Threshold = 0.55, Noise Std = 2

R'/MAX	Prob. of miss	# of False	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

TABLE II: Average Communication Cost (Noise Std:2 , Threshold: 0.55, R' = MAX)

MAX	With ONLi	Without ONLi
10	0.73	5.68
7	0.56	3.9

VI. Conclusion and Future Work

A collaborative processing scheme for sensor networks 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 considerations are presented to apply a spatial Prewitt filter to distributed data processing in sensor networks. The proposed scheme has sufficient robustness to 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 paper can also be used for tracking the temporal variations of signal distributions with low communication costs. As a continuation of contour detection in static case, the authors are working on an algorithm for tracking the speed, direction and deformation of contours using localized computations and collaborative processing in wireless sensor networks.

References

[1] K. K. Chintalapudi and R. Govindan, "Localized edge detection in sensor fields," in IEEE International Workshop on Sensor Network Protocols and Applications, pp. 59-70, 2003.

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

[3] W. R. Armstrong, "Localized contour detection in wireless sensor networks," MSEE Thesis, University of North Carolina at Charlotte, 2005.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] "Digital Image Processing, 3rd edition" by William K. Pratt, John Wiely & Sons, Inc., 2001.