NED: An Efficient Noise-Tolerant Event and Event Boundary Detection Algorithm in Wireless Sensor Networks

Guang Jin and Silvia Nittel
Department of Spatial Information Science and Engineering
University of Maine, Orono, Maine, USA
{jin,nittel}@spatial.maine.edu

Abstract

Wireless sensor networks provide an advanced platform to observe the physical world. Different users may be interested in different events derived from a same spatial phenomenon. The constrained and noisy environment of sensor networks, however, challenges successful in-network solutions to monitor and detect events and event boundaries. This paper presents an efficient algorithm, named NED, to support event and event boundary detection in wireless sensor networks. NED encodes partial event estimation results into variable length messages exchanged locally among neighboring nodes. Sensor nodes estimate events and event boundaries based on moving averages to eliminate noise effects. Thus, NED is resource-friendly to constrained sensor networks, and scales well to very large networks. Our experiment results illustrate that NED's communication cost is flexible and moderate to different noise levels, and NED provides high quality estimation results of event and event boundary detection.

1 Introduction

Wireless sensor networks provide an advanced miniature computing and sensing platform for more and more applications. Outstanding capabilities of sensor networks enable us to observe the physical world in a new way, which is today almost impossible through traditional means [1].

Except for quantity readings from individual sensor nodes [2], quality properties of physical phenomena [3] help us to intuitively understand the occurrences in the physical world. Qualitative information often require less resources from networks than quantitative readings, and therefore are more favorable to constrained sensor networks. Users are also more willing to receive reports about interesting events [4, 5] than a flood of quantitative raw readings from sensor networks.

Many physical phenomena are spatially continuous. Events based on a spatial continuous phenomenon, however, are complex and different. For example, in a household environment, we may define the place "where the temperature is greater than $200°C$" from the temperature field as a dangerous event, whereas the region "where the temperature is between $20°C$ and $25°C$" is cozy to most people. The constraints of sensor networks, such as sensing noise, lossy links and constrained energy resources, challenge efficient in-network event and event boundary detection algorithms.

This paper presents a novel algorithm, named "Noise-tolerated Event and Event Boundary Detection (*NED*)", for constrained wireless sensor networks. NED uses a variable-length event coding mechanism to save the communication cost of distributed computation. Based on established statistical models, NED can eliminate the sensing noise effect and estimate high-quality event and event boundary detection results. Furthermore, NED supports arbitrary event threshold settings on the same phenomenon. NED provides sensor networks a flexible, efficient and high-quality solution for in-network event and event-boundary detection.

The remaining parts of this paper are organized as follows: In Section 2, we explore the related work in the area. Section 3 introduces preliminary concepts and definitions. Section 4 proposes the principles and algorithms of NED. We illustrate our experiment results in Section 5, conclude in Section 6.

2 Related work

Current work on sensor networks focuses on the retrieval of quantitative data from individual sensor nodes. High-level systems, such as sensor database management systems [2, 6, 7, 8], are available to support declarative query languages for sensor readings. Current sensor DBMSs also support complex queries, such as aggregation [9, 7] and spatial aggregation queries [10]. [3] provides a framework to extract qualitative information in sensor networks. Based on a qualitative model, sensor networks are structured into triangular irregular networks (TIN) to support in-network query processing and qualitative information deriving. [3] also presents an algorithm to maintain and adapt the network topology to changes in the physical world. To estimate the local event information, however, is not discussed in detail by [3].

In-network event and event boundary detection algorithms [4, 5, 11] are important foundations for qualitative queries in sensor networks. [4] presents a distributed algorithm based on Bayesian theory. Event estimations are encoded into binary values, and exchanged among neighboring sensor nodes. A sensor node counts the vote for event, and compares the number with a predefined threshold to estimate the event status. [4] proves that the optimal decision should be made based on majority votes. The performance of the majority voting algorithm, however, is sensible to noise, although its communication cost is inexpensive. [5] provides another distributed algorithm based on the moving median method, in which sensor nodes exchange their sensor readings in float values among neighbors. The median reading is assumed to be the local average value of the underlying spatial phenomenon. Several statistical tests are applied to pick out the outliers first and then estimate the locations of event boundaries. The communication cost of the algorithm in [5], nevertheless, is 32 times or more higher than the cost of majority voting algorithm [4]. [11] uses a Quad-tree based algorithm to detect and approximate event boundaries. However, the tree pruning operation requires a high demand on resource consumption. On the other hand, different applications would define different events based on different threshold settings even over a same spatial continuous phenomenon. The algorithm in [5, 11] fails to estimate the boundary of such events since the algorithm's assumption that there is a detectable sharp change between event regions and non-event regions fails for spatially continuous phenomena.

3 Preliminaries

3.1 Phenomena and events

Although the real world is a 3D spatial world, in this paper, we model the world as a 2D space, R^2, which can satisfy most applications. A **phenomenon** is a spatial scalar field that represents the variation of a scalar property over a region of 2D space. A spatial field is a function from space coordinates to a scalar property. Formally, given a spatial framework S and a class of scalar values V, a *spatial scalar field* is a function F whose domain is S and codomain is V [3].

Definition 1 *Similarly to [3], in this paper, we define a phenomena as,*

$$Y : R^2 \to R. \tag{1}$$

Many phenomena are spatially continuous. In other words, the change of the scalar value is gradual over the space, and the first derivative of the phenomenon exists everywhere in the 2D space. As shown by Eq.1, we shall index the phenomena, Y, by a 2D point, p, to the phenomena reading, $Y(p)$ at the specific spatial point.

The scalar value of a phenomenon provides a quantitative description of space based on which we can derive qualitative information. As stated in [12], quantitative properties of space form a large, continuous domain, often modelled as real numbers (e.g., Eq.1), whereas qualitative properties form a small, discrete domain, often modelled as binary values.

Definition 2 *In this paper we model an* ***event*** *as a qualitative aspect of space, and define the event, E, at point p as following.*

$$E(p) = \begin{cases} 1, \textit{if } Y(p) \geqslant T \\ 0, \textit{else} \end{cases} \tag{2}$$

In Eq.2, T is a predefined scalar threshold value, which derives the event based on the phenomenon reading at the point p. If Y is a temperature field, then E based on $T = 200°\mathrm{C}$ defines a fire event. Users can define more complex events based on Eq.2, such as a "cozy event", $Cozy()$, where the temperature is between 20°C and 25°C, defined as, $Cozy(p) = AND(E_1(p), NOT(E_2(p)))$. Here E_1 is defined by the threshold, $T_1 = 20°\mathrm{C}$, and E_2 is defined by $T_2 = 25°\mathrm{C}$.

The **boundary** of an event is another type of qualitative information which separates the event space from the non-event space. An event boundary describes important information about an event, such as the shape, the size and the location of the event. The event threshold, T, provides a natural choice for boundary since the phenomenon is spatially continuous.

Definition 3 *Therefore, we define the boundary, $B()$, as following.*

$$B(p) = \begin{cases} 1, \textit{if } Y(p) = T \\ 0, \textit{else} \end{cases} \tag{3}$$

3.2 Sensor networks

3.2.1 Constraints of sensor networks

Several factors constrain sensor networks as stated by[13].

- The energy source of sensor nodes are very limited. In real applications, it is inconvenient or even impossible to replace the batteries of sensor nodes, which makes the problem more severe.

- The communication costs are by far higher than other resource consumptions in sensor nodes. For example, on Berkeley MIC2 model, transmitting one bit of data consumes as much as computing 800 commands on board.

- The communication range between sensor nodes are limited, typically in tens of meters.

Those factors make sensor networks in favor of localized data processing.

3.2.2 Noisy sensor readings

In this paper, we use s_i to indicate a sensor node, i. Sometimes, s_i also represents the 2D location of the sensor node. The reading of sensor s_i on the underlying phenomena is modelled as follows:

$$R(s_i) = Y(s_i) + \epsilon \tag{4}$$

Eq.4 reveals that the sensor reading, $R(s_i)$, is affected by noise. A common model of the error term assumes ϵ is a white normal random variable, $\epsilon = N(0, \sigma^2)$. In other words, the errors of different sensors are independent, but the variance is fixed because of the uniform manufacture of sensors, and the readings of sensors are unbiased to the real phenomenon.

3.3 In-network event detection

In-network event and event boundary detection, like other in-network qualitative information processing [3], attracts sensor network applications in different ways. First, the processing and communication cost of qualitative information are resource friendly to constrained sensor networks, since the data requirement of qualitative information is cheaper than that of quantitative data, as we can see from Eq.1 and Eq.2. Second, qualitative information is more meaningful than simple quantitative data. In-network event and event boundary detection can relax network burdens in both terms of communication and computation costs.

The constrains of sensor networks, however, challenge efficient in-network event and event boundary detection algorithms. A simple translation from quantitative readings into event detection results has to face the faulty results because of sensing noises. NED, on the other hand, in-network estimates event and event boundary against the effects of noise.

4 NED

4.1 Foundation of NED

In a dense sensor network deployment, it is impractical and unnecessary to construct a fully connected sensor network due to the constrained resources. Sensor networks favor localized data processing to relax the resource consumption.

Assumption 1 *In this paper, we assume that sensor nodes can hear from each other in a certain distance, and define the neighboring node set of a sensor node, s_i, as,*

$$N(s_i) = \{s_j | s_i \textit{ can directly hear from } s_j\}. \tag{5}$$

The sensor node, s_i, can only receive detailed readings from its neighbors, $N(s_i)$.

Assumption 2 *In this paper, we assume the noise term, ϵ, is a white normal noise $N(0, \sigma^2)$ as stated in Eq.4.*

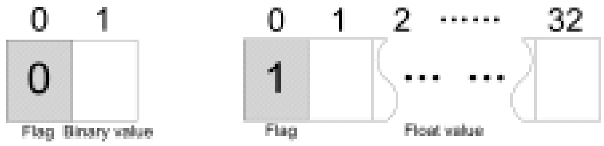
Assumption 3 *This paper assumes the underlying phenomena is a spatial continuous phenomena. We also assume the phenomena is isotropic and stationary.*

Assumption 4 *This paper assumes the sensor nodes in $N(s_i)$ are evenly distributed around the location of node s_i.*

4.2 Coding events

To process distributed events and event boundary detection algorithms, sensor nodes need to encode event estimation results into digital formats. In [4], sensor nodes uses a binary variable (1 bit) to represent a local event detection result, while sensor nodes encode a event reading as a float value (32 bits) in [5]. The binary event estimation is resource-efficient to constrained sensor networks, whereas the estimation result based on float values are more precise at the cost of significantly increasing the burden on networks.

The probability density of a normal random variable concentrates around the the mean value. For a normal variable, $N(\mu, \sigma^2)$, 95% probability falls within the range $[\mu - 1.96\sigma, \mu + 1.96\sigma]$. Since the sensing error ,ε , is a normal white noise (assumption 2), sensor readings have very different certainty powers on local event estimation results. If a sensor reading is much greater than the threshold T (e.g., the distance to the threshold is greater than 1.96σ), the event should be a significant event (at the certainty greater than 95%). Similarly, a sensor node may detect significant non-event (i.e., $T - R(s_i) > 1.96\sigma$) and insignificant event readings (i.e., $|T - R(s_i)| \leqslant 1.96\sigma$).



(a) Significant Event

(b) Insignificant Event

Figure 1. Event coding

To balance the communication cost and the estimation quality, NED uses a variable length coding mechanism to represent partial estimation results of individual sensor nodes as shown by Fig.1. Firstly, users set a significant level for NED according to different application requirements. In this paper, we use a general setting, 95%, in all examples and experiments. If a sensor node detects a significant event or a significant non-event reading, the node uses 2 bits of message to represent its reading. As explained by Fig.1(a), the first bit is a flag which is set to be 0 for significant event or non-event, and the second bit indicates the event estimation (i.e., 1 for a event, 0 for a non-event). If a sensor observes an insignificant event (i.e., the reading falls within the $[1.96\sigma, 1.96\sigma]$ range) , the node sets the flag to 1 and requires additional 32 bits to convey the original sensor reading, as shown by Fig.8.

4.3 Event and boundary detection

Based on the underlying assumptions, NED applies the moving mean method to estimate the phenomena readings at individual sensor nodes' locations as following.

$$\hat{Y}(s_i) = mean(R(N(s_i))) \tag{6}$$

In Eq.6, $mean(R(N(s_i)))$ is defined as,

$$mean(R(N(s_i))) = \frac{\sum(R(s_j))}{n}, \text{ where } s_j \in N(s_i). \tag{7}$$

In Eq.7, n indicates the number of nodes in the neighboring node set, $N(s_i)$. Another general used model is the moving median model which uses the median reading of neighboring nodes instead of the mean reading [5] to estimate the local average. The median value has several advantages (e.g., the median value is robust against outliers). Estimation results based on the moving mean method, however, are better than the results based on moving medians because of several facts. The first reason is that in sensor networks the number of neighboring nodes are limited. Second, underlying spatial phenomena are varying over space, and sensing noise are white normal variables. Our simulation results in section 5 also confirm our expectations.

NED uses binary values to indicate significant event and non-event readings. Before running the moving average method to remove noise effects, sensor nodes have to restore the float readings from binary values. After receiving messages from neighboring nodes, a node restores a significant event message into the phenomena reading, $T + 1.96\sigma$, and a significant non-event into $T - 1.96\sigma$ according to the significant level, 95%. Sensor nodes then estimate local events, $\hat{E}(s_i)$, based on the moving mean results as following.

$$\hat{E}(s_i) = \begin{cases} 1, \text{ if } \hat{Y}(s_i) \geqslant T \\ 0, \text{ else} \end{cases} \tag{8}$$

Since the variance of noise, σ^2, is pre-known and fixed, typically provided by the specification of sensing devices, Eq.6 is an estimation of local mean with the new estimated variance defined by Eq.9.

$$\hat{\sigma^2} = \frac{\sigma^2}{n} \tag{9}$$

We cannot directly apply Eq.3 based on the estimated local phenomenon reading, $\hat{Y}(s_i)$. The first reason is that sensor nodes may not be located on the exact boundary points. The second reason is the noise effects. Therefore, NED uses Eq.10 to estimate the boundary, $\hat{B}()$, for the 95% significant level.

$$\hat{B}(s_i) = \begin{cases} 1, \text{ if } T \in [\hat{Y}(s_i) - 1.96\hat{\sigma}, \hat{Y}(s_i) + 1.96\hat{\sigma}] \\ 0, \text{ else} \end{cases} \tag{10}$$

$\hat{B}(s_i)$ in Eq.10 indicates whether the sensor node, s_i, has the 95% confidence that the node is located on the boundary $Y(s_i) = T$. NED's coding mechanism symmetrically trims the sensor readings around the event threshold, T. Since the phenomenon is isotropic (i.e., there is no directional difference) and stationary (i.e., the change of phenomenon readings is scaled by the Euclidean distance), and the sensor nodes are evenly distributed, the estimated readings of sensor nodes around the event boundary are very close to the event threshold T. Thus in NED, the nodes around a event boundary have higher chances to report the boundary.

4.4 NED algorithm

Tab.1 illustrates the pseudocodes of NED. Since the variance of sensing noise is pre-known, we assume the variance value, σ^2, is cached by each node. In the beginning, users set the significant level, $SigLevel$, according to which sensor nodes set the confidence range. In practice, the time is divided into rounds for sensor nodes to communicate with each other [7]. In each round, each sensor node encodes its sensor reading into event messages as shown by Fig.1. After receiving neighboring messages, sensor nodes restore significant event and non-event messages (binary values) into sensor reading values (float values), average neighbors' readings to eliminate noise effects, and estimate the event and event boundary based on Eq.8 and Eq.10. Finally, sensor nodes report event and event boundary estimation results accordingly.

4.5 Discussion

NED ignores the detailed locations of neighboring nodes. If the variation of a phenomena is large, and sensor nodes are not evenly distributed one can also apply different weights on neighboring readings. For example, a weight can be a function of Euclidean distances between sensor nodes. Weighting readings, however, increases the message size because of additional location information, and furthermore aggravates the burden of networks.

If a phenomena changes very rapidly like a step function over space, as assumed by [5] that phenomenon readings are μ_1 in event regions and μ_2 in non-event regions, the boundary threshold T can be any values in (μ_1, μ_2) to separate events and non-events. NED can use the boundary threshold $T = \frac{\mu_2+\mu_2}{2}$ to keep the symmetry of sensor readings around the threshold, T, and detect this type of phenomena.

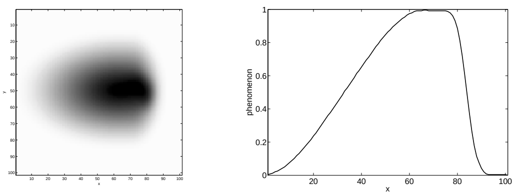
For spatially continuous phenomena, the variable length coding mechanism of NED allows nodes far from event boundaries to communicate by only 2-bit messages. Those nodes close to event boundaries use 33-bit messages to achieve high quality estimations. Hence, NED is resource efficient in the constrained environment of sensor networks.

5 Experimental results

We simulated NED in MatLab. To test the performance of NED, we used a graphic tool to generate several graylevel graphics and assumed the graylevel values as the underlying phenomenon. All graylevel values are from 0, pure white, to 1, pure black, without losing any generalization. The unit distance is 1 pixel in our experiments. Since this paper focuses on continuous phenomenon, a smooth

Table 1. Algorithm pseudocodes

1.	delta = setSignificantLevel(SigLevel);
2.	r = getMySensorReading();
3.	msg = codeEvent(r,T);
4.	msgList = communicateToNeighbors(msg);
5.	readingList = restore(msgList);
6.	avg = estAvg(readingList);
7.	sqrtVar = estSqrtVar(readingList);
8.	if (avg ⩾ T) estEvent = true;
9.	else estEvent = false;
10.	if (T ⩾ avg-delta*sqrtVar) AND (T ⩽ avg+delta*sqrtVar) estBoundary = true;
11.	else estBoundary = false;



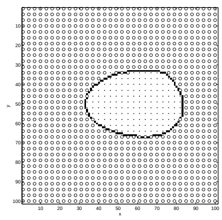
(a) A phenomenon (b) Cross section at y=50

Figure 2. A synthetic phenomenon

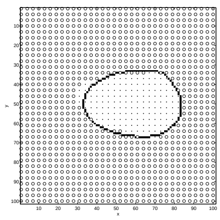
101×101 graphic, as shown by Fig.2, is used by most of our experiments. The phenomenon illustrated by Fig.2(a) continuously changes over the space. The cross section of phenomenon at $Y = 50$ clearly indicates the continuity of the phenomenon as shown by Fig.2(b).

We distributed sensor nodes across the graphic and took individual pixel gray values as the sensor readings, and apply a normal noise to each sensor reading. We also assume sensor nodes to be able to communicate with each other within the distance 5.

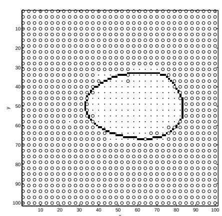
5.1 Event and boundary detection



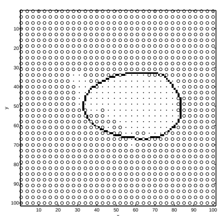
(a) $\sigma = 0.1$



(b) $\sigma = 0.2$



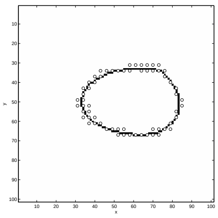
(c) $\sigma = 0.3$



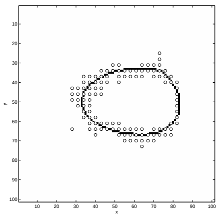
(d) $\sigma = 0.4$

Figure 3. Event detection with $T = 0.5$

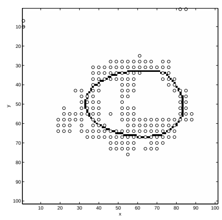
First, we distribute sensor nodes into a grid layout where the distance between neighboring two nodes is 3. Fig.3 shows the event detection results based on $T = 0.5$ with different levels of noise, where the event is located inside the solid line. The dots indicate the sensor nodes detecting an event whereas the circles indicates the nodes reporting non-events. Fig.3(a)-3(d) illustrate the results with noise variance settings $\sigma = 0.1$ to $\sigma = 0.4$ respectively. As we can see, NED almost perfectly estimates the event distribution with the noise, $N(0, 0.1^2)$. Although the noises with $\sigma = 0.4$ degrade the estimation result, the estimation quality is still acceptable.



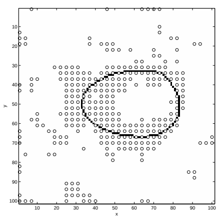
(a) $\sigma = 0.1$



(b) $\sigma = 0.2$



(c) $\sigma = 0.3$

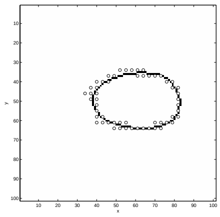


(d) $\sigma = 0.4$

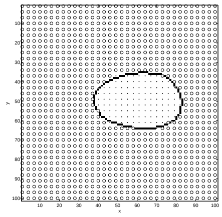
Figure 4. Boundary detection with $T = 0.5$

Another function of NED is the boundary detection as shown by Fig.4 where the solid line indicates the exact boundary and the circles are the sensor nodes reporting the boundary. Fig.4(a)-4(d) show the boundary detection results with noise variance settings $\sigma = 0.1$ to $\sigma = 0.4$, respectively. The boundary detection result based on $N(0, 0.1^2)$ is the best, and the result gets deteriorated while the noise variance increases. Since the noise level, $\sigma = 0.4$, is very high compared with the boundary $T = 0.5$, the result of $\sigma = 0.4$ is reasonable. One may apply a second round boundary estimation based on the estimated event results as shown by Fig.3. A second round estimation, however, may oversmooth the results and therefore miss some boundaries.

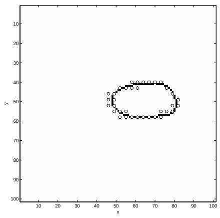
One of the advantages of NED is that it supports arbitrary event boundary settings. Fig.5 shows the boundary detection results based on different threshold settings with the noise setting $\sigma = 0.1$. For the moderate noise, NED shows very good results. The sensor nodes around the event boundaries successfully report the location of boundaries. Fig.5 indicates that the size of event $Y(p) \geqslant 0.8$ is smaller than the size of event $Y(p) \geqslant 0.6$, as we can observe from Fig.2.



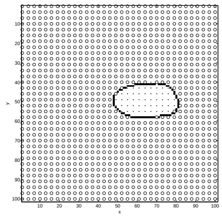
(a) Boundary on $T = 0.6$



(b) Event on $T = 0.6$

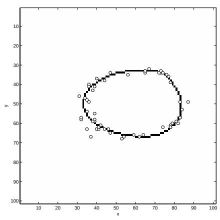


(c) Boundary on $T = 0.8$

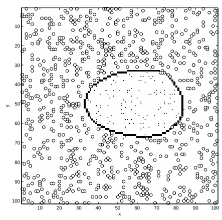


(d) Event on $T = 0.8$

Figure 5. Detection on arbitrary thresholds



(a) Boundary detection

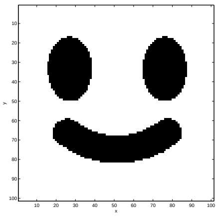


(b) Event detection

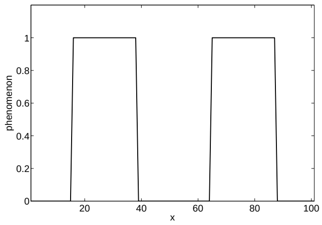
Figure 6. Detection based on random layouts

We run another test to simulate mobile sensor nodes. As illustrated by Fig.6, we randomly select $1,500$ pixels from the simulated phenomenon to provide the locations and readings of sensor nodes. A white normal noise with variance $\sigma = 0.1$ is also applied to each reading. As shown by Fig.6(a), NED returns a very good event boundary estimation based on the threshold $T = 0.5$ Fig.6(b) illustrates that the nodes almost perfectly report the event status.

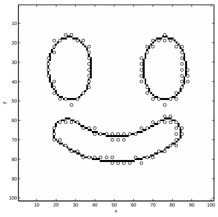
We also use a binary phenomenon as shown by Fig.7(a) to test the performance of NED on sharply changing phenomena. The cross section at $y = 30$, Fig.7(b), shows that the phenomenon is a step function across the space. We set



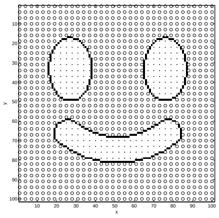
(a) A binary event



(b) Cross section at $y = 30$



(c) NED boundary detection



(d) NED event detection

Figure 7. NED results on a binary phenomenon

the boundary threshold $T = 0.5$ and the noise variance as $\sigma = 0.2$ to test the performance of NED. Fig.7(c) shows that the sensor nodes successfully report the event boundary, and the nodes almost perfectly estimate the event stati as illustrated by Fig.7(d). Overall, Fig.7 exemplifies the effectiveness of NED on discontinuous phenomena.

5.2 Estimation quality of NED

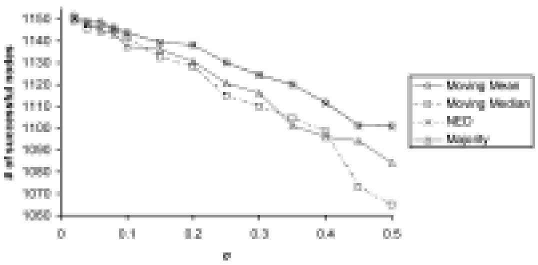
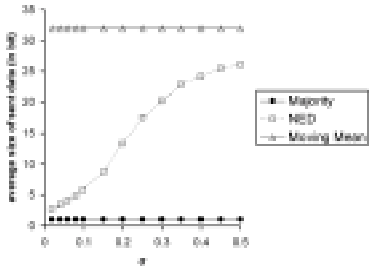
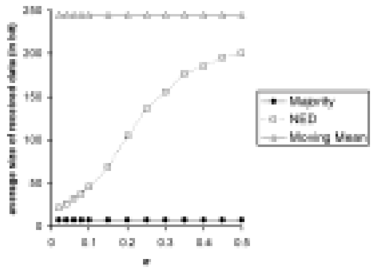


Figure 8. Estimation Quality

We tested different methods 5 times to get their average estimation performance. Fig.8 illustrates the average number of sensor nodes successfully reporting the local event status for $T = 0.5$ with different noise variance settings based on the grid-like network layout. Most nodes far away



(a) Average size of sent data



(b) Average size of received data

Figure 9. Data requirement of NED

the boundary have very small chances to make faulty estimation results, so the success rates of all methods are over 90%. We also test the performance of moving mean and median methods without transforming significant float readings into binary values. As shown by Fig.8, the estimation results of NED and moving mean method are almost the same and the best among those competitors. The moving median and majority voting methods report more faulty events than NED, because the limited number of neighboring nodes restricts the performance of moving median and majority voting methods.

5.3 Cost of NED

Wireless radio communication is the main bottleneck of sensor networks. We run NED 5 times to record the average data requirements for the event $T = 0.5$ with different noise variance settings based on the grid-like network layout, as shown by Fig.9. As explained by Fig.9(a), majority voting method only needs 1 bit to encode the event whereas normal moving mean method requires 32 bits for sensor readings. The average size of received data, however, depends not only on particular algorithms, but also on the wireless radio communication range. Since we assume sensor nodes can hear from each other in the distance of 5, the average size of received data, as shown by Fig.9(b), is greater than the size of sent data, but still explains similar results to Fig.9(a). The data requirement of NED is between the two methods, and closer to the requirement of majority voting method for smaller noise interferences.

Our experiment results show very good event and event boundary estimation results of NED. Compared with other available methods, NED is resource-efficient to sensor networks and can adapt the data requirements to different noise levels.

6 Conclusion and future work

This paper presents a novel distributed event detection algorithm, NED, for sensor networks. NED supports event and event boundary detection based on arbitrary event threshold settings. Although NED focuses on spatial continuous phenomena, our experiment results show the effectiveness of NED on discontinuous phenomena. The experiment results also illustrate the high-quality estimation results of NED and the high-efficiency of NED in constrained wireless sensor networks.

Future work includes the design of an efficient geometry data structure for event boundaries to represent them efficiently.

References

[1] G. Tolle, J. Polastre, R. Szewczyk, D. Culler, N. Turner, K. Tu, S. Burgess, T. Dawson, P. Buonadonna, D. Gay, and W. Hong, "A macroscope in the redwoods," in *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems*, (New York, NY, USA), pp. 51–63, ACM Press, 2005.

[2] P. Bonnet, J. Gehrke, and P. Seshadri, "Querying the Physical World," *IEEE Pers. Commun*, vol. 7, pp. 10–15, 2000.

[3] M. Duckham, S. Nittel, and M. Worboys, "Monitoring dynamic spatial fields using responsive geosensor networks," in *Proceedings of the 13th ACM International Conference on Advanced in Geographic Information Systems (ACM-GIS05)*, pp. 51–60, ACM Press, 2005.

[4] B. Krishnamachari and S. S. Iyengar, "Distributed bayesian algorithms for fault-tolerant event region detection in wireless sensor networks," *IEEE Trans. Computers*, vol. 53, no. 3, pp. 241–250, 2004.

[5] M. Ding, D. Chen, K. Xing, and X. Cheng, "Localized fault-tolerant event boundary detection in sensor networks," in *IEEE INFOCOM*, vol. 2, pp. 902 – 913, IEEE, March 13-17 2005.

[6] A. Demers, J. Gehrke, R. Rajaraman, N. Trigoni, and Y. Yao, "The Cougar Project: A work-in-progress report," *SIGMOD Record*, vol. 32, no. 4, pp. 53–59, 2003.

[7] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "Tag: a tiny aggregation service for ad-hoc sensor networks," *SIGOPS Oper. Syst. Rev.*, vol. 36, no. SI, pp. 131–146, 2002.

[8] P. Bonnet, J. Gehrke, and P. Seshadri, "Towards sensor database systems," in *Proceedings of the Second International Conference on Mobile Data Management*, pp. 3–14, Springer-Verlag, 2001.

[9] J. M. Hellerstein, W. Hong, S. Madden, and K. Stanek, "Beyond average: Toward sophisticated sensing with queries," in *2nd International Workshop on Information Processing in Sensor Networks (IPSN)*, vol. 2634, pp. 63–79, Springer Lecture Notes in Computer Science, 2003.

[10] M. Sharifzadeh and C. Shahabi, "Supporting spatial aggregation in sensor network databases," in *Proceedings of the ACM International Conference on Advances in Geographic Information Systems (ACM-GIS'04)*, pp. 166–175, 2004.

[11] R. Nowak and U. Mitra, "Boundary estimation in sensor networks: Theory and methods," in *2nd International Workshop on Information Processing in Sensor Networks (IPSN)* (F. Zhao and L. J. Guibas, eds.), vol. 2634, pp. 80–95, Springer Lecture Notes in Computer Science, 2003.

[12] A. Galton, *Qualitative Spatial Change*. Oxford, England: Oxford University Press, 2000.

[13] D. E. David Culler and M. Srivastava, "Overview of sensor networks," *Computer*, vol. 37, no. 8, pp. 41–48, 2004.