# Belief-based Cleaning in Trajectory Sensor Streams*

Sitthapon Pumpichet[1], Niki Pissinou[2], Xinyu Jin[1] and Deng Pan[2]

[1]Department of Electrical and Computer Engineering
Florida International University
Miami, USA
Emails: {spump001, xjin001}@fiu.edu

[2]School of Computing and Information Sciences
Florida International University
Miami, USA
Email: {pissinou, pand}@fiu.edu

*Abstract* – **The imprecision in data streams received at the base station is common in mobile wireless sensor networks. The movement of sensors leads to dynamic spatio-temporal relationships among sensors and invalidates the data cleaning techniques designed for stationary networks. As one of the first methods designed for mobile environments, we introduce a novel online method to clean the imprecise or dirty data in mobile wireless sensor networks. Our method deploys a belief parameter to select the helpful neighboring sensors to clean data. The belief parameter is based on sensor trajectories and the consistency of their streaming data correctly received at the base station. The evaluation over multiple mobility models shows that the proposed method outperforms the existing data cleaning algorithms, especially in sparse environments where the node density in the system is low.**

***Keywords: mobile wireless sensor networks; online data cleaning; trajectory sensor data cleaning***

## I. Introduction

The advantages of mobile wireless sensor networks (MSN) proffer the feasibility of promising applications, such as traffic monitoring [5], wildlife tracking [7], civil planning [6], and epidemic surveillance [8]. The success of these applications heavily depends on the quality of the collected sensor data. In MSN, it is, however, common that the sensor data received at the base station are not as precise as data measured by sensors before the data is transmitted. The imprecise or "dirty" data caused by inherent limited resources of sensors, weak wireless multi-hop communications and node mobility can be noticed as noisy data, missing data, non-ordered data, and outliers, etc. For example, only approximately 40% of sensor data are successfully delivered as experimented with stationary sensors at the Intel research laboratory [4]. This amount of delivered data and its precision will be further reduced by node mobility that causes the node isolation and intermittent connectivity. Such dirty data need to be corrected or cleaned for better data analysis in MSN applications.

There are attempts to clean the sensor databases [14], [15]. They are the offline cleaning methods which need to have the sensor data published in databases as a pre-processing step. As an inherent feature of continuous query in sensor applications, an online method is needed instead. Many popular statistical, probabilistic, machine learning and logic reasoning including Bayesian theorem [1], Markov theorem [12, 16], neural networks [11], moving average method [13] are deployed to clean sensor streams in a real-time fashion. In addition, several frameworks are also proposed a pipeline [2], a belief-based [3] and a model-based [17] fashion to clean sensor streams. However, these methods operate by mainly utilizing the static spatio-temporal relationships among sensors. The node mobility in MSN, which presents the dynamic network topology, then invalidates all existing methods assuming the static sensor relationships.

To our best knowledge, the method in [9] deploying the concept of virtual sensor and adaptive filter techniques is the only method designed to clean data in mobile environments. Nevertheless, this method does not consider the non-synchronization of sampling time among sensors, and its performance is limited by the node density in the system. Thus, we are motivated to clean MSN sensor data with an online method to satisfy the real-time applications. Our main contributions are:

- We introduce a belief-based sensor selection method to identify the group of sensors that is helpful in cleaning data based on their current trajectories and the quality of their data streams.
- We present a novel online data cleaning method designed for the dynamic environment in MSN applications. Our evaluation results show that the cleaning performance of our method outperforms those of virtual sensor-based method in [9] and a method designed for stationary sensor networks in [13].

The problem statement and assumptions are described in section II. Section III explains our proposed method in detail. The evaluation and analysis is then illustrated in section IV. The conclusion is finally shown in section V.

## II. Problem Statement and Assumptions

Assuming that there is a pre-process operating to detect the dirty data, such as outliers, non-ordered data sequence, out-of-date data and missing data, etc., such dirty data is discarded by the system. We develop an online algorithm to clean the dirty data in MSN streams. The designed cleaning process is centralized based architecture, i.e., all cleaning mechanisms including the detection of dirty data and data stream management are conducted at the base station where all sensor data streams are forwarded.

In practice, the trajectory data expressing the time-location information and the sensor measurements from a

*The work was supported in part by the US National Science Foundation under grants CNS-0851733, CNS-0843385 and the Department of Homeland Security.

sensor could be delivered to the base station via the different channel as an out-of-bound transmission. Although the sensor measurements are dirty and need to be cleaned, we assume that the trajectory data is correctly received at the base station.

We focus on cleaning the dirty data from sensors, which are moving in a pre-defined area of interest. We assume that multiple sub-areas form up the area of interest. The level of reading in the same sub-area is similar and different from that of adjacent sub-areas. The boundaries among sub-areas are also assumed to be known.

## III. Proposed Method

Our proposed data cleaning method is an area-based approach assuming a priori knowledge of sub-area boundaries. The cleaning process computes the replacement of dirty data by utilizing the readings from a group of sensors that are believed to be offering enough reliable readings from a specific sub-area. In this section, we explain our proposed cleaning method in detail. We first discuss how a group of neighboring sensors is selected for collaborating in the cleaning process. We then describe how the dirty sample is cleansed based on the distance function in both time and location of sensors.

### A. Belief-based Sensor Selection

With the number of deployed sensors in practice, brute-force methods to select the most correlated data readings are not practical. Based on a priori knowledge of sub-area boundaries, each sub-area has been indexed and matched with a belief table. Our approach is using the belief table, which contains the updated belief degree of each sensor for each sub-area. For a sub-area, the belief degree of each sensor represents how trustworthy a sensor could help cleaning the dirty readings measured within the sub-area at a specific time. It is based on two parameters, which are (1) alibi degree and (2) detection rate of dirty data, explained as follows:

#### 1) Alibi degree (A)

The alibi degree is computed at a specific time to show the accommodation level that a sensor experiences and reads the dedicated measures within a sub-area. At a specific time, the higher the alibi degree of a sensor, the more the sensor operates within the corresponding sub-area. The alibi degree is computed from residence vector and the frequency of existence in the sub-area.

The residence vector expresses a series of existence of a sensor located in a sub-area. The sensor existence in each sub-area is computed from the trajectory data of each sensor received by the base station. The members of the vector are stored in the allocated window of memory space. They are of Boolean type; 1 when the sensor is located within the corresponding sub-area and 0 when the sensor stays outside that sub-area. For example, illustrated in Fig. 1, a sensor is traversing across a sub-area. If the allocated window size equals 9, the residence vector from time sequence $t_1$ to $t_9$ will be [0 1 1 0 1 1 0 0 0].

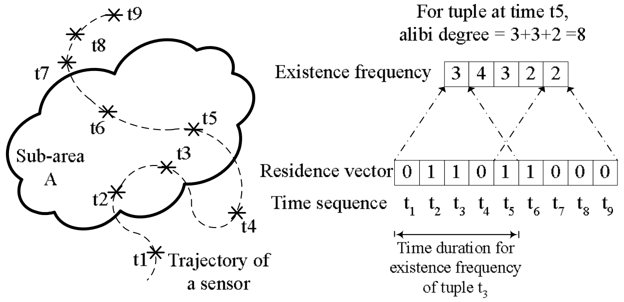


Figure 1. Alibi degree calculation

While the residence vector is updated, the frequency vector is also computed and stored in another window of memory. For a sensor, each member of the frequency vector represents the frequency of the sensor existence within the corresponding sub-area. The frequency of the sensor existence is calculated per time duration. For instance, Fig. 1 shows that the time duration for calculating the frequency of existence is set to 5 samples. The existence frequency at time instance $t_3$ is the sum of members of residence vector from instance $t_1$ to $t_5$; that of tuple $t_4$ is sum of members of residence vector from instance $t_2$ to $t_6$, and so on. Note that existence frequency at the time sequence $t_3$ contains residence information in the following tuples, which are those of $t_4$ and $t_5$. As it would be later explained, the existence frequency at time instance $t_3$ is needed to clean a dirty sample at sequence $t_3$. The cleaning process for the instance at $t_3$ would be delayed by half of the user-defined length of time duration. The higher frequency value implies a greater chance of the sensor having experience within the corresponding sub-area.

With the allocated window size of 9 and time duration of 5 tuples as shown in Fig. 1, the existence frequency vector would be fulfilled after the trajectory data of tuple $t_9$ is received by the base station. The alibi degree can then be computed as a dot product of residence vector and frequency vector. The maximum value of alibi degree is equal to length of existence frequency vector times its time duration in samples. At the time of tuple $t_5$, the alibi degree would then be equal to 3+3+2 =8, and equals 8/25 after normalized.

#### 2) Detection rate of dirty data (D)

Although two sensors are in the same sub-area, their different trajectories can lead to different environments affecting the quality of data delivery. Here, we present the detection rate of dirty data to inversely represent the reliability of the data stream of each sensor. As the area-based parameter, the detection rate of dirty data shows the quality of streaming data received from a sensor residing in the corresponding sub-area. As an online method, we propose to calculate the detection rate of dirty data as a ratio of the cumulative number of detected dirty samples to the number of all samples that the sensor measured within the corresponding sub-area.

Note that we assume that a pre-processing module to detect the dirty data exists and correctly detects the corrupted samples. Intuitively, the lower the detection rate of dirty data, the more reliable the data stream of the sensor residing in a particular sub-area.

### 3) *Belief degree and sensor selection*

At a time instance, the belief degree of each sensor will be calculated and updated to the belief table specific each sub-area. The belief degree would be proportional to the alibi degree but inversely proportional to the detection rate of dirty data. The derivation could be shown as in (1). The high-level description in updating the belief table of all sub-areas is illustrated in Table I.

$$\beta = (\alpha \cdot A_N) + (1-\alpha)\cdot\ (1-D) \tag{1}$$

where $\beta$ : belief degree
$\alpha$ : belief coefficient
$A_N$: normalized alibi degree
$D$ : detection rate of dirty data

Our approach to clean a corrupted sample utilizes the readings from sensors, which are reliable enough. The sensors with the $\beta$ value higher than a belief threshold ($\beta_{th}$) would then be selected to collaborate in the cleaning process. The proper values of belief threshold ($\beta_{th}$) and belief coefficient ($\alpha$), ranging between 0 and 1, are depending on applications and the nature of measurements of the system. For example, if the performance of the dirty data detection module offers a large uncertainty, the belief coefficient would be set close to 1.

## *B. Belief-based Cleaning Method*

After a group of sensors is selected to help cleaning the dirty data for the target sensor, a cleaning process will compute the cleansed value to replace the value of the dirty sample. The calculation of cleansed data considers (1) the time difference between the time that each available data of the selected sensors are sampled and the time when the target sensor senses the dirty sample, and (2) the distance between the selected sensors and the target sensor when the target sensor senses that dirty sample. Only readings of the selected sensors sensed in the same sub-area where the dirty data is measured are eligible to be deployed in our proposed cleaning process.

We assume that the lower the sampling time difference and the location distance between the selected sensors and the target sensor, the more similar the data from selected sensors would be to the actual measure of the target sensor. We here propose that a cleansed value will be equal to a weighted average that is indirect to a distance function in sampling time and location, as shown in (2).

$$d_c = \frac{\sum_{i=1}^{k} d_i \cdot \frac{1}{\Delta t(d_d, d_i)} \cdot \frac{1}{\Delta L(d_d, d_i)}}{\sum_{i=1}^{k} \frac{1}{\Delta t(d_d, d_i)} \cdot \frac{1}{\Delta L(d_d, d_i)}} \tag{2}$$

Where k : the number of data samples of selected sensors residing in the sub-area
$d_c$: cleansed data of the target sensor
$d_i$ : the eligible data from selected sensors
$\Delta t(d_d, d_i)$: difference in sampling time of the dirty sample $d_d$ and the eligible data $d_i$
$\Delta L(d_d, d_i)$: location distance of target sensor and selected sensors when the target sensor senses the dirty sample $d_d$

# IV. Evaluation and Analysis

In this section, we summarize our experiment analysis to evaluate the performance of our proposed algorithm. To our best knowledge, the virtual sensor-based method (VS) in [9] and the proposed belief-based method (BB) are the first algorithms attempting to clean dirty data in MSN environments; therefore, the performance of the proposed method will be compared with the VS method and another designed to clean data in static WSN based on the moving average method [13].

The performance of algorithms is evaluated in a simulated scenario in which there are *n* sensors moving randomly and sensing the temperature data. The 200 x 200 $m^2$ area of interest is divided into 9 sub-areas, as shown in Fig. 2. These 9 sub-areas will be classified into 3 categories based on the area characteristics: (1) Indoor area, (2) Shaded outdoor area and (3) Outdoor area.

Each category exposes temperature values based on a normal distribution with a different mean but the same standard deviation of 0.5ºC. The average temperature value of each category evolves by time according to the change of data trend collected from the Asheville Regional Airport, North Carolina, from January 1-15, 2007 [10]. The mean temperature in indoor areas is roughly 7ºC lower than that of shaded area and 13ºC lower than that of outdoor areas.

In each round of simulation, each sensor randomly starts sensing data during 0-30th second, and it would constantly sample the data every 30 seconds. With a variety of node densities, each sensor senses 1200 samples as a referenced data set. As we assumed that the dirty samples are detected before progressing to the proposed cleaning module, we randomly assigned a fixed percentage of all samples as the detected dirty data that need to be cleaned. The window size of the residence vector equals 9, and time duration for the existence frequency is set to 5.

TABLE I. Belief Table Update

```
// Input: The location data of sensors in window space at time $t_k$
// Output:   The updated belief table (T) of all sub-areas
// Update the belief table of each sub-area, one by one
Procedure Belief_update
for subA = 1 to S          // S is number of all sub-areas
  for i = 1 to N           // N is number of all deployed sensors
    Calculate the alibi degree;
    Calculate the detection rate of dirty data;
    Calculate the belief degree as shown in Eq.(1);
    Update the belief degree matched with sensor(i)  in T;
  end
end
end procedure
```

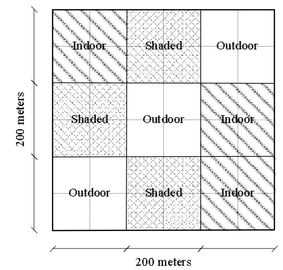


Figure 2. Layout of tested area of interest

We considered three mobility models – (1) random waypoint, (2) nomadic, and (3) random street in our evaluation. The random waypoint [18] is a classic mobility model that each node will move from its current location to a randomly selected new location with a random speed and it will pause before moving to another new location. Instead of the independent random movements, the nomadic mobility [20] represents groups of sensors that collectively move from one location to another. This mobility suits to scenarios of, for example, a class of students touring in a museum. The random street [19] is a newly established mobility model that mimics scenarios when there are path constraints such as walls, buildings and motorways presented as in a real map.

We used the Bonnmotion mobility scenario generator [21] to generate the trajectory data for all mobility models. In nomadic settings, the number of nodes per group is at 10 nodes with deviation of 2 nodes and the maximum group radius is at 15 meters. For the random street, we selected a real area with path constraints in Germany as defined in the GIS reference as the EPSG code: 31466; Gauss-Kruger zone 2. The maximum pause time is set at 60 seconds as similar to that in the random waypoint settings.

The performance of cleaning methods is evaluated by a ratio of the number of "successfully cleaned" samples to the number of whole detected dirty data. This ratio is referred to as the cleaning rate. A dirty sample would be successfully cleaned only when the absolute difference between the output of cleaning process and the referenced data is bounded under a user-defined error threshold.

For BB method, we experimented as the alibi degree and detection rate of dirty data are equally significant. We then set $\alpha$ equal to 0.5 and experiment with $\beta_{th}$ at 0.7. As we assume that the effective average transmission range of a sensor node is around 20-25 meters, the coverage of VS is then set at 22.5 meters.

We first compared the cleaning performance with various densities of sensor nodes moving in the area of interest as shown in Fig. 3. For all tested mobility models, the performance of our proposed method is superior to that of the VS method especially when the node density is low. In random waypoint models with 0.2 nodes/100 $m^2$, the cleaning rate of the BB method exceeds that of the VS method for at least 50% at 0.5 error threshold.

We also evaluated the cleaning rate when the percentage of detected dirty data is varied as shown in Fig. 4. For all mobility models, the cleaning rate of the proposed method surpasses at least 25% compared to other tested methods.

Scenarios with different average node speed of 2 mph (human walking), 8 mph (biking) and 20 mph (car slowly running) were also experimented. The result in Fig. 5 shows that the proposed method outperforms the VS method for all mobility types. Although the cleaning rate of the proposed method is degraded faster than the VS method in the nomadic mobility model, the superior performance is remaining up to the speed of car slowly running.

## V. Conclusions

In this paper, we have presented a novel simple method of data cleaning suited to MSN applications. Rather than relying on the static spatio-temporal relationships among sensors, which is invalid to MSN, we analyzed the area-based trajectory features as the residence pattern and existence frequency to reveal how a neighboring sensor can help in the cleaning process. Moreover, the cumulative detection rate of dirty data is also utilized to grade the trustworthy level of a data stream within per particular sub-area. The superior performance compared to that of the existing cleaning methods is demonstrated for various mobility models, dirty data rate and average sensor speed.

This work is the very first solution to clean dirty data in MSN. There are more challenging limitations to overcome. The trajectory information can also be dirty or imprecise. Also, the area classification might be unknown and dynamic. Our research direction is to find solutions to cope with such complex situations.

## Reference

[1] E. Elnahrawy and B. Nath, "Cleaning and querying noisy sensors," in Proc. of the 2nd ACM international conference on Wireless sensor networks and applications, 2003.

[2] S. R. Jeffery, G. Alonso, M. J. Franklin, Wei Hong, and J. Widom, "A pipelined framework for online cleaning of sensor data streams," in Proc. of the 22nd IEEE International Conference on Data Engineering, Atlanta, GA, USA. April 03-07, 2006.

[3] B. Q. Ali, N. Pissinou, and K. Makki, "Belief based data cleaning for wireless sensor network," Journal of Wireless, 2009.

[4] Intel Lab Data. [Online].
Available: http://berkeley.intel-research.net/labdata/.

[5] B. Hull, et al., "CarTel: a distributed mobile sensor computing system," in Proc. of the 4th International Conference on Embedded Networked Sensor Systems, Boulder, Colorado, USA, November 1-3, 2006.

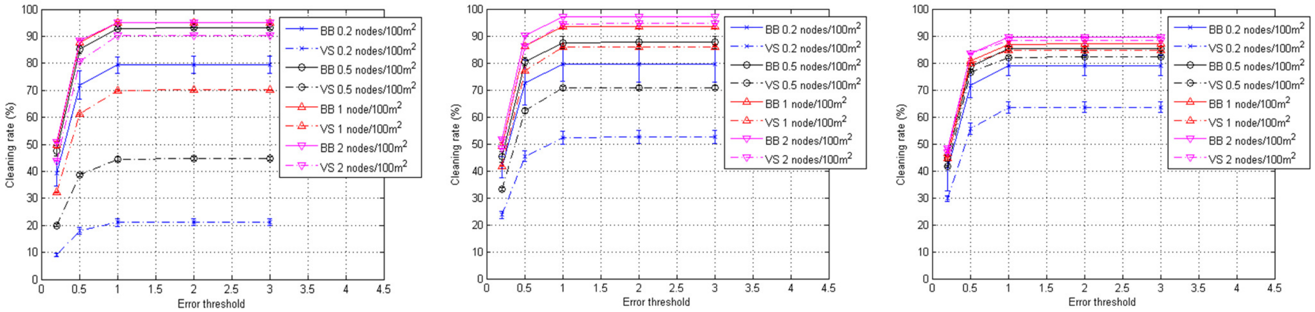
[6] J. Eriksson et al., "The Pothole Patrol: Using a Mobile Sensor Network for Road Surface Monitoring," in Proc. of the 6th ACM International Conference on Mobile Systems, Applications and Services, Breckenridge, CO, USA, June 2008.

[7] P. Juang et al., "Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with Zebranet," in Proc. of Architectural Support for Programming Languages and Operating Systems, San Jose, CA,USA, 2002.

[8] G. Hartvigsen et al., "Reusing Patient Data to Enhance Patient Empowerment and Electronic Disease Surveillance," Journal on Information Technology in Healthcare, vol. 7, no. 1, pp.4–12, 2009.

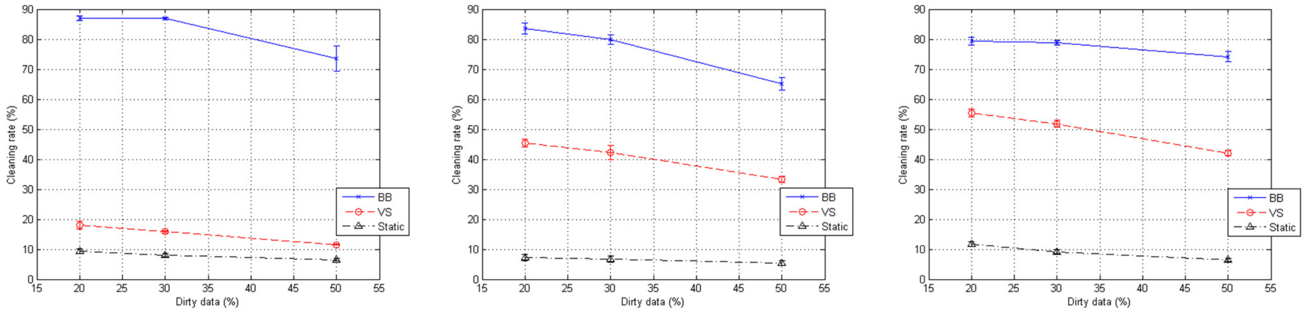
[9] S. Pumpichet and N. Pissinou, "Virtual sensor for mobile sensor data cleaning," in Proc. of the IEEE International Conference on Global Communications, Miami, FL, USA, December 2010.

[10] Online Climate Data Directory, National Climatic Data Center, US Department of Commerce. (NOAA). [Online].
Available: http://cdo.ncdc.noaa.gov/qclcd/qclcdhrlyobs.htm



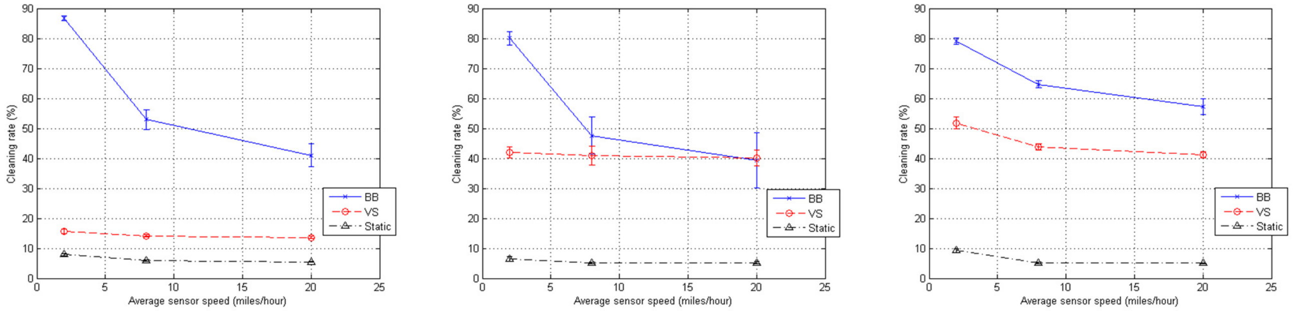
(a) Random waypoint (b) Nomadic (c) Random street

Figure 3. Cleaning performance with varying node density in different mobility models and dirty data of 20%



(b) Random waypoint (b) Nomadic (c) Random street

Figure 4. Cleaning performance with varying percentage of missing data in different mobility models at 2 mph average speed



(c) Random waypoint (b) Nomadic (c) Random street

Figure 5. Cleaning performance with varying average speed of sensors in the area in different mobility models and dirty data of 30%

[11] A. Petrosino and A. Staiano, “A neuro-fuzzy approach for sensor network data cleaning,” In Proc. of the 11th International Conference on Knowledge- Based and Intelligent Information & Engineering Systems in Conjunction with XVII Italian Workshop on Neural Networks, Vietri sul Mare, Italy, September 2007.

[12] F. Chu, Y. Wang, D. Parker, C. Zaniolo, “Data cleaning using belief propagation,” in Proc. of the 2nd International ACM SIGMOD workshop on Information Quality and Information Systems, Baltimore, MD, USA, June 2005.

[13] Y. Zhuang, L. Chen, X. Wang and J. Lian, “A weighted moving average-based approach for cleaning sensor data,” in Proc. of the 27th IEEE International Conference on Distributed Computing Systems, Toronto, Canada, June 2007.

[14] C. Mayfield, J. Neville and S. Prabhakar, “ERACER: A database approach for statistical inference and data cleaning,” in Proc. of ACM International Conference on Management of Data (SIGMOD), Indianapolis, Indiana, USA, June 2010.

[15] R. Cheng, J. Chen and X. Xie, “Cleaning uncertain data with quality guarantees,” in Proc. of ACM Very Large Databases (VLDB), Auckland, New Zealand, August 2008.

[16] B. Kanagal and A. Deshpande, “Online filtering, smoothing and probabilistic modeling of streaming data,” in Proc. of the 24th IEEE International Conference on Data Engineering, Washington, DC, USA, 2008.

[17] H. Jeung, S. Sarni, I. Paparrizos, S. Sathe and K. Aberer, “An end-to-end system for cleaning sensor data: model-based approaches,” unpublished.

[18] D. Johnson and D. Maltz, “Dynamic source routing in ad hoc wireless networks,” in Mobile Computing, T. Imelinsky and H. Korth (Eds.), Kluwer Academic Publishers, Norwell, MA, 1996, pp.153-181.

[19] N. Aschenbruck and M. Schwamborn, “Synthetic map-based mobility traces for the performance evaluation in opportunistic networks,” in Proc. of the 2$^{nd}$ ACM International Workshop on Mobile Opportunistic Networking, Pisa, Italy, February 2010.

[20] T. Camp, J. Boleng and V. Davies, “A survey of mobility models for ad hoc network research,” in Wireless Communications and Mobile Computing, vol. 2, pp. 483-502, 2002.

[21] “BonnMotion – a mobility scenario generation and analysis tool,” University of Bonn, Germany, 2002. [Online]. Available: http://net.cs.uni-bonn.de/wg/cs/applications/bonnmotion/,