MAKING SENSOR DATA WORK FOR EMERGENCY RESPONSE

Yanning Yang

Computer Science Department, Loughborough University, UK

Andrew May

Ergonomics & Safety Research Institute, Loughborough University, UK

Shuang-Hua Yang

Computer Science Department, Loughborough University, UK[1]

Keywords

Emergency Information Management, Incident Response, Sensor Data Processing, Sensor Data Storage, Sensor Data Cleaning

Abstract

Wireless Sensor Networks have shown great potential in filling in a gap with current Emergency Response (ER) - the lack of real-time information on the status of an incident. Sensors deployed in the monitoring field can transmit real-time data from the network wirelessly, and can therefore report emergency situations (such as fires) quickly and reliably. However, it is difficult for humans to directly interpret large sensor data streams because sensor data often contains noise. In addition, sensor data is often meaningless unless it is associated with time and location information. This paper investigates the process of making sensor data work effectively for ER. The requirements for designing the suitable sensor data processing methods for ER applications have been analyzed. The findings demonstrated that in order to make sensor data work for ER: (a) it needs to be properly pre-processed, (b) it must be stored and managed efficiently, and (c) meaning must be extracted from the data prior to its presentation to the emergency responders. A data storage mechanism design has been proposed for the storage and management of real-time sensor data streams and their associated time and location information. A database schema and associated support for query efficiency have been devised. The system can adapt itself to the different stages of a developing emergency incident. A sensor data cleaning method has been proposed to reduce noise and data outliers and quickly reflect real environmental changes. A sensor state model has been proposed, and detection rules for each state have been defined, in order that outliers can be distinguished from real abnormalities. Finally, simulations have been carried out to evaluate the proposed approaches for data storage and data cleaning. Test results show that the method is valid and is a promising basis for further extraction of meaning from sensor data.

[1] Prof. Shuang-Hua Yang, Computer Science Department, FK03 Holywell Park, Loughborough University, Loughborough, LE11 3TU UK Tel: +44 (0)1509 635670 E-mail: s.h.yang@lboro.ac.uk

1. Introduction

One of the most important goals for incident commanders during an emergency is 'Situation Awareness', the process of obtaining information to comprehend the nature of the emergency and project its status in the near future. Researchers have identified that both 'outside building information' (e.g. the best access to the sites) and 'inside building information' (e.g. the occurrence and spread of fire incidents in the building) are required to provide a full picture in the event of an emergency. Some of the latest Emergency Response Systems (ERS) have integrated technologies such as Global Positioning System (GPS) to help emergency responders arrive on site as quickly as possible. However, scarcely any information from the inside of the building is provided by current ERS. On arrival at the incident sites, incident commanders have to assess the situation by observing from the outside, asking people who have been in the building or people who have the knowledge of the inside, and by using the simple facilities such as operational panels provided by commercial buildings to roughly identify in which zone is the fire incident. However, very limited information can be obtained by these methods and they are often time-consuming and inaccurate.

Wireless Sensor Networks (WSN) have shown their potential in providing the "inside building" information. Sensors that are battery-powered, light and cheap can be deployed in the building to monitor the environment, and they can transfer real-time information about the occurrence and spread of an incident (such as a fire) to outside of the building. Information on, for example, temperatures, smoke ingress, gaseous composition, can provide great benefits to the situation awareness of the incident commanders.

It is, however, difficult for humans to directly interpret large sensor data streams because sensor data often contains large amount of redundancy, which would be hard for even a professional data analyst to interpret. Sensor data also frequently contains 'noise', which can be difficult to separate out from 'real' data. In addition, sensor data is usually meaningless unless it is associated with time and location information. This paper mainly focuses on the steps and technologies required to make sensor data work for emergency response.

The paper is structured as follows. Section 2 analyzes the requirements of WSN-based ER applications, the features of sensor data and the challenges of designing suitable sensor data processing methods. Section 3 describes the data storage mechanism, including the data storage structure and associated management. The proposed sensor data cleaning algorithm is presented in Section 4. Section 5 discusses the simulation that has been carried out to evaluate the proposed data storage mechanism and data cleaning algorithm. Section 6 concludes by discussing current findings and future work.

2. Requirement Analysis

The sensor data processing techniques that are required to make sensor data work for Emergency Response depend on the ER application and the nature of sensor data that feeds those applications.

2.1 Application level requirements

Emergency Response demands some fundamental features from any technology applied in this area.

<u>Timeliness</u>

The nature of Emergency Response requires timely and efficient response to incidents, because any delay may result in the loss of life and property. "In the practitioner community, emergency managers have learned and stated that accurate and timely information is as crucial as is rapid and coherent coordination among the responding organizations."(Walle and Turoff, 2007) Therefore, it is required that WSN-based ER applications be resilient, real-time systems, and support swift queries.

<u>Accuracy</u>

The accuracy of the sensor data is essential for Emergency Response applications, because the accuracy of the sensor data may influence the correctness of any decisions made by incident commanders. For example, if alarms generated by noises or outliers are not filtered, the false alarms will result in a significant waste of time and personnel resource responding to non-existent emergencies. However, if a real emergency is filtered by mistake, the result could be the loss of life or property. Sometimes, the decisions made based on sensor data can be a matter of life or death, hence it is important to maximise the quality of sensor data.

2.2 Features of Sensor Data

In comparison with traditional data, data from WSN has special features, which bring challenges in managing and processing sensor data.

<u>Streaming feature</u>

Traditional data is usually static entries input by a human, whereas sensor data is automatically generated streaming data. "Data streams have different characteristics from the data of traditional data processing."(Kim et. al., 2005) The streaming feature requires real-time online data processing. The streaming feature also means that the total volume of sensor data rises as the time duration increases. Therefore, a data storage mechanism which can maintain high query efficiency in spite of the increase in volume of data is required.

<u>High temporal and spatial correlations</u>

Sensors are usually deployed at a certain density so that they can cover the entire monitoring field. In normal situations, the environment doesn't change or changes very slowly, therefore "the readings observed at one time instant are highly indicative of the readings observed at the next time instant, as are readings at nearby devices" (Jeffery et. al., 2006). This high temporal and spatial correlation can be used to detect outliers and improve the quality of sensor data. In fact, sensor readings only have meaning if they are associated with time and location.

<u>Redundancy</u>

The strong spatial and temporal correlations typically present in sensor data can result in significant data redundancy in a database, yet it also reveals that the redundancy could be used to predict missing values and to detect outliers. A certain level of redundancy can improve the accuracy of database query results. Therefore, the solution to the redundancy issue of sensor data is not simply removing redundant data but to maintain it at a level that provides confidence in its data without producing unnecessary storage demands.

<u>Noisy</u>

Sensors are designed to be low cost and have low power consumption, however, this design focus may result in the accuracy of sensors being limited. In addition, sensors are normally deployed in harsh environment with background interference, and consequently sensors can experience internal fault or damage during emergencies such as fires. Research has stated that sensor data often contains errors (due to sensor function) and noise (due to other environmental interference) (Elnahrawy and Nath, 2003). This feature indicates that sensor data should be cleaned before being stored in any database.

As a result of the application level requirements and the special features of sensor data,

- A suitable data storage mechanism is required to store and manage sensor data efficiently,
- Data cleaning should be executed before saving data into a database,
- Sensor data should be integrated with time and location information to present meaning to Emergency Responders.

3. Data Storage Mechanism

The Data Storage Mechanism (DSM) is the way that data is stored and managed. It has a major impact on the efficiency of further data processing. However the importance of the DSM for sensor data does not seem to be highly recognised in other research on sensor data management. Existing sensor data storage research has focused on purely technical issues such as energy consumption (Mathur et. al., 2006) or storage placement (Ganesan et. al., 2003). Our aim is to design a DSM suitable for ER applications, which

- Stores and manages sensor readings, time-related information and location-related information,
- Supports timely response to queries regardless of the increase in volume of data,
- Is dynamic and flexible enough to accommodate different needs at different incident stages.

Database schema

The data that need to be stored can be classified into three categories: sensor information (including sensor identifier and sensor readings), time-related information and location-related information.

The overall database structure designed is as shown in Figure 1. The SensorReading table stores and manages streaming data received from the WSN. The SensorDeployment table stores and manages associated sensor location. Building information is stored in two tables, Building and FloorMap.

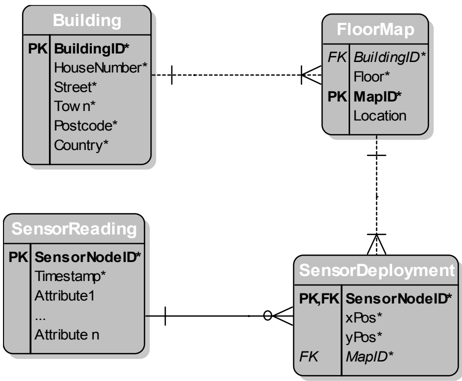

Figure 1. Overall database schema.

Each tuple in the SensorReading table specifies the source of the sensor data (SensorNodeID), the time instance that the sensor data is generated (Timestamp), and several sensor readings (Attributes 1 to n). Each sensor node can carry a number of sensors, each of which is considered to generate an attribute of the sensor node and assigned an attribute name, such as temperature, smoke, flame, etc. Atrribute1 to n will be replaced by attribute names in real applications. A sensor node ID and timestamp uniquely determines its readings, written as [SensorNodeID, Timestamp] → [A1, A2…An]. SensorNodeID is the foreign key that links to the primary key on the SensorDeployment table.

Each tuple in the SensorDeployment table specifies a sensor node (SensorNodeID), where it is deployed (Map), and its relative coordinates on the map (x, y). A sensor node ID uniquely determines where it's deployed, written as [SensorNodeID] → [Map, x, y].

Building information is organized as a table containing entries for each building address, and a table of floor maps. Each tuple in the Building table specifies a building (BuildingID), and its address (HouseNumber, Street, Town, Postcode, Country). A building ID uniquely determines its address, written as [BuildingID] → [HouseNumber, Street, Town, Postcode, Country]. Each tuple in the FloorMap table specifies what a map describes (BuildingID, Floor) and where the map is stored (Map). A building and floor pair uniquely determines its map, written as [BuildingID, Floor] → [Map].

In the case of an emergency, the building data can be retrieved by searching by its address and postcode from the central database, then the required floor map can be retrieved and displayed. Time-related information is associated with sensor data in the SensorReading table, and the real-time sensor readings can be associated with its location on the floor map through the bridge of the SensorDeployment table.

Time-driven data management

Sensor data is real-time streaming data, which means that as the time duration increases, the size of the database will grow, therefore, the query efficiency will drop. A time range based partitioning is used to maintain the query efficiency.

Partitioning is the division of a database or its element into smaller, more manageable parts, according to the partition key, so that only the partitions that contain the answer to the query will be scanned rather than the whole dataset. Partitioning gives benefits such as manageability, performance and scalability.

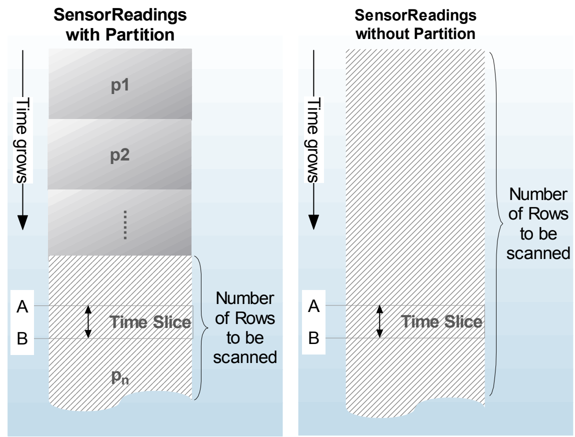

Figure 2. Time range based database partition.

In the scenarios that we are concerned with, typical queries are time range based queries such as: "what is the distribution of sensor readings in the area S between time A and time B?" Partitioning the table SensorReadings by time range will store the data in more manageable parts according to the timestamp, therefore although only the number of partitions will increase as the size of the dataset grows, the number of rows that need to be scanned for any query will be kept relatively stable, since only partitions that contain answers to such queries will be scanned instead of the whole dataset, as shown in Figure 2. Therefore, the database can still maintain query efficiency as the volume of data increases.

Dynamic support for different incident stages

The system is also designed to collect data adaptively according to different incident stages. Before an incident, data collection can be configured at a low frequency. Sensors can be in sleep mode most of the time and only wake up once in a while to monitor the environment. When a suspicious state is detected, data collection frequency can be increased. During incidents, sensor readings will be saved in real-time. Finally, the data collection frequency can be set back to normal after the incident.

The benefits of the adaptive behaviour are twofold, it reduces the energy consumption of the sensor network, and it reduces the amount of data storage space that is required (as shown in Figure 3).

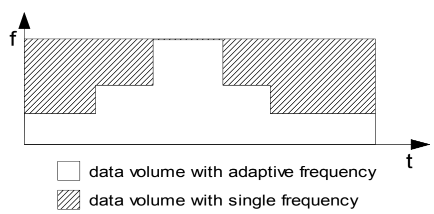

Figure 3. Adaptive sampling frequency.

4. Data Cleaning

Sensor data cleaning mainly deals with three data quality problems: noise, outliers, and missing values.

Researchers have used approaches such as Bayesian Theory (Elnahrawy and Nath, 2004), Neural Network (Petrosino and Staiano, 2007), Wavelets (Zhuang and Chen, 2006), Kalman Filter and Weighted Moving Average (Zhuang, Y. et. al., 2007) for sensor data cleaning. However, using probability theory to predict the most likely range that sensor sampling falls in has limitations in application to ER, because emergencies are unpredictable events, and using probabilities learned from data collected from one environment to predict sensor samplings in another environment may not be appropriate. Approaches based on Neural Networks are usually based on theories and simulations, but the possibility of implementing such a complex system in practical situations has not been envisaged. Wavelet transformation based methods require an initial dataset large enough to be able to separate noise from real data, and is therefore more suitable for the analysis of historical data rather than real time data. Compared to the above, Weighted Moving Average is easy to implement, however its flexibility is limited and the neighbour tests phase might be an issue because it may result in neighbours checking each other, causing overflowed checking requests. Comparing to these, Kalman Filter method was chosen for further research because of its good performance in filtering data and lightweight implementation.

Accuracy and timeliness are vital for ER. Therefore, removing outliers in the sensor data as well as quickly reflecting any real changes in the sensed values is the main focus of the sensor data cleaning approach designed. The designed algorithm adopted a Kalman Filter to filter noises, and integrated a proposed sensor state model and neighbour support with the Kalman Filter to achieve the goal of removing outliers as well as quickly reflecting real changes.

4.1 Kalman Filter

A Kalman Filter can be applied to smooth noisy sensor input for further processing. It is represented by the system model $x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}$, which calculates the state x_k from its previous state x_{k-1} and control input u_{k-1}, with a measurement model $z_k = Hx_k + v_k$ .The random variables w_{k-1} and u_{k-1} represent the process and measurement noise (respectively). They are assumed to be independent (of each other), white, and with normal probability distributions: $p(w) \sim N(0,Q)$, $p(v) \sim N(0,R)$.

In a broad high-level overview, the on-going Kalman Filter cycle can be summarized as:

Step 1: Time Update. The filter calculates the estimated state x_k^- from its previous state using equation $x_k^- = Ax_{k-1} + Bu_{k-1}$, (1)

and calculates the error covariance of the estimation $P_k^- = AP_{k-1}A^T + Q$. (2)

Step 2: Measurement Update. It adjusts the projected estimation by an actual measurement at the time, denoted as the adjusted estimation $x_k = x_k^- + Kg_k(z_k - Hx_k^-)$, (3)

where $Kg_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$, (4)

and then updates the error covariance of the estimation $P_k = (I - Kg_k H)P_k^-$. (5)

The updated x_k and P_k will be taken as feedback to calculate the estimation for the next time.

More about Kalman Filters can be found in the reference (Welch and Bishop, 2006).

4.2 Sensor State Model

For each sensor being monitored, there are four states.

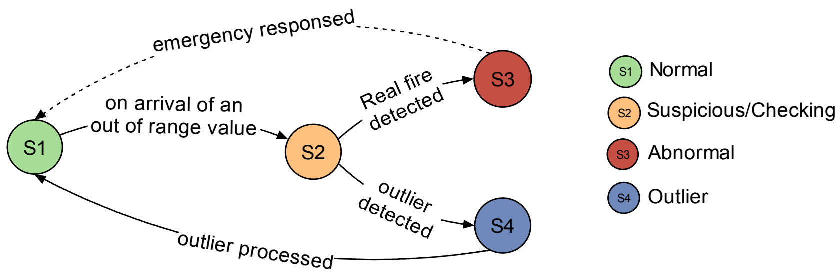

Figure 4. Sensor State Model.

S1 (Normal): A state s is considered to be normal when the incoming reading z_k is within a preset range Tv, and the rising rate is within a preset range Tr, denoted as
$s \in S_1 \Leftrightarrow z_k \in Tv \wedge (z_k - z_{k-1}) \in Tr$ (i)

S2 (Suspicious/Checking): A suspicious state appears when the current reading is beyond the preset range or the rising rate is beyond the preset range, denoted as
$s \in S_2 \Leftrightarrow z_k \notin T_r \vee (z_k - z_{k-1}) \notin T_v$ (ii)

When a suspicious state occurs, the system will enter into a checking state. It will check the Neighbour Support, which is the ratio of the number of neighbours who are in the same suspicious state to the number of neighbours in total. $NS = \frac{N_{S_2}}{N_n}$ (iii)

S3 (Abnormal): If the Neighbour Support is above a threshold, it indicates that an incident is detected, denoted as $s \in S_3 \Leftrightarrow s \in S_2 \wedge NS > th$ (iv)

S4 (Outlier): If the Neighbour Support is not enough, it is considered to be an outlier. Outliers will be logged, and the sensor will go back to its previous state, denoted as
$s \in S_4 \Leftrightarrow s \in S_2 \wedge NS < th$ (v)

4.3 Sensor State Model integrated Kalman Filter

A Kalman Filter is efficient to reduce noise. However, it takes time to reflect a change in the environment and it does not separate outliers from real environmental changes to deal with them differently. Therefore, the sensor state model is integrated with a Kalman Filter in our proposed algorithm. The integrated Kalman Filter adjusts the projected estimation according to the different state of the sensor. Therefore, it can remove outliers as well as quickly reflect the environmental changes. The on-going cycle of the integrated Kalman Filter can be described as follows:

Step 1: Estimation/Suspicious Detection The filter calculates the estimation x_k^- and its error covariance P_k^- using equation (1) and equation (2) respectively, at the same time detecting for a suspicious state using rule (ii).

Step 2: Measurement Update If a suspicious state is found, Neighbour Support is integrated to decide whether it is a real abnormal state or an outlier. If it is an abnormal state, the current sensor reading is used instead of the adjusted estimation, and P_k is increased to quickly reflect the change; If it is an outlier state, the adjusted estimation is replaced by the last normal data. If a normal state is found, the normal Kalman Filter update is used.

5. Performance Evaluation

Two groups of experiments have been carried out to evaluate the performance of the proposed Data Storage Mechanism and the Data Cleaning Algorithm.

<u>Data storage mechanism experiment</u>

The main purpose of the data storage experiment was to evaluate the query efficiency of our database design whilst real-time data continues feeding into the database.

A data generator, databases and a query engine were implemented for this experiment, as shown in figure 5. The data generator generated random data at certain time intervals. The real-time data generated was then stored in both a database partitioned by range of timestamp and a second database without partitioning. The query engine performed the same time range queries on both databases.

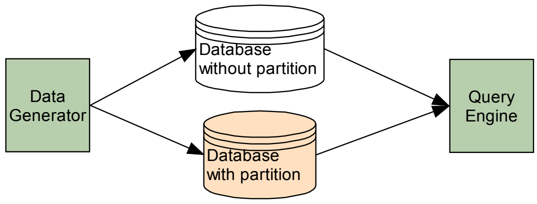

Figure 5. Database storage mechanism experiment.

The query efficiency was expected to remain the same regardless of how large the database was becoming.

The experimental results are shown in table 1.

Size of database(rows)	Query Efficiency					
	Database with partition			Database without partition		
	Rows scanned	Partitions scanned	Time needed	Rows scanned	Partitions scanned	Time needed
5124	1524	p2	0s	5124	all	0s
60586	1800	p32	0s	60586	all	3s
70139	1800	p37	0s	70139	all	6s

Table 1. Database query experimental result.

As the database becomes larger, for the same range query, the Database Management System (DBMS) has to scan more rows to find answers to the query from the database without partitioning, however the number of rows that the DBMS scanned in the partitioned database remains the same, because only the partition that contains the data was scanned.

This experiment showed that the proposed data storage mechanism can maintain the query efficiency regardless of the overall size of the dataset. Consequently, it could be advantageous in Emergency Response Systems, as efficient querying can lead to an efficient response to emergencies.

Data Cleaning Experiment

The Data cleaning experiment aimed to test three things: a) Reducing the noise in the sensor data b) Removing outliers c) Quickly reflecting real changes in the environment.

Data from a 5*5 network for the duration of 300 time frames was simulated using MatLab, with random noise and outliers (spikes) added. The test environment considered the scenario of fire occurrence (represented by room temperature normally at around 20°C suddenly rising to around 60°C) at a random time frame and at a random location, and spreading to the nearby area. The proposed data cleaning algorithm is expected to be able to remove noises and outliers as well as quickly reflect the real change in the environment that indicates a potential fire. The result is shown in figure 6.

Comparing graph (a) with graph (b), the noises were reduced by the Kalman Filter, but there was an obvious delay in reflecting the fire occurring at Timeframe=186, and the outlier at Timeframe=278 were smoothed rather than being removed. Comparing graph (b) with graph (c), the noises and outliers existing in the raw data were effectively removed by the Sensor State Model integrated Kalman Filter, and the real environmental change was quickly reflected.

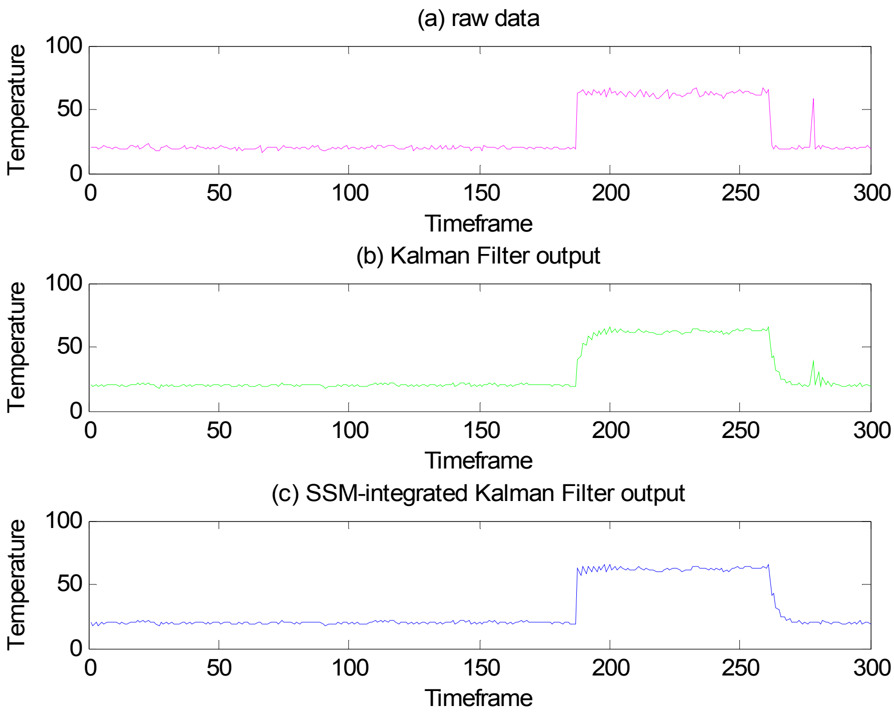

Figure 6. Data Cleaning Experiments.

6. Conclusion

In conclusion, to make sensor data work for emergency responses, sensor data needs to be properly pre-processed and stored and managed efficiently. This must take place before further data processing /fusion can be done on the post-processed data to help extract meaning for the incident commanders, an example of this would be projected failures of infrastructure due to critical temperatures.

In this paper, we have analyzed the requirements and challenges in designing such Wireless Sensor Network based Emergency Information Management System, and proposed a data storage mechanism and a data cleaning approach. The proposed data storage mechanism stores and manages both real-time sensor readings and static information required by emergency response applications. It provides dynamic support to different incident stages, and maintains query efficiency during the incident in spite of the incident duration. The proposed Sensor State Model integrated Kalman Filter algorithm reduces noise, removes outliers as well as quickly reflects real abnormalities in the monitoring environment.

During the incident, different areas in the building can have different Levels of Seriousness. A higher Level of Seriousness suggests a potentially higher risk area. The incident commanders have confirmed that a real-time visual distribution of this information with colour code can increase their situation awareness on site.

Future work will develop a location-based meaning extraction algorithm on top of the sensor data, including means of determining the Level of Seriousness from the sensor data, and how to integrate this information with floor maps.

References

Elnahrawy, E. and Nath, B. (2003) Cleaning and querying noisy sensors. In proceedings of 2nd ACM international conference on Wireless sensor networks and applications, (San Diego, CA, USA, 2003), 78-87.

Elnahrawy, E. and Nath, B. (2004) Context-aware sensors. In H. Karl, A.Willig, A.Wolisz (Eds.): Wireless Sensor Networks, pp. 77–93, 2004.

Ganesan, D. et. al. (2003) An evaluation of multi-resolution storage for sensor networks. In SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems, Los Angeles, California, USA, Anonymous ACM, New York, NY, USA, 89-102.

Jeffery, S.R. et. al. (2006). A Pipelined Framework for Online Cleaning of Sensor Data Streams. In proceedings of the 22nd International Conference on Data Engineering,(April 03-07, 2006)

Kim, C. H. et. al. (2005). Architectures for streaming data processing in sensor networks. In Proceedings of the ACS/IEEE 2005 international Conference on Computer Systems and Applications (January 03 - 06, 2005). AICCSA. IEEE Computer Society, Washington, DC, 59-I.

Mathur, G. et. al. (2006) Ultra-low power data storage for sensor networks. In IPSN '06: Proceedings of the fifth international conference on Information processing in sensor networks, Nashville, Tennessee, USA, Anonymous ACM, New York, NY, USA, 374-381.

Petrosino, A. and Staiano, A., (2007) A Neuro-fuzzy Approach for Sensor Network Data Cleaning. In Knowledge-Based Intelligent Information and Engineering Systems. pp.140-147.

Walle, B.V.D. and Turoff, M., (2007). Emergency Response Information Systems: Emerging Trends and Technologies. Communications of the ACM, 50(3), pp. 29-31.

Zhuang, Y. et. al. (2007) A Weighted Moving Average-based Approach for Cleaning Sensor Data. In proceedings of 27th International Conference on Distributed Computing Systems.(25-27 June 2007) pp. 38-48.

Zhuang Y, Chen L. (2006) In-network outlier cleaning for data collection in sensor networks. In *Clean DB Workshop: ACM SIGMOD*, 2006.

Author Biography

Ms Yanning Yang is currently a PhD research student in the Computer Science Department, Loughborough University, UK. She received the MSc degree from Loughborough University in 2005, and BEng degrees from Beijing Institute of Technology, China in 2004. Her research interests include information retrieving and presentation, sensor data processing, and real time data transfer over the Internet.

Andrew May is a Research Fellow at the Ergonomics and Safety Research Institute, Loughborough University. He has a BEng in mechanical engineering (Birmingham University), an MSc in Human Factors (University College London) and is currently completing a PhD in the human factors of mobile location services. He has published widely in human factors journals. His research interests are adding value through user-centred design, mobile location services and research methodologies.

Professor Shuang-Hua Yang received a B.Sc. degree in instrument and automation, M.Sc. degrees in process control from University of Petroleum in 1983 and 1986 respectively, and a Ph.D. degree in intelligent systems from Zhejiang University in 1991. He is Professor of Networks and Control and the Director of the Networks and Control Research Group at Loughborough University. He is also an overseas professor in Central China Normal University and a guest professor in Huazhong University of Science and Technology. He is a fellow of the Institute of Measurement & Control (FInstMC), a senior member of IEEE, the chairman of the East Midlands Section of InstMC, and a Chartered Engineer (CEng) in the UK. He is an associate editor of the International Journal of Systems Science and a member of the editorial advisory board of the International Journal of Information and Computer Security, and an editorial committee member of the Journal of the Institute of Measurement and Control. His research interests include wireless sensor networks, networked control, safety critical systems, and real time software maintenance.