Towards Adaptive Sensor Data Quality Improvement based on Context Models

Aboubakr Benabbas, Simon Steuer, Daniela Nicklas
Chair of Mobile Systems, University of Bamberg, Bamberg, Germany
{aboubakr.benabbas, simon.steuer, daniela.nicklas}@uni-bamberg.de

***Abstract*—Pervasive applications use context information for decision making and to adapt to new circumstances at run time. We can derive this context from sensors (dynamic) or from the deployment information (static). Some applications rely on a combination of context information sources to evaluate the quality of the data received. Besides, different context sources enable the applications to adjust to changes in the configuration as soon as they happen. The possible adaptation can include a change in the data processing to incorporate data quality improvement approach to provide a better context to the application. In this paper, we offer an adaptive data quality improvement based on a combination of context sources that we model using a domain-specific ontology.**

***Index Terms*—Data Quality, Context Modelling, Semantic Sensor Models, Data Stream Processing**

I. Introduction

Sensors have become omnipresent. Sensor-based applications use those sensor-generated data in different situations and for different purposes. Pervasive applications have different use cases like health monitoring [1], traffic congestion estimation [2], crowd counting [3] and pervasive sensing in smart cities [4].

Sensor data is often faulty, Also, as reported by Lohr [5], data scientists spend between 50-80% of their time dealing with data quality problems. Therefore, application developers need to find ways to assess the quality of the data before they use it. However, quality of sensors is context-dependent, i.e., it changes depending on the conditions sensed environment. Context information is a source for data quality assessment [6]. Context information can either be sensed (sensor measurement), static (defined in deployment), profiled (user configuration), or derived (from other context sources) [7]. Examples of context-dependent applications include but are not limited to: Elderly care [8], human activity recognition [9], adaptive activity recognition based on dynamic context and different sensor data [10] [11], and model-based prediction using derived context [12].

If we decide to use sensor data without any quality assessment and/or correction, we simply use data that is mostly inaccurate or even false. However, if context information is used while collecting data online, we can avoid the loss of context data by storing it and using it to perform data quality correction. Although the creation of elaborate context models can be a complex task, we can use it to store all the static context information and enrich it with additional functions to react to various context changes. Domain-specific ontologies like SSN (Semantic Sensor Network) [13] ease the modelling task by providing most of the necessary terms to describe the sensors and the different context information.

Since sensors can deliver data continuously as streams, the option of using a *Data Stream Management System* like *Odysseus* [14] to handle the incoming data provides a good option: Such systems provide a rich set of data processing functions called operators. Operators can be combined in processing units called queries to process the data according to the application needs. We can use the combination of context models and data stream processing to adapt the data processing, when changes in the context occur.

The contributions of this paper include:

- How we use an ontology to model the static, sensed, profiled and derived context information.
- How we define data processing patterns and we use them to assess and improve sensor data quality in a Data Stream Management System
- How we evaluate our approach with a real world use case to compare the impact on the application with and without data quality assessment.

We argue that complete sensor models should provide a description of the context. They can also include defined processing patterns that can be triggered if the context information changes at run time. This makes the model adaptive to changes in the context and enables the application to include data quality improvements when needed.

The rest of the paper is organized as follows: In Section II, we introduce our use case. In Section III, we introduce the sensor model to describe the context and parts responsible for the adaptive behaviour to improve the quality of the sensor measurements. Afterwards, we evaluate our method by comparing the results with and without the introduced quality assessment in Section V. In Section VI, we discuss related work for adaptive context-aware modelling. Finally, we wrap up in Section VII.

II. Use Case

To emphasize the importance of data quality in pervasive and sensor-based application, we take the use case of a sensor-based application that would benefit from data quality improvement. The use case shows examples of data quality problems. Besides, we point out the data processing patterns that work to improve the quality.

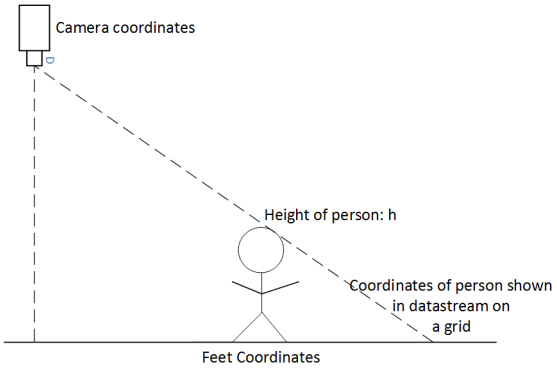

Fig. 1. The position calculation process

Crowded indoor environments provide organizers and security personnel alike with huge challenges. They aim for a high security level and the best possible visitor experience. Airports, fairs and expositions are examples of such environments, where people tend to have a repeated behaviour like queuing up for checks or standing before a stall or exhibit.

Information about the visitors, their sizes, and hotspots is very valuable to take safety and logistic measures. For this end, the use of cameras to track the human flows and their concentration points is the obvious choice. However, with the increasing fears around people's privacy and how to prevent any unlawful infringement on their personal information imposes high restriction on the collected data. No personal photos or videos are allowed to be stored without the consent of the concerned person.

To remedy this problem, privacy-preserving overhead cameras have been developed that track people, without storing any visual data of the persons seen.

A. Camera Function

Figure 1 shows how the camera captures the coordinates of a moving person. The camera mounted on the ceiling of the room scans the area and captures the moving objects. It takes the pictures of moving persons, measures the height, records the coordinates and gives a unique ID to the object. The camera functions as a black box for the developers of the sensor-based applications. The pictures are processed to extract the persons and coordinates are produced.

The camera has two modes of counting: a head-mode and a feet-mode. While the feet-mode is more accurate in locating the positions of the moving objects, it shrinks the tracking area of the camera. The head-mode, however, enables a large tracking area and thus, a possibly more accurate count.

These cameras deliver continuous streams of data containing information about the person. One stream is called 'Object Stream': It gives the size of the moving person and its current position. The second stream is called 'Event Stream': It produces *LINECROSS* events, when the moving persons cross preconfigured counting lines. Additionally, the camera has a web interface that delivers aggregated data about counting zones and lines like the number of line crosses. Due to privacy regulations, no images or footage are stored. As a consequence, we cannot perform offline data video processing. In addition, the cameras do not perform facial recognition (overhead), therefore, it is difficult to deal with some of its problems like *the inaccuracy problem* and *The missing counts problem*

B. Camera Problems

The inaccuracy problem: The height problem is caused by the distortions incurred by the increasing distance from the camera. This is reflected in the data generated by the cameras. This problem can be observed and modelled to reflect a *distance to sensor* pattern to capture the effect of the sensed context (distance of the moving person to the camera) on the error in her measured heigh.

The missing counts problem: The missing counts problem occurs when the preconfigured counting lines of the camera fail to identify *LINECROSS* events and count their occurrences. This happens mostly when the camera is set to head-mode, where the coordinates in the data stream are not the feet coordinates, but rather deviated coordinates (see Figure 1). The wrong coordinates make the intersection with the counting lines impossible at times, causing the camera to deliver wrong counts. This behaviour implies a wrong count due to a change in the profiled context (settings). This change in the context can trigger a change in the data processing by introducing a *data correction* pattern, in which new data processing queries are started to correct the wrong coordinates and report the missing counts.

In the next section, we see how the context model manages to capture the different context information to enable the adaptive data quality improvements through processing patterns.

III. The Sensor Model

First, we introduce the processing patterns. Afterwards, we show how they are expressed in the model. We have two processing patterns, that describe how is data from different sources is combined to assess and/or improve the quality of the data:

1) The *distance to sensor* pattern: this pattern uses a dependency between the context sensed by the sensor and the distance to the sensor. This dependency can be defined through user-defined functions.
2) The *data correction* pattern: this pattern uses user-defined functions to correct the data. The functions can be plugged into the flow of data processing based on the model description and the context.

The processing patterns are not specified explicitly. They are described by data quality conditions, the current context, and the quality improvement processes.

The context model should include all the needed context information about the camera and capture the adaptive behaviour dictated by the change in the sensed and profiled context. The model contains description of processing patterns to assess the quality of the data. The adaptation to changes in the context will trigger processes to improve the quality of the data.

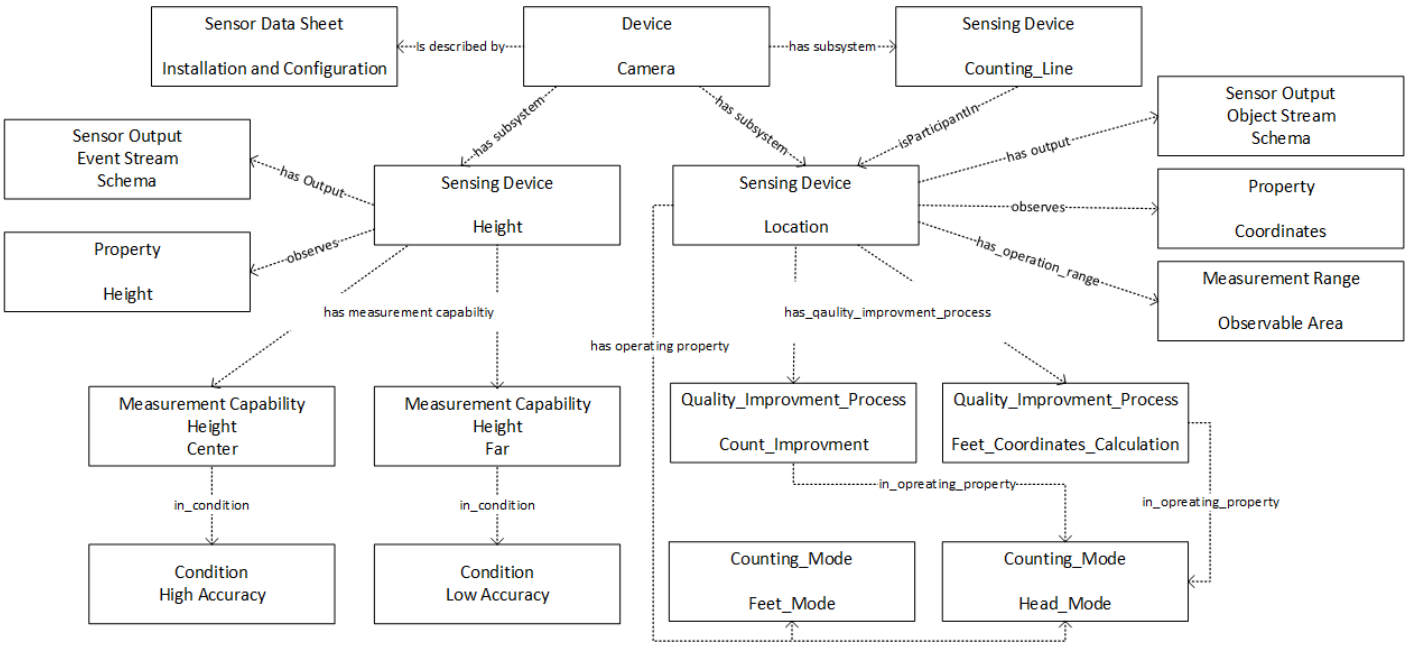

Fig. 2. SSN model of the camera

The sensor model that manages to describe the various context information about the sensor is depicted in Figure 2. The static context can be obtained from the 'Sensor Data Sheet'. Here, we set the location of the camera and the mounting height. This context information does not change and allows us to always determine the coverage area of the camera. The camera has two data streams that can be modelled as subsystems; one for the height measurement of the tracked person and one for her coordinates. The profiled context is the set of user defined settings that can be changed at any time during the operation of the sensor. Such settings include The counting lines, which provide the number of persons that cross it. Since these lines provide a count, they are considered as a subsystem too.

The derived context that influences the accuracy of the height measurement can be defined as a set of 'Measurement Capabilities' with the conditions specified, in which they occur. The 'High Accuracy' condition sets the distance to the camera, in which the height measurements are good and do not need any correction. The 'Low Accuracy' condition sets the distance, in which the height measurements are faulty. The sensed context provided by the 'Location' is used to compute the distance to the camera and choose which condition is met. This model description of the dependency between the location of the moving person and the accuracy of the height measurement describes the *distance to sensor* pattern. When the condition of 'Low_Accuracy' is satisfied, the data stream management system uses a set of operators in a query to perform a height measurement correction.

The adaptive data quality improvement part is described by the 'Quality_Improvement_Processes'. These are switched on and off based on the current 'Operating Property'. The 'Operating Property' specifies the Counting_Modes the camera has. If the 'Counting_Mode' is set to 'Feet_Mode' by the user, then, no action is taken and no quality improvement processes are triggered. If the profiled context changes, when a user switches the 'Counting_Mode' to 'Head_Mode', a reconfiguration signal is sent to the system and the model activates the 'Count-Improvement' and the 'Feet_Coordinates_Calculation' processes. The processes constitute the data correction pattern. In the following section, we show how is the process of data quality improvement is performed by using data processing patterns in a data stream management system.

IV. Implementation

In this section, we show processing patterns to assess and perform correction on the data and the architecture of the system. In Figure 3, we have the Sensor Quality Management (SQM) component that receives the context information from the sensor model and creates the data processing queries. It also creates quality-aware queries when the model contains quality improvement processes and processing pattern. The data stream management system (DSMS) has a component to manage deployment and configuration information. When the sensors change their configuration or deployment settings, the DSMS sends a signal to the SQM to adapt the queries to reflect the change. in the profiled context (configuration).

A. The 'distance to sensor' pattern for height measurements

Through testing and data collection, we discovered that the height of the tracked people is not accurate. The data analysis showed a high correlation value of 0,86 between the distance of the tracked person to the camera and the error percentage of the height measurement. This can be used to create a linear

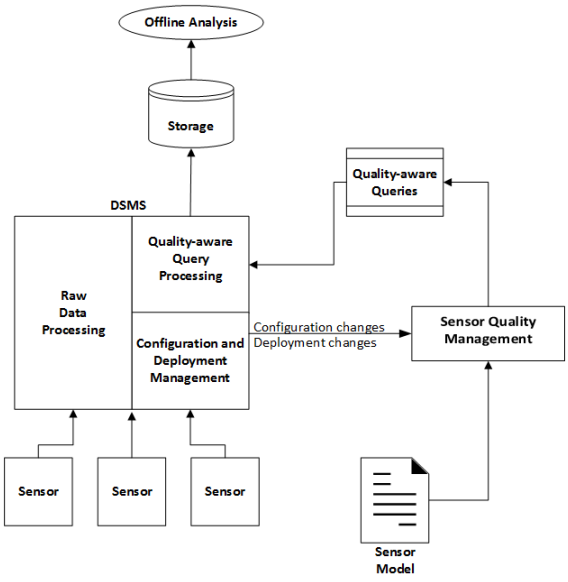

Fig. 3. The System Architecture

regression model that describes the influence of the distance to the camera on the height error of the measurement as follows:

$$err = a(dtc) - \epsilon$$

Where *dtc* is the distance to the camera and *err* is the error rate in the recorded height. This derived context information from the data analysis can be added to the sensor model as a quality constraint. This linear regression model is good for estimating the error when the person is at least half a meter or more away from the camera. We take the measurement of the camera when the person is less than half a meter away. As illustrated in Figure 4, the query represents the processing pattern. The join operator combines the streams coming from the two sensors, computes the distance to the camera and computes the accuracy based on the regression.

B. The 'data correction' pattern for line count improvement

In order to solve the problem of missing line counts, we need to determine the feet position of the moving person first. We compute the feet position and then, check for any overlooked *LINECROSS* events and output them. In order to identify such events, we need to continuously build a vector from the current position and the last position of the moving person and check for a potential intersection with the closest counting line. This data correction pattern is presented in the query depicted in Figure 5.

V. Evaluation

To evaluate the impact of our approach in terms of the data quality improvement, we carried out experiments to gather data and compare it with the results of the processing pattern queries.

TABLE I
Mean Average Error and Mean Average Percentage Error of LINECROSS-events

MAE Query	MAE Camera	MAEP Query	MAEP Camera
21,5	35,1	0,42	2,37

A. Improving the accuracy of height measurements

We apply the *distance to sensor* pattern on the camera data and make corrections of height values. To get a measure of the impact that this correction has on the data, we compared the height measurements made by the camera and took always the best average value as the average height given by the camera for a moving person. Then, we took the corrected height measurements and computed the best average corrected value from the corrected height values for a moving person produced by the query in Figure 4. We compare the accuracy results of both height values with each other and without any correction. We see in Figure 6 that the "best average value approach" gives a low error percentage up to 1 meter away from the camera. In distances longer than 1 meter, we see how the error percentage grows further. The proposed approach of "best corrected average value" keeps the error percentage low, even up to a distance of 6 meters.

B. Improving the accuracy of counting lines

To measure how the count line improvement works, we designed experiments to evaluate the feet coordinates computation and the line count improvement. In the first one, we defined positions, in which a test person should stand and asked him to move around inside the area observed by the camera. In Figure 7, we see the deviation between the ground truth coordinates used for the experiments and the coordinates computed by the improvement query. The results show an average deviation from the ground truth coordinates of 1,43% of the x-coordinate and 2,52% of the y-coordinate. The average deviation of the x-coordinate remains fairly low, whereas the y-coordinate gets higher after 2,5 meters.

In the second experiment, we set different paths for test persons with many counting lines set in different positions. The results shown in Table I show the Mean Absolute Error and the Mean Absolute Percentage Error on the *data correction* pattern in comparison to the counts made by the camera. We see that the pattern manages to improve the counts by reporting more counts than the camera.

From the results of the experiments, we see that the queries manage to improve the accuracy of the data by providing a better height measurement and giving better line counts.

VI. Related Work

Earlier research in context-aware applications focused on context modelling and reasoning [15]. This laid the groundwork for further research in the area of context-aware sensor applications. Huebscher and McCann proposed an adaptive middleware design for context-aware applications [16]. This is

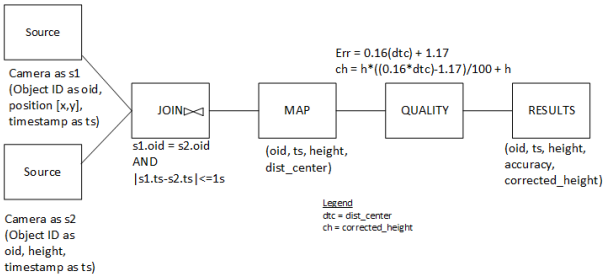

Fig. 4. Query plan to compute the accuracy of the height and correct it

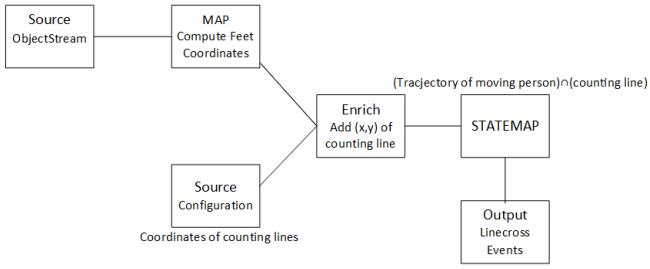

Fig. 5. Query plan to improve the counts of the counting lines

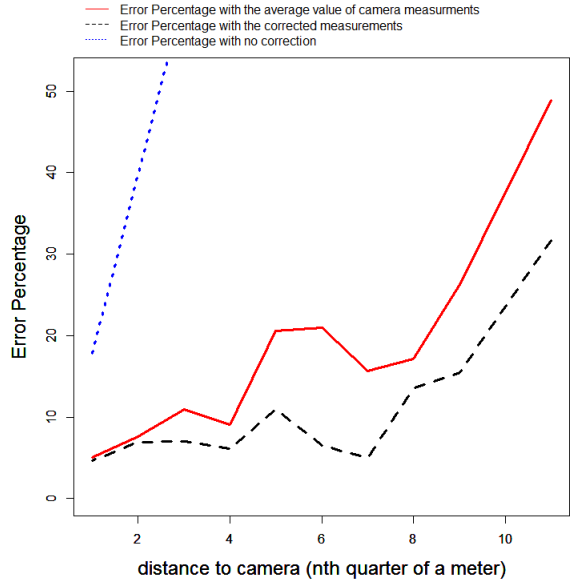

Fig. 6. Comparison of the error rate to the no correction approach

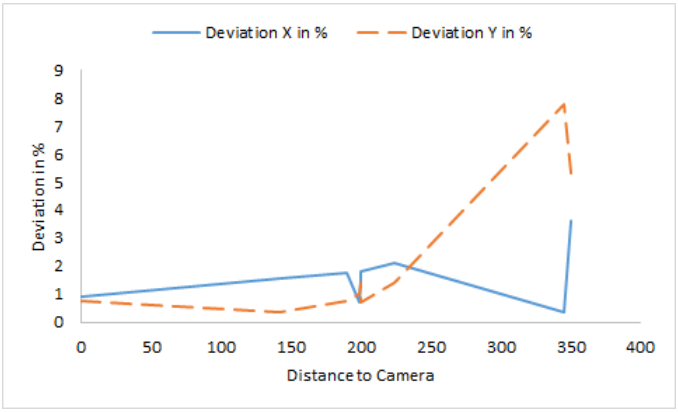

Fig. 7. Results for the feet-position of first test person in controlled positions

one of the earliest relevant works that abstract the applications from the provided sensor context. Only few works in literature demonstrate the sensor node itself as a a context-aware device. Taherkordi et al. introduced a self-adaptive context processing framework for wireless sensor networks (WSN) [17]. The mentioned contributions tried to create entities to provide context information, either by having a middleware or by the sensor node itself. However, these contributions did not emphasize much the issue of the context quality collected by the sensors. Also, the context-based adaptability does not refer to the quality of the sensor data.

Many authors have discussed the issue of data quality. Yates et al. examined data quality for pervasive sensing systems [18]. They focused on the impact of latency and caching on data quality. Sheikh et al. proposed a middleware that decouples applications from the data producing sensors [19]. The middleware aims to make up for the quality limitations of the context sensors. Schiffer defines quality of context as "any information that describes the quality of information that is used as context information" [20]. Batini et al. offered an interesting view on the data quality dimensions and their respective definitions in [21]. Batini et al. defined clear dimensions of quality like accuracy and completeness. From the related work described above, we see that work in the domain of data quality focused on describing the quality dimensions with the use of different parameters and the value range for each parameter. The research in this area did not examine the impact of data quality on the processing results and focuses on data quality without considering methods to improve it.

Work on Quality of Service in data stream processing aims at producing Data Stream Management Systems that are quality-aware. Schmidt et al. developed a deterministic data stream processing system called QStream that offers QoS parameters to users to choose from [22]. Abadi et al. proposed a dynamic optimization model for operators to optimize different QoS metrics across a combined server and sensor network [23]. Klein and Lehner presented a flexible model for the propagation and processing of data quality in a stream processing network for sensor data in a smart environment [24]. These contributions described the influence of the different operators on data quality but did not target the context-induced data quality issues and how to improve it.

Camera-based approaches are widely used for crowd counting. Some of the contributions include the work of Belongie and Rabaud, where crowd counting is done through motion recognition [25]. Another approach uses low level feature detection to track people [26]. These approaches do not use anonymizing cameras and do not process the data online to track people in real time.

From the above discussion we highlight the importance of our work, where we present our approach of combining either static context with sensed context information or profiled context and sensed context information to trigger adaptive data quality processing improvement. The use of domain-specific ontology enables the reuse of the context model on a larger scale.

VII. Conclusion

In this paper, we propose the use of context models to improve the accuracy of the data produced by sensors by adapting the processing to the changes in context. We showed the feasibility of our approach through the use case of anonymizing people counting cameras and how these provide inaccurate height measurements and counts. Through different variants of context, we managed to provide processing patterns that monitor the quality of the data and trigger data quality improvements when needed. This work comes as a follow-up to a previous work [27], in which we monitored the quality of low-cost particulate matter sensors using semantic sensor models. In the future, we plan to extend the ontology with rules to infer those patterns through reasoning and automatically generate the data processing queries.

Acknowledgement

We would like thank Philipp Grandl[1] for his help and valuable input in the process of writing this paper. He has done the data analysis and conducted the experiments and carried out a part of the evaluation.

References

[1] U. Varshney, "Pervasive healthcare and wireless health monitoring," *Mob. Netw. Appl.*, vol. 12, no. 2-3, pp. 113–127, Mar. 2007.

[2] R. Bhoraskar, N. Vankadhara, B. Raman, and P. Kulkarni, "Wolverine: Traffic and road condition estimation using smartphone sensors," in *2012 Fourth International Conference on Communication Systems and Networks (COMSNETS 2012)*, 2012, pp. 1–6.

[3] S. Depatla and Y. Mostofi, "Crowd counting through walls using wifi," in *2018 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2018, pp. 1–10.

[4] F. Al-Turjman, "Mobile couriers' selection for the smart-grid in smart-cities' pervasive sensing," *Future Generation Computer Systems*, vol. 82, pp. 327 – 341, 2018.

[5] S. Lohr, "For big-data scientists, 'janitor work' is key hurdle to insights," *The New York Times*, 2014.

[6] L. Bertossi, F. Rizzolo, and L. Jiang, "Data Quality Is Context Dependent," in *Enabling Real-Time Business Intelligence*, ser. Lecture Notes in Business Information Processing, M. Castellanos, U. Dayal, and V. Markl, Eds. Berlin, Heidelberg: Springer, 2011, pp. 52–67.

[7] K. Henricksen and J. Indulska, "Modelling and using imperfect context information," in *IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second*, March 2004, pp. 33–37.

[8] C. Stamate, G. Magoulas, S. Kueppers, E. Nomikou, I. Daskalopoulos, M. Luchini, T. Moussouri, and G. Roussos, "Deep learning Parkinson's from smartphone data," in *2017 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. Kona, Big Island, HI, USA: IEEE, Mar. 2017, pp. 31–40.

[9] T. Sztyler and H. Stuckenschmidt, "Online personalization of cross-subjects based activity recognition models on wearable devices," in *2017 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. Kona, Big Island, HI, USA: IEEE, Mar. 2017, pp. 180–189.

[10] J. Wen, J. Indulska, and M. Zhong, "Adaptive activity learning with dynamically available context," in *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. Sydney, Australia: IEEE, Mar. 2016, pp. 1–11.

[11] Chongguang Bi, G. Xing, T. Hao, Jina Huh, Wei Peng, and Mengyan Ma, "FamilyLog: A mobile system for monitoring family mealtime activities," in *2017 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. Kona, HI: IEEE, Mar. 2017, pp. 21–30.

[12] M. D. Tomaras, M. I. Boutsis, and V. Kalogeraki, "Modeling and Predicting Bike Demand in Large City Situations," *IEEE International Conference on Pervasive Computing and Communications*, p. 10, 2018.

[13] Michael Compton et al., "The SSN ontology of the W3C semantic sensor network incubator group," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 17, pp. 25 – 32, 2012.

[14] H.-J. Appelrath, D. Geesen, M. Grawunder, T. Michelsen, and D. Nicklas, "Odysseus: A highly customizable framework for creating efficient event stream management systems," in *Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems*, ser. DEBS '12. New York, NY, USA: ACM, 2012, pp. 367–368.

[15] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, and D. Riboni, "A survey of context modelling and reasoning techniques," *Pervasive and Mobile Computing*, vol. 6, no. 2, pp. 161–180, Apr. 2010.

[16] M. C. Huebscher and J. A. McCann, "Adaptive Middleware for Context-aware Applications in Smart-homes," in *Proceedings of the 2Nd Workshop on Middleware for Pervasive and Ad-hoc Computing*, ser. MPAC '04. New York, NY, USA: ACM, 2004, pp. 111–116, event-place: Toronto, Ontario, Canada.

[17] A. Taherkordi, R. Rouvoy, Q. Le-Trung, and F. Eliassen, "A Self-adaptive Context Processing Framework for Wireless Sensor Networks," in *Proceedings of the 3rd International Workshop on Middleware for Sensor Networks*, ser. MidSens '08. New York, NY, USA: ACM, 2008, pp. 7–12, event-place: Leuven, Belgium.

[18] D. J. Yates, E. M. Nahum, J. F. Kurose, and P. Shenoy, "Data quality and query cost in pervasive sensing systems," *Pervasive and Mobile Computing*, vol. 4, no. 6, pp. 851–870, Dec. 2008.

[19] K. Sheikh, M. Wegdam, and M. Van Sinderen, "Middleware support for quality of context in pervasive context-aware systems," in *Pervasive Computing and Communications Workshops, 2007. PerCom Workshops' 07. Fifth Annual IEEE International Conference on*. IEEE, 2007, pp. 461–466.

[20] Thomas and M. Schiffers, "Quality of context: What it is and why we need it," in *In Proceedings of the 10th Workshop of the OpenView University Association: OVUA 03*, 2003.

[21] C. Batini and M. Scannapieco, *Data Quality: Concepts, Methodologies and Techniques (Data-Centric Systems and Applications)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[22] S. Schmidt *et al.*, "Qstream: Deterministic querying of data streams," in *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, 2004.

[23] Abadi *et al.*, "The Design of the Borealis Stream Processing Engine." in *CIDR*, 2005.

[24] A. Klein and W. Lehner, "Representing data quality in sensor data streaming environments," *J. Data and Information Quality*, vol. 1, no. 2, 2009.

[25] S. Belongie and V. Rabaud, "Counting Crowded Moving Objects," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1. Los Alamitos, CA, USA: IEEE Computer Society, 2006, pp. 705–711.

[26] Z.-S. J. Liang, A. B. Chan, and N. Vasconcelos, "Privacy preserving crowd monitoring: Counting people without people models or tracking," in *2008 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, 2008, pp. 1–7.

[27] A. Benabbas, M. Geißelbrecht, G. M. Nikol, L. Mahr, D. Nähr, S. Steuer, G. Wiesemann, T. Müller, D. Nicklas, and T. Wieland, "Measure particulate matter by yourself: data-quality monitoring in a citizen science project," *Journal of Sensors and Sensor Systems*, vol. 8, no. 2, pp. 317–328, 2019.

[1] philipp.grandl@doubleslash.de