EAI Endorsed Transactions
on Ubiquitous Environments

Research Article **ICST.ORG**

PRICAPS: A System for Privacy-Preserving Calibration in Participatory Sensing Networks

Kevin Wiesner*, Florian Dorfmeister, Claudia Linnhoff-Popien

Ludwig-Maximilians-Universität München (LMU Munich), Mobile and Distributed Systems,
Oettingenstr. 67, 80538 Munich, Germany

Abstract

By leveraging sensors embedded in mobile devices, participatory sensing tries to create cost-effective, large-scale sensing systems. As these sensors are heterogeneous and low-cost, regular calibration is needed in order to obtain meaningful data. Due to the large scale, on-the-fly calibration utilizing stationary reference stations is preferred. As calibration can only be performed in proximity of such stations, uncalibrated measurements might be uploaded at any point in time. From the data quality perspective, it is desirable to apply backward calibration for already uploaded values as soon as the device gets calibrated. To protect the user's privacy, the server should not be able to link all user measurements. In this article, we therefore present a privacy-preserving calibration system that enables both forward and backward calibration. The latter is achieved by transferring calibration parameters to already uploaded measurements without revealing the connection between the individual measurements. We demonstrate the feasibility of our approach by means of simulation.

Keywords: participatory sensing, mobile sensing, on-the-fly calibration, user privacy

Received on 17 January 2014, accepted on 24 October 2014, published on 17 November 2014.

Copyright © 2014 S. K.Wiesner *et. al.*, licensed to ICST. This is an open access article distributed under the terms of the Creative Commons Attribution licence (http://creativecommons.org/licenses/by/3.0/), which permits unlimited use, distribution and reproduction in any medium so long as the original work is properly cited. 5

doi: 10.4108/ue.1.3.e5

1. Introduction

Today, mobile phones already include an increasing set of embedded sensors. Currently available phones come with built-in accelerometers and gyros, as well as location, audio, and image sensors. Even thermometers and hygrometers are embedded into the newest models. With this development, mobile phones evolve from standard phones, intended for personal communication only to ubiquitous sensing devices that are globally distributed.

These devices can be utilized to form a new kind of sensor network, so-called participatory sensing networks (PSN) (also referred to as *mobile phone sensing* [14], *people-centric sensing networks* [3], or *mobile crowdsensing* [8]), where people serve as carriers for mobile phone-based sensing devices. PSNs allow for large-scale, global data collection and real-time information display. In future, they could be used, e.g., to monitor environmental pollution, temperature or the noise intensity of urban areas. The main advantage of PSNs is that data can be collected on a large-scale with automatically deployed and virtually always-on, consumer-paid and continuously recharged sensor nodes.

Leveraging the sensors built into mobile phones as information source typically entails two main problems: On the one hand, those sensors are heterogeneous, due to the great number of different manufacturers and device models. On the other hand, sensors embedded in mobile phones are low-cost hardware. Consequently, calibration is necessary in order to obtain meaningful data and poses a crucial aspect for the success of PSNs. In general, there a two types of calibration: manual and on-the-fly. The former is typically performed by field experts and is used for high precision instruments, especially if manageable amounts of sensors have to be calibrated. On-the-fly calibration describes an online process, in which sensors are automatically calibrated while being deployed and running. It is done by utilizing stationary reference stations, whose measurements are used as ground-truth. For large-scale PSNs, manual calibration is too elaborate and time-consuming, and thus on-the-fly calibration is preferred.

A calibration process can only be performed if a mobile phone user comes sufficiently close to one of those reference stations. As the mobility of users cannot be controlled, this can lead to the upload of uncalibrated measurements, especially in case of long intervals without a user's encounter with a reference station. Hence, in order to improve the system's overall quality of information, it is desirable

*Corresponding author. Email: kevin.wiesner@ifi.lmu.de that the server can apply backward calibration for already uploaded values, as soon as the calibration process is carried out for a client, i.e., the server adjusts previously uploaded measurement values with the newly determined calibration parameters. In order to protect the user's privacy, though, the server should not be able to link all conducted measurements of a client, as this could reveal the user's entire mobility trace. In other scenarios, this could be achieved by using changing pseudonyms in combination with MIX networks [4] to avoid the traceability of users and their measurements. But the quasi uniqueness of the calibration parameters would allow to link calibrated measurements of a user.

In this article, we present our on-the-fly calibration system PRICAPS (Privacy-Preserving Calibration for Participatory Sensing) that allows for both forward and backward calibration in a privacy-preserving way. The latter is achieved by transferring carefully selected calibration parameters to already uploaded measurements in a way that completely blurs the connection between the individual measurements.

The remainder of this article is organized as follows. In Section 2, we describe our problem statement and motivate the need for a privacy-preserving calibration approach in PSNs. Section 3 discusses related work. In Section 4, we introduce the calibration model, followed by the description of PRICAPS in Section 5. Then, we evaluate our approach in Section 6, and finally conclude in Section 7.

2. Motivation & Problem Statement

In this section, we want to emphasize the need for a privacy-preserving calibration system for participatory sensing. We first motivate the need for backward and forward calibration. Then, we outline the arising problem.

Participatory sensing creates large-scale, low-cost sensor networks that allow for comprehensive data collection in urban or densely populated areas. These networks "enable public and professional users to gather, analyze and share local knowledge" [1] by creating participatory sensing campaigns and tasking mobile devices.

In order to allow for multi-purpose usage, data should be accurate and accessible in real-time. For instance, statistic applications need very accurate data, however, not necessarily the freshest. In contrast, live applications need up-to-date information, which in return does not have to be perfect. For instance, a routing service that calculates the most ecological route, such as *Eco Routing* [6], requires knowledge about the current situation, even if provided data is slightly inaccurate.

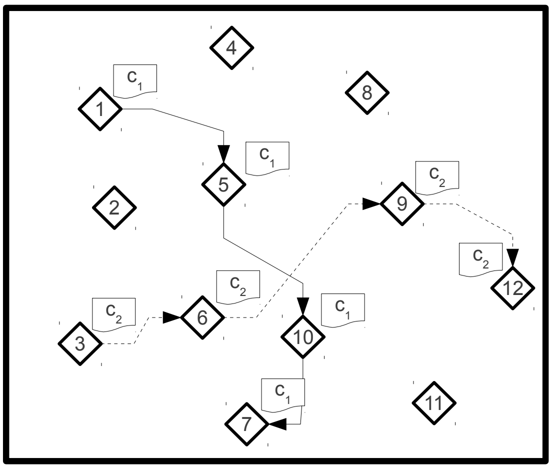


Figure 1. Applying exact backward calibration parameters (here: c_1 and c_2), can reveal the link between uploaded measurements (indicated with diamonds).

The provision of accurate and instantly accessible data through PSNs is *per se* not possible, as collected data is typically inaccurate due to the heterogeneous low-cost sensors built into mobile devices. As a consequence, calibration is needed in order to extract meaningful information out of the provided raw data. As calibration in PSNs is typically done on-the-fly with the help of ground-truth reference stations [9], an instant calibration is, in general, not possible. We therefore propose a calibration system that supports forward and backward calibration. This allows for both uploading uncalibrated data immediately and correcting uploaded values (*ex post*) if more precise data is available through a recent calibration.

However, the backward calibration poses a privacy problem: If a user has uploaded measurements that she wants to correct due to a recent calibration, she has to let the server know about the new calibration and the measurements that should be corrected.

A user u has to send the calibration parameters c_u and the set of measurement identifiers that should be adapted $(m_i, ..., m_{i+j})$. Even if split into $j+1$ separately sent tuples $< c_u, m_i >, ..., < c_u, m_{i+j} >$, the server could link all measurements to user u due the quasi uniqueness of c_u, as calibration parameters typically differ from device to device. If all measurements $m_i, ..., m_{i+j}$ can be linked to user u, the server also knows about the mobility trace of this users in this interval. This is illustrated in Figure 1. Here two users u_1 and u_2 upload their calibration parameters c_1 and c_2 respectively, which allows the server to reconstruct the mobility traces, indicated by the arrows.

Thus, a calibration system for participatory sensing needs to allow for backward calibration of already uploaded measurements in way that does not breach the users' privacy by allowing a reconstruction of mobility traces.

3. Related Work

There is a lot of research work related to participatory sensing. Most work focuses on approaches and techniques that enable data collection with mobiles phones ([1, 3, 7, 14, 21]), but neglect calibration issues. In addition, there is also a wide range of work dealing with sensor calibration in general. For instance, Bychkovskiy et al. [2] presented a post-deployment calibration technique, designed especially for dense sensor networks. In a first step, the algorithm exploits the temporal correlation of signals received at neighboring nodes to derive relative calibration relationships between each pair of neighbors. In a second step, the consistency of these calibration functions is maximized. White and Culler [20] proposed a calibration approach based on parameter estimation, which was primarily developed for sensor and actuator networks, in which both sensors and actuators require calibration. However, this kind of approaches generally cannot be applied to participatory sensing, as dense networks of static, resource-constrained or actuator nodes are assumed.

Miluzzo et al. proposed CaliBree [16], a distributed self-calibration system for mobile wireless sensor networks. Mobile sensors compare their data with those of ground-truth nodes when they experience the same environment, i.e., upon reception of locally broadcasted ground-truth information. As their nodes do not possess any positioning capabilities, they are dependent on the broadcasted information. In our approach, we assume that mobile phones are able to determine their position (e.g., using GPS), which allows for a more precise determination of whether nodes should experience the same environment. Furthermore, no direct wireless communication link between ground-truth stations and sensors is necessary, thereby facilitating the integration of already existing measurement stations and avoiding investments in new hardware. In contrast to the distributed CaliBree calibration, Honicky [11] presented an centralized approach, where the automatic calibration of sensors embedded into mobile phones is achieved by using Gaussian process regression. Through the cloud-based approach, global information about all of the sensors in the system can be integrated into the calibration process. Hasenfratz et al. [10] introduced new calibration algorithms, i.e., backward and instant calibration for on-the-fly calibration of low-cost gas sensors. The focus of the article lies on applying the algorithms on actual data and no mechanisms for the exchange of data between the entities is described.

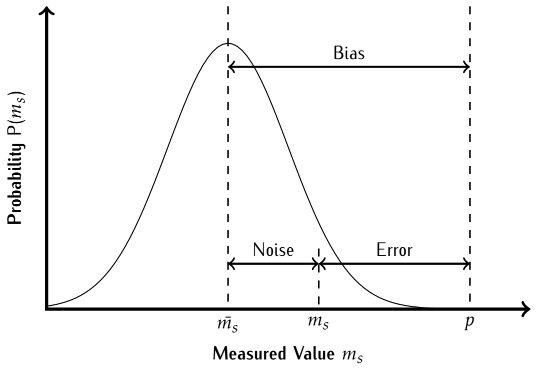


Figure 2. Systematic and random measurement errors (adapted from Bychkovskiy et al.[2])

These approaches either neglect the privacy aspect as a central instance knows about all measurements of the nodes [11] or do not take into account that nodes pass by reference stations infrequently. The latter leads to the upload of possibly uncalibrated measurements. To the best of our knowledge, our approach is the first that preserves the users' privacy and allows for backward and forward calibration.

4. Calibration Model

We assume mobile phones to be equipped with low-cost gas sensors, which we aim to calibrate with our system. In this section, we therefore introduce the underlying calibration model.

PSNs can be seen as a special type of sensor network. Sensor networks usually aim to monitor one or multiple phenomena of interest. In order to be able to detect a phenomenon P, there needs to be a measurable signal $p : T \to D$ that arises from P, with $T \subseteq \mathbb{R}^+$ being the time and $D \subseteq \mathbb{R}$ being the value domain. Let $m_s(t_i)$ be the measurement of a sensor s at time $t_i \in T$, and $p(t_i)$ the actual value of the phenomenon at that time. If sensor s is a *perfect sensor*, $m_s(t_i) = p(t_i)$ is true for any point in time and no calibration is necessary.

However, sensors are typically not behaving perfectly, and especially for low-cost gas sensors there is a significant precision loss due to sensor aging [12] and influencing contextual settings (e.g., humidity) [13]. Typically two types of measurement errors occur (see Figure 2): the *Bias* describes an offset in the mean amplitude of the readings $\bar{m}_s$ from the true value p, whereas the *Noise* describes the random component in the error. The aim of calibration is to remove the systematic *Bias*, whereas the *Noise* can typically compensated by repeated measurements. Calibration of sensors can hence be described as the process of minimizing the deviation of the measured values $m_s(t_i)$ from the actual values $p(t_i)$, which is achieved by applying a *calibration curve* ϕ to the measured values. We use a polynomial of order k as a representation of ϕ :

$\mathbb{R}^{k+1} \times D \rightarrow D$ with a vector of *calibration parameters* $c = (c_0, c_1, ..., c_k) \in \mathbb{R}^{k+1}$ and x as the measurement input:

$$\phi(c,x) = \sum_{n=0}^{k} c_n * x^n. \tag{1}$$

As a sensor can be calibrated several times, we denote $\omega : T \rightarrow \mathbb{R}^{k+1}$ as the function returning the effective calibration parameters at a certain point of time. As a result, the calibrated value $\tilde{m}_s(t_i)$ of a sensor s at time t_i is

$$\tilde{m}_s(t_i) = \phi(\omega(t_i), m_s(t_i)) = \sum_{n=0}^{k} \omega(t_i)_n * m_s(t_i)^n. \tag{2}$$

For a *perfect sensor* s that needs no calibration, it is $\forall t_i \in T : \omega(t_i) = (0, 1, 0, 0, ..., 0) \in \mathbb{R}^{k+1}$ and $m_s(t_i) = p(t_i)$. By means of calibration we aim for *perfectly calibrated* sensors that behave like *perfect sensors* from a point t_c in time onwards, so that $\forall t_i \geq t_c, t \in T : \omega(t_{i+1}) = \omega(t_i)$ and $\tilde{m}_s(t_i) = p(t_i)$. This ideal state is typically not reached, as sensors continuously degrade and thus do not remain perfectly calibrated. However, by continuously repeating the calibration process an approximation of the ideal state can be reached.

In order to determine the above introduced *calibration curve* ϕ, a set C (with $|C| \geq (k+1)$) of calibration tuples $< m_s(t_i), p(t_i) >$ is needed, i.e., for a certain number of measurements we need to know the actual value of the phenomenon of interest. For this purpose, we utilize stationary reference stations, as we assume those sensors to be perfectly calibrated at any point. For each measurement $m_s(t_i)$ and actual value $p(t_i)$, we store the time t_i and the location l_i of the mobile phone, respectively of the reference station, so that we have a set of measurements M, consisting of tuples of the form $< t, m_s(t), l(t) >$, and a set of actual values S, consisting of tuples of the form $< t, p(t), l(t) >$. To access the different parts of these tuples, we use the dot notation, e.g., $m.l$ for the location of a tuple $m \in M$. Hence, the set of calibration tuples C can be written as

$$\begin{aligned} s \in S, m \in M : C = \{(s.p, m.m_s) | &|s.t - m.t| \leq \delta_t \\ &\wedge |s.l - m.l| \leq \delta_l\}, \end{aligned} \tag{3}$$

with δ_t and δ_l being parameters describing the temporal and spatial distance between ground-truth and mobile measurements, which have to be adapted according to the phenomenon of interest.

5. PRICAPS: Privacy-Preserving Calibration for Participatory Sensing

In this section, we will describe our system for Privacy-Preserving Calibration for Participatory Sensing (PRICAPS). As proposed by Christin et al., we use the term Participatory Sensing "to designate applications using mobile phones as sensors (or as data sink for interfaced sensors) where participants voluntarily contribute sensor data for their own benefit and/or the benefit of the community" [5]. The process of data collection and upload is described in Section 5.1. Calibration refers to the process of minimizing the deviation of measurement values from actual values by determining a calibration curve (cf. Section 4). PRICAPS is an on-the-fly calibration system, i.e., it calibrates sensors while they are in use by utilizing stationary reference stations providing ground-truth data. Many cities already deployed stationary sensor stations measuring the air quality in use. For instance, Zurich has four stations[1], and in Munich there are even 10 stations deployed[2]. We assume such reference stations to be available and that their measurements are accessible through well-defined web service interfaces.

Figure 3 illustrates the calibration pipeline of our system. By comparing reference measurements to the user's measurement data, instant *forward calibration* can be performed. Forward calibration refers to the process of determining a calibration curve on a user's mobile device that is applied to future measurements before uploading those. In contrast, *backward calibration* refers to the process of adjusting previous measurements by applying a newly determined calibration curve to already uploaded data. In the following, we shortly describe the measuring and upload process, before the two calibration phases are described in more detail.

5.1. Measurements & Data Upload

In order to obtain data that can be calibrated, measurements have to be taken first. We assume that users conduct measurements using their mobile phones and upload their data to a server, which is responsible for storing all measurements. The upload is done via MIX networks with users utilizing self-generated pseudonyms for communicating their measurements and change those on a regular basis. Users can even use a new pseudonym for each measurement. These pseudonyms, in the following also denoted as ps_{id}, are necessary in order to be able to reference specific measurements within the backward calibration process. In addition, the location of the measurement is transmitted, resulting in upload tuples of the form $< ps_{id}, m_s, l >$.

To avoid timing-based attacks, these tuples do not include the (local) time at which the measurement was taken. Instead, time is divided into intervals t_{int} that match the required measurement frequency, e.g., $t_{int} = 15min$ if the measurement frequency is 4x per hour, and measurements are uploaded at a random point of time within these intervals. The server records the arrival time t_{arr} of the incoming measurements and stores the combined tuples $< t_{arr}, ps_{id}, m_s, l >$ in its database.

[1] http://www.ostluft.ch/

[2] http://maps.muenchen.de/rgu/luftmessstationen

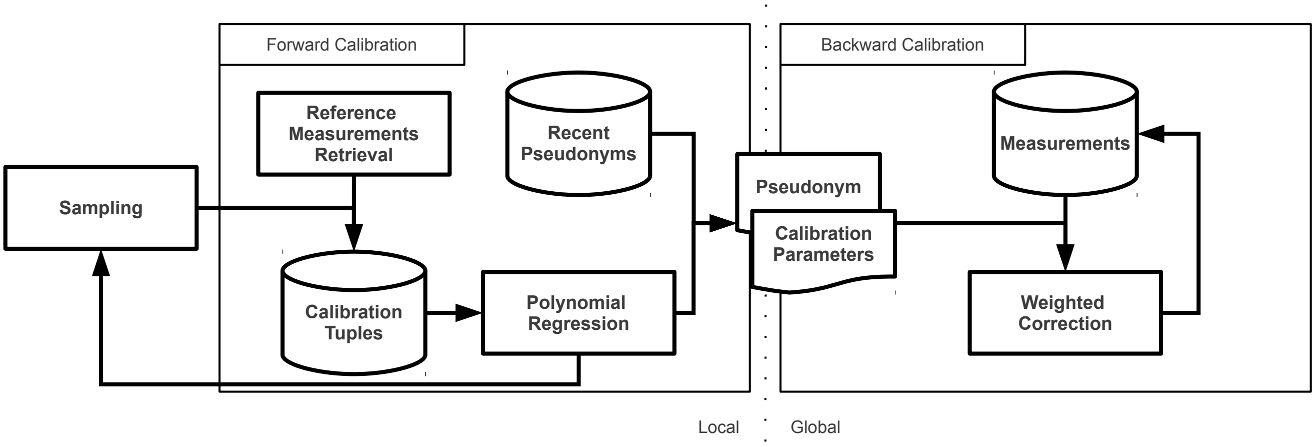


Figure 3. Calibration pipeline showing the two calibration phases

5.2. Forward Calibration

In the forward calibration process, a calibration curve is determined based on the comparison of recent measurements of both the mobile phone and a reference station. First, the user's device (hereafter referred to as the *client*) needs to be aware of any reference stations within its area. Therefore, the server provides a list of reference stations together with their locations and the accessible data interface for the reference measurement retrieval. This list is requested as soon as the client enters an unknown area, and is refreshed by periodical updates.

Knowing the locations of nearby reference stations, the client checks for each measurement, whether it is in proximity of one of those. If so, the reference measurements are retrieved. As mentioned in the previous section, the temporal and spatial ranges stating what is to be considered as "proximity" depend on the phenomena of interest and have to be specified by adapting the parameters δ_l and δ_t in Equation 3.

If reference measurements are only downloaded when users are in proximity of a station, the station's operator might draw conclusions about the number of users that performed a calibration within a calibration period, especially in scenarios with only a few users that are calibrating. To avoid this, in each calibration period a certain percentage Γ of users perform fake data request, i.e., they request data from reference stations without being close to such a station. Each user u draws a random number γ_u from $[0.0; 1.0]$ and if $\gamma_u < \Gamma$ a fake data request is performed at a random time within the calibration period. Since no matching user-collected measurement exist, retrieved responses to fake data requests are simply discarded by the users.

For real reference retrievals, the locally recorded measurements and the reference measurements are then combined and the calibration tuples are formed through a temporal and spatial filtering process (cf. Section 4). Basically, this step combines measurements that were taken at approximately the same time and location. These calibration tuples are then used to determine a calibration curve that is specific to the current state of a mobile user's sensing equipment. In order to avoid distorted or premature calibrations, PRICAPS takes the following countermeasures: First, forward calibration is only performed if a predefined minimal number of calibration tuples ($C_{MinCount}$) exist in order to reduce the impact of possible outliers within the calibration tuples. Second, calibration is only started if a certain value range within the calibration tuples is covered ($C_{MinRange}$), to avoid a calibration optimized for a limited value range. Third, in order to avoid unnecessary calibrations, the calibration process is only started if a certain timeout has been exceeded since the last calibration ($C_{Timeout}$). The actual determination of the calibration curve parameters is done by polynomial regression. The model is fitted using the method of least squares, which minimizes the sum of the squares of the deviations between reference and mobile sensor measurements. The determined calibration tuples are then used to correct future measurements before uploading them (see Figure 4). In a discretized form, they are also used during backward calibration to correct already uploaded measurements.

5.3. Backward Calibration

In the backward calibration process, already uploaded measurements should be adjusted with a newly determined calibration curve. As already mentioned, users change their pseudonyms on a regular basis in order to protect their privacy. As a result, only the users themselves know which pseudonyms the calibration curve should be applied to. Thus, a client that has locally determined a new calibration curve has to inform the server about the pseudonyms and

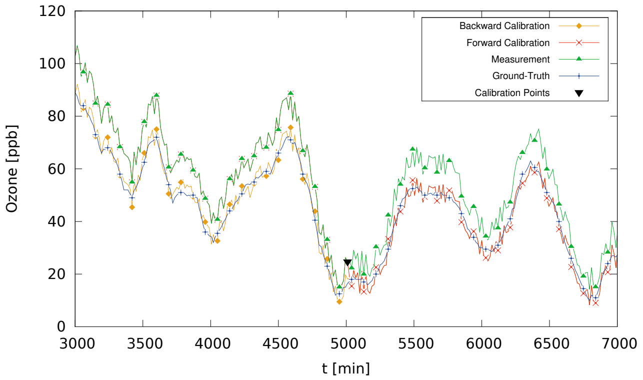


Figure 4. Example excerpt of simulated and calibrated measurements of a node over time.

the calibration parameters. A naive approach would be to send tuples consisting of the pseudonym to be adjusted and the calibration vector c. However, this would naturally lead to a breach of the user's privacy: as the exact calibration parameter vector typically differs from phone to phone, sending c could reveal the link between the different pseudonyms of a user (see Figure 1).

In PRICAPS, this is counteracted by incorporating the concept of *k-anonymity* [19]. To obfuscate the exact calibration parameter, the client discretizes the calibration parameters before uploading them to the server. By this, the probability of having the same calibration vector c as other clients and achieving k-anonymity is increased. For this process, a discretization function $\psi : \mathbb{R}^{k+1} \times \mathbb{R}^{k+1} \longrightarrow \mathbb{R}^{k+1}$ is used, which returns a discretized (and thereby generalized) calibration vector $\tilde{c}$:

$$\tilde{c} = \psi(c, d) = \begin{pmatrix} \lceil \frac{c_0}{d_0 * \theta(c)} \rfloor * (d_0 * \theta(c)) \\ \vdots \\ \lceil \frac{c_k}{d_k * \theta(c)} \rfloor * (d_k * \theta(c)) \end{pmatrix}, \tag{4}$$

where $d \in \mathbb{R}^{k+1}$ is the discretization vector that is known system-wide (i.e., all clients use the same d) and $\lceil x \rfloor$ denotes the rounding function to the nearest integer. θ describes a factor for adjusting the discretization granularity to the extent of the deviation δ of c from the perfect sensor s: $\theta(c) = 2^{max(\lceil \lg \delta(c) - \varphi \rceil, 0)}$, with δ being the degree of deviation $\delta(c) = \|c - s\|_2 = (\sum_{n=0}^{k} (\frac{c_n - s_n}{d_n})^2)^{\frac{1}{2}}$, and φ being a constant for determining the steps of adjustment. To clarify this step, we illustrate the discretization with an example: We assume a calibration vector $c = (9.3292, 0.8567)$ and a discretization vector $d = (2.0, 0.1)$ with $\varphi = 2$. This leads to $\delta(c) = 4.8798$ and $\theta(c) = 2$, and finally to the discretized calibration vector $\tilde{c} = (8.0, 0.8)$.

Naturally, as c is distorted, the discretization process leads to a loss of precision, with the amount of distortion depending on d. However, the error introduced should be relatively small compared to the gain of precision achieved by calibrating and adjusting $m_s(t_i)$ to $\tilde{m}_s(t_i)$, even with deliberately distorting the calibration parameters. Furthermore, as $\tilde{c}$ is only used within the backward calibration process, the error does not propagate to future measurements.

To avoid privacy attacks based on the upload time, backward calibration parameters are only uploaded at certain specified times, resulting in so-called "calibration bursts". By this, all users that want to apply backward calibration to their measurements, upload their parameters for the total interval since the last calibration burst. As done before, the upload of $\tilde{c}$ to the server is carried out via a MIX network, so that the updates cannot be linked to the physical device.

The last step is the weighted correction of former measurements by the server. This is done by applying the received calibration parameters and calculating a new measurement value. Ideally, this new value and the former value should be combined to a corrected measurement value by using weights that depend on the point of time within the last calibration period of the corresponding node. Measurements closer to the calibration point at which the backward calibration parameters have been determined should be stronger affected by the correction than measurements closer to the previous calibration point. The idea behind this is that it is typically not reasonable to alter measurements that have just been (forward) calibrated by applying a much later determined backward calibration. However, as the server does not know the actual calibration times of a node, only an approximation can be calculated. Instead of using the actual calibration times, the server uses weights that depend on the point of time within the calibration burst. The corrected value $\tilde{m}_s(t_i)$ is calculated with the following formula

$$\begin{aligned} \tilde{m}_s(t_i) = & \frac{(t_i - cb_{n-1}) * \phi(\omega(cb_n), m_s(t_i))}{cb_n - cb_{n-1}} \\ & + \frac{(cb_n - t_i) * \phi(\omega(cb_{n-1}), m_s(t_i))}{cb_n - cb_{n-1}}, \end{aligned} \tag{5}$$

where cb_n and cb_{n-1} denote the times of the current calibration burst and the previous calibration burst respectively. As this might heavily deviate from the ideal weighted correction, the client calculates the ideal weighted correction $\hat{m}_s(t_i)$ itself before uploading the backward calibration parameters

$$\begin{aligned} \hat{m}_s(t_i) = & \frac{(t_i - ct_{n-1}) * \phi(\omega(ct_n), m_s(t_i))}{ct_n - ct_{n-1}} \\ & + \frac{(ct_n - t_i) * \phi(\omega(ct_{n-1}), m_s(t_i))}{ct_n - ct_{n-1}}, \end{aligned} \tag{6}$$

with ct_n and ct_{n-1} denoting the actual calibration times of that node. Only if a backward calibrated value is closer to the ideally corrected value, i.e., if $|\hat{m}_s(t_i) - m_s(t_i)| > |\hat{m}_s(t_i) - \tilde{m}_s(t_i)|$, the client uploads the calibration parameters and initiates the backward calibration process.

6. Evaluation

We evaluated our concept by means of simulation. As ground truth data for our simulated measurements, we used real ozone measurements of 14 days collected at stationary stations in Munich (cf. Footnote 2, p. 4). We interpolated this data in the time domain to increase the resolution from 1/hour to 1/minute, as well as in the spatial domain, in order to have a ground truth value for each position within the simulation area. For the latter, we employed Shepard's method for Inverse Distance Weighting [18] with the power parameter $p = 2$.

To simulate the deviation of mobile sensors, we used the model for ozone measurements presented in [10]: the authors deployed sensors with *MiCS-OZ-47* ozone sensing heads, and found that the measurement errors are normally distributed, if they are only initially calibrated. They observed a normal distribution $\mathcal{N}(\mu, \sigma^2)$ with $\mu \sim \mathcal{U}(-9, 9)$ ppb and $\sigma \sim \mathcal{N}(3, 1)$ ppb over the period of a day. For our simulations, we applied this model to generate artificial data, i.e., based on this model we determined an error curve for each sensor node. The error curve was set to an order of 1, i.e., a polynomial of the form $a * x + b$, where a was set to a random value ranging from $[-8.0, 8.0]$ and b to a value ranging from $[-0.2, 0.2]$, as those values closely modeled the mentioned behavior. We also integrated an aging factor of 0.2 ppm/day (as in [10]) to account for the loss of precision over time. As a result, a measurement was simulated by applying the error curve on the ground truth value, adding the deviation arising from sensor aging, and finally adding some noise from the aforementioned distribution.

We then conducted simulations with the setup stated in Table 1. The backward calibration was performed once per week. The calibration curve ϕ was set to an order of 1, thus c, $\tilde{c}$, and $d \in \mathbb{R}^2$. In our evaluations we used the following discretization parameters: $d_0 = \{1.0, 1.5, 2.0\}$, $d_1 = \{0.05, 0.1, 0.15, 0.2\}$, and $\varphi = \{2, 3, 4\}$, resulting in 36 different discretization combinations. In the following, discretization parameter combinations are written in the form $d_0, d_1; \varphi$.

6.1. K-Anonymity

In a first step, we analyzed our approach regarding the level of k-Anonymity. We therefore run simulations with each of the above mentioned discretization combination and analyzed how often k-Anonymity was reached for $k = \{2, 3, ..., 10\}$.

Figures 5a-c show the achieved k-Anonymity for 1000 nodes. It is obvious that more fine-grained discretization vectors, i.e., vectors with small discretization steps (such as $1.0, 0.05; 4.0$) perform worse than more coarse-grained vectors (such as $2.0, 0.2; 2.0$). It can be seen that especially the discretization parameter d_1 is decisive, and that discretizations with $d_1 = 0.15$ or $d_1 = 0.2$ reached the desired k-Anonymity level significantly more often. The results also show that smaller values for φ have a more positive impact on the anonymity level than larger values, as the discretization parameters are adapted more rapidly and thus become more coarse-grained. For $k = 5$, the k-Anonymity level was reached in more than 80% of the time with 28 out of the 36 discretization combinations. For $k = 10$, 23 discretization combinations reached the specified level in more than 60% of the time. We then selected the worst and the best performing discretization from the former results and simulated it with varying node numbers, i.e., $\#nodes = \{1000, 1500, 2000\}$. The results are shown in Figure 5d. It can be seen that especially in the worst case, the increase of participating nodes significantly increases the percentage of achieved k-Anonymity.

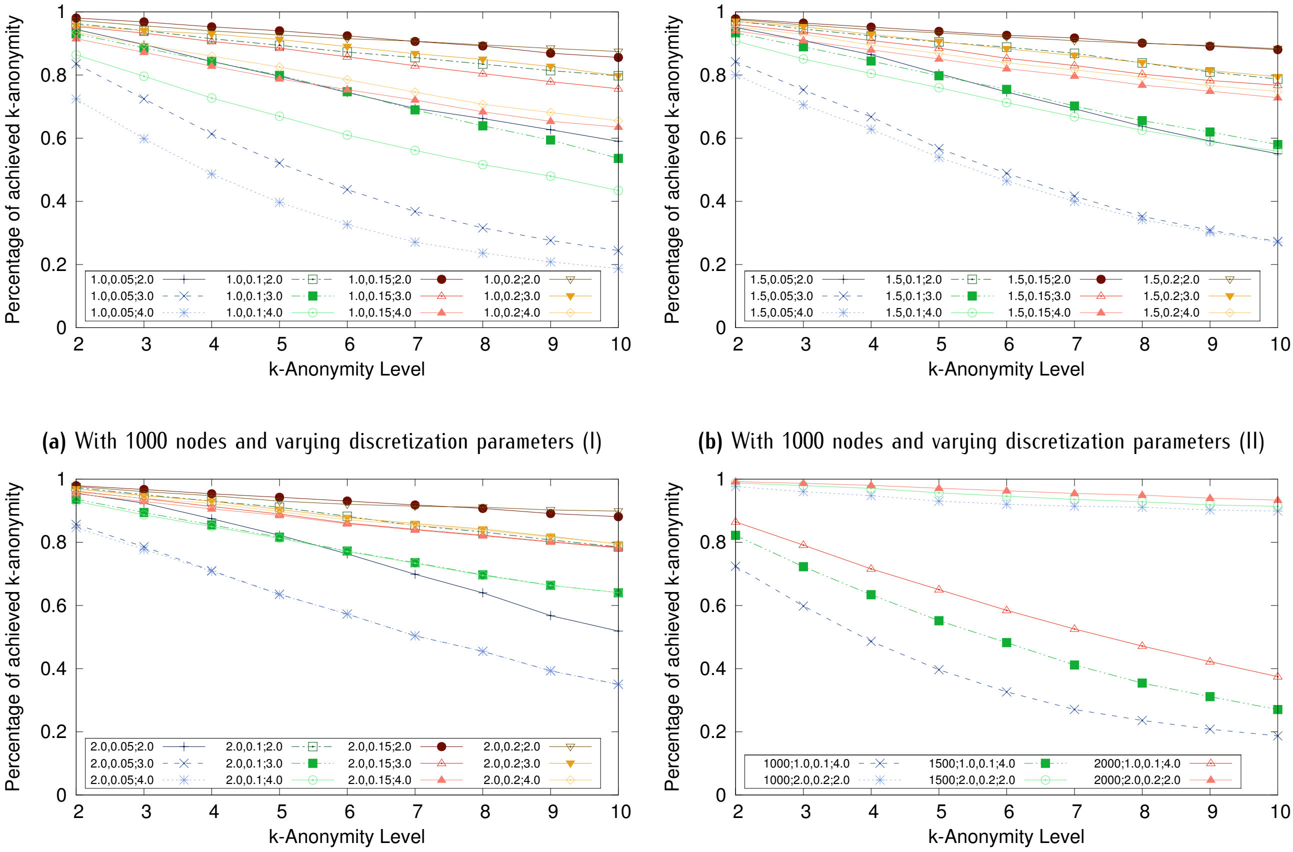
6.2. Discretization Error

In a next step, we analyzed the error introduced by discretizing the calibration parameters in the backward calibration process. In this step, we only considered discretization parameters that achieved a k-Anonymity level of 10 at least 60% of the time. Figures 6a,b show the average discretization error in relation to the average calibration gain (the average was calculated only over the amount of nodes that performed a calibration). For the former, we compared the results using the discretized calibration vector $\tilde{c}$ with those using the exact calibration parameters c (in relation to the ground truth value). The calibration gain is the average gain in precision when applying the discretized calibration curve $\tilde{c}$, compared to results without calibration. Here, the results are obviously orthogonal to the aforementioned results: the most fine-grained discretization results in the lowest error and the highest gain. It can be seen again that especially the choice of d_1 and φ are decisive for the result. Even though a few exceptions resulted in a negative backward calibration gain, i.e., the discretization of the calibration lead to a worse result than without the calibration, with most parameters a positive result was achieved.

We further examined the calibration gain for each calibration period, which is the time interval between two calibration points, e.g., the first calibration period (C_1) is the time interval from the simulation start until the first calibration. More precisely, we define the set of calibration periods as follows: $\{\ i \in 1, ..., n+1 : C_i =$

Table 1. Simulation setup

No. of Nodes	1000, 1500, 2000	Simulation time	14 days
Mobility Model	Random Walk	Max. speed	8.33 $\frac{m}{s}$
Measurement frequency	4x per hour	No. of reference stations	5
δ_l	250 m	$C_{MinCount}$	5
$C_{MinRange}$	30 ppm	$C_{Timeout}$	5 days



(a) With 1000 nodes and varying discretization parameters (I)

(b) With 1000 nodes and varying discretization parameters (II)

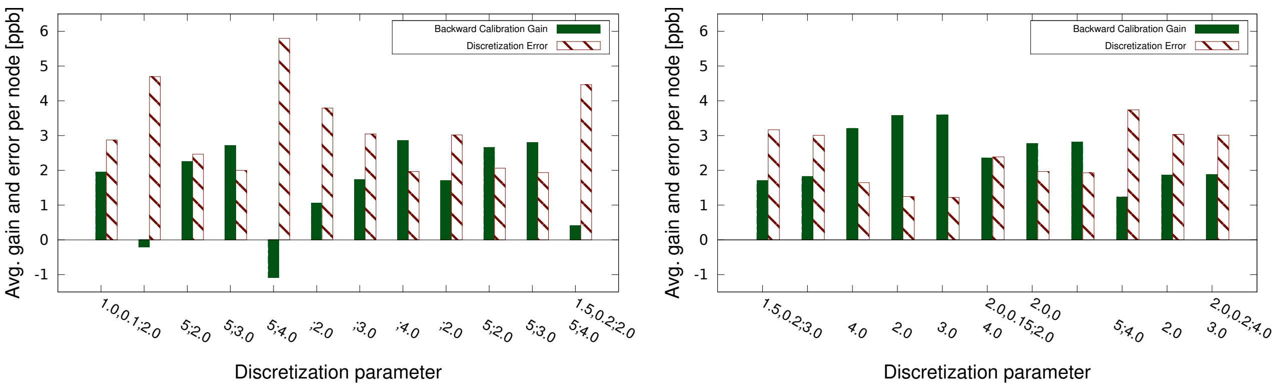
(c) With 1000 nodes and varying discretization parameters (III)

(d) With varying numbers of nodes and worst/best parameters

Figure 5. Achieved k-Anonymity level for discretization parameters

$[t_{c_{i-1}}; t_{c_i}]\}$, with $\{t_{c_1}, t_{c_2}, ...t_{c_n}\}$ being the set of calibration times. As we set $C_{Timeout} = 5$ for our simulations, a maximum of three calibration points was possible and consequently a maximum of four calibration periods (C_1 to C_4).

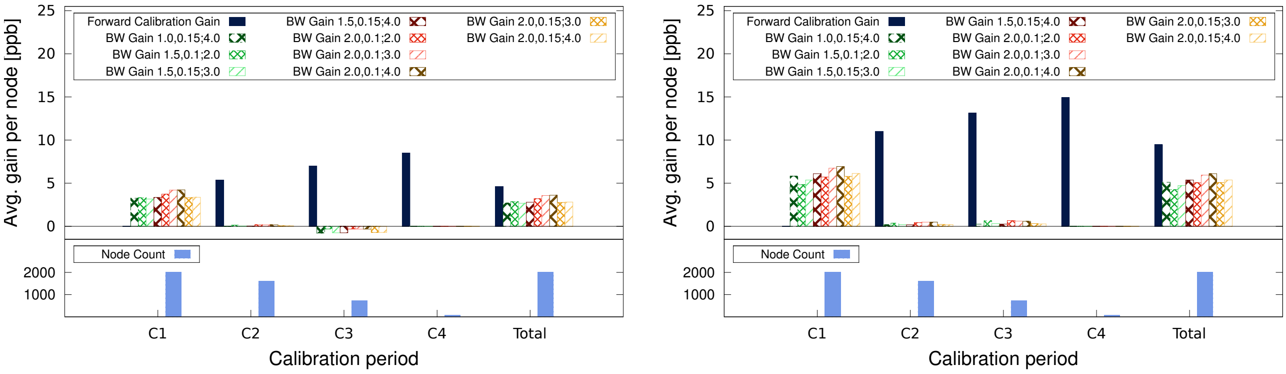
The upper parts of Figures 7a,b show the average calibration gain for the individual calibration periods and the overall gain, whereas the lower parts show the number of nodes that were calibrated in the individual round. In each figure, the forward calibration gain was only plotted once, since forward calibration does not depend on discretization parameters. We illustrated the results for 2000 nodes and chose those discretization parameters, whose backward calibration gain was higher than the discretization error (see Figure 6a,b). In Figure 7a, the results with the aforementioned aging factor of 0.2 ppb/day are illustrated. In period C_1 no forward calibration gain is achieved, as forward calibration adapts only future measurements, i.e., from t_{c_1} onwards. But for the following rounds, an increasing forward calibration gain can be observed, however, with a strongly decreasing number of nodes. The backward calibration has the highest impact in C_1, as uploaded values in this period are completely uncalibrated. In the following rounds, the backward calibration is comparatively small and in the third



(a)

(b)

Figure 6. Average backward calibration gain and discretization error for varying discretization parameters over the simulated 14-day period.

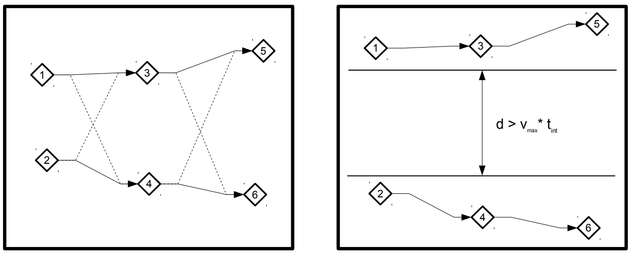


(a) With an aging factor of 0.2 ppb/day

(b) With an aging factor of 1.2 ppb/day

Figure 7. Comparison of forward and backward calibration gain per calibration period with varying aging factors.

round even negative. This stems from the relatively short time interval between the calibration points. In C_3, the sensors have already been calibrated twice and the aging factor does not distort the measurements strongly enough within this calibration interval, so that the discretized backward calibration is not reasonable in this case. In Figure 7b, we increased the aging factor to 1.2 ppb/day. This simulates a stronger aging of the sensors, but can also be interpreted as longer periods between the calibration points with a constant aging factor (i.e., 6 times longer calibration intervals with an aging factor of 0.2 ppb/day). It can be seen that both the forward and the backward calibration gain increased; the latter now results in a positive gain in each round. As could be expected, this shows that backward calibration is reasonable if the calibration interval is long enough for the sensors to significantly deviate from their former calibration.



(a) Traces with different probabilities

(b) Measurements that are too far apart to belong to a trace

Figure 8. Example scenarios that illustrate possible limitations.

6.3. Discussion & Challenges

Discussion. In this section, we want to discuss the limitations of our approach. As the presented results above showed, PRICAPS cannot guarantee a certain level of anonymity but can rather be seen as a "best effort" approach depending on the number of participating users and their mobility.

Further, our approach does not really incorporate means for coping with different probabilities of certain traces. A highly simplified example is shown in Figure 8a. The solid traces might be more likely than the dotted traces, where both users would take a detour. However, in PRICAPS we assume the sampling rate to be very low, so that each measurement could have been conducted by a large portion of the users and, as a result, there are plenty of possible trace combination, so that a reliable reconstruction of the trace should not be possible.

Another aspect that might weaken the privacy level is the possibility that measurements are too far apart so that it is obvious that they do not belong to the same user. In Figure 8b, a possible scenario is shown for two users. In this case, it seems as there are two users and their traces could be reconstructed. However, the server does not know how many users are currently participating. It could also be the case that this are four different users, so again a reliable trace reconstruction is not possible.

Notice that in all our results, we stated the *worst-case* k-anonymity level, i.e., we calculated the k-anonymity level as if it was known how many users are calibrating. If there are n users with the same calibration vector c and each users adapts m measurements, there are in total $n * m$ updated measurements. In our results, we stated this as k-anonymity level of n. In fact, the server is not aware of the actual amount of users and from the server perspective the updates could originate from a group ranging from 1 to $n * m$ users. As a result, the privacy level should be even higher than our results indicate.

To further improve the results, PRICAPS could be extended by *gamification* features, i.e., users could be incentivised to adapt their mobility. As proposed in [15], users could be rewarded, if they adapt their route in a specified way. This could be used to prompt participants to visit reference stations more often, which would lead to better results regarding data quality and user privacy.

Challenges. A major challenge of realizing PRICAPS is the necessity of appropriate reference stations. This entails that a sufficient amount of stations is required and that those stations have to be reasonably located within the investigation area, so that users pass these sites frequently. Further, as mentioned in 5, we assume reference measurements to be accessible through well-defined web service interfaces. As a consequence, existing stations have to be upgraded or new stations have to be deployed in order to fulfill this requirements. However, building up this infrastructure is very costly and probably takes time.

Another challenge not tackled yet is the consideration of a phone's context when initiating a calibration process. If a mobile phone is in a pocket or bag when approaching a reference station, it is obvious that its measurements deviate from those collected by the station. As a result, calibration tuples should only be recorded if reference station and mobile phone experience the same context. Therefore, a recognition system for the phone's context as in [17] should be incorporated.

7. Conclusion & Future Work

We presented PRICAPS, a system for privacy-preserving calibration system in participatory sensing networks that enables forward as well as backward calibration, while simultaneously protecting the users' privacy. We proposed a pseudonym-based system that allows for transferring calibration parameters to other pseudonyms without revealing the connection between those. Our analysis shows that we can achieve a high degree of anonymity, but only at the price of sacrificing precision. More precisely, the anonymity level and the backward calibration gain are negatively correlated, i.e., an increase of the one leads to a decrease of the other. Our results show that there are several discretization parameters that lead to promising results for both, however, the "optimal" setting depends on the application scenario and the subsequent weighting of anonymity in relation to precision. As the loss of precision is small in relation to the overall gain, we believe that PRICAPS represents a valid concept for privacy-preserving calibration in PSNs.

In future work, we want to evaluate our concept with more extensive simulations using a realistic urban simulation environment and implement a prototype to evaluate the concept in real-life settings.

References

[1] Burke, J., Estrin, D., Hansen, M., Parker, A., Ramanathan, N., Reddy, S. and Srivastava, M.B. (2006) Participatory Sensing. In *Proceedings of WSW'06 at SenSys '06*.

[2] Bychkovskiy, V., Megerian, S., Estrin, D. and Potkonjak, M. (2003) A collaborative approach to in-place sensor calibration. In *Information Processing in Sensor Networks* (Springer Berlin / Heidelberg), *LNCS* **2634**, 556–556.

[3] Campbell, A., Eisenman, S., Lane, N., Miluzzo, E., Peterson, R., Lu, H., Zheng, X. *et al.* (2008) The Rise of People-Centric Sensing. *Internet Computing, IEEE* **12**(4): 12–21. doi:10.1109/MIC.2008.90.

[4] Chaum, D. (1981) Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM* **24**(2): 84–90.

[5] Christin, D., Reinhardt, A., Kanhere, S.S. and Hollick, M. (2011) A survey on privacy in mobile participatory sensing applications. *Journal of Systems and Software* **84**(11): 1928 – 1946. doi:10.1016/j.jss.2011.06.073, URL http://www.sciencedirect.com/science/article/pii/S0164121211001701.

[6] Duchon, M., Wiesner, K., Müller, A. and Linnhoff-Popien, C. (2012) Collaborative sensing platform for eco routing and environmental monitoring. In Martins, F., Lopes, L. and Paulino, H. [eds.] *Sensor Systems and Software* (Springer Berlin Heidelberg), *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering* **102**, 89–104. doi:10.1007/978-3-642-32778-0_8, URL http://dx.doi.org/10.1007/978-3-642-32778-0_8.

[7] Eisenman, S.B., Lane, N.D., Miluzzo, E., Peterson, R.A., seop Ahn, G. and Campbell, A.T. (2006) Metrosense project: People-centric sensing at scale. In *Proceedings of WSW at Sensys'06*.

[8] Ganti, R., Ye, F. and Lei, H. (2011) Mobile crowdsensing: current state and future challenges. *IEEE Communications* **49/11**: 32–39. doi:10.1109/MCOM.2011.6069707.

[9] Hasenfratz, D., Saukh, O., Sturzenegger, S. and Thiele, L. (2012) Participatory Air Pollution Monitoring Using Smartphones. In *Proceedings of the 2nd Workshop on Mobile Sensing: From Smartphones and Wearables to Big Data*.

[10] Hasenfratz, D., Saukh, O. and Thiele, L. (2012) On-the-Fly Calibration of Low-Cost Gas Sensors. In *Wireless Sensor Networks* (Springer Berlin / Heidelberg), *LNCS* **7158**, 228–244.

[11] Honicky, R.E. (2007) *Automatic calibration of sensor-phones using gaussian processes*. Tech. Rep. UCB/EECS-2007-34, EECS, UC Berkeley. URL http://www.eecs.berkeley.edu/Pubs/TechRpts/2007/EECS-2007-34.pdf.

[12] Huan, C., Zhiyu, L. and Gang, F. (2011) Analysis of the aging characteristics of SnO2 gas sensors. *Sensors and Actuators B: Chemical* **156**(2): 912 – 917. doi:10.1016/j.snb.2011.03.005, URL http://www.sciencedirect.com/science/article/pii/S0925400511001870.

[13] Kamionka, M., Breuil, P. and Pijolat, C. (2006) Calibration of a multivariate gas sensing device for atmospheric pollution measurement. *Sensors and Actuators B: Chemical* **118**(1-2): 323 – 327. doi:10.1016/j.snb.2006.04.058, URL http://www.sciencedirect.com/science/article/pii/S0925400506003108.

[14] Lane, N., Miluzzo, E., Lu, H., Peebles, D., Choudhury, T. and Campbell, A. (2010) A survey of mobile phone sensing. *IEEE Communications* **48/9**: 140–150.

[15] McCall, R., Kracheel, M. and Koenig, V. (2012) Reducing Congestion through Persuasive Gaming. In *Proceedings of "The Car as an Arena for Gaming" Workshop at MobileHCI 2012* (San Francisco, CA,).

[16] Miluzzo, E., Lane, N., Campbell, A. and Olfati-Saber, R. (2008) Calibree: A self-calibration system for mobile sensor networks. In *Distributed Computing in Sensor Systems* (Springer Berlin / Heidelberg), *LNCS* **5067**, 314–331.

[17] Miluzzo, E., Papandrea, M., Lane, N.D., Lu, H. and Campbell, A.T. (2010) Pocket, Bag, Hand, etc.-Automatically Detecting Phone Context through Discovery. In *Proceedings of the International Workshop on Sensing for App Phones (PhoneSense 2010)* (Zurich, Siwtzerland): 21–25.

[18] Shepard, D. (1968) A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the ACM '68* (New York, NY, USA: ACM): 517–524. doi:10.1145/800186.810616, URL http://doi.acm.org/10.1145/800186.810616.

[19] Sweeney, L. (2002) k-anonymity: A model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* **10**(5): 557–570. doi:10.1142/S0218488502001648, URL http://dx.doi.org/10.1142/S0218488502001648.

[20] Whitehouse, K. and Culler, D. (2002) Calibration As Parameter Estimation in Sensor Networks. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA '02)* (New York, NY, USA: ACM): 59–67. doi:10.1145/570738.570747, URL http://doi.acm.org/10.1145/570738.570747.

[21] Wiesner, K., Duchon, M. and Dürr, M. (2012) Distributed Multi-Head Clustering for People-Centric Networks. In *Proceedings of the 6th International Conference on Sensor Technologies and Applications (SENSORCOMM 2012)* (Rome, Italy): 53–58. URL http://www.thinkmind.org/index.php?view=article&articleid=sensorcomm_2012_3_10_10142.