Opportunistic Multiparty Calibration for Robust Participatory Sensing

Francoise Sailhan, Valérie Issarny, Otto Tavares Nascimento

Opportunistic Multiparty Calibration for Robust Participatory Sensing

Françoise Sailhan*†, Valérie Issarny*, Otto Tavares Nascimento*
*MiMove Team, Inria Paris, France
firstname.lastname@inria.fr
†Cedric Laboratory, CNAM Paris, France
Email: firstname.lastname@cnam.fr

***Abstract*—While bringing massive-scale sensing at low cost, mobile participatory sensing is challenged by the low accuracy of the sensors embedded in and/or connected to the smartphones. The mobile measurements that are collected need to be corrected so as to accurately match the phenomena being observed. This paper addresses this challenge by introducing a *multi-hop, multiparty calibration* method that operates in the background in an automated way. Using our method, sensors that are within a relevant sensing (and communication) range coordinate so that the observations of the participating (previously) calibrated sensors serve calibrating the other participants. As a result, our method is particularly well suited for participatory sensing within crowd meetings, as as for instance within public spaces. Our solution leverages multivariate linear regression, together with robust regression so as to discard the measurements that are of too low quality for being meaningful. To the best of our knowledge, we are the first to introduce a multiparty calibration algorithm, while previous work in the area focused on pairwise calibration. The paper further introduces a supporting prototype implemented over Android, and related experiment in the context of noise sensing. We show that the proposed multiparty calibration system enhances the accuracy of the mobile noise sensing application.**

I. Introduction

Mobile participatory sensing applications are increasingly becoming popular (e.g., see [14], [13], [12] for a survey). Among benefits, the applications allow collecting in situ measurements and thereby offering personalized environmental reports together with aggregating knowledge across space and time (e.g., [7], [8]). However, the low cost and miniaturization of sensors embedded in (or connected to) today's smartphones come with noises and drifts that often trigger inaccuracies, and thus lead to communicate incorrect knowledge (e.g., see [10]). A well-known approach to reduce such errors consists in calibrating sensors. Although manual calibration is a well-proven approach for expensive stationary sensors that are either regularly calibrated by experts or inherently resistant to errors, it is not robust enough in the case of mobile phone sensing. Indeed, users will hardly perform the necessary periodic calibration, and unlikely have the necessary expertise to do so in an accurate way. Performing collaborative and automatic calibration among nearby sensing smartphones then appears as a promising alternative solution to the problem of correcting the bias of faulty sensors.

The main contribution of this paper lies in introducing a continuous automated *multiparty calibration* method for crowd-sensing systems, which overcomes the need for manual calibration. The proposed distributed calibration method compensates reading errors by automatically leveraging the knowledge of the crowd and without requiring: (i) a centralized calibration system or (ii) the involvement of the end user. Methods for the automated opportunistic calibration of sensors have previously been introduced in the literature (e.g., [18], [3], [20]). However, to the best of our knowledge, the proposed solutions are all limited to a pairwise process (i.e., the cooperation of one uncalibrated sensor with one calibrated sensor). Our method generalizes the opportunistic calibration to the multiparty case so as to leverage the contributions of the multiple nearby sensors and thereby improve the efficacy of automated calibration. The multiparty calibration is particularly well suited for participatory sensing where large crowds meet, as in public spaces.

We formulate the multiparty calibration problem as a minimization problem where: (i) the calibration parameters are expressed as a sum of mean square differences between the outputs of the mobile sensors that coordinate, and (ii) the residual calibration error has to be kept to a minimum. Assuming a set $\mathcal{S}$ of already calibrated sensors and a non-calibrated sensor s, which are all within a *shared sensing range* (i.e., their respective distance to the sensed phenomenon is such that they do measure the same), the proposed method calibrates the sensor s using the calibrated measurements of the subset $\mathcal{S}'$ of $\mathcal{S}$ that result in the lowest calibration error. Identifying the relevant subset $\mathcal{S}'$ requires analyzing the quality of the successive calibrations that have been taking place so far. To do so, we model the history of *multiparty calibration coordinations* (aka *multiparty calibration rendezvous*) among smartphones as a weighted directed hypergraph. The hypergraph then serves electing the smartphones with which to calibrate at a given time so that the new multiparty calibration results in the lowest cumulated calibration error. Going one step further, we generalize our solution to the case of *multi-hop, multiparty calibration* that coordinates a sequence of multiparty calibration rendezvous. We have implemented the proposed calibration system over Android and have applied it to the case of mobile noise sensing. Experimentation results show that the proposed calibration system increases the accuracy of mobile participatory sensing.

Summarizing, the contributions of our paper are as follows:

- We introduce a novel *multiparty calibration* method that leverages the multiple sensors of the relevant shared sensing (and communication) range for enhancing the efficacy of automated collaborative calibration (Section II).
- We specifically frame calibration as a parameter estimation problem and we use multivariate linear regression that generalizes simple regression methods so as to account for several predictor variables instead of a single one. We also apply robust regression so as to overcome the inherent presence of outliers (Section III).
- We have implemented a fully decentralized prototype of the calibration system for Android (Section IV).
- We have run experiment in the context of noise participatory sensing using the smartphone's microphone. Results show that the proposed automated calibration method leads to reduce the observation errors produced by the sensors (Section V).

Finally, related work (Section VI) and conclusions (Section VII) are presented.

II. Multiparty Calibration

As presented in the literature, for many sensors, there is a close to linear dependence between the measurements produced by the sensor and the sensed phenomenon (e.g., see [15], [2]). Accordingly, it is assumed that a sensor s can be provided with the relevant calibration coefficients/parameters for the linear function, which returns the calibrated measurements $\hat{y_s}(t)$ given the sensor measurements $y_s(t)$ at time t. In the same way, the measurements of a non-calibrated sensor have a linear dependence with those of a calibrated sensor, as illustrated in Figure 1 with the specific example of noise sensing using the smartphones' microphones. More formally, any reading $y_s(t)$ of the uncalibrated sensor s (at time t) can be expressed as the following function: $y_s(t) = \beta_0 + \beta_1 x_c(t)$ with $x_c(t)$ corresponding to the reading of the calibrated sensor c. It follows that a calibrated measurement is obtained by applying $\frac{y_s(t)-\beta_0}{\beta_1}$ on any raw reading $y_s(t)$ of s. The coefficient β_0 and β_1 of the linear equation are then computed using linear regression. The calibration of s through the coordination with c is called a *pairwise calibration rendezvous* (e.g., see [20]).

The computation of calibration parameters using calibrated measurements is necessarily less accurate than compared to the ground truth, especially when the calibrated sensor is a low-cost sensor. We thus generalize the pairwise approach to a *multiparty calibration rendezvous* so as to leverage the measurements provided by the multiple calibrated smartphones within the *shared sensing range* of the uncalibrated sensor.

A. From Pairwise to Multiparty Calibration

Figure 2 illustrates various scenarios of distributed collaborative calibrations, from the pairwise to the multiparty case.

As, e.g., in [20], we distinguish 2 types of sensors: *Reference sensors* that are calibrated at all times and low-cost sensors that require periodic calibration. Sensors of the former type are typically high-cost, high-quality sensors, such as high-end sonometers in the case of noise measurements.

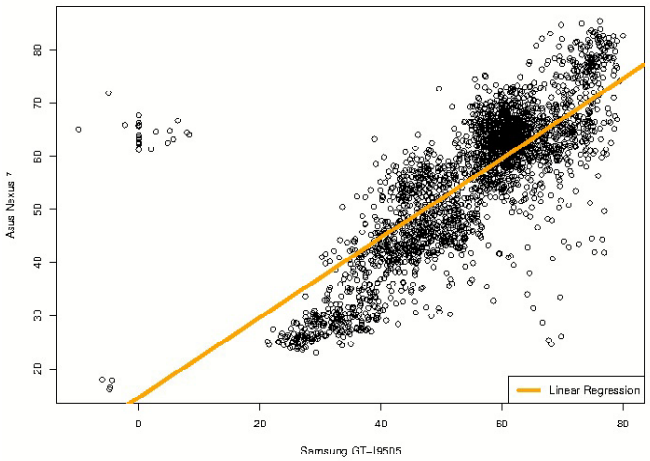


Fig. 1: *Linear relationship between raw and calibrated measurements:* Sound Level (dB(A)) of an uncalibrated Asus Nexus 7 expressed as a function of the Sound Level (dB(A)) sensed by a calibrated Samsung GT-I9505.

The distributed collaborative calibration requires rendezvous, although infrequent, with reference sensors for the calibration of a network of low-cost sensors.

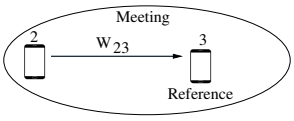
Figures 2a and 2b depict the pairwise case supported in the literature [18], [3], [20]. The latter in particular highlights the *multi-hop calibration*, based on a sequence of pairwise calibration rendezvous with the initial calibration starting with the Reference sensor. Figures 2c and 2d then introduce the general case of multi-hop, multiparty calibration.

B. Problem Formulation

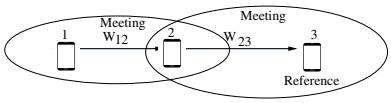
Consider n mobile smartphones, each with a built-in sensor, which are deployed across a large scale area $\mathbb{R}^2$. The smartphones/sensors are denoted using the indices $1, \cdots, n$. The raw readings/measurements obtained by any sensor i ($1 \leq i \leq n$) correspond to a time-ordered sequence of discrete measurements $y_i(t)$ taken at time $t_1, \cdots, t_p$. In order to assess the relationship between the measurements provided by multiple nearby devices, we rely on multivariate linear regression [6], from which to infer the calibration parameters for a sensor i given the calibrated measurements provided by sensors within the relevant shared sensing range.

Precisely, assume a non-calibrated smartphone i ($1 \leq i \leq n$) and K ($1 \leq K \leq n-1$) calibrated smartphones. We say that there is a *multiparty calibration rendezvous* between i and the K smartphones in the case of both spatial and temporal proximity/meeting over a time period $t \in [t_1..t_p]$. That is, if the following conditions are met:

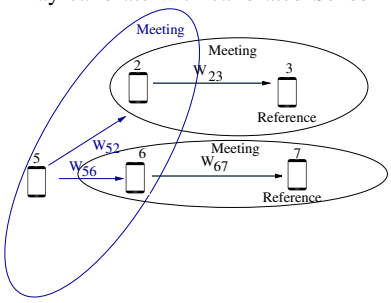
- The K sensors are in communication range with i during the overall calibration period $[t_1..t_p]$.
- The K sensors are in a *shared sensing range* with i, i.e., the distance between i and any sensor k of K is less than the maximum calibration distance D, which is defined



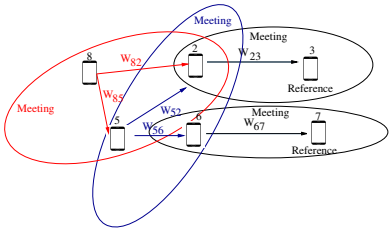
(a) *Pairwise calibration:* Sensor 2 is not calibrated and meets with Reference sensor 3 within a relevant sensing (and communication) range. Sensor 3 then coordinates with Sensor 2 for automated calibration.



(b) *Multi-hop pairwise calibration:* There is a sequence of pairwise calibrations. Sensor 2 first calibrates with the Reference Sensor 3. Then, Sensor 1 may calibrate with calibrated Sensor 2.



(c) *Multi-hop pairwise followed by multiparty calibration:* Sensors 2 and 6 calibrate through pairwise calibrations. Then, multiparty calibration occurs for Sensor 5.



(d) *Multi-hop multiparty calibrations:* Two concurrent pairwise calibrations are performed followed by a sequence of 2 multiparty calibrations, i.e., the calibration of Sensor 5 and then Sensor 8.

Fig. 2: *Calibration rendezvous:* From pairwise to multiparty.

based on the sensing phenomenon and environment being considered. Formally, we have: $|L_i(t) - L_k(t)| \leq D$ with $L_i(t)$ (resp. $L_i(t)$) defining the location of sensor i (resp. k) at time $t \in [t_1..t_p]$.

During the rendezvous, the uncalibrated sensor i attempts to calibrate using the calibrated measurements provided by the K nearby sensors, for which we leverage multivariate linear regression.

III. Multiparty Calibration using Multivariate Linear Regression

Given the multiparty calibration rendezvous of i with K smartphones, the purpose of the multivariate linear regression is to compute the calibration coefficients/parameters for i based on the calibrated measurements provided by the K mobile sensors. In particular, the calibrated measurement of sensor i is defined as the following linear function:

$$\hat{y_i}(t) = \beta_0 + x_1(t)\ \beta_1 + x_2(t)\beta_2 + \cdots + x_K(t)\beta_K + e_i(t) \quad (1)$$

where: $x_j(t)$ $(1 \leq j \leq K)$ denote the calibrated measurements of the K sensors; $\beta_0, \cdots \beta_K$ represent the unknown (and fixed) regression coefficients; and e_i is the residual noise, with $t \in [t_1..t_p]$. The model assumes that e_i is a normally distributed random variable with 0 mean and a constant standard deviation σ which is unknown [6]: $\mathbb{E}(e_i) = 0$, $var(e_i) = \tau^2$, $cov(e_i, e_k) = 0$ for all $i \neq k$.

Equation 1 can be compactly written into vectors and matrix as:

$$\hat{Y_i}(t) = X(t)\beta + E(t) \quad (2)$$

with: $\hat{Y_i}$ a p-dimensional vector, β a $K+1$ dimensional vector, X is $p \times (K+1)$ matrix, and E is p-dimensional vector.

We thus want to find the regression coefficients $\beta = (\beta_0, \cdots, \beta_K)^T$ of $\hat{Y_i}$, where ^{T} denotes the transpose of the matrix. For this, we apply the method of the least square, which minimizes the sum of the squared differences between the actual values y_i (resp. Y_i) and the calibrated values $\hat{y_i}$ (resp. $\hat{Y_i}$), using the linear regression defined in Equations 1 and 2. Formally, the estimate of the regression coefficient denoted $\hat{\beta} = (\hat{\beta_0}, \cdots, \hat{\beta_K})^T$ is computed such that $\sum_{t=t_1}^{t_p}[y_i(t) - \hat{y_i}(t)]^2$ is minimized. This difference $\sum_{t=t_1}^{t_p}[y_i(t) - \hat{y_i}(t)]^2 = \sum_{t=t_1}^{t_p}(y_i(t) - \beta_0 - \beta_1 x_1(t) - \beta_2 x_2(t) - \cdots - \beta_K x_K(t))^2$, is minimized by setting: $\hat{B} = (X^T X)^{-1} X^T Y$. It follows that the fitted value $\hat{Y}$ verifies:

$$\hat{Y} = X\hat{B} = X(X^T X)^{-1} X^T Y \quad (3)$$

The residual is given by:

$$\hat{E} = Y_i - \hat{Y_i} = (I - X(X^T X)^{-1} X^T)Y \quad (4)$$

After fitting a linear model, one may wonder how well it fits. There are characteristics of the sample multivariate regression that are of primary importance. They relate to the variance-covariance matrix of $\hat{B}$, the error deviation, their standard errors and their correlation. The variance-covariance matrix of the sample coefficient $\hat{\beta}$ is a symmetric $p \times p$ square matrix and is given by $V(\hat{\beta}) = \hat{\sigma}^2 (X^T X)^{-1}$. The diagonal values are the variances of the sample coefficients $Var(\hat{\beta_i})$, the covariance between two estimates, written as $cov(\hat{\beta_i}, \hat{\beta_j})$ is given as an off-diagonal value. Finally, the correlation between two coefficient estimates, written $Corr(\hat{\beta_i}, \hat{\beta_j})$, is given by the covariance divided by the product of standard deviation. The error deviation is estimated as $\hat{\sigma} = \sqrt{\sum_{j=t_1}^{t_p} \frac{(Y(t) - \hat{Y}(t))^2}{p-1}}$.

A. Removing Outliers using Robust Regression

Although we generally observe linear dependences between raw and calibrated sensor measurements, the existence of outliers in the measurements distorts the relationship (e.g., see Points at the Top-Left of Figure 1).

We thus remove outliers using the *robust regression* algorithm presented in [19], which iterates on the following steps:

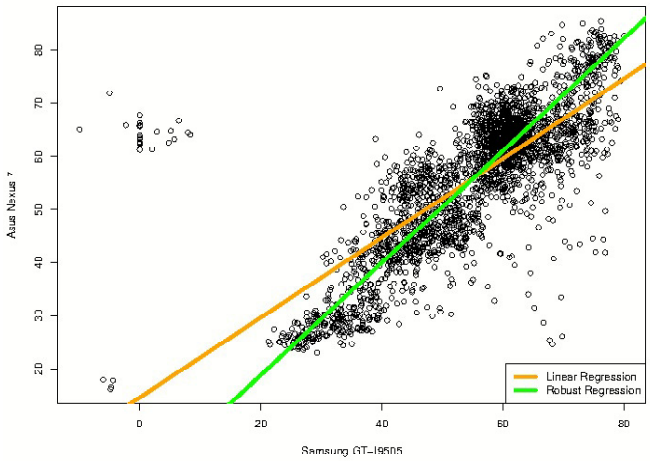


Fig. 3: *Multivariate with vs without robust regression:* Sound Level (dB(A)) of an uncalibrated Asus Nexus 7 expressed as a function of the Sound Level (dB(A)) sensed by a calibrated Samsung GT-I9505. Multivariate linear regression depicted with and without robust regression.

1) Randomly select few samples;
2) Fit the model to these few samples using the aforementioned multivariate regression;
3) Evaluate the quality of the fit on the remaining points using the median, which is a well known robust estimator. The subset that is characterized by the best fit is kept.

Finally, the model is re-fitted to all the data that are sufficiently close to the model and the data that are not close enough (i.e., the outliers) are removed. Figure 3 illustrates the multivariate regression without and with robust regression to remove the outliers. In what follows, we refer to the former as *multivariate (linear) regression*, and to the latter as *robust (linear) regression* although it is meant for *robust multivariate linear regression*.

B. Assessing the Relevance of a Calibration RendezVous

Thanks to the above multivariate and robust (linear) regression, the calibration coefficients of an uncalibrated smartphone i may be estimated based on the readings $x_1, \cdots, x_K$ provided by the surrounding K sensors that are met during a multiparty rendezvous. However, Sensor i should not systematically (re-)calibrate under the occurrence of a calibration rendezvous. It must do so only if the quality of the regression is sufficient. In particular, a high residual error (i.e., e_i in Equation 1) reflects a poor correlation between the readings provided by the surrounding sensors, and in such a case, the conditions are not met for an effective calibration. For instance, this may occur when although the sensors are in a shared sensing range, they have a different sensing context. Considering the noise sensing example, poor correlation typically occurs when some smartphones are in bags/pockets and others are handheld.

Another case to consider for assessing the relevance of a given multiparty calibration relates to the history of past calibrations. We must compare the quality of the calibration parameters computed in the current rendezvous against the quality of the previous calibrations (if any). We thus maintain the history of multiparty calibrations using a weighted directed *hypergraph*: a multiparty rendezvous between Sensor i and K sensors is represented by an *hyperedge* between node i and the K nodes. The quality of the regression established by i based on the readings provided by the K nodes is reflected by the weight of the directed hyperedge between i and the K nodes. Using the hypergraph, we introduce an algorithm that determines the best calibration parameters to apply by comparing the respective weights of the possible calibrations, while further supporting multi-hop calibration.

C. Multi-hop, Multiparty Calibration

Formally, we represent the multiparty rendezvous occurrences using a directed hypergraph $G = N(U, V)$, where U denotes the set of nodes (i.e., sensors/smartphones) and V the set of hyperedges representing the multiparty rendezvous. An hyperedge between a node i and K calibrated nodes is then weighted such that each of the inner-edges between i and one of the K nodes provides the properties of the regression established by i using the readings provided by the calibrated node. This is illustrated by the weights (w) provided over the (inner-)edges in Figure 2. Thus, $V : U * U- > R^q$ denotes the set of all the (inner-)edges such that $w_{ij}(t)$ for i, j in V is defined as a q-dimensional vector that depicts:

- The characteristics of the meeting between i and j, which includes, e.g., the time period during which the calibration is taking place;
- The properties of the calibration of sensor i, which is established based on the readings provided by sensor j. This includes $\hat{y_i}$ as well as the parameters that determine the quality of the regression, including the variance-covariance matrix (as previously defined) and the regression error.

Following, consider the hypergraph that is locally maintained by any participant node i, i.e., the graph depicts the history of past rendezvous together with the rendezvous that are currently eligible for the calibration of i. The K nodes in the sensing range of i similarly maintain their hypergraph of calibration rendezvous. The various nodes may then exchange their respective hypergraphs, which allows each of them to compute a path of rendezvous that minimizes their own calibration error (i.e., as estimated by the regression qualities that weight the edges). Specifically, any sensor i may independently compute the shortest path (in terms of the edge weights) towards a reference sensor, for which we implement the Dijkstra algorithm [5]. Ultimately, the elected calibration is the one that results in the lowest cumulated weight. Assuming that there are M meetings, and ordering the hypergraph as a self-balancing binary search tree, the algorithm requires $\Theta(|M||E|(|E| + |V|)log|V|)$ time to establish the shortest paths.

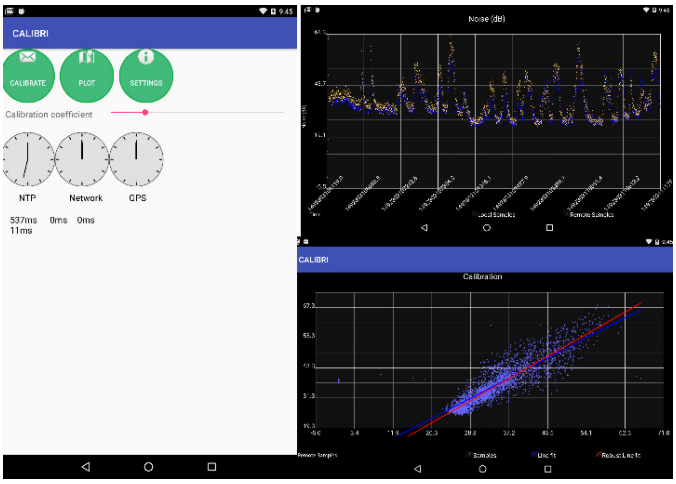


Fig. 4: *Visualization:* The Android App GUI.

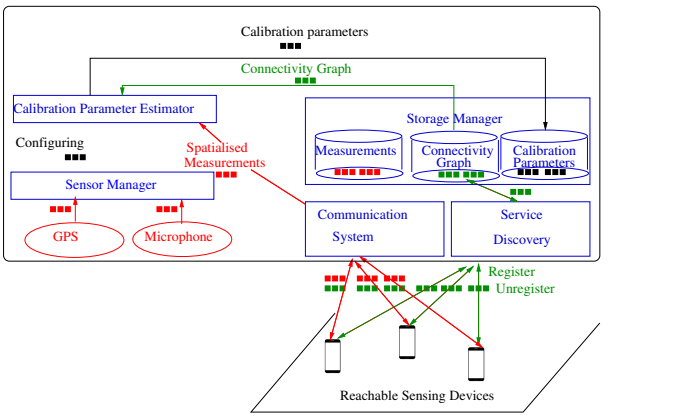


Fig. 5: *The distributed calibration system:* Architecture.

D. The Multiparty Calibration Process

Summarizing the proposed multiparty calibration method, any mobile i participating in the collaborative calibration system periodically runs the following algorithm:

1) Node i detects the presence of nearby sensing device(s), i.e., devices in the shared sensing range.
2) If any eligible rendezvous, i exchanges its sensing measurements (i.e., time series) in a synchronized manner so as to establish the linear relationship between the measurements of the nearby sensors and the raw measurements obtained locally.
3) The best calibration path is determined and the calibration function is set.

IV. Prototype Implementation

We have implemented a prototype of our opportunistic calibration system as an Android Application Package (APK). Figure 4 illustrates the GUI of the application, which plots the measurements and the regression line. The system is intended to enable pervasive calibration in a fully decentralized way, and thus does not require any central authority. While our approach supports the automated calibration of diverse sensor types, we have focused on noise sensing for our prototype and experiment, which leverages our prior experience with mobile noise sensing at the urban scale using the Ambiciti app (formerly called SoundCity – cf ambiciti.io) [10].

A. System Architecture

Figure 5 depicts the architecture of the calibration system that is deployed on any participating node.

The architecture features the following main components:

- *Service Discovery* tracks the nearby sensing devices with which to calibrate (rendezvous) in a fully decentralized way.
- *Communication System* establishes the connectivity with these nearby devices and orchestrates the exchange of information, while relying on device-to-device communication enabled by Wifi direct [1].
- *Sensor Manager* interacts with the local sensors, i.e., the positioning system and the microphone so as to gather spatialized and discretized sound measurements. In particular, the sound is recorded using a mono-channel 16-bits Pulse Code Modulation (PCM) with a sample rate of 44100Hz. The position is also used to determine wether the devices are within the relevant calibration distance.
- *Calibration Parameter Estimator* implements the multivariate linear and robust regressions so as to compute the calibration parameters. The component takes as input the measurements that have been received and the local measurement, and regress this information. The measurements include the local sound and the sounds that have been provided by the nearby devices. Note that sounds are stored only for the duration of the calibration rendezvous.
- *The Storage Manager* holds the configuration parameters and the measurements along with the rendezvous hypergraph.

B. Distributed Service Discovery

The discovery of the nearby phones that offer a calibration service is central to the opportunistic multiparty calibration, while we aim at a fully decentralized solution that does not require any central registry/discovery service. Our service discovery component is then implemented using Wifi direct [1], which supports both: (i) device-to-device communication without going through an access point, and (ii) service discovery at the link layer. In particular, Wifi direct forms a so-called P2P group that is equivalent to Basic Service Set (BSS) and negotiates a group owner that acts as an Access Point (AP). The resulting discovery process consists of the following phases:

1) The registration of the calibration service with a service advertisement including the service name as well as some details about the service.
2) The service discovery phase that is based on bonjour [9], where the device sends probes for service discovery.

3) The selection of the nearby device(s) that offer(s) the calibration service.
4) Once the nearby services are discovered, the P2P group formation takes places with a selection of the group owner.
5) The calibration rendezvous is initiated.

C. Core Calibration Service

Once devices offering the multiparty calibration service enter in a rendezvous (meet), thanks to the distributed service discovery, they exchange the following information towards calibration:

- The rendezvous hypergraphs that provide the respective histories of past rendezvous. This allows each participant to update its local hypergraph.
- The time to schedule the sound recording as well as the duration of the recording. In order to schedule the sound recording, the sensing device that acts as an AP disseminates a scheduling order including the starting time and the duration of the recording.
- The digital audio that has been recorded during the specified time period.

Based on the multiple measurements that have been provided by the nearby devices, the multivariate and robust regressions are performed as defined in the previous section, and the best path can be established, leading to the establishment of the calibration function.

D. Dealing with Synchronization

One practical problem related to sensing is the synchronization of the mobile sensors that enter into a rendezvous, which is needed to ensure that readings are correctly time stamped and can be adequately compared. We need a high accuracy synchronization given that 44 samples are taken during 1 millisecond with an acoustic sensor operating at 44100Hz.

Towards the above, we experimented several synchronization methods based on the Global Positioning System (GPS), the network-enabled Network Identity and Time Zone (NITZ), and Network Time Protocol (NTP). Although the time provided by the Global Positioning System (GPS) seems to be the most promising approach with an error in the order of nano seconds [4], such synchronization suffers from several limitations. In particular, the GPS primarily works outdoor due to weak satellite signal that does not easily penetrates building. Also, we experienced a weak accuracy (i.e., in the order of few hundreds of milliseconds) on some buggy Android implementations. Meanwhile, phones with telephony capabilities have a time synchronization provisioned by network operators using the NITZ mechanism which is part of the GSM specification. We observed significant offsets and drifts likely due to the fact that synchronization is in practice very rare (in the order of weeks). Finally, we made some tests using the NTP protocol, which is widely used in the Internet and shows a tile offset around few milliseconds. We therefore use NTP, which constitutes the most reliable solution in our case.

As we anticipate time-delayed and possibly noisy sensor readings, our objective lies in precisely determining the delay by cross-correlating sensors readings and identifying the delay when the peak of cross-correlation is obtained. In particular, we use the zero mean normalized cross correlation where the input readings are normalized with their mean and standard deviation over the lag region. Given two time series y_i and x_k, and τ samples, the normalized *cross-correlation* between the pair of series is defined by:

$$Cross_{y_i x_k}(\tau) = \frac{1}{(p-1)\sigma_{y_i}\sigma_{x_k}} \sum_{t=t_1}^{t_p} [y_i(t)-\mu_y].[x_k(t-\tau)-\mu_x] \quad (5)$$

where μ_x and μ_y are the means of each time series and there are p samples in each.

V. Evaluation

We report on the empirical evaluation of our multiparty calibration system.

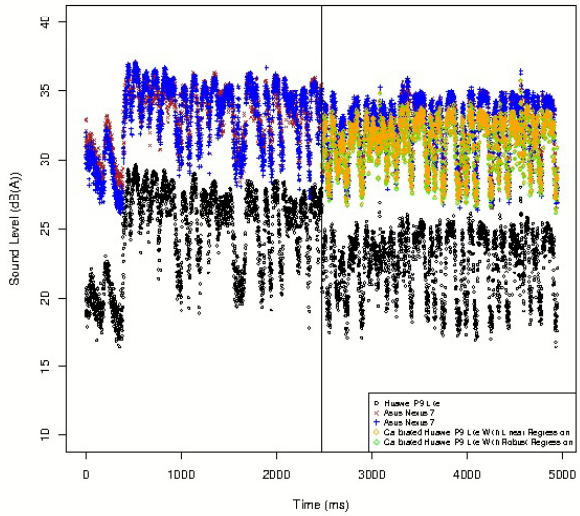
A. Multiparty Calibration RendezVous with 3 Participants

Our first test involves an uncalibrated phone (Huawei P9 Lite) and two calibrated phones (Asus Nexus 7). The three phones are next to each other and coordinate within a multiparty rendezvous for 5 seconds in an indoor environment.

Results are reported in Figure 6, which depicts on the right hand side: the calibrated measurements of the Asus Nexus 7s, the raw measurements of the Huawei, and the calibrated measurements of the Huawei after 2.5 seconds using multivariate and robust regression. The table on the right hand side further gives the Goodness of Fit for both regression. We note that the robust regression provides a better fit to the reference sound level than does the multivariate linear regression alone, thanks to the removal of 208 outliers. The mean deviation, which is expressed as the mean difference between the readings of the uncalibrated phone and the two calibrated phones was of about 8.34dB(A) before the calibration and is reduced to 0.78 dB(A) (resp. 0.64 db(A)) after the calibration using multivariate linear regression alone (resp. robust regression). The experiment confirms the hypothesis we formulated on the regression in Eq.1: the mean of the residual error is negligible (-7.27.10^{-15} for a linear regression alone and -2.35.$10-15$ with robust regression) and its standard deviation remains constant (resp. 1.23 and 0.85). In addition, the correlation between the readings y and the calibrated ones $\hat{y}$, which is represented by the coefficient of determination denoted R^2, is conveniently very high and superior with robust regression.

B. Impact of the RendezVous Duration

We now assess the impact of the rendezvous duration on the calibration (Figure 7). We note that while a relatively short meeting (i.e., in the order of 1 second) is sufficient to calibrate, a shorter meeting does not allow collecting enough observations for guaranteeing a calibration with a high R^2 and small residual error.



	Linear Regression	Robust Regression
Mean deviation before calibration	8.34 dB(A)	8.34 dB(A)
Mean deviation after calibration	0.78 dB(A)	0.64 dB(A)
R^2	0.85	0.91
Adjusted R^2	0.84	0.92
Mean of the residual error	$-7.27.10^{-15}$	$-2.35.10-15$
Median of the residual error	0.04	0.03
Std deviation of the residual error	1.23	0.85
Number of measurements	2480	2272

Fig. 6: *Sound level (dB(A)) sensed by 3 smartphones within a shared sensing range*: The uncalibrated Huawei P9 Lite adjusts its calibration parameters, relying on multivariate linear regression with and without robust regression, and using the calibrated measurements provided by the two Asus Nexus 7.

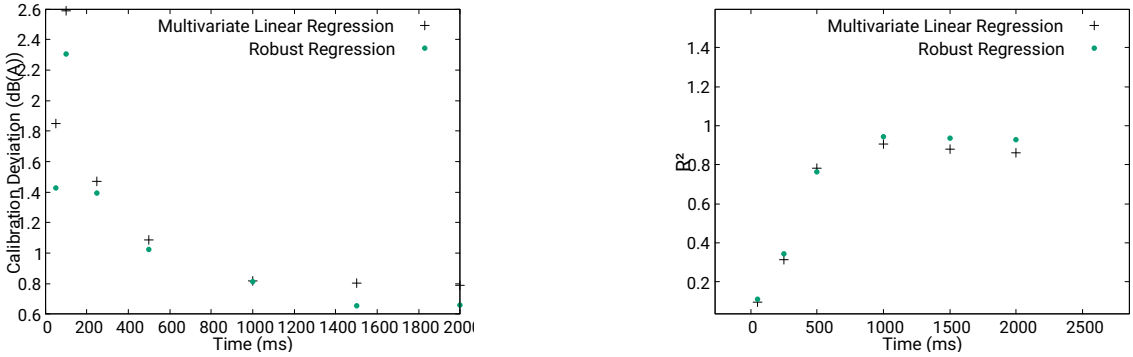


Fig. 7: *Impact of the rendezvous duration*: Quality of the calibration according to the duration of the rendezvous in terms of *Calibration Deviation* (Left) and *Coefficient of Determination* (Right).

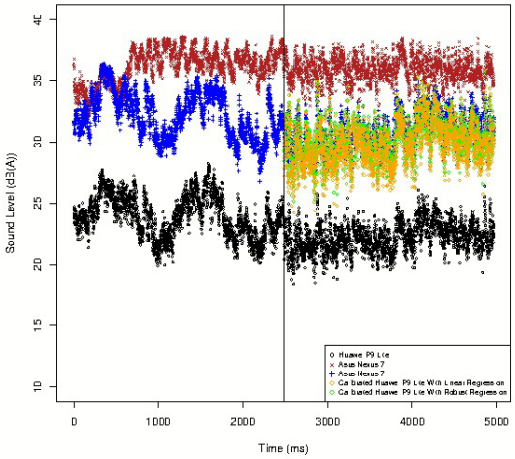
Distance	R^2 of a linear regression	R^2 of a robust regression
5	0.44	0.5
10	0.18	0.18
20	0.06	0.06

TABLE I: Goodness of fit expressed as a function of the distance separating three smartphones in an indoor environment.

C. Accounting for the Diverse Sensing Contexts

In general, a calibration rendezvous must lead to change the calibration parameters of a sensor only when the context is appropriate. The goodness of fit of the regression enables discarding non relevant rendezvous. As an illustration, Table I gives the value of R^2 according to the distance of the smartphones from each other. While the value of R^2 is as high as 0.85 in Figure 6 that corresponds to a short range rendezvous, the value of R^2 becomes too low for a large range and leads to discard the associated calibration parameters. Note that this is inferred without requiring precise location data.

Our last experiment focuses on the case where one of the calibrated phones participating to a rendezvous is in a pocket and experiences friction, while the two other phones (one calibrated and one not calibrated) are handheld. Figure 8 shows that the calibration is not impacted by the wrong sensing context of the phone in the pocket: after calibration, the sound level sampled by the newly calibrated Huawei smartphone is close to that of the handheld calibrated smartphone and far from that of the smartphone in the pocket. In practice, the multivariate regression operates in such a way that it reduces the impact of the uncorrelated measurements by lowering accordingly the value granted to the related β coefficient. It follows that the multivariate regression compensates faulty sensors.



	Linear Regression	Robust Regression
Deviation before calibration	10.25 dB(A)	10.25 dB(A)
Deviation after calibration	2.69 dB(A)	2.5 dB(A)
R^2	0.52	0.57
Adjusted R^2	0.52	0.57
Mean of the residual error	$3.95.10^{-15}$	$-3.93.10^{-15}$
Median of the residual error	-0.038	-0.03
Std deviation of the residual error	1.16	1.099
Number of observations	2487	2436

Fig. 8: *Sound level (dB(A)) sensed by 3 smartphones within a shared sensing range but inconsistent calibration context*: The uncalibrated Huawei P9 Lite, which is handheld adjusts its calibration parameters, relying on linear regression with and without robust regression, and using the calibrated measurements provided by the two Asus Nexus 7, one handheld (blue measurements), and one in a pocket and experiencing friction (red measurements on top).

VI. Related Work

Although calibration remains a prerequisite to the wide deployment of sensors, a relatively small body of research has tackled the issue of adjusting the calibration parameters at scale. Existing work subdivides into micro and macro calibration approaches, depending on the number of sensors involved.

A. Micro Calibration

The problem of calibrating a small set of sensors, which is also known as *micro calibration*, is usually addressed using a controlled stimuli serving as ground-truth data. The micro-calibration can be further performed manually or automatically.

Few recent studies [11], [16], [22] investigate the micro calibration of the microphone readings as part of applications that monitor the noise level. In practice, the calibration is performed manually in a laboratory by generating a noise using dedicated loudspeaker. The noise measured by the monitoring application is further confronted to a professional sound level meter.

For instance, as part of the Ambiciti/SoundCity noise sensing application [22], the noise accuracy of mobile microphones is assessed using pink noise (low frequency ranging from 20Hz to 20kHz) with a sound pressure ranging from 40 to 95 dB(A) as well as narrowband noise with a center frequency of 125, 500 and 8000Hz and a respective bandwidth of 12.5, 50, 200 and 800Hz. Confronting the app measurements (using the microphone) to a Cirrus Optimus red class of sound level meter meeting the IEC61672 international standards, the study investigates the range, accuracy and repeatability sensibility inside a lab considering various sensing conditions (i.e., device orientation, speed, mitigation, friction and wind). Experiments show that although there is usually a good linearity in phone response, for some smartphones, non linearities appear above a certain threshold (75 dB(A)) most likely due to saturation. Results also show that there is a limited variability of the bias considering a given model of smartphone [10]. However, as also reported in [11], there is a high variance in measurements of similar apps between different device models.

In [16], authors use broadband white noise in a $125m^3$ ISO 3741 compliant room. The output voltage was adjusted in Pulse to produce a uniform sound field at 50 dB(A), 70 dB(A), and then 90 dB(A). While testing the capability of the devices for measuring environmental noise, authors showed that IOS apps perform better than Android-based apps and that the age of the phone impacts the accuracy perhaps due to the deterioration of microphone over time. The accuracy of the apps varied widely across apps [17]. While manual calibration of inbuilt microphone using a controlled stimuli is routinely performed, automatic and blind calibration that involves observing unknown signals is also investigated.

B. Macro Calibration

Macro-calibration eliminates the need for calibrating each sensor individually by jointly calibrating sensors in the field as a whole and allowing calibrated sensors to calibrate non-calibrated ones. Macro calibration implies that the sensor density is sufficient to ensure that spatially-correlated readings are almost identical. Macro-calibration is commonly addressed without relying on a controlled stimuli and as such can be qualified as *blind calibration*.

Recent results in macro calibration focus on collaborative and distributed aspects. In collaborative blind calibration [21], calibration among nearby sensors is treated as a consensus problem where the proposed algorithm achieves an asymptotic

agreement for sensor gains and offsets in the mean square sense.

There are few macro-calibration methods that cope with mobile sensing [18], [3], [20]. All scenarios assume pairwise calibration and deal with sensors values that vary smoothly (as it is the case with low-cost gas, temperature, ozone and carbon monoxide sensors). Based on the readings provided by nearby sensors, a pairwise calibration is performed using linear regression. Finally, multi-hop calibration is provided using the model of pairwise rendezvous introduced in [3], which is expressed as a matrix where non-zero edge represents a pairwise calibration. In [3], time series are collected in a synchronized manner while linear correlation is used to weight all data points and filter out irrelevant/uncorrelated points. Then, simple linear regression is used to establish the pairwise calibration parameters. Finally, all the paths are exhaustively enumerated using breadth-first search, discarding paths with a confidence level below a threshold. The path with the highest confidence level, which is obtained by multiplying confidence levels along each segment of the path, is privileged.

Our work goes one step further by introducing multi-hop multiparty calibration, which leads to the increased efficacy of automated calibration. Further, the proposed approach is particularly well suited for crowd sensing scenarios where the crowd meets, as in public spaces.

VII. Conclusion

Novel mobile applications spark the recognition of a new calibration problem where numerous devices calibrate without requiring the involvement of end-users or expensive sensor technologies. To tackle this new problem, we introduce an opportunistic macro calibration system that pervasively compensates the reading error of an uncalibrated sensor. While opportunistic calibration in mobile sensing has been so far addressed as a pairwise process involving only two sensors, we leverage the high density of smartphones in the urban environment to introduce automated, multiparty calibration. Our multiparty calibration system coordinates multiple surrounding smartphones and leverages multivariate linear regression where calibration parameters are expressed as a sum of mean square differences between the reading of the neighboring sensors. Further, our system handles outliers using robust regression. We have implemented and experimented our solution. Initial results show that our system contributes to enhancing the accuracy of mobile phone sensing. In particular, our evaluation shows that multiparty calibration leads to accurately calibrate phones as they meet, while discarding non-relevant sensing context (e.g., measurements provided by a calibrated phone in a pocket). Future work includes studying the multiparty calibration method in relation with user mobility and crowd meeting, so as to assess the most relevant contexts for running multiparty calibration as opposed to continuously running the calibration process in the background.

References

[1] Wifi Alliance. Wifi peer-to-peer services (p2ps), technical specification (for wifi direct certification) version 1.2. In *http://www.wi-fi.org/discover-wi-fi/wi-fi-direct*, 2014.

[2] S. Burke. Regression and calibration. In *Statistics and data analysis*, LC.GC European online supplement, 2011.

[3] V. Bychkovskiy, S. Megerian, D. Estrin, and M. Potkonjak. A collaborative approach to in-place sensor calibration. In *Proceedings of the 2nd International Conference on Information Processing in Sensor Networks*, 2003.

[4] P.H. Dana. Global positioning system (gps) time dissemination for real-time applications. *Real Time Systems*, 12, 1997.

[5] E.W. Dijkstra. A note on two problems in connexion graphs. In *Numerische Mathematik*, 1959.

[6] N.R. Draper and H. Smith. *Applied Regression analysis*. Wiley Series in Probability and Statistics, 2014.

[7] S. Hachem, V. Mallet, R. Ventura, A. Pathak, V. Issarny, P. G. Raverdy, , and R. Bhatia. Monitoring noise pollution using the urban civics middleware. In *2015 IEEE First International Conference on Big Data Computing Service and Applications (BigDataService)*, 2015.

[8] S. Hachem, G. Mathioudakis, A. Pathak, V. Issarny, and R. Bhatia. Sense2health: A quantifed self application for monitoring personal exposure to environmental pollution. In *Proceedings of SENSORNETS*, 2015.

[9] Apple Inc. Bonjour overview. In *https://developer.apple.com/bonjour/*.

[10] Valerie Issarny, Vivien Mallet, Kinh Nguyen, Pierre-Guillaume Raverdy, Fadwa Rebhi, and Raphael Ventura. Dos and Don'ts in Mobile Phone Sensing Middleware: Learning from a Large-Scale Experiment. In *ACM/IFIP/USENIX Middleware 2016*, Proceedings of the 2016 International Middleware Conference, Trento, Italy, 2016.

[11] C.A. Kardous and P.B. Shaw. Evaluation of smartphone sound measurement applications. *Journal of Acoustical Society of America Express Letter*, 135(4), 2014.

[12] W. Z. Khan, Y. Xiang, M. Y. Aalsalem, and Q. Arshad. Mobile phone sensing systems: A survey. *IEEE Communications Surveys Tutorials*, 15(1):402–427, First 2013.

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

[14] Nicholas D. Lane, Shane B. Eisenman, Mirco Musolesi, Emiliano Miluzzo, and Andrew T. Campbell. Urban sensing systems: Opportunistic or participatory? In *Proceedings of the 9th Workshop on Mobile Computing Systems and Applications*, HotMobile '08, pages 11–16, New York, NY, USA, 2008. ACM.

[15] J. Majewski and O. Boyko. Application of regression methods to humidity sensors calibration. *Proceedings of electrotechnical institute*, 252, 2011.

[16] E. Murphy and E. King. Testing the accuracy of smartphones and sound level meter applications for measuring environmental noise. *Applied Acoustics*, 135, May 2016.

[17] D.R. Nast, W.S. Speer, and C.G. Le Prell. Sound level measurements using smartphone "apps": Useful or inaccurate? *Noise and health Journal*, 106(72), 2014.

[18] G.P. Picco and W. Heinzelman. On-the-fly calibration of low-cost gas sensors. In *European conference on Wireless Sensor Networks*, 2012.

[19] P. Rousseeuw. Least median of squares regression. *Journal of the American Statistical Association*, 79(388), 1984.

[20] O. Saukh, D. Hasenfratz, and L. Thiele. Reducing multi-hop calibration errors in large-scale mobile sensor networks. In *14th International Conference on Information Processing in Sensor Networks*, 2015.

[21] M.S. Stankovi, S.S. Stankovi, and K.H. Johansson. A consensus-based distributed calibration algorithm for sensor networks. *Serbian journal of electrical engineering*, 135(4), 2016.

[22] R. Ventura, V. Mallet, V. Issarny, P-G Raverdy, and R. Rebhi. Evaluation and calibration of mobile phones for noise monitoring application. 2017. Submitted for publication.