Model-Driven Public Sensing in Sparse Networks

Damian Philipp(✉), Jarosław Stachowiak, Frank Dürr, and Kurt Rothermel

Institute of Parallel and Distributed Systems,
University of Stuttgart, Stuttgart, Germany
{damian.philipp,jaroslaw.stachowiak,
frank.duerr,kurt.rothermel}@ipvs.uni-stuttgart.de

Abstract. Public Sensing (PS) is a recent trend for building large-scale sensor data acquisition systems using commodity smartphones. Limiting the energy drain on participating devices is a major challenge for PS, as otherwise people will stop sharing their resources with the PS system. Existing solutions for limiting the energy drain through model-driven optimizations are limited to dense networks where there is a high probability for every point of interest to be covered by a smartphone. In this work, we present an adaptive model-driven PS system that deals with *both* dense and sparse networks. Our evaluations show that this approach improves data quality by up to 41 percentage points while enabling the system to run with a greatly reduced number of participating smartphones. Furthermore, we can save up to 81 % of energy for communication and sensing while providing data matching an error bound of 1 °C up to 96 % of the time.

1 Introduction

Public Sensing (PS) is a recent trend for building flexible and large-scale sensor data acquisition systems, facilitated by the proliferation of commodity smartphones [3]. Modern smartphones feature various sensors such as camera, light intensity, and positioning sensors like GPS. In addition, they offer capabilities for processing and communicating sensor data. Thus, sensor data can be obtained without having to support a fixed sensor network.

In building such PS systems, we face several challenges. On the device side, the main issue is a limited energy supply. While smartphone batteries are frequently recharged, keeping the energy consumption for PS minimal, i.e., ensuring that the battery still makes it through a whole day, is a key requirement as otherwise participants may be unwilling to support PS. On the data side, problems are to specify tasks and to deliver data with sufficient quality. Due to node mobility, it is likely that each time data is requested, a different device is best suited to take readings for the task at hand. However, if we want to minimize the energy consumption on participating devices, querying all smartphones for readings or even proactively collecting location information for all devices is prohibitive.

I. Stojmenovic et al. (Eds.): MOBIQUITOUS 2013, LNICST 131, pp. 17–29, 2014.
DOI: 10.1007/978-3-319-11569-6_2

These challenges require careful planning which smartphones should take readings where and when to ensure that useful data of sufficient quality is delivered to a client of the system while keeping the energy consumption minimal.

To address the problem of how to specify interesting data and thus enable flexible PS systems, the concept of *virtual sensors* (v-sensors for short) was introduced [12]. V-sensors provide a mobility-transparent abstraction of the PS system. They are configured to report a set of readings at a client-defined sampling rate at a given position, thus presenting a view on a static sensor network. The PS system then selects nearby smartphones to provide readings for a v-sensor.

However, due to node mobility, some v-sensors may not have any smartphone nearby and may thus be unable to report data readings. Model-driven approaches can be used to fill these gaps with a value inferred from available data [10] and to improve the energy consumption by leaving out v-sensors where values can be inferred with sufficient accuracy [13], thus making large-scale PS viable.

However, these approaches are tailored towards dense networks where most v-sensors are well covered (and thus available), e.g., in a busy city center or a business area at lunchtime. For model-driven approaches to provide accurate inferred readings, a minimum set of input data from available v-sensors is required. Collecting this minimum set of data is a problem in (partially or completely) sparsely populated areas, e.g., business areas during off-hours or housing areas during business hours, where the density of smartphones is overall low, or when the most interesting v-sensors are unavailable while many less interesting v-sensors are available.

We address this challenge by presenting an approach for optimized model-driven PS that works in both dense and sparse networks. To this end, we extend our previous model-driven approach. The basic idea is to derive knowledge on v-sensor availability from ongoing query executions. This knowledge is then used in a multi-round approach to iteratively refine the set of v-sensors to query.

In detail, the main contributions of this paper are: (1) An approach for building knowledge on v-sensor availability without extra energy cost. (2) An adaptive query execution model that exploits this knowledge to compensate for unavailable v-sensors, thus making optimized PS viable in both dense and sparse networks. (3) Evaluations analyzing the performance of our approach and showing significant improvements compared to previous approaches.

The quality of data obtained by our system is improved by up to 41 percentage points while at the same time useful data can be provided with a greatly reduced number of participating smartphones. Furthermore, we show that we can save up to 81 % of energy for communication and sensing while providing inferred readings matching an error bound of 1 °C up to 96 % of the time. As a by-product, our system is privacy-friendly, i.e., it provides data readings of good quality without tracking the position of individual smartphones.

The remainder of this work is structured as follows. Section 2 presents the system model and problem statement. In Sect. 3 we present the model-driven PS system before we describe the extensions for sparse networks in detail in Sect. 4.

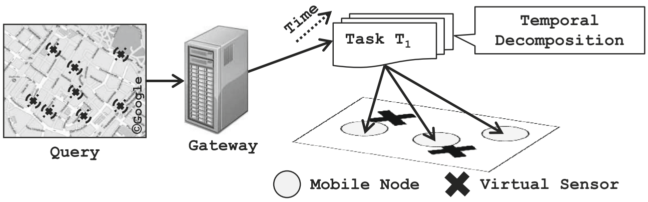


Fig. 1. Overview of sensing task execution

Evaluation results for our system are discussed in Sect. 5. Section 6 compares our approach to related work while Sect. 7 concludes this work.

2 System Model and Goals

First, we present our system model and formulate the problem to be solved by our enhanced PS system.

2.1 System Model and Architecture

Following the general design of PS systems, our system consists of two kinds of components: *mobile smartphones* and a *gateway server* (see Fig. 1). Each smartphone features a positioning sensor such as GPS, has constant Internet access, e.g., via 3G, and has access to a set of environmental sensors (sound, temperature, air pollution, etc.) that may be built-in or connected via Bluetooth. We assume that each smartphone has access to all sensors necessary to satisfy any request posted to the system. Users of mobile smartphones are assumed to be walking with no further assumptions about their mobility. The *gateway server*, located on the Internet, serves as an interface for clients to request data from the system and redistributes these requests to the smartphones. Note that for scalability the gateway may be implemented as a distributed service.

To request data, clients submit a query $Q = (V, p, QoS)$ to the gateway, consisting of a set of virtual sensors $Q.V$, a sampling period $Q.p$, and a set of quality parameters $Q.QoS$. The sampling period dictates the interval at which readings for all v-sensors should be provided. The quality parameters control the operation of our algorithm and will be explained in the corresponding sections. *Virtual Sensors* are attributed with a type of reading *v.type* and a position *v.loc*, thus specifying where to take data readings. Furthermore, each v-sensor has a coverage area $v.area$ defined relative to its location. When a smartphone is located in $v.area$, it may take a reading for v and we say that v is *available*. Otherwise, v is *unavailable*. Coverage areas of v-sensors in a query $v \in Q.V$ must be pairwise disjoint to ensure a unique mapping of smartphones to v-sensors, but may otherwise be chosen arbitrarily.

Each v-sensor v can provide either an *effective reading* or an *inferred reading*. An effective reading is taken by a smartphone in $v.area$ whereas an inferred reading is computed at the gateway using a data-driven model without interaction with any device.

2.2 Problem Statement

Our goal is to efficiently provide sensor data on spatially distributed environmental phenomena according to a client-defined quality bound $Q.QoS$, independent of the current distribution of smartphones in the observed area. We want to minimize the number of requested effective readings while at the same time compensating for unavailable v-sensors and maximize the number of v-sensors $|V'|$ for which the quality constraints are fulfilled.

3 Optimized Query Execution in Dense Systems

In this section we first introduce the multivariate Gaussian distribution model used by our approach. We then present the basic model-driven execution for energy-efficient PS systems (DrOPS), based on [13], that will be extended with adaptive algorithms for compensating for unavailable v-sensors in later sections.

3.1 Multivariate Gaussian Distribution

Multivariate Gaussian Distributions (MGD) have been shown to be a suitable model for inferring values for spatially distributed phenomena, e.g., in [4,5,10]. Their advantage over other methods, e.g., spatial interpolation approaches such as linear interpolation, is that they capture the correlation of observed values rather than relying on indirect criteria, e.g., spatial distance. Note that other types of phenomena, e.g., discrete events, may require a different model. In our system, an MGD model is used in two ways: Inferring missing values from a set of incomplete observations and selecting the best set of v-sensors to observe.

Given a model MGD_V over a set V of v-sensors and a vector of effective readings V_{eff} at v-sensors $V_{\text{eff}} \subset V$, we can infer the most likely current values $\mu_{u|P_{V_{\text{eff}}}}$ at (currently unobserved) v-sensors $u \in V_{\text{inf}} = V \setminus V_{\text{eff}}$ as

$$\mu_{u|P_{V_{\text{eff}}}} = \mu_u + \Sigma_{u,V_{\text{eff}}} \Sigma^{-1}_{V_{\text{eff}},V_{\text{eff}}} (P_{V_{\text{eff}}} - \mu_{V_{\text{eff}}}) \tag{1}$$

$$\sigma^2_{u|V_{\text{eff}}} = \sigma^2_u - \Sigma_{u,V_{\text{eff}}} \Sigma^{-1}_{V_{\text{eff}},V_{\text{eff}}} \Sigma_{V_{\text{eff}},u} \tag{2}$$

where μ_V is the vector of mean values for all $v \in V$ and $\Sigma_{V,V}$ is the matrix of (co)variances between all v-sensors in the model. The output is a Gaussian distribution where $\sigma^2_{u|V_{\text{eff}}}$ indicates whether the observations V_{eff} were a good choice for inferring $\mu_{u|P_{V_{\text{eff}}}}$.

To optimize the operation of our system, we strive to minimize the size of V_{eff} while ensuring good data quality, i.e., limiting $\sigma^2_{u|W}$ to a client-defined threshold $Q.QoS.\sigma^2_{max}$. Finding the smallest V_{eff} that still achieves a given quality of inferred values is an NP hard problem, for which the near-optimal heuristic

GREEDY algorithm was proposed [8]. GREEDY iteratively selects a fixed number of v-sensors. Initially, $V_{\text{inf}} = V$. In each iteration, the v-sensor v with the maximum mutual information is moved to V_{eff}, i.e., the v-sensor that reduces the uncertainty about the values at v-sensors in $V_{\text{inf}} \setminus \{v\}$ the most. For a detailed discussion of this algorithm and the mutual information criterion, see [8].

To adapt the algorithm to selecting a set of v-sensors based on the requested result quality rather than a predetermined fixed number, we change the termination criterion: in our system, MODIFIEDGREEDY adds v-sensors to V_{eff} until $\forall u \in V_{\text{inf}} : \sigma^2_{u|V_{\text{eff}}} \leq Q.QoS.\sigma^2_{max}$.

Note that the achievable degree of optimization depends on the magnitude of the correlations found in the data. If only weak correlations exist, MODIFIEDGREEDY will select $V_{\text{eff}} = V$. Furthermore, the accuracy of the selection as well as the inference relies on the accuracy of the MGD. As we will show, our system ensures that the MGD in use always reflects current data.

3.2 Model-Driven Query Execution

Next, we look at how to apply the model-based optimization in a PS system.

The operation of DrOPS is driven by the gateway. Given a query $Q = (V, p, QoS)$, in each sampling period, the gateway creates a sensing task $T = (V_{\text{eff}}, QoS), V_{\text{eff}} \subseteq V$ as depicted in Fig. 1. T is then broadcast to all smartphones. On receiving T, each smartphone samples its position and determines whether it is located in the coverage area of any v-sensor $v \in V_{\text{eff}}$. If so, it takes a reading of the requested type and returns the reading along with the identity of the v-sensor to the gateway. Should there be more than one effective reading reported for a v-sensor v, only the reading that was taken closest to $v.loc$ is retained. All other readings for v are discarded.

To optimize data acquisition, DrOPS alternates its operation between two phases. In *Basic Operation Phases*, V_{eff} is equal to V, i.e., no optimization is performed. Data is gathered to build or update an MGD model of the phenomenon observed in this query and only effective readings for available v-sensors are reported to the client. To keep the optimized operation phase short, an online learning algorithm is used [13]. When an MGD model is available, the system switches to an *Optimized Operation Phase*. In this phase, MODIFIEDGREEDY is used to minimize the size of V_{eff} and inferred readings are provided for all v-sensors $v \in V_{\text{inf}} \cup$ *unavailable v-sensors*, i.e., where no effective reading was taken. In parallel, an online model validity check algorithm determines whether the current MGD has become inaccurate and if so, switches the system back to a basic operation phase.

4 Alternate Virtual Sensor Selection for Sparse Networks

The optimized query execution presented in the last section *assumes*, that most or all of the v-sensors are constantly available. This assumption does not hold in a

Require: V, MGD_V, V_{unav}, V_{avl}, $Q.QoS.\sigma^2_{max}$
$V_{\text{eff}} = V_{\text{avl}}, V_{\text{inf}} = V \setminus V_{\text{eff}}$
while $\exists v \in V : \sigma^2_{v|V_{\text{eff}}} > Q.QoS.\sigma^2_{max}$ and $V_{\text{eff}} \neq V \setminus V_{\text{unav}}$ **do**
$u = argmax_{u \in V_{\text{inf}} \setminus V_{\text{unav}}}$ MUTUALINFORMATION(u, V, MGD_V, V_{eff})
$V_{\text{eff}} = V_{\text{eff}} \cup \{u\}, V_{\text{inf}} = V_{\text{inf}} \setminus \{u\}$
end while
return V_{eff}

Fig. 2. AdaptiveGreedy algorithm

sparse network setting, which is characterized by a low probability for each individual v-sensor to be available. Therefore, we begin by introducing the AdaptiveGreedy algorithm that includes *knowledge* about the (un)availability of v-sensors in the selection. Finally, we present how to use AdaptiveGreedy in our *Round-based Alternate V-Sensor Selection* to extend DrOPS to compensate for unavailable v-sensors.

4.1 Adaptive Greedy Algorithm

Compared to the previously described ModifiedGreedy algorithm, AdaptiveGreedy depicted in Fig. 2 takes two additional parameters: A set of v-sensors *known* to be unavailable $V_{\text{unav}} \subseteq V$ and a set of v-sensors *known* to be available $V_{\text{avl}} \subseteq V, V_{\text{avl}} \cap V_{\text{unav}} = \emptyset$. The availability of v-sensors not contained in $V_{\text{avl}} \cup V_{\text{unav}}$ is unknown. Using an optimistic strategy, AdaptiveGreedy *assumes* these v-sensors to be available, although they may turn out to be unavailable during task execution. A pessimistic strategy would need to probe the availability of all v-sensors beforehand by querying all smartphones for their position. This would cause the PS system to use as much energy as an approach without any optimization just for probing v-sensor availability, thus voiding the entire optimization approach.

Given these parameters, AdaptiveGreedy computes a new selection of v-sensors V_{eff} analogous to ModifiedGreedy under the additional constraints that no v-sensor known to be unavailable is selected and that all v-sensors known to be available are selected, i.e., $V_{\text{eff}} \cap V_{\text{unav}} = \emptyset$ and $V_{\text{avl}} \subseteq V_{\text{eff}}$. Forcibly selecting all of V_{avl} is warranted by the fact that in our system detecting the availability of v-sensor v coincides with getting an effective reading for v (see Sect. 4.2). Thus, not selecting all of V_{avl} would be a waste of effort.

4.2 Round-Based Alternate Virtual Sensor Selection

We now introduce the *Round-based Alternate V-Sensor Selection*, depicted in Fig. 3, where the gateway subdivides each sampling period into a number of *Q.QoS.rounds* rounds. The duration of each round is $\frac{Q.p}{Q.QoS.rounds}$. At the beginning of each round we first update our knowledge about current v-sensor availability. Based on this knowledge, we then select a new set of v-sensors for which effective readings should be acquired. Note that for long sampling periods $Q.p$,

Require: V, MGD_V, $Q.QoS$
$V_{\text{avl}} = \emptyset, V_{\text{unav}} = \emptyset, E_0 = \emptyset, V_{\text{eff},0} = \emptyset$
for $i = 1..Q.QoS.rounds$ **do**
$V_{\text{avl}} = V_{\text{avl}} \cup E_{i-1},\ V_{\text{unav}} = V_{\text{unav}} \cup (V_{\text{eff},i-1} \setminus E_{i-1})$
$V_{\text{eff},i} = \textsc{AdaptiveGreedy}(V,\ MGD_V,\ V_{\text{unav}},\ V_{\text{avl}},\ Q.QoS.\sigma^2_{max})$
$V_{\text{eff},i} = V_{\text{eff},i} \setminus \bigcup_{j=1}^{i-1} V_{\text{eff},j}$
if $V_{\text{eff},i} = \emptyset$ **then**
return $\bigcup_{j=1}^{i-1} E_j$
end if
$E_i = \textsc{executeTask}(V_{\text{eff},i}, Q.QoS)$
end for
return $\bigcup_{j=1}^{Q.QoS.rounds} E_j$

Fig. 3. Round-based alternate v-sensor selection

round duration should be limited to, e.g., 5 s each to ensure that smartphones cannot move too much between individual rounds. Otherwise, the availability of v-sensors may significantly change during each round, thus voiding the knowledge on v-sensor availability built so far. For the same reason, we do not carry over knowledge from past sensing periods, as nodes may have moved significantly between sensing periods.

In the first round, $V_{\text{avl}} = \emptyset = V_{\text{unav}}$, thus we assume all v-sensors to be available. Therefore, the initial selection of $V_{\text{eff},1}$ is identical to using ModifiedGreedy as in the non-adaptive system. In fact, when setting $Q.QoS.rounds = 1$, the system behaves exactly as previously presented in Sect. 3. The resulting subtask T_1 is distributed to the smartphones. For all v-sensors in $V_{\text{eff},1}$ that are actually available an effective readings will be reported to the gateway. All readings received in this round are stored in set E_1.

In subsequent rounds $i = 2 \ldots Q.QoS.rounds$, we first update our knowledge on v-sensor availability by setting $V_{\text{avl}} = V_{\text{avl}} \cup E_{i-1}$ and $V_{\text{unav}} = V_{\text{unav}} \cup (V_{\text{eff},i-1} \setminus E_{i-1})$. Thus, all v-sensors for which a reading was requested but no effective reading was received are known to be unavailable for the remainder of the sampling period. Based on this new knowledge we then compute a new selection $V_{\text{eff},i}$ using AdaptiveGreedy. A new subtask $T_i = (V_{\text{eff},i} \setminus \bigcup_{j=1}^{i-1} V_{\text{eff},j})$ is then distributed to the smartphones. We repeat this process until either the maximum number of rounds has been reached or no additional v-sensors were selected. At this point, inferred readings are computed from all effective readings that have been collected.

5 Evaluation

We evaluated our approaches based on real-world environmental measurements and generated mobility traces. In the following, we will first present the setup of our evaluation before discussing the results in detail.

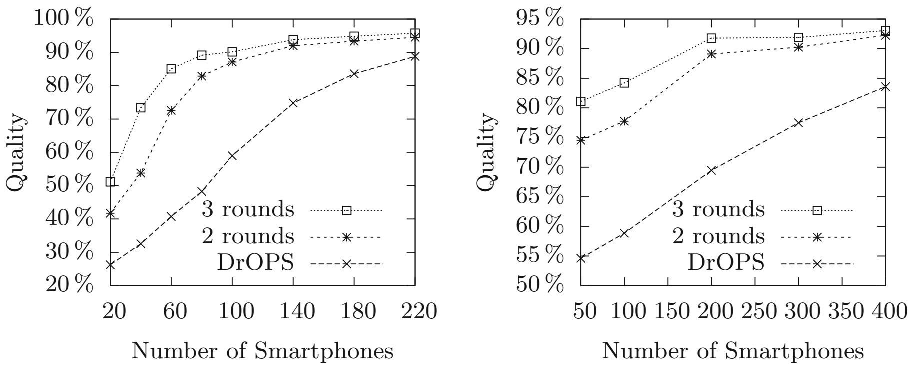
5.1 Simulation Setup

We evaluate our algorithms in a simulated PS system, implemented using Omnet++, driven by two real-world datasets containing temperature measurements: LAB data from 50 fixed sensors deployed in an indoor lab [5] and LUCE data from over 100 fixed sensors from an outdoor deployment [11]. Using real-world data readings is important to make the performance of the model-driven optimization comparable to that of a real deployment of our system, i.e., to observe realistic correlations of individual v-sensors. Queries are generated by replicating the fixed sensors of each data set as v-sensors in order to generate a temperature map of the observed area. For our PS system, we generated mobility traces for a varying number of smartphones, following the available paths in each deployment area. Energy cost is modeled using empirical energy models for communication [2] and sensing [14]. We do not consider energy for positioning, as it is amortized over other location-based applications frequently running on a smartphone. Each simulation runs for 6 simulated hours with a time offset between simulations increasing in steps of 3 h from the start of each data set. Quality parameters are set to $Q.QoS.\sigma^2_{max} = 0.1$ for the AdaptiveGreedy algorithm and $Q.T = 1\,^\circ\mathrm{C}$ as an absolute acceptable error threshold for the model validity check algorithm.

We analyze the performance of our system under three metrics: *Quality*, *Broken Queries*, and *Relative Energy Consumption*. We compare the performance of our system to a naive algorithm without optimization, i.e., $V_{\text{eff}} = V$ always, and the original DrOPS system for dense networks.

5.2 Quality

The *Quality* metric, depicted in Fig. 4, is defined as the fraction of queries in which the QoS-constraints are met out of all queries for which at least one effective reading was received, thus characterizing the data quality a client can expect from the system. Values are averaged over all simulation runs for each number of mobile smartphones.



(a) LAB data (b) LUCE data

Fig. 4. Results for quality metric. Fraction of queries in which the QoS constraints are met.

Under the DrOPS system, quality is good at just under 90 % for both datasets in a dense system, i.e., when using the maximum number of smartphones, but quickly degrades to under 60 % for 100 smartphones or less. Using our adaptive approach, the quality increases to over 90 % in a dense system. Furthermore, it is far more robust to a decreasing number of smartphones. In the LUCE data, for example, using 3 rounds we can still provide 81 % quality using 50 smartphones, whereas using DrOPS requires 400 smartphones to match this quality.

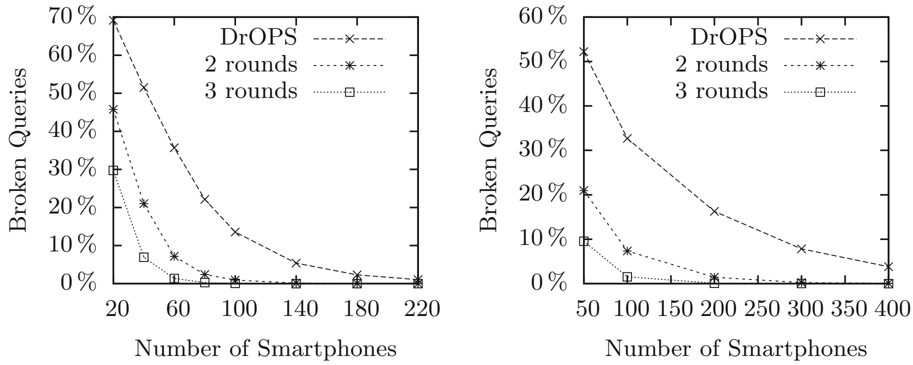
5.3 Broken Queries

Next, we analyze results for the *broken queries* metric, denoting the fraction of queries for which no effective readings were received at the gateway, i.e., characterizing how both approaches perform at finding available v-sensors.

Evaluation results are depicted in Fig. 5. Again, values are averaged over all simulation runs for each number of smartphones. Similar to the quality metric, the number of broken queries using DrOPS drastically increases for a decreasing number of smartphones, while our extended algorithm is much more robust. Under the LAB data, for a single round the fraction of broken queries increases to 5 % for 140 smartphones whereas using 3 rounds, we can provide 7 % of broken queries with only 40 smartphones. For the LUCE data, DrOPS cannot match the fraction of broken queries when using 3 rounds and at least 100 smartphones.

5.4 Relative Energy Consumption

Finally, we use the *relative energy consumption (REC)* metric to characterize the energy consumption. As the absolute energy consumption varies greatly for different time offsets, e.g., due to a varying number of sensing tasks, the REC is computed by normalizing the energy consumption for each node in a simulation



(a) LAB data (b) LUCE data

Fig. 5. Results for broken queries metric. Fraction of queries for which no effective readings were obtained.

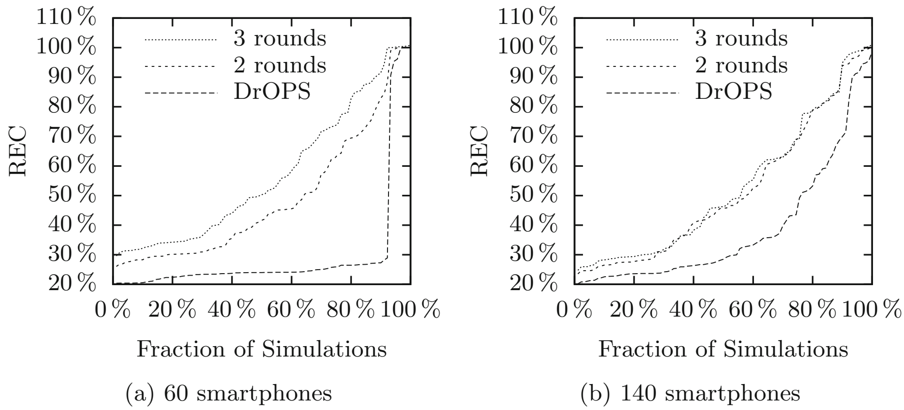


Fig. 6. Cumulated relative energy consumption, LAB data

by the average energy consumption per node using the naive algorithm for the same simulation parameters. Figure 6 shows the cumulated average REC per simulation for the LAB data. Results for the LUCE data are similar and thus not shown due to space constraints. Note that the energy drain is nearly uniformly distributed among all nodes in a simulation. The maximum difference between nodes in a simulation was 14.4 percentage points.

We see that additional communication for additional rounds increases the energy consumption. The difference is greatest in a sparse setting, where few effective readings are collected in early rounds, i.e., most work is done in later rounds. In a denser setting, the difference diminishes, as later rounds add fewer readings and thus less energy is used in later rounds. Note the sharp increase in REC for DrOPS in Fig. 6a. For about 90 % of simulations, hardly any data is collected, i.e., only few available v-sensors are found and thus little energy is spent (cf. Figs. 4a, 5a), whereas for the few cases where available v-sensors are found, only a weak model can be derived, i.e., V_{eff} is very large. As the round-based approach is better at finding available v-sensors, it does not exhibit this behavior. Using our round-based approach, we still can save up to 77 % of energy (compared to 81 % for DrOPS). When the system contains at least as many smartphones as v-sensors, energy consumption is at most that of the naive approach. When fewer smartphones are present, the round-based approach may use up to 6 % more energy.

In summary, using our round-based alternate v-sensor selection strategy will vastly improve the robustness of the system regarding a reduced number of participating devices by increasing the number of opportunities to gather data. Thus, it allows for operation in sparse networks. Even in a dense network, the quality of results returned to the client is improved. Furthermore, in a sparse network much of the energy consumed by DrOPS goes to waste, as no data is obtained for that energy. In the round-based approach, the increased energy consumption results in many more useful data readings and thus less wasted energy. Finally, robustness and result quality further increase when using additional rounds, while the energy cost for using additional rounds only increases when the additional rounds provide an actual benefit.

6 Related Work

The idea of Public Sensing (PS) has spawned a growing research interest over the last few years [3]. To address hardware and software systems challenges, several prototype architectures have been proposed, e.g., [6,7]. However, none of these systems deals with possible optimizations of the data acquisition process.

Reddy et al. analyze past mobility and participation of smartphones to recommend which devices to query in the future [15]. While this might be feasible in sparse networks, it requires long setup times and manual operator intervention.

Several works explored how to increase the efficiency of PS systems. An approach for location-centric task execution at a single v-sensor is proposed by Lu et al. [9] while in our previous work we show how to efficiently sample fixed sensors using mobile smartphones [16] and extend this idea to task execution at multiple v-sensors in parallel [12]. Furthermore, there are extensions for sampling along road segments [17] and updating road-maps [1]. Optimizations presented in these works are targeted at densely populated systems and are limited to individual v-sensors. Mendez et al. showed how a model-driven approach, a well-researched topic in fixed sensor networks [5,8], can improve result quality [10] while we presented how to use the model to optimize large-scale data acquisition in PS [13]. All of these approaches assume densely populated networks. Krause [8] presents an algorithm for selecting most informative v-sensors in the presence of unavailable v-sensors. This algorithm assumes that only a true subset of the selected v-sensors is unavailable, which does not hold in a sparse PS system.

7 Conclusion

In this work, we presented an adaptive extension for a model-driven public sensing system to enable operation in sparse networks, where most v-sensors are unavailable. Model-driven data acquisition systems can reduce the energy consumption of PS systems by requiring fewer effective readings. However, to provide sufficient result quality, a minimum number of effective readings is required. With our extended round-based v-sensor selection algorithm, we can find the required readings even when the majority of v-sensors is unavailable.

Our evaluations show that we can enable the system to work with a greatly reduced number of smartphones and that result quality is improved by up to 41 percentage points. Furthermore, we can save up to 81 % of energy for sensing and communication while providing inferred readings matching an error bound of 1 °C up to 96 % of the time.

In future work we plan to further evaluate our algorithm in a real-world deployment and to extend our approach by including a hybrid 3G/WiFi ad-hoc routing scheme to further reduce energy for communication.

Acknowledgements. This work is partially funded by the German Research Foundation under the ComNSense project (www.comnsense.de).

References

1. Baier, P., Weinschrott, H., Dürr, F., Rothermel, K.: MapCorrect: automatic correction and validation of road maps using public sensing. In: 36th Annual IEEE Conference on Local Computer Networks, Bonn, Germany, October 2011
2. Balasubramanian, N., Balasubramanian, A., Venkataramani, A.: Energy consumption in mobile phones: a measurement study and implications for network applications. In: Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement, pp. 280–293. ACM, New York (2009)
3. Campbell, A.T., Eisenman, S.B., Lane, N.D., Miluzzo, E., Peterson, R.A., Lu, H., Zheng, X., Musolesi, M., Fodor, K., Ahn, G.-S.: The rise of people-centric sensing. IEEE Internet Comput. **12**(4), 12–21 (2008)
4. Cressie, N.A.C.: Statistics for Spatial Data. Wiley-Interscience, New York (1993)
5. Deshpande, A., Guestrin, C., Madden, S.R., Hellerstein, J.M., Hong, W.: Model-driven data acquisition in sensor networks. In: Proceedings of the 30th International Conference on Very Large Databases, VLDB Endowment, pp. 588–599 (2004)
6. Hull, B., Bychkovsky, V., Zhang, Y., Chen, K., Goraczko, M., Miu, A., Shih, E., Balakrishnan, H., Madden, S.: Cartel: a distributed mobile sensor computing system. In: Proceedings of the 4th International Conference on Embedded Networked Sensor Systems, pp. 125–138. ACM, New York (2006)
7. Kanjo, E., Benford, S., Paxton, M., Chamberlain, A., Fraser, D.S., Woodgate, D., Crellin, D., Woolard, A.: MobGeoSen: facilitating personal geosensor data collection and visualization using mobile phones. Pers. Ubiquit. Comput. **12**, 599–607 (2008)
8. Krause, A.: Optimizing Sensing - Theory and Applications. Ph.D. thesis, Carnegie Mellon University (2008)
9. Lu, H., Lane, N.D., Eisenman, S.B., Campbell, A.T.: Bubble-sensing: binding sensing tasks to the physical world. Pervasive Mob. Comput. **6**(1), 58–71 (2009)
10. Mendez, D., Labrador, M., Ramachandran, K.: Data interpolation for participatory sensing systems. Pervasive Mob. Comput. **9**(1), 132–148 (2013)
11. Nadeau, D., Brutsaert, W., Parlange, M., Bou-Zeid, E., Barrenetxea, G., Couach, O., Boldi, M.-O., Selker, J., Vetterli, M.: Estimation of urban sensible heat flux using a dense wireless network of observations. Environ. Fluid Mech. **9**, 635–653 (2009)
12. Philipp, D., Dürr, F., Rothermel, K.: A sensor network abstraction for flexible public sensing systems. In: Proceedings of the 8th International IEEE Conference on Mobile Ad-hoc and Sensor Systems, pp. 460–469, October 2011
13. Philipp, D., Stachowiak, J., Alt, P., Dürr, F., Rothermel. K.: DrOPS: model-driven optimization for public sensing systems. In: IEEE International Conference on Pervasive Computing and Communications, San Diego, CA, USA, March 2013, pp. 185–192. IEEE Computer Society (2013)
14. Priyantha, B., Lymberopoulos, D., Liu, J.: Eers: Energy efficient responsive sleeping on mobile phones. In: International Workshop on Sensing for App Phones, Zurich, Switzerland (2010)
15. Reddy, S., Estrin, D., Srivastava, M.: Recruitment framework for participatory sensing data collections. In: Floréen, P., Krüger, A., Spasojevic, M. (eds.) Pervasive 2010. LNCS, vol. 6030, pp. 138–155. Springer, Heidelberg (2010)

16. Weinschrott, H., Dürr, F., Rothermel, K.: Efficient capturing of environmental data with mobile RFID readers. In: Proceedings of the 10th International Conference on Mobile Data Management, Washington, DC, USA, 2009, pp. 41–51. IEEE Computer Society (2009)
17. Weinschrott, H., Dürr, F., Rothermel, K.: StreamShaper: coordination algorithms for participatory mobile urban sensing. In: Proceedings of the 7th IEEE International Conference on Mobile Ad-hoc and Sensor Systems, San Francisco, CA, USA, November 2010. IEEE (2010)