User-Centric Incentive Design for Participatory Mobile Phone Sensing

Wei Gao and Haoyang Lu
Department of Electrical Engineering and Computer Science
University of Tennessee at Knoxville
{weigao, hlu9}@utk.edu

ABSTRACT

Mobile phone sensing is a critical underpinning of pervasive mobile computing, and is one of the key factors for improving people's quality of life in modern society via collective utilization of the on-board sensing capabilities of people's smartphones. The increasing demands for sensing services and ambient awareness in mobile environments highlight the necessity of active participation of individual mobile users in sensing tasks. User incentives for such participation have been continuously offered from an application-centric perspective, i.e., as payments from the sensing server, to compensate users' sensing costs. These payments, however, are manipulated to maximize the benefits of the sensing server, ignoring the runtime flexibility and benefits of participating users. This paper presents a novel framework of user-centric incentive design, and develops a universal sensing platform which translates heterogenous sensing tasks to a generic sensing plan specifying the task-independent requirements of sensing performance. We use this sensing plan as input to reduce three categories of sensing costs, which together cover the possible sources hindering users' participation in sensing.

1. INTRODUCTION

Smartphones, which are nowadays the key device in mobile computing and communication architecture, are programmable and equipped with a rich set of on-board sensors including GPS, accelerometer, gyroscope, microphone, camera, etc. Collective utilization of these sensors enables monitoring of humans' activities and surrounding environments, and opens the door for sensing applications in various domains, including environmental monitoring,[13,16,23,28] social interaction,[8,14,21] healthcare,[4,9,22] and transportation.[5,15,26]

These mobile phone sensing applications rely on participation of individual smartphone users. Participating users transmit their sensed data via 3G or WiFi communications to a sensing server residing in the remote cloud, in which the sensed data is accumulated, processed, and published. Most designs of mobile phone sensing assume voluntary user participation,[3,8,13] which is unrealistic in practice. A sensing task incurs a heterogeneous variety of costs, which makes a user unwilling to contribute her smartphone's local resources. We classify such sensing costs into two categories: i) *energy consumption* for generating, processing, and transmitting the sensed data, and ii) *local storage* occupied by the sensed data. Users may exhibit various levels of tolerance to different categories of sensing costs according to the specific sensing task, environmental context, and system resource conditions. Such heterogeneity hence makes it challenging to maximize user benefits and reduce their reflectance against mobile sensing tasks via modeling, analyzing, and fulfilling users' demands and preferences.

A straightforward solution to user incentive design is to offer payments from an application-centric perspective that compensate users' sensing costs.[11,29] This method, however, does not prioritize the benefits of participating users, and is incapable of effectively stimulating user participation in practice. The users' only way of receiving payments is to submit their sensing plans as bids to an auction, which is operated by the sensing server in a centralized fashion. The server takes advantage of users' competition for server's payments, and receives the sensed data with the minimum amount of payments to users. A participating user, however, is disallowed to change her sensing plan at runtime after having made a deal with the server.

In this paper, we present a novel user-centric design of user incentives in participatory mobile sensing applications. In response to the above challenges, the fundamental idea of this design is to enable a user to express her generic tolerance to sensing costs by flexibly adjusting her preferred Level of Participation in Sensing (LoPS) at runtime, and further to exploit reduction of sensing costs with respect to a user's LoPS as the user's incentive. The key insight behind this counter-intuitive design is the selfish but rational nature of human beings. A user's willingness to participate in sensing is negatively proportional to her sensing costs, and a user consents to participate in a sensing task which is cost-free even if no incentive is provided at all. The lower a user's LoPS is, the more sensing costs are saved to retain the user's participation in sensing.

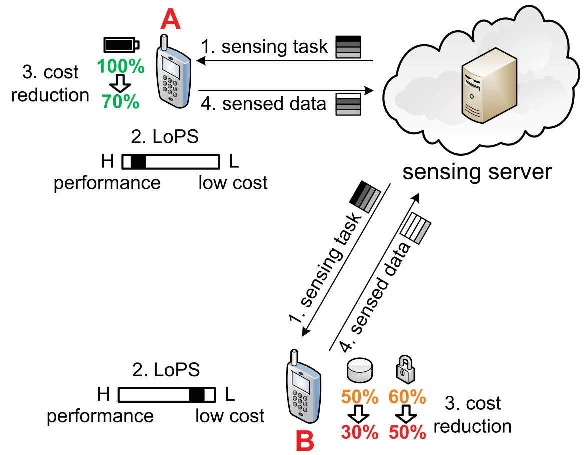

Figure 1. User-centric incentive framework for participatory mobile phone sensing

Moreover, the sensing costs of a user are closely correlated with the sensing performance provided by the user. More dedicated participation in sensing improves sensing performance, but also incurs higher sensing costs. For example, a user consumes more energy to report her current GPS location at a high frequency for better accuracy of tracking her movement. Our design will balance between a user's sensing costs and the sensing performance provided by the user, based on the user's specified LoPS.

The rest of this paper is organized as follows. Section 2 briefly describes the related work. Section 3 describes our design at a high level. Section 4 and 5 present the details of our proposed universal sensing platform and sensing incentive design. Section 6 concludes the paper.

2. RELATED WORK

A large variety of mobile systems have been developed to implement sensing functionality for environmental monitoring,[16,28] social interaction,[14,21] healthcare,[9,22] and transportation.[26,30] The sensed data was initially used to monitor the noise level[16] or city traffic conditions.[26] Later studies further employ classification and supervised learning techniques to infer from raw sensed data the users' contexts and behavior patterns,[10,17] which are used to facilitate users' social interaction.[14,21] Mobile phone sensing also aims to improve the health status of human beings.[9,18,22] However, each scheme is limited to improve the accuracy, timeliness, and reliability of its specific sensing application. A generic framework for application-independent sensing is missing, but important. Most schemes assume unlimited local resources and voluntary user participation, which are unrealistic in practice, either.

The universal sensing platform, the key component of our user-centric incentive design, is partially inspired by existing work on application-independent programming and monitoring framework for mobile phone sensing.[3,10,12] Kang et al. in[10] realized generality among sensing tasks via interaction with the remote cloud, and Das et al.[3] facilitated the development of sensing applications by encapsulating sensing software components as generic executable binaries. However, most schemes aim to improve the sensing performance or reduce the overhead of sensing system development and execution. None of them focuses on user incentive design nor considers runtime support of user flexibility in sensing tasks.

Our incentive framework is closely related to the few prior work on incentive design for mobile phone sensing.[11,29] In,[11] Lee and Hoh designed a reverse auction based incentive mechanism, where users sell their sensed data to the sensing server with users' claimed bid prices. Yang et al.[29] improved this design by ensuring the auction truthfulness. However, both schemes aim to maximize the benefits of the sensing server, ignoring users' benefits and flexibility when participating in sensing. A user's participation in sensing was also over-simplified as his/her sensing time, which is insufficient to depict the practical user behaviors in heterogeneous sensing tasks.

3. OVERVIEW

Our new design is demonstrated by the following example. In Figure 1, each user independently determines her LoPS after having received the sensing task. User A has her smartphone fully charged, and is hence willing to provide sensed

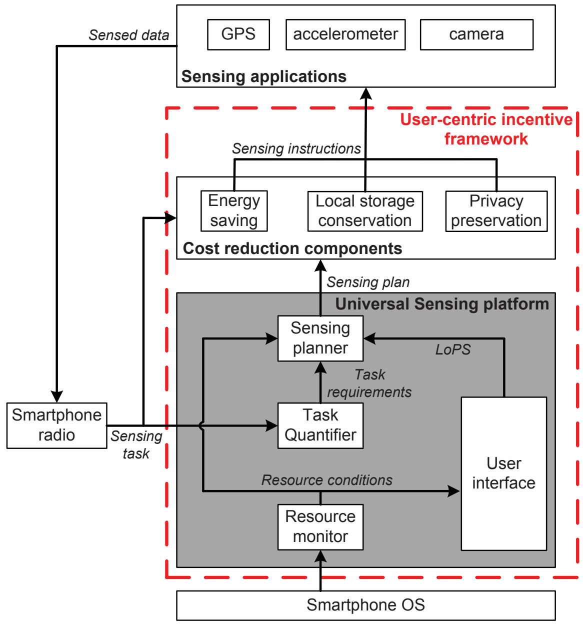

Figure 2. The architecture of user-centric incentive framework

data with high quality. In this example, our design controls the energy consumption of A's within 30%, which retains A's further participation in sensing. In contrast, user B stored a large amount of personal documents on her smartphone. B only has 50% of local storage available for caching sensed data, and expects a merely 60% chance of preventing her private personal information from being leaked in sensing tasks. B hence opts for a low LoPS. Our design reduces the sensing costs, say, to be below 20% to ensure B's normal use of her smartphone, and B is able to stop participating in sensing at any time to avoid depletion of her local resources or risk of privacy breach.

4. UNIVERSAL SENSING PLATFORM

Our incentive framework, as illustrated in Figure 2, is deployed as a middleware between the OS and sensing applications at a user's smartphone, and fundamentally alters the way sensing applications are operated. Instead of directly receiving instructions from the sensing server via the smartphone radio,[8,14,16,21] sensing applications follow the instructions from our incentive framework, which realizes users' runtime flexibility of maximizing their own benefits in sensing tasks.

The universal sensing platform, a key component of such framework, takes as input the requirements of a sensing task and a user's LoPS, and outputs a task-independent sensing plan to balance between sensing costs and performance. This sensing plan enforces a user's runtime flexibility in a sensing task by denoting the reduced requirement of sensing performance to be satisfied in practice, which is determined based on the smartphone's resource conditions and user's LoPS. It is then used as a unified instruction to reduce various categories of sensing costs.

The main challenge of designing this platform is the generality of the sensing plan over heterogeneous sensing tasks. Conventional wisdom[2,22,26,28] focuses on system design for supporting specific sensing tasks. Some schemes suggest to realize generality among sensing tasks via interaction with the remote cloud,[10] or facilitate the development of sensing applications by encapsulating sensing software components as generic executable binaries.[3] However, users' demands, preferences, and benefits are generally ignored by these schemes, which are hence inapplicable to our framework.

We tackle this challenge via generic quantification of the requirements of sensing performance specified by heterogeneous sensing tasks. The sensing plan is then generated from such generic performance requirements without taking the specifics of a sensing task into account. This sensing platform consists of the following components:

- **User interface:** It displays the current conditions of the smartphone's local resources to the user, and receives user's input about her preferred LoPS.

- **Task quantifier:** It receives the specifications of a sensing task from the sensing server, and quantifies the task's requirements of sensing performance in a generic fashion.
- **Resource monitor:** It adaptively monitors the up-to-date conditions of the smartphone's local resources corresponding to various categories of sensing costs.
- **Sensing planner:** It produces the generic sensing plan based on inputs from the above three components.

The last three components form the core of the universal sensing platform. We will describe our design of these components in more detail in the rest of this section.

4.1 Task Quantifier

Generic quantification of the requirements of sensing performance is challenging due to the heterogeneity of sensed data. For *numerical data* such as GPS[5,26] or accelerometer[4,14] readings, sensing accuracy is ensured by a sufficiently high sensing frequency. For *rich-text data* such as sound,[13,23] photo,[28] or video clip,[2] the quality of sensed data is determined by the resolution of audio/video recording.

We tackle such heterogeneity via a dimensionless metric, which measures the relative quality of sensed data that satisfies the requirement of sensing performance, compared to the maximum quality that the sensing device can provide. In particular, the actual quality of sensed data improves in a non-linear manner when the requirement of sensing performance increases, and such non-linear relationship is task-dependent. For example, the maximum sampling frequency of 3-D accelerometer equipped by iPhone 4 is 100Hz, but a sampling frequency of 50Hz will be enough for accurate characterization of the user movements in most cases. Therefore, we quantify the performance requirement of 50Hz as close to 1 instead of 50Hz/100Hz=0.5. Similarly, the maximum resolution of the front camera of iPhone 4 is 5 megapixels, but a photo taken with a resolution of 1 megapixels may be enough for recognizing the colors of clothes that people in the photo are wearing.

The key question is then how to analytically formulate such non-linear relationship. We formulate such non-linear relationship as a k-order polynomial:

$$m = f(p, \mathbf{d}) = \sum_{i=0}^{k} a_i(\mathbf{d}) p^i, \tag{1}$$

where m is the dimensionless metric, p indicates the original requirement of sensing performance, and $\mathbf{d}$ indicates the sensed data. For numerical data, $\mathbf{d}$ is a scalar representing the current sensor reading. For rich-text data, $\mathbf{d}$ is a feature vector representing data characteristics.

4.2 Resource Monitor

We then monitor the remaining percentage of smartphone battery as an indicator of energy consumption, and to take all the storage media of smartphone into account when evaluating the percentage of local storage occupation. The major challenge is that the smartphone OS only provides system APIs for such resource monitoring to user applications. For example, the Android OS broadcasts system event notifications if the phone battery or local storage is depleted, and a notification can be received by any user application via its broadcast receiver. However, such notification is difficult to reach a middleware which is deployed within the Android application framework and is not executed as an instance of the Davlik virtual machine providing the system APIs. To tackle this challenge, we develop efficient resource monitoring algorithms at the OS level. These algorithms are implemented at the Android application framework and directly interact with the Android runtime library. In contrast, Other mobile OS such as iOS or Windows mobile poses various limitations on the system customizability. For example, iOS does not provide open accessability to the OS kernel, and forbids any user application from running at background as system services.

4.3 Sensing Planner

A generic sensing plan specifies the requirement of sensing performance to be satisfied in practice with the dimensionless metric described in Section 4.1, and is calculated by a function $m' = f(m, p, r)$ where m indicates the original requirement of sensing specified by a sensing task, p is the user's LoPS, and r indicates the current conditions of system resources. Both m' and m are specified with the dimensionless metric. For example, suppose we have $m = 0.8$ for a sensing task which requires a user's smartphone to report GPS readings every second. If $r = 80\%$ and $p = 0.8$, we may have $m' = f(m, p, r) = 0.6$, which is lower than the original performance requirement (m) due to the reduced LoPS of user.

Having received such a sensing plan, the various cost reduction components, as illustrated in Figure 2, minimize the different categories of sensing costs while satisfying the performance requirement specified by m'.

We interpret the user's LoPS as her tolerance to sensing costs. In particular, for a user's LoPS $p \in [0, 1]$, $p = 1$ indicates that a user can tolerate the sensing costs corresponding to the original requirement (m) of sensing performance, and $p = 0$ indicates that a user does not allow any local resource to be consumed by a sensing task. Based on such interpretation, we plan to further substantiate the design of sensing planner in our future work by addressing the following research issues:

- **Sensing planning:** We develop a probabilistic framework for determining a sensing plan. Our basic idea is to generate m' as a random sample drawn from a population of one-sided truncated Gaussian distribution* $\mathcal{N}_t(\mu, \sigma^2; \mu)$, where $\mu = m \cdot p$ and $\sigma^2 = g(1 - r)$. When r =100%, we have $g(0) = 0$, and the value of m' is deterministically set as $m \cdot p$. Otherwise, m' is having a higher probability to be smaller than $m \cdot p$ when r is decreasing, and we plan to further analyze the impact of different forms of the function $g(\cdot)$ on the effectiveness of sensing planning.
- **Dynamic adaptation:** The sensing plan needs to be adaptive to the dynamic changes of system resource conditions, so as to avoid resource depletion. When r quickly decreases, we enable the sensing planner to automatically reduce m' before the user reacts and manually reduces her LoPS. Obviously, this adaptation can not be efficiently supported by the above probabilistic framework, in which r is only used to determine σ^2 in the probabilistic population drawing m'. Instead, we plan to devise optimization techniques which ensure that m' is appropriately adjusted to mitigate system resource consumption. When the value of r is small, reduction of r leads to more decrease of m'.

5. SENSING INCENTIVES

Built on the above universal sensing platform, in this section we present our methods of exploiting the reduction of various types of sensing costs as user incentives for participatory mobile sensing applications.

5.1 Energy Saving

A sensing task consumes the limited battery power of users' smartphones. We take as input the generic sensing plan generated by the universal sensing platform, and reduce the energy consumption of sensing with respect to the requirements of sensing performance specified by the sensing plan.

The key insight behind energy saving is that users' sensor readings are correlated in time and such correlation indicates redundancy among sensed data. Some sensor readings are unnecessary and can be omitted to reduce the energy consumption of sensing. For example, the current GPS location of a user is correlated with the user's GPS locations in the past. We exploit the temporal correlation between sensor readings of a user in different times, so as to adaptively reduce the user's sensing frequency. The requirement of sensing performance specified by the generic sensing plan is satisfied by controlling the amount of redundancy being eliminated.

Our approach to such energy saving is based on our previous work on Hidden Markov Model (HMM) formulations.[6] We assume that a user obtains sensor readings s_t in slotted time t. The temporal variation of $c_t = f(s_{t-1}, s_t)$ will be formulated based on HMMs. Being different from a normal Markov process which consists of a discrete state space $\mathbf{S} = \{s_1, s_2, ..., s_N\}$, a state transition probability matrix $\mathbf{A} = \{a_{ij}\} \in \mathbb{R}^{N \times N}$, and an initial state distribution $\mathbf{\Pi} = \{\pi_i\}$, a HMM hides its states behind a set of observation PDFs $\mathbf{B} = \{b_i(x)\}$ where each $b_i(x)$ is associated with a state s_i. A HMM $\mathcal{H} = (\mathbf{S}, \mathbf{A}, \mathbf{B}, \mathbf{\Pi})$ calculates the occurrence probability of an observation sequence $\mathbf{O} = o_1 o_2 ... o_L$ with the state sequence $\mathbf{I} = i_1 i_2 ... i_L$ as $\mathbb{P}(\mathbf{O}|\mathbf{I}, \mathcal{H}) = \prod_{k=1}^{L} b_{i_k}(o_k)$.

We bridge the gap between continuous values of c_t and discrete Markovian states using the observation PDFs of HMMs, and to represent the variation of c_t using a number ($N = |\mathbf{S}|$) of "correlation stages". The k-th correlation stage indicates a specific value range of $c_t \in [(k-1)/N, k/N]$, and is associated to a Markovian state s_k via the corresponding observation PDF $b_k(x)$. Variations of c_t are hence described in the form of transitions among Markovian states. We use Gaussian

*Suppose $X \sim \mathcal{N}(\mu, \sigma^2)$ has a Gaussian distribution and lies within the interval $(-\infty, T]$, then X conditional on $X \leq T$ has an one-sided truncated Gaussian distribution $\mathcal{N}_t(\mu, \sigma^2; T)$.

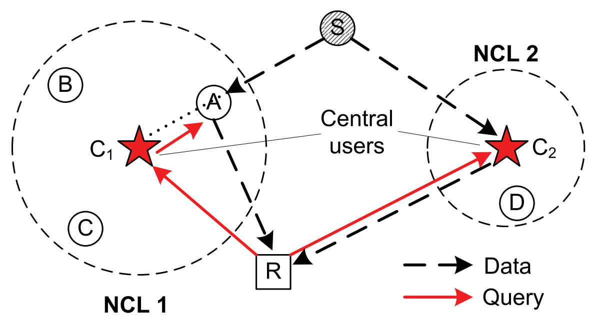

Figure 3. Intentional caching based on Network Central Locations (NCLs)

distribution as the form of observation PDFs and characterize the variation of c_t by adaptively re-estimating the parameters of observation PDFs at runtime.

The reduction of sensing cost then builds on predictions of temporal correlation between sensor readings on both steady-state and transient-state time scales, being supported by the above HMM formulation. First, steady-state prediction estimates the cumulative correlation between sensor readings over a long period of time in the future as

$$c_{avg} = \sum_{i=1}^{N} \psi_i \cdot \mu_i, \tag{2}$$

where $\psi_i = \mathbb{P}(s_i|\mathcal{H})$ is the stationary distribution of a HMM $\mathcal{H} = (\mathbf{S}, \mathbf{A}, \mathbf{B}, \mathbf{\Pi})$, and μ_i is the mean value of the observation PDF $b_i(x) \in \mathbf{B}$. Second, transient-state prediction estimates the correlation c_{t+1} in the next time slot in the future. Suppose the most recent sequence of observed correlations of $\mathcal{H}$ is $\mathbf{C} = c_1 c_2 ... c_L$, we are able to find the best state sequence $\mathbf{I} = i_1 i_2 ... i_L$ which maximizes $\mathbb{P}(\mathbf{O}|\mathbf{I}, \mathcal{H})$, using the Viterbi algorithm.[20] c_{t+1} is then estimated by a probabilistic distribution as

$$p_{t+1}(x) = \sum_{j=1}^{N} \mathbb{P}(q_{t+1} = s_j | q_t = i_L) \cdot b_j(x), \tag{3}$$

where q_t and q_{t+1} indicate the current state of $\mathcal{H}$ at time t and $t+1$, respectively.

5.2 Local Storage Conversation

The next important category of sensing costs is the local storage space of users' smartphones that is occupied to locally cache the sensed data before sending the data to the sensing server. A large variety of sensing applications require peer-to-peer collaboration among sensing users, who cooperatively access the sensed data being cached from each other for better sensing efficiency[12,24] or more complicated sensing objectives such as environmental context inference.[1,2] Sensed data can be pushed by the data source to its caching locations, from where the data can be pulled (queried) by other users with less delay.

The key insight behind conservation of smartphones' local storage is that users involved in collaborative sensing exhibit skewed patterns of accessing the sensed data cached by each other. Only a small portion of popular data is frequently accessed, and can be intentionally cached to ensure timely response to user queries. For example, sensor readings are exchanged among users for inferring their environmental contexts[19,24,25,27] or social relationship with each other.[8,12] In these scenarios, the sensed data from few users, such as a user passing the points of interest in urban sensing or an active participant in a social party, is likely to be more frequently accessed. We develop cooperative caching techniques which build on such skewness of data access patterns, and to analytically answer the questions of i) **where to cache**, ii) **how many to cache**.

We develop distributed methods which dynamically determine the most appropriate locations for caching sensed data, so as to answer the above questions. Built on our previous work,[7] our basic idea is to incorporate data popularity into NCL selection as shown in Figure 3, so as to determine the caching locations for each data item according to its popularity. The more popular a data item is, the more copies of this data item are cached at different NCLs to satisfy the requirements of sensing performance.

Distributed caching framework: As a prerequisite, we develop a practical caching framework enabling distributed NCL selection among users themselves with local network knowledge. Each user autonomously calculates her own centrality and broadcasts this value to others. After a specific broadcasting period, each user autonomously selects the users with the K highest centrality values as the central users representing NCLs.

The key problem of distributed NCL selection is the possible inconsistency of NCLs selected by different users, due to the uncertainty of opportunistic data transmission and the subsequent heterogeneity of local network knowledge available at different users. Assuming that the information of a user C with high centrality has a probability p to be received by user A but not B, there is a probability no larger than $1-(1-2p(1-p))^K$ for A and B to have inconsistent NCL selections.

Data-specific caching locations: Instead of employing a fixed number (K) of NCLs, we adaptively vary K for each data item according to its popularity, by developing a data-specific NCL selection metric and taking both data popularity and users' LoPS into account. First, we enable a central user to calculate the popularity of a data item as the probability that this data item will be queried again in the future, based on the past k queries to this data item it has received. Second, the more popular a data item is, the more NCLs are selected for caching this data, and data with high popularity is prioritized to be cached at users with high centrality.

In our design, users need to coordinate with each other in a fully distributed fashion, so as to estimate the current amount of data copies being cached and make their own caching decisions. Such distributed coordination, however, is challenging due to the opportunistic connectivity between caching users. We address this challenge by developing a probabilistic framework. Our basic idea is to first assume homogeneous data access patterns in a global network scope, and then develop statistical models to emulate the temporal and spatial randomness of such patterns. Based on such models, each user independently estimates the number of data copies cached in the network. This estimation is further amended every time who two caching users contact, during which the caching locations are adaptively adjusted.

Workload balancing: Central users may quickly consume their local storage and are hence unwilling to continue their participation into sensing. First, such caching workload has been implicitly balanced among central users by our proposed data-specific NCL selection, which decreases the centrality of a user with a low LoPS. Second, we adaptively migrate the functionality of a central user to another user in the network. In practice, an existing central user C is responsible for selecting the new central user C' when C is low in storage. This migration will happen when C contacts C' and transfers its cached data to C'.

6. CONCLUSIONS

This paper presents a design of an incentive framework for participatory mobile phone sensing from a user-centric perspective, which motivates users to participate in sensing tasks by granting users the runtime flexibility of maximizing their own benefits. Such new design concept makes a fundamental shift from conventional payment-based user incentives offered from an application-centric perspective, and completely removes the control of the sensing server over users. It attempts to offer incentives as users' runtime flexibility of adjusting their participation in sensing and reducing their sensing costs. In the future, we plan to further perform extensive experiments evaluating the effectiveness of such user incentive design in realistic mobile sensing applications with involvement of actual mobile users.

REFERENCES

[1] X. Bao and R. Choudhury. Vupoints: Collaborative sensing and video recording through mobile phones. *ACM SIGCOMM Computer Communication Review*, 40(1):100–105, 2010.

[2] X. Bao and R. Roy Choudhury. Movi: Mobile phone based video highlights via collaborative sensing. In *Proceedings of the 8th Int'l Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 357–370, 2010.

[3] T. Das, P. Mohan, V. N. Padmanabhan, R. Ramjee, and A. Sharma. Prism: Platform for remote sensing using smartphones. In *Proceedings of the 8th Int'l Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 63–76, 2010.

[4] R. de Oliveira and N. Oliver. Triplebeat: Enhancing exercise performance with persuasion. In *Proceedings of the 10th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI)*, pages 255–264, 2008.

[5] R. Ganti, N. Pham, H. Ahmadi, S. Nangia, and T. Abdelzaher. Greengps: A participatory sensing fuel-efficient maps application. In *Proceedings of the 8th Int'l Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 151–164, 2010.

[6] W. Gao and G. Cao. Fine-Grained Mobility Characterization: Steady and Transient State Behaviors. In *Proceedings the 11th ACM Interational Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 61–70, 2010.

[7] W. Gao, G. Cao, A. Iyengar, and M. Srivatsa. Supporting cooperative caching in disruption tolerant networks. In *Proceedings of International Conference on Distributed Computing Systems (ICDCS)*, 2011.

[8] S. Gaonkar, J. Li, R. R. Choudhury, L. Cox, and A. Schmidt. Micro-blog: Sharing and querying content through mobile phones and social participation. In *Proceeding of the 6th International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 174–186, 2008.

[9] P. Jarvinen, T. Jarvinen, L. Lahteenmaki, and C. Sodergard. Hyperfit: Hybrid media in personal nutrition and exercise management. In *Proceedings of 2nd International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth)*, pages 222–226, 2008.

[10] S. Kang, J. Lee, H. Jang, H. Lee, Y. Lee, S. Park, T. Park, and J. Song. Seemon: Scalable and energy-efficient context monitoring framework for sensor-rich mobile environments. In *Proceedings of the 6th Int'l Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 267–280, 2008.

[11] J. Lee and B. Hoh. Sell your experiences: A market mechanism based incentive for participatory sensing. In *Proceedings of IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 60–68, 2010.

[12] Y. Lee, Y. Ju, C. Min, S. Kang, I. Hwang, and J. Song. Comon: Cooperative ambience monitoring platform with continuity and benefit awareness. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 43–56, 2012.

[13] H. Lu, W. Pan, N. Lane, T. Choudhury, and A. T. Campbell. Soundsense: Scalable sound sensing for people-centric applications on mobile phones. In *Proceedings of the 7th Int'l Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 165–178, 2009.

[14] E. Miluzzo, N. D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. B. Eisenman, X. Zheng, and A. T. Campbell. Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application. In *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems (SenSys)*, pages 337–350. ACM, 2008.

[15] P. Mohan, V. Padmanabhan, and R. Ramjee. Nericell: Rich monitoring of road and traffic conditions using mobile smartphones. In *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems (SenSys)*, pages 323–336, 2008.

[16] M. Mun, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, R. West, and P. Boda. Peir, the personal environmental impact report, as a platform for participatory sensing systems research. In *Proceedings of the 7th Int'l Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 55–68, 2009.

[17] S. Nath. Ace: Exploiting correlation for energy-efficient and continuous context sensing. In *Proceedings of the 10th Int'l Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 29–42, 2012.

[18] S. Nirjon, R. Dickerson, Q. Li, P. Asare, J. Stankovic, D. Hong, B. Zhang, X. Jiang, G. Shen, and F. Zhao. Musicalheart: A hearty way of listening to music. In *Proceedings of the 10th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2012.

[19] M. R. Ra, B. Liu, T. F. La Porta, and R. Govindan. Medusa: A programming framework for crowd-sensing applications. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 337–350, 2012.

[20] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of IEEE*, 77(2):257–286, 1989.

[21] K. Rachuri, C. Mascolo, M. Musolesi, and P. . Rentfrow. Sociablesense: Exploring the trade-offs of adaptive sampling and computation offloading for social sensing. In *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 73–84, 2011.

[22] K. K. Rachuri, M. Musolesi, C. Mascolo, P. J. Rentfrow, C. Longworth, and A. Aucinas. Emotionsense: A mobile phones based adaptive platform for experimental social psychology research. In *Proceedings of the 12th ACM International Conference on Ubiquitous Computing (UbiComp)*, pages 281–290, 2010.

[23] R. Rana, C. Chou, S. Kanhere, N. Bulusu, and W. Hu. Ear-phone: An end-to-end participatory urban noise mapping system. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 105–116, 2010.

[24] X. Sheng, J. Tang, and W. Zhang. Energy-efficient collaborative sensing with mobile phones. In *Proceedings of IEEE INFOCOM*, pages 1916–1924, 2012.

[25] N. Thepvilojanapong, S. Konomi, Y. Tobe, Y. Ohta, M. Iwai, and K. Sezaki. Opportunistic collaboration in participatory sensing environments. In *Proceedings of the 5th ACM International Workshop on Mobility in the Evolving Internet Architecture (MobiArch)*, pages 39–44, 2010.

[26] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson. Vtrack: Accurate, energy-aware road traffic delay estimation using mobile phones. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, pages 85–98, 2009.

[27] H. Weinschrott, F. Durr, and K. Rothermel. Streamshaper: Coordination algorithms for participatory mobile urban sensing. In *Proceedings of the 7th IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS)*, pages 195–204, 2010.

[28] T. Yan, V. Kumar, and D. Ganesan. Crowdsearch: Exploiting crowds for accurate real-time image search on mobile phones. In *Proceedings of the 8th Int'l Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 77–90, 2010.

[29] D. Yang, G. Xue, X. Fang, and J. Tang. Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing. In *Proceedings of the 18th ACM International Conference on Mobile Computing and Networking (MobiCom)*, 2012.

[30] J. Yang, S. Sidhom, G. Chandrasekaran, T. Vu, H. Liu, N. Cecan, Y. Chen, M. Gruteser, and R. Martin. Detecting driver phone use leveraging car speakers. In *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 97–108, 2011.