9-2013

ConferenceSense: Monitoring of public events using phone sensors

Vigneshwaran SUBBARAJU
Singapore Management University, vigneshwaran@smu.edu.sg

Amit KUMAR
IIIT, Delhi

Vikrant Nandakumar
IBM Research, India

Sonali Batra
State University of New York, Buffalo

Salil Kanhere
University of New South Wales

See next page for additional authors

DOI: https://doi.org/10.1145/2494091.2499775

Citation

SUBBARAJU, Vigneshwaran; KUMAR, Amit; Nandakumar, Vikrant; Batra, Sonali; Kanhere, Salil; De, Pradipta; Naik, Vinayak; Chakraborty, Dipanjan; and MISRA, Archan. ConferenceSense: Monitoring of public events using phone sensors. (2013). *UbiComp'13 Adjunct: Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing, Zurich, Switzerland, September 8-12*. 1167-1174. Research Collection School Of Information Systems.
Available at: https://ink.library.smu.edu.sg/sis_research/1959

Author
Vigneshwaran SUBBARAJU, Amit KUMAR, Vikrant Nandakumar, Sonali Batra, Salil Kanhere, Pradipta De, Vinayak Naik, Dipanjan Chakraborty, and Archan MISRA

ConferenceSense: A Case Study of Sensing Public Gatherings using Participatory Smartphones

S.Vigneshwaran
Singapore Management University
vigneshwaran@smu.edu.sg

Pradipta De
SUNY Korea
pradipta.de@sunykorea.ac.kr

Amit Kumar
Indraprashtha Institute of Information Technology
Delhi, India
amit1262@iiitd.ac.in

Vinayak Naik
Indraprashtha Institute of Information Technology
Delhi
naik@iiitd.ac.in

Vikrant Nandakumar
IBM Research India
vikrant.nandakumar@in.ibm.com

Dipanjan Chakraborty
IBM Reseach India
cdipanjan@in.ibm.com

Sonali Batra
State University of Newyork
Buffalo

Archan Misra
Singapore Management University
archanm@smu.edu.sg

Salil Kanhere
University of New South Wales
salilk@cse.unsw.edu.au

UbiComp '13 Adjunct, Sept 8-12, 2013, Zurich, Switzerland.
ACM 978-1-4503-2139-6/13/09...$15.00.

Abstract

We explore the use of a participatory sensing paradigm, where data generated from individual smartphones is used to extract and understand collective properties of temporary public gatherings and events (e.g., concerts & conferences). We focus on the use of this paradigm at a technical conference, and describe the design, implementation and deployment of *ConferenceSense*, an application that uses multiple sensor and human-generated inputs from attendees' smartphones to infer context, such as the start time of a session or the degree of interaction during a tea break. Based on data collected from multiple attendees at a 3-day conference, we explore how ConferenceSense can be used for monitoring and collecting event statistics, and describe challenges and open questions.

Introduction

Community-driven or *participatory* sensing is an established sensing paradigm [1,2], where individual users sense and share data related to their individual environment. This enables the collective discovery of environmental state, patterns or anomalies. Most prior research in this area has focused on applications such as pollution, traffic and road condition or public health monitoring [3]. In such scenarios, there is (a) a focus on collecting information from a specific sensor that is suitable for the associated application (e.g., accelerometer and GPS for pothole monitoring), and (b) an implicit assumption that the underlying data is generated solely from voluntary, uncoordinated reports provided by individuals.

In this paper, we focus on a new category of applications that utilize such community-oriented sensing: detecting and understanding attributes of temporary public gatherings, such as trade shows, music concerts and academic conferences. Our application space differs from the previously mentioned examples of participatory sensing in two ways:

(a) Our focus is on trying to derive various *high-level* attributes of the event (e.g., how well did attendees interact with one another or which conference talks failed to capture the audience attention?) that are impacted by the crowd, as opposed to attributes of the physical world (e.g., is the room cold?).

b) The quality of the participatory sensing is affected by the localized movement, behavior of individuals in indoor, public spaces, in contrast to prior work that looks at outdoor or city-scale deployments.

We believe that such a smartphone-based, community-driven approach can be a compelling way for sensing temporary public events, as the alternative approach of extensive infrastructure instrumentation is not only expensive, but largely unworkable from a deployment perspective. More specifically, this work describes our experiences with building and deploying a prototype community-sensing based application, called *ConferenceSense*, over participants at a representative technical academic conference (IEEE MDM). We emphasize that this work is preliminary, and principally reports us an understanding of the issues involved in utilizing such a participatory platform (what worked and what did not?), rather than providing a definitive solution architecture. ConferenceSense combines both implicit (sensor-generated) and explicit (human entered) inputs to try and predict various properties of interest (e.g., which sessions/events attracted highest participation, who are the participants with whom the other attendees have the most interaction, when did sessions actually start and end, etc.). We also describe the nature of data gathered from voluntary participants, and discuss take-aways and recommendations for employing such a sensing paradigm, given the behavioral characteristics of these participants. Generally, our study with the ConferenceSense prototype seeks to answer the following key research questions: (1) What types of data should ConferenceSense gather, and how willing are participants to share the data from various phone sensors? (2) What insights/context could we reliably extract from the gathered data, and why did certain forms of context prove difficult to infer? (3) What enhancements and capabilities would be needed, in future versions of ConferenceSense?

Related Work

Current Mobile Crowd-sensing applications (MCS) [1] are divided into three classes *Environmental*, *Infrastructure* and *Social*, referring to the type of phenomena being sensed. Our work falls in the social MCS category. Prior works here [4] typically compute and aggregate similar context across individuals (e.g., recording one's exercise routine and comparing it with that of the overall community) and aim to provide individual-level feedback. In contrast, our focus is on detecting the *aggregate context* of the event, by fusing possible disparate context from different individuals. As such, architecting general and special-purpose crowdsensing platforms [5] for effective data collection from the community is a new area of research.

Effective participatory sensing must address challenges related to both privacy and resource-usage; specifically, the energy of the devices and the network bandwidth. The problem of energy consumption during mobile sensing is addressed via approaches such as [6], which achieved a significant reduction in energy overheads by intelligently duty-cycling the relevant sensors.

There has been a few prior works on collaborative sensor fusion from smartphones. For example, the Darwin phone [7] performs collaborative sensing for applications such as audio fingerprinting, pollution monitoring and radio fingerprinting. Darwin uses an evolving classifier that continuously tunes the received data from various users so that it is robust to environment changes in sensing the same phenomena. However, Darwin is designed primarily for extracting environmental context, rather that high-level social context. The Tagsense application [8] uses collaborative sensing to tag pictures based on events extracted from smartphones of surrounding users. While inferring such context, Tagsense employs a direct phone-to-phone communication paradigm to share such context among neighboring phones. In contrast, ConferenceSense focuses on extracting insights that may require the fusion of sensor data from nodes from non-neighboring nodes as well, and thus employs a more server-coordinated infrastructure.

System Overview

In this section, we describe the system architecture of ConferenceSense. ConferenceSense is designed as a client-server application, where the client side runs as an Android application. The client side running on the user's mobile device collects the sensor data and uploads the data to the server. The server is designed to collect and store the data uploaded by multiple users, and perform real time analytics and query processing. The data channel between the client app and the server is typically the Wi-Fi connection provided free of cost at any event site. Thus users are not required to consume bandwidth from their metered data plans. Figure 1 gives an overview of the design.

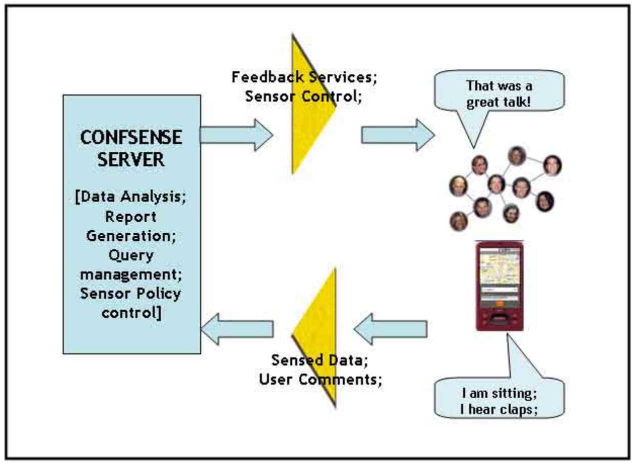

Figure 1: Overview of the ConferenceSense system for Event Monitoring

The Android application at the client side has two key functions: (a) collect raw sensor data from accelerometer, Wi-Fi, Bluetooth, and microphone (b) periodically upload the data to the server such that client side storage is not overused. It is also possible for the user to submit asynchronous feedback on any event, which effectively acts as tags for events. Fig. 2 shows the interface to collect user feedback. While collecting sensor data, it is important to ensure that the device does not run out of battery during the duration of the conference. The app is designed to trigger the data collection only when it detects that it is within the conference venue, indicated by the detection of the pre-defined conference Wi-Fi SSID. The server is placed within the Wi-Fi network Local Area Network (LAN) and collects data from clients. A rolling log with a fixed size is used to collect the sensor data such that there is always a bound on the local storage used.

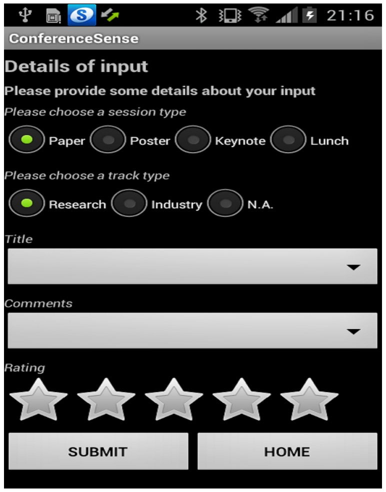

Figure 2: Reporting screen to send user reviews.

The ConferenceSense server acts as (a) the data repository, and (b) the policy manager that directs the data collection on each device. The raw data sent from each user is uniquely identified by the device IMEI number, and stored in a database. The analysis module processes the data to push information of interest to the devices. The policy manager is invoked when a device registers for the first time with the server. The policy manager specifies the default duty cycle of each sensor. If the battery level on any device drops below a threshold, the information is sent to the server. The server reacts by reinitializing the duty cycles to reduce the energy consumption. Another challenge is that the collected data must be time synchronized. The start of the event is considered as the start time, and pushed to each node at initialization. The timestamp on each data point collected is relative to the start time of the conference, thus synchronizing all data points across multiple devices.

The data collected from the participants are the raw accelerometer readings along x, y and z directions, the SSIDs, BSSIDs of the visible access points along with their signal strengths, the names of the visible Bluetooth devices, battery levels, the audio data from the microphone in AMR format and finally the asynchronous user comments and feedback from the reporting screen.

Evaluation: Sensing pulses of the Event and the Crowd

Next, we focus on analyzing the data to understand what our experiments suggest towards extracting attributes of the event via community sensing. First, we provide an overview of the user recruitment and interaction process.

System Deployment and User Recruitment

In this section, we describe the system deployment and how the campaign was socialized to recruit participants for the study. We had a table set up at the entrance with “ConferenceSense”written in a small table-top triangular shaped banner. We could have prepared a big banner but we did not want to make the sensing appear as one of the central items in the conference. We believe that sensing should be by participation and better done passively.

The *ConferenceSense* volunteers personally approached the conference participants to explain them about this sensing effort and also ran a quick walkthrough of the application using our demo smartphone. The helpdesk volunteers assisted the interested users to understand the application and made them aware of how they could start sensing while they are at the conference location. After obtaining the consent from the user, the installation of the application was performed on their smartphone. In order to reiterate the primary motive behind participatory sensing, the participants were given an option to switch on only those sensors, from which they were willing to share data for the sensing exercise.

To start with, around 26% of the participants had concerns regarding their battery consumption while using the application. The volunteers kept these participants informed about the results of the battery drain tests, which were conducted prior to this deployment. In addition to this, the volunteers also informed the participants to reach out to the helpdesk if they face any issues with respect to the application. 66% of the participating users was not comfortable to turn the microphone sensor ON, although they were aware of the fact that it collects only discrete samples of audio. 80% of the users was unwilling to provide user-generated feedback for the different sessions of the conference. The *ConferenceSense* volunteers had to constantly remind them to submit updates from the user input screen of the application. Unfortunately, about 3% of the enthusiastic participants could not sign up successfully because, the application was incompatible with their phones (the application is compatible on phones running Android OS v2.2 and above).

One of the major issues faced during the sensing exercise was to handle the intermittent Wi-Fi network connectivity at the conference location. The location staff helped us in placing the wireless routers at different designated points in the room. Unfortunately, due to inadequate plug points in the conference rooms, few attendees unplugged the power supply to the wireless routers and started to charge their draining laptop batteries. In addition to the Wi-Fi issue, the conference venue’s network infrastructure was unstable due to unexpected overload in the network. This eventually led to the back-end server being inaccessible for a few times during the first two days.

A total of 16 users contributed with data. We observed that power users are present in this form of crowdsensing. Power users are users who contribute a significant fraction of the total data. Next, we discuss results from a preliminary analysis of this data to recover potentially important and useful patterns.

Bluetooth Encounters

Every Bluetooth device logs information about its Bluetooth neighbors periodically. This information can be used for computing unique Bluetooth encounters for a device ($encounters_{bt}$), i.e., all other Bluetooth devices that it has seen at any point of time. A particular device might see the same device multiple times, but to keep the encounters unique it will be treated as only one encounter. Figure 3 includes only those devices that have $encounters_{bt} \geq 3$, across all conference days. The encounter information (time, volume) can be used to model dynamism of the social vicinity and infer attributes like whether the vicinity is static or dynamic.

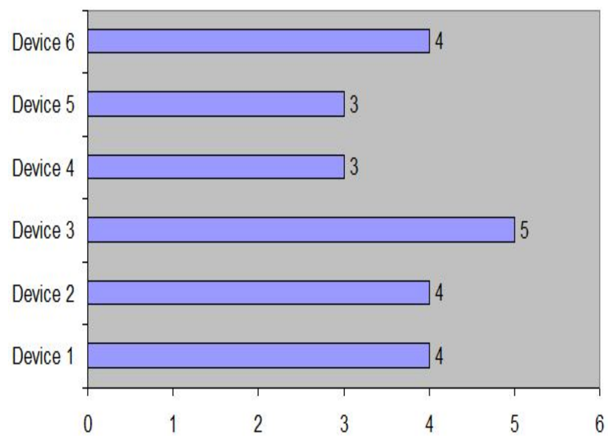

Figure 3: Bluetooth Encounters: Y-axis represents a Bluetooth device and X-axis represents the total number of unique Bluetooth devices encountered by the corresponding Bluetooth device.

Temporal Participation Statistics

In this subsection, we map the data collected from multiple sensors on a particular day with the schedule of the conference on that day. Figure 4 plots the number of distinct users, who have contributed to any type of data (bluetooth, wifi, accelerometer, and user input) with respect to the activity slots in the conference schedule. First four bars in a time slot represent users contributing a particular type of data and the bar reading *Total Users* in every time slot is representing *union* of all users corresponding to other four bars in the same time slot. It is clear the participation varies with time slots, implying that quality of inference of attributes will also get affected. This also indicates that, for reliable extraction of event attributes, the data collection needs to be programmed to maximize quality of data being collected for the attributes of importance.

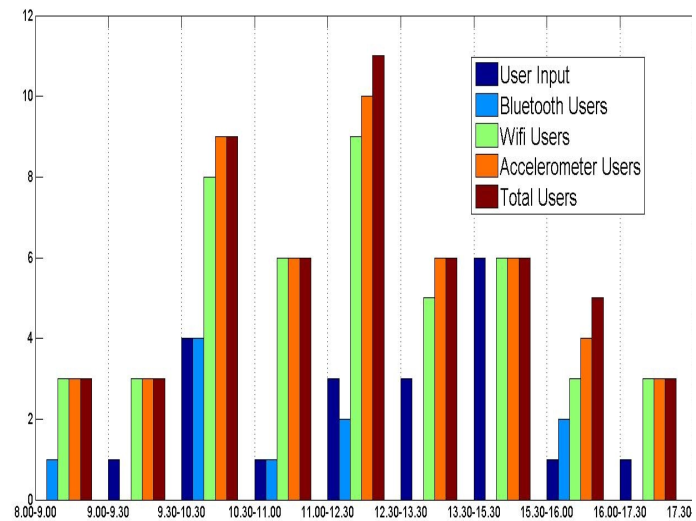

Figure 4: Temporal Participation Statistics: X-axis represents the scheduled activity slots in the conference and Y-axis represents the distinct number of users, who have contributed with data. Plot represents data for a day in the conference.

Movement Statistics

While the users roam around during the day, i.e., attending sessions, workshops, tea breaks, and lunch breaks, the accelerometer of the phone logs their locomotions, as acceleration components along the 3-axes of the smartphone (x, y, z). Figure 5 represents the magnitude of acceleration ($\sqrt{x^2 + y^2 + z^2}$) averaged over all users for each activity slot in the conference schedule. Since the accelerometer always has a reading in Z-axis due to gravity, the minimum acceleration over all the days is near to 9. This aggregate statistic gives an indicator of average movement levels of the users.

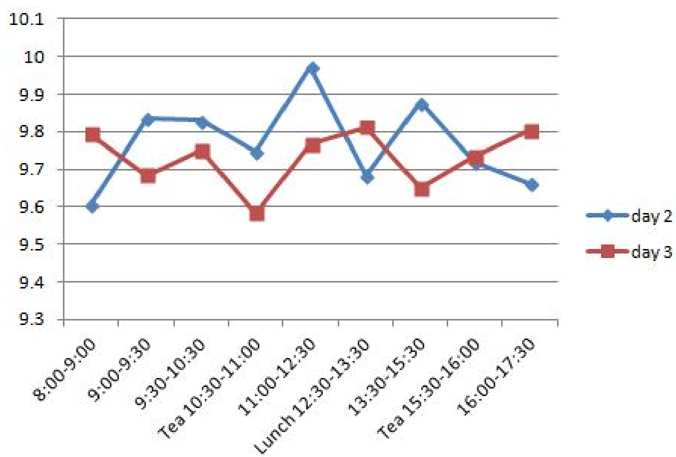

Figure 5: Movement statistics: Y-axis represents the average magnitude of acceleration of all users and X-axis represents the scheduled slots of the conference.

As can be seen from the graph, on day 3 more movement is observed during the lunch time as compared to day 2; while at the tea break before lunch time, day 2 depicts more movement than day 3. However, the quality of this statistic depends on the volume and diversity of the movement patterns. We are currently investigating whether such data can reveal event attributes like “session X: users trickling in”. This requires an information-theoretic view of mining and fusing data from multiple users.

Take-Aways

This section summarizes our initial take-aways from the analysis we have conducted.

Central Control: While collecting crowdsensed data, we believe that a larger control has to be exercised by the central server. This is necessary to accommodate data quality variations across the devices, preferences of the participants, and the nature of the attributes of importance. When we have centralized control, we can influence the individual sensing behavior, while being able to track the global state and quality of data being submitted. We need to focus on two questions: (a) which sensors are relevant to a particular participant or at a particular time and (b) what are the fidelity of the data needed for the query that needs to be processed.

Human Factors: There are human factors, which make quality of the sensed data, time, space and activity dependent. For example, some participants are more concerned about the power consumed, while some others are concerned about their privacy. Moreover, different users have different roles. Students would have a certain sense of engagement while a conference organizer or a professor might have different objectives, which would guide their individual movement patterns. The mixture of community participating in the collection plays a role in detecting event attributes reliably. In spite of all this, our study indicates that it might be possible to infer basic parameters of an event, such as mobility characteristics and interaction characteristics of the users, if participation can be programmed to collect high fidelity data.

Personalization: While conducting the campaign, it became apparent that the sensing policies have no personalization to reflect individual user's concerns. We need to investigate the implications in applying a personalized policy for the participants, considering their individual behavior. Further, in order to improve the reliability, we need to incorporate quality guarantees on query responses as well as control the community sensor network to minimize resource usage by changing the data fidelity depending on the query being processed.

Conclusions and Future Work

The implications of applying the community sensing paradigm to sense large events like conferences are explored in this work with the help of a simple setup that was deployed at a technical conference. Our preliminary analysis was oriented towards understanding movement statistics and the distribution of data submitted by users. We find that, while it might be possible to infer some basic attributes of the event, a cloud coordinated approach with a personalized policy framework can improve both reliability of inference, and the user experience. Our future work is oriented towards developing reliable algorithms to estimate key performance indicators of such events, from crowdsensed data, and by detecting and controlling the collection of high value data.

References

[1] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges." *IEEE Communications Magazine*, no. 11, pp. 32–39, 2011.

[2] D. Christin, A. Reinhardt, S. S. Kanhere, and M. Hollick, "A survey on privacy in mobile participatory sensing applications," *Journal of Systems and Software*, vol. 84, no. 11, pp. 1928–1946, 2011.

[3] K. Aberer, S. Sathe, D. Chakraborty, A. Martinoli, G. Barrenetxea, B. Faltings, and L. Thiele, "Opensense: Open community driven sensing of environment," in *Proceedings of the ACM SIGSPATIAL International Workshop on GeoStreaming*. ACM, 2010, pp. 39–42.

[4] H. Lu, J. Yang, Z. Liu, N. D. Lane, T. Choudhury, and A. T. Campbell, "The jigsaw continuous sensing engine for mobile phone applications," in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2010, pp. 71–84.

[5] 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*. ACM, 2012, pp. 337–350.

[6] Y. Wang, J. Lin, M. Annavaram, Q. A. Jacobson, J. Hong, B. Krishnamachari, and N. Sadeh, "A framework of energy efficient mobile sensing for automatic user state recognition," in *Proceedings of the 7th international conference on Mobile systems, applications, and services*, ser. MobiSys '09, 2009.

[7] E. Miluzzo, C. T. Cornelius, A. Ramaswamy, T. Choudhury, Z. Liu, and A. T. Campbell, "Darwin phones: the evolution of sensing and inference on mobile phones," in *Proceedings of the 8th international conference on Mobile systems, applications, and services*, ser. MobiSys '10, 2010.

[8] C. Qin, X. Bao, R. Roy Choudhury, and S. Nelakuditi, "Tagsense: a smartphone-based approach to automatic image tagging," in *Proceedings of the 9th international conference on Mobile systems, applications, and services*, ser. MobiSys '11, 2011.