

Demo: AD-Sense—Activity-driven Sensing for Mobile Devices

Jian Cui

CS Dept., Tsinghua University
TNLIST, Tsinghua University
Beijing, 100084, China
cuijian.hh@gmail.com

Guodong Sun

CS Dept., Tsinghua University
TNLIST, Tsinghua University
Beijing, 100084, China
sgdcs@tsinghua.edu.cn

Bin Xu

CS Dept., Tsinghua University
TNLIST, Tsinghua University
Beijing, 100084, China
xubin@tsinghua.edu.cn

Abstract

This paper designs, implements, and evaluates the AD-Sense system, an activity-driven sensing framework for mobile devices, which aims at providing good accuracy of sensing while reducing the energy consumption. AD-Sense first accurately extracts user's activity, with only acceleration data from user-held device. Then, according to the activity information, AD-Sense dynamically determines sensor-probing pattern, such as localization, in order that it may use as less energy as possible while meeting the application fidelity. Analysis and testbed experiments show that AD-Sense efficiently tradeoffs the energy consumption and the accuracy requirement, and reduces the hand-operated overhead of user.

Categories and Subject Descriptors

C.2.4 [Computer-communication Networks]: Miscellaneous

General Terms

Design, Implementation, Experimentation

Keywords

Mobile devices, Activity analyzing, Energy efficiency

1 Background and Motivation

Embedded with diverse communication interfaces and sensors, mobile device becomes more and more smart recently. A mobile phone, for example, can trace where you are, whom you call, and to which music you often listen; it also can access the Internet, either with the equipped Wi-Fi interface or the telecom channel. Besides communication modules, tiny sensors are built in mobile devices and are able to measure temperature, light, acceleration, and so on. Those on-device sensors will be able to acquire more user-centric information. In the literature, a critical problem is enabling the feasible and efficient sensing. One obvious method of on-phone sensing, as used in MIT's Funf Sneak Peak [2], is to continuously sample the sensors with a regular period configured by user. We argue that such method is neither efficient in energy-accuracy tradeoff nor flexible in management of user.

First, it is very hard for periodic sensing to achieve good tradeoff between energy and application fidelity. This is

largely because the space and activity of users often vary with time, and then the requirement for sensor data are possibly different in terms of fidelity. For example, for a location-tracing application, the location readings with the time interval of five minutes are generally enough to trace a person who is attending a meeting in office. If the sensing rate is set to be once per minute, unnecessary energy will be consumed, because the redundant data will not provide more views on the user's location in the office. But, to precisely trace runners or subway passengers, the location readings at one-minute interval are possibly insufficient.

Second, intensive sensing may bother the user with fast battery drainage. To benefit from mobile sensing which is usually energy consuming, the user often has to recharge his device more frequently, troubling him and reducing the cycle of the battery. In another instance, when the user wants to start some apps that needs significant energy, like call making or video taking, but only finds the battery capacity is not enough to supply these apps, he will have to navigate the running sensing tasks and stop one or many of them, by carefully tapping some buttons displayed on phone screen. Obviously intervention-free or even intervention-less sensing apps are promising in practice.

In this work, we design, implement, and evaluate an activity-driven sensing framework, called AD-Sense, for mobile devices with built-in sensors. Different from the traditional sensing schemes with a period adjusted by user, AD-Sense dynamically determines when, how often, and how long to probe on-phone sensors, according to user's activities which can be efficiently extracted only with local knowledge of mobile phone. AD-Sense efficiently tradeoffs the energy and the accuracy requirements and reduces the hand-operated overhead of user, by dynamically determining sensor-probing patterns.

2 Overview of AD-Sense

As a sensing framework, AD-Sense now considers most of hard sensors built in mobile phones, covering, say, WiFi, GPS, and accelerometer, and some soft sensors monitoring the installed apps, the music often selected by users, etc.; it can of course be extended to monitor other external sensors such as RFID readers, heart-beat sensor, etc.. In general, AD-Sense provides a framework for data collection, data analysis and data visualization of mobile application.

As shown in Fig.1, on the phone side, AD-Sense involves

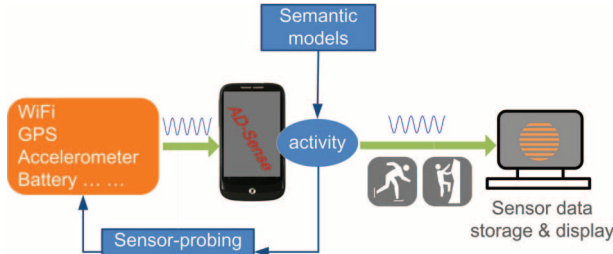


Figure 1. Architecture of AD-Sense

a data collection module and an activity recognition module. The activity recognizer works with a series of semantic models closely related to human activities; it now outputs four major categories of human activities: sitting, walking, standing, and running. According to the current activity of user, AD-Sense determines a sensor-probing pattern that controls the data collector in terms of the rate, duration of sampling.

On the back-end server, the raw sensor data from mobile phone is stored and analyzed. When logging into the server, a user can review his daily history about his location, place visits, exercises and so on, in a visualized way. In addition, a registered user, if he wish, can grant other persons or developers to access all or part of his data. Through the shared data, developers using AD-Sense can design more powerful apps, which benefit users in turn.

3 Design

One challenge in designing AD-Sense is to accurately identify user's activities as well as their variation. We use a series of semantic models [4, 5] to map the sensing data of WiFi APs, accelerometer, and GPS, into user activities.

Semantics-model-based Activity Recognizer. The semantic models we use includes two components: the activity recognizer and the place detector which is used to identify user's location and the handoff of places. The activity analysis stage pre-processes sensor data, extract some statistic features of data, and classify personal activity according to [1, 5]. As the activity of user changes, his location as well as place will possibly change. In this work, we use WiFi map and footprint to evaluate the variation of user's location. Detailedly, AD-Sense first scans WiFi data around user, and then transforms the data into WiFi fingerprint which contains more information about WiFi APs. By comparing the maximal similarity of fingerprint with scanned results, AD-Sense can effectively determine whether user's location changes. Furthermore, AD-Sense employs the WiFi map and the scanning results of user's WiFi signal to deduce their movement from one place to another, by measuring the variation of fingerprint similarities. Here we consider two categories of place handoffs: entrance and departure.

Activity-driven Data Collector. In order to achieve high accuracy, the application often samples sensors with a small regular period. But this way is very energy-hungry because starting and reading sensors both consume significant energy. Here we use an activity-driven strategy to adaptively control data collection, with sampling rates as low as possible and without loss of fidelity. When, for example, a user is sitting at his desk for working, AD-Sense will stop moni-

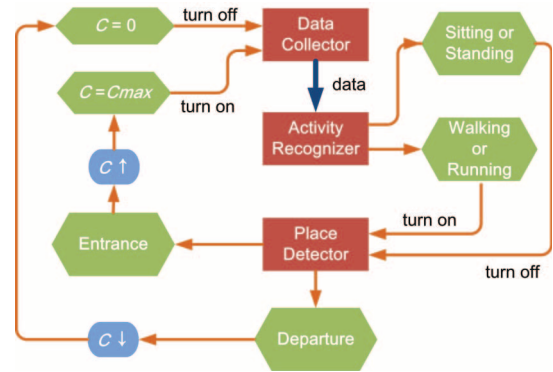


Figure 2. Data collection strategy of AD-Sense

toring surrounding WiFi APs and calculating the GPS value, while leaving only accelerometer in working.

Fig.2 shows the basic flow of AD-Sense's data collection. When the program starts, it samples all sensors and inputs these samples to activity recognizer. If the state is identified either as sitting or standing, AD-Sense turns off place detector. If walking or running state is determined, place detector is turned on. If place detector finds that the user's state is in entrance or departure, variable C will be increased or decreased with one. When $C = 0$, meaning the user's state is stable, AD-Sense switches off data collector; when $C = C_{max}$, meaning the user's state varies significantly, AD-Sense has to switch on the data collector to probe on-phone sensors.

4 Demonstration Proposal

During th demo stage, we will present the automatic feature and the visualization of AD-Senses, by displaying user's trace and his activity distribution.

Once a user registers on-line the AD-Sense website [3], he can download, install, and run AD-Sense codes on his Android phone. The sensor data is delivered to the web server through WLAN, GSM, or GSM. During the sensing, user does not have to do any management, until he wants to stop AD-Sense. From the server side, a user can see not only the raw data such as his location, but also his activity distribution information, which is more helpful for physical exercise planning or healthcare considerations.

5 Acknowledgements

This work was supported by two NSFs of China, respectively with No.60803124 and No.61170212.

6 References

- [1] L. Bao and S. Intille. *Activity recognition from user-annotated acceleration data*, chapter 1, pages 1–17. LNCS 3001. Springer-Verlag, 2004.
- [2] <http://funf.media.mit.edu/sneak.html>. 2011.
- [3] <http://iweb.cs.tsinghua.edu.cn/PICD/>. 2011.
- [4] D. Kim, Y. Kim, D. Estrin, and M. Srivastava. Sensloc: sensing everyday places and paths using less energy. In *SenSys*, pages 43–56, Zurich, Switzerland, 2010.
- [5] H. Lu, J. Yang, Z. Liu, N. Lane, T. Choudhury, and A. Campbell. The jigsaw continuous sensing engine for mobile phone applications. In *SenSys*, pages 71–85, Zurich, Switzerland, 2010.