An Energy Efficient User Context Collection Method for Smartphones

Yoonseon Han*, Joon-Myung Kang†, Sin-seok Seo‡, Ahmed Mehaoua*§¶, and James Won-Ki Hong*‡

*Division of IT Convergence Engineering, POSTECH
{seon054, jwkhong}@postech.ac.kr

†Department of Electrical and Computer Engineering, University of Toronto
joonmyung.kang@utoronto.ca

‡Department of Computer Science and Engineering, POSTECH
sesise@postech.ac.kr

§Department of Mathematics and Computer Science, University of Paris Descartes
¶Laboratoire Bordelais de Recherche en Informatique, CNRS LaBRI, France
ahmed.mehaoua@parisdescartes.fr

***Abstract*—Recently, user context information has been utilized for efficient and effective network and service management. In this situation, there is a strong need to collect user context information. One of the proprietary methods to collect user context information is to use sensors equipped inside a smartphone such as accelerometer, gyroscope, digital compass, and microphone. However, user context collection on a smartphone easily leads to rapid drain of the smartphone battery. To overcome the problem, this paper proposes a novel user context information collection model for efficient smartphone battery management. This model is based on two strategies: scheduling and dynamic sensor reconfiguration. The proposed model is implemented on Galaxy Nexus Android smartphone. The Performance evaluation shows that the proposed model reduces the energy consumption by a ratio of 42 percent compared to the current periodic sensor reading model.**

***Index Terms*—Context-Awareness, Context Collection, Energy Efficiency, Smartphone**

I. INTRODUCTION

User context information can be utilized for efficient and effective network and service management [1], [2]. User context information provides clues to estimate users' service and resource demands in near future by analyzing past usage history, user preferences, and resource demanding patterns. By estimating the future resource demands, we can improve the quality of services and utilization of resource usage. In this situation, there is a strong need to collect user context information. To collect user context information, a lot of devices are available such as PCs, laptops, tablets, and smartphones. Smartphones are personal devices embedding multiple sensors, such as accelerometer, gyroscope, compass, and microphone, that are suitable for collection of user context information. Moreover, users carry a smartphone during most of time. Using the sensors, we can collect user context information by inferring or analyzing data collected from the sensors. However, this method causes of rapid drain of the smartphone battery.

One of the simplest models to collect user information is that turns on and reads sensors periodically, and analyzes the sensed values to decide that the condition for a user context is satisfied or not. In this case, we can configure two main parameters related to sensor performances: sampling rate and duty cycle. By adjusting these two parameters, we can trade off battery consumption with sensing accuracy. One of the fundamental principles to minimize the energy consumed by unrelated sensors is to decrease sensor accuracy when a sensor is not necessary in a given situation. For example, when a user has a meeting, accelerometer is not an essential sensor. In this case, we can turn off or degrade accelerometer's sensing accuracy. Therefore, the main challenge to reduce battery consumption is to decide which sensors are necessary in a certain situation and which are not.

This paper proposed a novel battery efficient user context information collection model based on two methods: sensor scheduling and dynamic sensor reconfiguration. The main idea of the proposed method is pull-based sensor reading, which activates sensors only when they are required. We can reduce energy consumption by eliminating unnecessary activation and degrading accuracy for not related sensors in a given user situation. The method is implemented on Android smartphone, Galaxy Nexus. Moreover, we show that the proposed methods reduced battery consumption to about 42 percent compared with a periodic sensor reading method.

The remainder of this paper is organized as follows. In

This research was supported by World Class University program funded by the Ministry of Education, Science and Technology through the National Research Foundation of Korea (R31-10100) and MSIP(Ministry of Science, ICT and Future Planning), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency)(NIPA-2013-H0301-13-3002).

Section II, we present related work. In Section III, we describe the overall procedure of our proposed model. In Section IV, we introduce our context collection method based on sensor scheduling. In Section V, we describe our method to reconfigure sensors dynamically for reducing energy consumption. In Section VI, we present the design and implementation of the proposed model and evaluate its performance in terms of energy efficiency and accuracy. Finally, we conclude and give some perspectives in Section VII.

II. RELATED WORK

User context information is defined as any information that can be used to characterize the situation of an entity [3]. In the definition, an entity is a person, place, or object that is considered relevant to the interaction between a user and an application. The same definition is applied in this research. There are many research work related with user context information collection and user behavioral pattern recognition. [4] pointed out the idea and vision of mobile phone as a sensor as well as the introduction of on-going research issues. The most of research efforts are based on a singular sensor, and the most important issue is to accurately recognize and classify user context in a certain situation. In many cases, smartphones are utilized as an interconnection hub to connect various sensors or an analyzer to process sensor values. Moreover, there are research efforts to utilize sensors equipped inside smartphone such as accelerometer [5]–[7], microphone [8], [9], and GPS [10].

One of the well-known side effects of user context collection is rapid battery discharging due to frequent sensor activation and analysis. To reduce battery consumption, several fundamental methods are proposed such as minimum sensor selection, sensor duty cycle adjustment, sensor sampling rate adjustment, and sharing a common information with others. The basic principal is to exchange sensor accuracy with battery consumption by capturing which sensor is necessary or not depending on a users' situation. For example, if a user is staying on his/her office desk, then GPS and accelerometer provide less meaningful information to figure out user's status. Therefore, deciding which and when sensors should be activated with an optimal configuration is the most important issue. To decide sensor activation time and configuration, Wang et al. [11] proposed a method to exploit user states inferred from sensor value inference, and Lu et al. [12] proposed a method to utilize characteristics of each sensor. In Paek et al. [10], a method based on user's past history of location and speed changes to decide GPS information update time is proposed. From user's past history logged with event location and time, the method estimates the current location uncertainty according to a user's different situation. In Lee et al. [13], a location information sharing method to reduce energy consumption is introduced based on the fact that common users spend most of time with others. However, the most of research only focused on a single sensor optimization not entire energy consumption of smartphone.

III. OVERVIEW

The idea to use sensors for collecting user context information is that sensor values differ according to current user activity. For example, an accelerometer monitors the acceleration values of device in terms of three axis directions, and the values show different patterns depending on user behavior. Fig. 1 shows that the changes of the sensor values by user behaviors: stay, walking, step up, and step down. By analyzing the value changes, we can detect and recognize user context information related with behavior and environment. To extract meaningful features from sensor values, several analyzer is introduced such as mean, standard deviation, and etc.

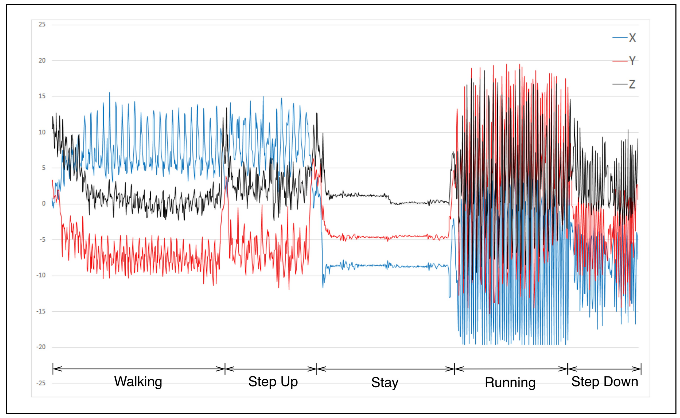

Fig. 1. Different patterns of accelerometer sensor according to user behaviors

The proposed method consists of two main steps: sensor activation scheduling and dynamic sensor reconfiguration. The former is to recognize what user context information should be collected and which sensors are used. In this step, unnecessary sensors are excluded by interactively generated sensor schedule after activation of a sensor to reduce battery consumption based on context specifications. The latter is for reducing the energy consumption on context collection process based on dynamic sensor reconfiguration. In this step, the sensors judged as not important will be dynamically reconfigured to reduce energy consumption by adjusting sampling rate and sensor activation interval using two model: user state and past-history.

IV. USER CONTEXT INFORMATION COLLECTION METHOD

The starting point of this research is to develop a real-time user context information collection model for smartphones. First of all, we need to figure out which user contexts should be collected. To provide necessary information, context specification is introduced. This specification contains the conditions to decide whether a user context is satisfied or not based on values from several sensors. A user context specification consist in a set of conditions represented as 4 tuples, <S, A, T, V>, where S is a sensor name, A is an analyzer name, T is time window, and V is a sensor value range. For example, a condition, <Accelerometer, Mean, 1000, 5>,

```
context:
    name : “walking”,
    condition :
        sensor : “accelerometer”,
        time frame : “5000”,
        analysis method : “Standard Deviation”,
        value : “1.0, 3.0”,
```

Fig. 2. A context specification example

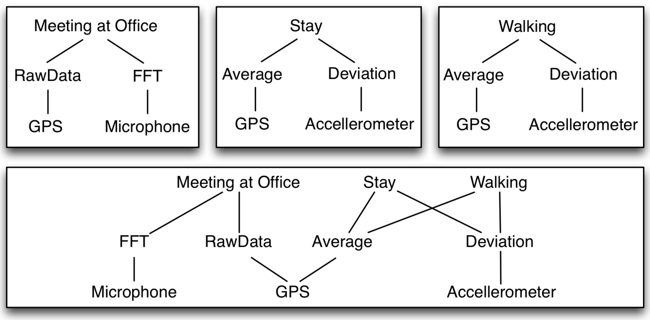

Fig. 3. An example of context collection tree building process

can be interpreted as a condition that is satisfied when the mean value of accelerometer sensor collected during 1000 milliseconds is equal to 5. If all conditions for a context are satisfied, then we decide that a specific context is valid for a given situation. Fig. 2 shows a context specification example.

Based on the context specification, we know what kinds of user contexts should be collected and which sensors should be triggered. However, each context specification contains the information only about a context, not all contexts to be collected. Therefore, we need to synthesize each fragment of context specifications into a unified model. In this paper, a context collection tree, a tree structured model to represent the relationships among contexts, sensors, and analyzers, is proposed. The structure of a context collection tree is three tiers: context, analyzer, and sensor. Each path from top to bottom represents a condition. Fig. 3 shows a context collection tree building an example with three context specifications: meeting at office, stay, and walking. A context collection tree provides information to support sensor and analyzer scheduling.

A. Sensor activation scheduling

In this paper, we use a pull-based sensor reading method which activates sensors only when they are needed. Therefore, a next step is to determine a schedule of sensor activation. The basic principle to make a sensor schedule is that an energy efficient sensor should be activated first. The reason to do this scheduling is that unnecessary sensor activation will be prevented when a condition for deciding collection of a context is not satisfied by previously activated sensors. If a condition is denied by previously activated sensors, then the context cannot be satisfied regardless of remaining conditions are satisfied or not. For example, a context “Stay” in Fig. 3 can’t be satisfied when a condition related with GPS is not satisfied regardless of the result of accelerometer.

TABLE I
THE NOTATIONS USED FOR SENSOR ACTIVATION SCHEDULING

Name	Description
N_C	A number of contexts on a context collection tree.
N_S	A number of sensors on a context collection tree.
C_i	ith context on a context collection tree.
S_j	jth sensor on a context collection tree.
$U(S_j)$	A utility function returns a score value used for sensor scheduling.
$SR(S_j)$	A function returns current sampling rate of sensor S_j.
$ES(S_j)$	A function returns energy consumption of jth sensor per a second.
$T(C_i, S_j)$	A function returns required time frame for jth sensor to collect ith context. If ith context is not related with jth sensor, 0 is returned.

The notations for sensor scheduling and context collection procedure are summarized in Table. I. To deicide the sequence of sensors, a utility function is defined as follows.

$$U(S_j) = \frac{\sum_{i=0}^{N_C}\sum_{k=0}^{N_S} ES(S_j)\cdot SR(S_j)\cdot T(C_i,S_j)}{ES(S_j)\cdot SR(S_j)\cdot Max(T(C_0,S_j),...,T(C_{N_C},S_j))}$$

The utility function, U(S_j), is designed to return a high score value when a sensor is related with many contexts and has low energy consumption. Overall procedure to collect user context information is summarized in Fig. 4.

Overall context collection procedure

Input: a set of context specifications to be collected
Output: a set of contexts which all conditions are satisfied

1. Build a context collection tree based on input context specifications
2. Calculate score values for all sensors in the context collection tree using the utility function, $U(S_j)$
3. Select a sensor, S_j, which have the highest score value
4. Activates S_j during time frame, $Max(T(C_0,S_j),...,T(C_{N_C},S_j))$
5. Execute corresponding analyzers for the sensor
6. Remove contexts which the condition is not satisfied by the activated sensor from the context collection tree
7. Remove the activated sensor and related conditions from the context collection tree
8. Iterate step 2-7 until there is no sensor to remove
9. Return remaining context on the context collection tree after confirmation that all conditions are satisfied

Fig. 4. Overall procedure of context information collection

V. Dynamic Sensor Reconfiguration

Usually, sensing accuracy and battery consumption of sensors are in a trade-off relationship. To raise the accuracy, a sensor should collect more samples, and be activated longer time. These actions require more energy consumption. The main idea to design an energy efficient method for user context information is to use the trade-off relationship. The method should continuously collect user context information anytime and anywhere. However, the importance of a sensor changes depending on a user behavior or environment. For example, if a user is having a meeting in the office, then the value of GPS sensor is less important than that of other sensors such as microphone. Therefore, we can reduce energy consumption by degrading accuracy of unrelated sensors at a given user situation.

Two main sensor configuration parameters related with energy consumption are sampling rate and sensor activation interval. Sampling rate is the number of samples per unit of time, and sensor activation interval is a time from end time of activation to start time of next activation. To configure them, two configuration methods are introduced: user state based and past-history based sensor reconfiguration. These methods are based on two important observations of user context information.

- **Continuity:** Most of human behaviors and environments are maintained in a some time period. In the view of sensors, the current user context sustains a certain amount of time. For example, a human behavior, “walking”, continues until he or she arrives at a destination, and it takes some time depending on the distance. Using this observation, we can figure out a set of sensors having a high relation with a given situation after recognizing a set of common contexts.
- **Life pattern:** The patterns of human behaviors or environments are very dependent on the time. For example, a typical user has a sequence of behaviors such as going to work, working, lunch, working, and going to home. Based on this observation, we can have a clue how often sensor are activated depending on a certain time.

By applying the observations, we can find out the proper sensor configuration parameters to reduce energy without reducing precision of user context collection. Fig. 5 shows the relationships among sensor configurations and proposed methods. A set of sensors are activated, and sensors are reconfigured by two reconfiguration methods. Sampling rate of sensors is reconfigured according to a user state, and sensor activation interval is adjusted by referring past-history.

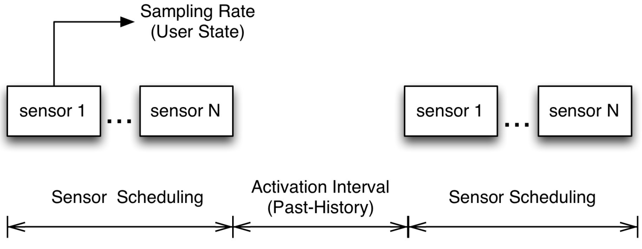

Fig. 5. Sensor configuration methods

A. User state based sensor reconfiguration

To extract the common sensor patterns from contexts having sustainability, a user state specification is introduced. A user state is defined as a set of user contexts, and it describes how to reconfigure sensors when a user situation is judged to fulfill the user state. The basic idea behind user state specification is that some sensors that is not related with current sustaining user behaviors or environments will be reconfigured to reduce energy consumption during next amount of time. The sensors having high inter-relationship will not be degraded by the reconfiguration. The degradation of sensors is adjusted by changing sampling rate of sensors. Fig. 6 shows an example for user state specification to describe the “meeting” user state. The meaning of the specification is that accelerometer will be degraded to low sampling rate when three contexts, *stay, speech, and located at the office,* are collected.

```
user state:
    name : “meeting”,
    condition context : “stay”
    condition context : “speech”
    condition context : “located at the office”
sensor reconfiguration :
    user state : “meeting”
    sensor : “accelerometer”,
    sampling rate : low
```

Fig. 6. A context specification template

B. Past-history based sensor reconfiguration

As explained previously, the change rate of common contexts is stable after entering a certain state for a while. However, when a state transition occurs, it means that a set of common contexts is changed rapidly. To capture a user context information accurately, we need to pay attention to the moment of a state transition. By utilizing a life patten of a user, we can roughly estimate when a state transition will occur. In this paper, the number of state transitions are logged in a time slot. A day in divided into 48 time slots by 30-minute unit. When a time slot has a higher number of state transitions than an average number, sensor activation interval will be shorten to collect contexts frequently. Sensor activation interval is calculated by the equation (1), where D(i) is sensor activation interval at the time slot i, N(i) is the number of state transitions at the time slot i, α is a normalization value to set mean as 1, and $D_{default}$ is default sensor activation interval cycle value.

$$D(i) = (\frac{Mean(N(0),...,N(48))}{N(i)} + \alpha) \cdot D_{default} \qquad (1)$$

VI. Evaluation

A. Implementation

We designed a system based on the class diagram as shown in Fig. 7. This architecture is implemented on Galaxy Nexus with Android version 4.1.1 and Kernel version 3.031.

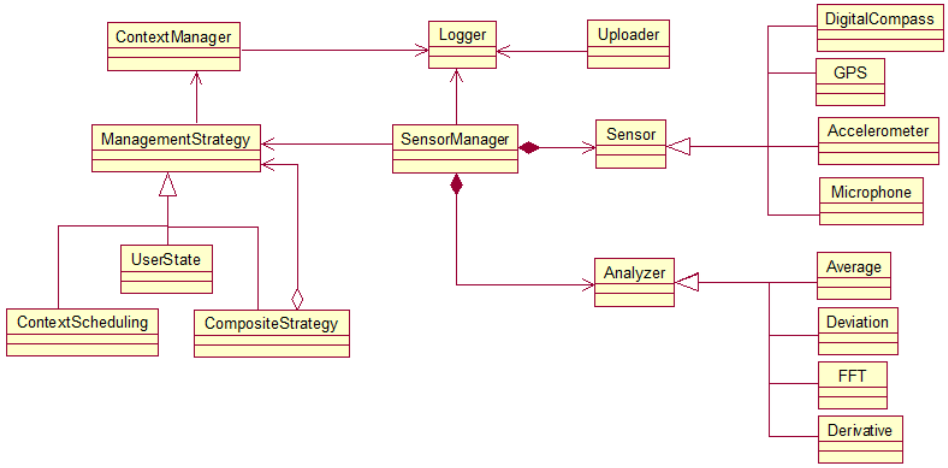

Fig. 7. The class diagram to describe an overall system architecture

The system architecture is composed of 7 main components. *Sensor* is the super class of all physical and logical sensors. *Analyzer* analyzes or infers information from raw sensor data. *SensorManager* makes relations between *Sensor* and *Analyzer* to interconnect them. *ManagementStrategy* has all responsibilities to decide parameters related with sensor reconfiguration. *ContextManager* decides which contexts exist based on analyzers' results. *Logger* stores information or data to database or files. *Uploader* sends information or data to external entities.

B. Experiments

1) Parameter Setup: Default activation interval of sensing is set to 90 seconds. To schedule sensor, we need to determine the return values of $SR(S_j)$ and $ES(S_j)$ described in the section IV. For the simplicity of experiments, we defined two levels of sampling rate denoted as low and normal. It means that sensors are activated with only a pre-defined low or normal sampling rate. For setting $ES(S_j)$, we measured energy consumption ratio among sensors using Android battery API. The detailed values are described in Table II. In case of GPS, Android does not allow to reconfigure sampling rate.

TABLE II
$SR()$ AND $ES()$ FUNCTIONS' RETURN VALUES

Parameter	SR()	ES()
Accelerometer - Low	5 Hz	1
Accelerometer - Normal	16 Hz	1.59
Microphone - Low	11025 Hz	1.65
Microphone - Normal	16000 Hz	3.54
GPS	1Hz	5.15

2) Context and user state specifications: For actual context collection, we specified seven contexts and four user states. The detailed specifications are summarized in Table III for contexts and Table IV for user states. "Home" and "Office" are more specific version of "Location" context. To solve the cold start problem of past-history method, we executed the same scenario twice and measured energy changes only at the second trial.

TABLE III
SPECIFIED CONTEXTS

Name	Sensor	Analyzer
Stay	Accelerometer	Standard Deviation
Walk	Accelerometer	Standard Deviation
Run	Accelerometer	Standard Deviation
Silence	Microphone	Decibel
Speech	Microphone	Decibel, Silence Ratio
Music	Microphone	Decibel, Silence Ratio
Location	Accelerometer GPS	Standard Deviation Range Matching
Meeting	Accelerometer Microphone	Standard Deviation Decibel, Silence Ratio

TABLE IV
SPECIFIED USER STATES

Name	Condition contexts	Sensor	Reconfiguration
Default	None		
Working	Office, Meeting Office, Stay	Accelerometer	Low
Rest	Home, Stay Home, Silence	Accelerometer Microphone	Low Low
Moving	Walking Running	Microphone	Low

3) Experimental scenario: To compare the proposed model with previous solutions, we recorded the sensor values from four situations: going to work, working, go home, and taking rest. The recorded sensor values are repeated during 30, 120, 30, and 120 minutes for each situation as the same sequence with the recording. When sensors are activated, the system will replace the sensor values with the recorded values. By replacing the sensor values, we can measure sensors with a planed scenario without a care of the uncertain user behavior. We compared the proposed model with a periodic sensor reading model with 90-second interval.

4) Sensor activation interval and sampling rate: To verify the proposed method, we measured sensor activation five times with sampling rate while conducing the experimental scenario. The number of activations and average sampling rate of each sensor is summarized in Table V. The results show that GPS activations are screened out by scheduling, and average sampling rate is decreased by sensor reconfiguration.

TABLE V
THE NUMBER OF ACTIVATIONS AND SAMPLING RATE

	Accelerometer	**Microphone**	**GPS**
Activations (periodic)	1044	1044	1044
Activations (proposed)	1043	1043	209
Average SR (periodic)	16Hz	16000Hz	1Hz
Average SR (proposed)	7.2Hz	15007.9Hz	1Hz

5) Energy consumption: We measured energy consumptions five times while conducting the scenario on the smartphone using the Android battery API. Fig. 8 shows the energy consumption of each model: no collection, the proposed method, and periodic sensing method. The result shows that the proposed model reduces battery consumption to 42 percent compared to a periodic sensing model.

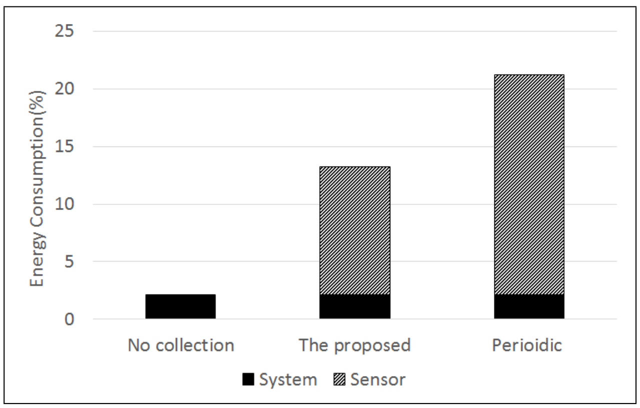

Fig. 8. Energy consumption comparison: no collection, the proposed model, and periodic sensing model

VII. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a user context information collection model with the consideration of battery efficiency using smartphone sensors such as accelerometer, microphone, and GPS. The proposed method mainly consists of two procedures: sensor activation scheduling and dynamic sensor configuration. When a set of user contexts to be collected is specified, the proposed method activates related sensors according to a schedule to improve energy efficiency. The dynamic sensor configuration is based on two simple observations: continuity and life pattern. By utilizing the observations, the method dynamically reconfigure sensor activation interval and sensors' sampling rate. To evaluate the proposed method, we designed an experiment scenario, and it shows that the method reduced 42 percent of battery consumption than a periodic sensor reading model.

For future work, we need to deploy the proposed method for common users. The experiment result presented in the paper is based on a use case scenario, not real situations. By deploying it, we can confirm that the proposed method correctly will reduce energy consumption for various situations faced by common users. Moreover, by installing accurate external energy measurement devices, we need to evaluate the proposed method more accurately with various metrics in terms of performance, latency, and accuracy. We also plan to develop an automated specification generator for the user context specification and user state specification. To make those specification, a developer must decide low level parameters such as sampling rate and sensor activation interval. By developing a tool to assist specifications, developers can focus more on high level specifications to collect user context information.

REFERENCES

[1] P. TalebiFard and V. Leung, "A dynamic context-aware access network selection for handover in heterogeneous network environments," in *Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on*, 2011, pp. 385–390.

[2] B. Fuentes, A. Bantouna, L. Bennacer, G. Calochira, M. Ghader, G. Katsikas, and F. Yousaf, "On accomplishing context awareness for autonomic network management," in *Future Network Mobile Summit (FutureNetw), 2012*, 2012, pp. 1–9.

[3] A. K. Dey, "Understanding and using context," *Personal Ubiquitous Comput.*, vol. 5, no. 1, pp. 4–7, Jan. 2001.

[4] N. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. Campbell, "A survey of mobile phone sensing," *Communications Magazine, IEEE*, vol. 48, no. 9, pp. 140 –150, sept. 2010.

[5] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman, "Activity recognition from accelerometer data," in *Proceedings of the 17th conference on Innovative applications of artificial intelligence - Volume 3*, ser. IAAI'05. AAAI Press, 2005, pp. 1541–1546.

[6] Z. Yan, V. Subbaraju, D. Chakraborty, A. Misra, and K. Aberer, "Energy-efficient continuous activity recognition on mobile phones: An activity-adaptive approach," in *ISWC'12*, 2012, pp. 17–24.

[7] Y.-S. Lee and S.-B. Cho, "Activity recognition using hierarchical hidden markov models on a smartphone with 3d accelerometer," in *Proceedings of the 6th international conference on Hybrid artificial intelligent systems - Volume Part I*, ser. HAIS'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 460–467.

[8] S. Chu, S. Narayanan, and C.-C. J. Kuo, "Environmental sound recognition with time-frequency audio features," *Trans. Audio, Speech and Lang. Proc.*, vol. 17, no. 6, pp. 1142–1158, Aug. 2009.

[9] S. Chu, S. Narayanan, and C.-C. Jay Kuo, "Environmental sound recognition using mp-based features," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, 31 2008-april 4 2008, pp. 1 –4.

[10] J. Paek, J. Kim, and R. Govindan, "Energy-efficient rate-adaptive gps-based positioning for smartphones," in *Proceedings of the 8th international conference on Mobile systems, applications, and services*, ser. MobiSys '10. New York, NY, USA: ACM, 2010, pp. 299–314.

[11] 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. New York, NY, USA: ACM, 2009, pp. 179–192.

[12] 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*, ser. SenSys '10. New York, NY, USA: ACM, 2010, pp. 71–84.

[13] 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*, ser. MobiSys '12. New York, NY, USA: ACM, 2012, pp. 43–56.