

Article

## Analysis of Characteristics of Power Consumption for Context-Aware Mobile Applications

Meeyeon Lee, Deok-Ki Kim and Jung-Won Lee \*

Department of Electrical and Computer Engineering, Ajou University, 206, World cup-ro, Yeongtong-gu, Suwon-si, Gyeonggi-do 443-749, Korea; E-Mails: mylee@ajou.ac.kr (M.L.); gcross07@ajou.ac.kr (D.-K.K.)

\* Author to whom correspondence should be addressed; E-Mail: jungwony@ajou.ac.kr; Tel.: +82-31-219-1813; Fax: +82-31-212-9531.

External Editors: Neil Y. Yen and Hwa-Young (Michael) Jeong

Received: 13 August 2014; in revised form: 31 October 2014 / Accepted: 11 November 2014 /

Published: 14 November 2014

---

**Abstract:** In recent years, a large portion of smartphone applications (Apps) has targeted context-aware services. They aim to perceive users' real-time context like his/her location, actions, or even emotion, and to provide various customized services based on the inferred context. However, context-awareness in mobile environments has some challenging issues due to limitations of devices themselves. Limited power is regarded as the most critical problem in context-awareness on smartphones. Many studies have tried to develop low-power methods, but most of them have focused on the power consumption of H/W modules of smartphones such as CPU and LCD. Only a few research papers have recently started to present some S/W-based approaches to improve the power consumption. That is, previous works did not consider energy consumed by context-awareness of Apps. Therefore, in this paper, we focus on the power consumption of context-aware Apps. We analyze the characteristics of context-aware Apps in a perspective of the power consumption, and then define two main factors which significantly influence the power consumption: a sort of context that context-aware Apps require for their services and a type of ways that a user uses them. The experimental result shows the reasonability and the possibility to develop low-power methods based on our analysis. That is, our analysis presented in this paper will be a foundation for energy-efficient context-aware services in mobile environments.

**Keywords:** power consumption; energy efficiency; low power; context-awareness; smartphone applications (Apps)

---

## 1. Introduction

Mobile devices, in particular, smartphones have the advantages in context-awareness and intelligent services [1]. They have the high performance in the aspect of hardware such as various embedded sensors like GPS and Accelerometer. Besides, smartphones are everywhere in the hands of millions of people all over the world. Users of smartphones actively use them without any coercion or discomfort all the time. Due to these strong points, applications of smartphones for context-aware services have been actively developed [2,3]. Context-awareness is an essential technique which can detect changes of context and provide useful relevant information [4–8]. A Context-aware smartphone application (App) means a computer program running on smartphones and provides various customized services based on context-awareness. In general, a process of context-aware Apps has three phases as follows [9]:

- (1) Context data acquisition: context-aware Apps collect low-level sensing data and simply pre-process them.
- (2) Context analysis: they infer the high-level context by modeling, training, analyzing and mining low-level data.
- (3) Service integration: they provide their own services according to the inferred real-time context.

Recently, in the area of context-aware services in mobile environments, many studies presented methods to gather sensing data from embedded sensors and to infer the high-level context by applying various data mining methodologies [2,10–12]. They aimed just for the precise inference of the real-time context.

However, context-awareness in mobile environments has some difficulties due to limitations of devices such as limited computing, limited storage capabilities, and limited battery power. Among these issues, the power or energy problem is the most critical one that has to be seriously considered [13]. In order to improve the energy-efficiency in mobile environments, some studies proposed their own methods to reduce power consumed by display or LCD [14], and by CPU [15], and by GPS module [16]. In addition, there are many research about the power consumption in the related areas like wireless sensor networks [17,18]. That is, previous works just have focused on the low-power or energy efficiency of H/W modules or networks. A few studies recently started to do research on S/W-based approaches in order to improve the power consumption. As one of them, Smartphone Energizer is a context-aware technique to extend battery life [19]. It aims to save the device's energy by offloading computation according to the contextual information. However, it should work with an offloading server and still has some issues like communication and privacy. Also, these previous works for measuring or reducing the consumed power did not target individual applications of smartphones. A study [20] tried to calculate the power consumption of mobile device as well as mobile applications. It presented a power model to meter a few applications and showed its accuracy with experimental results. A power estimation method based on battery traces [21] was proposed to calculate accurate power dissipation rate for smartphone applications. In recent research [22], the power consumption of various mobile cloud computing applications

was investigated by using existing energy and performance measurement tools. However, these studies just measured the power consumed by each application, but did not analyze the patterns or causes.

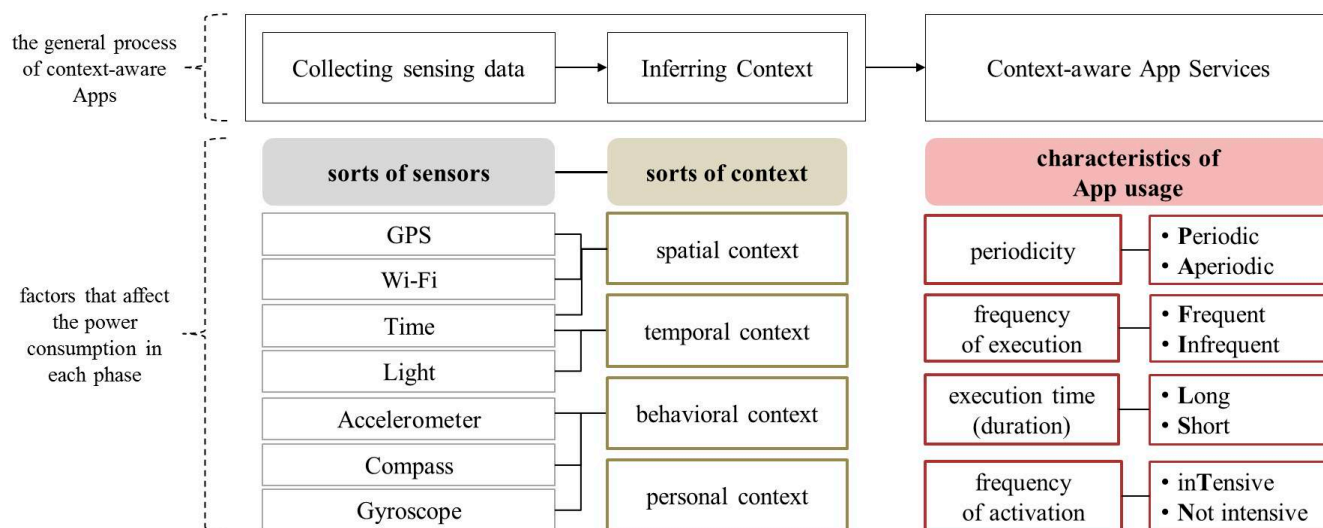
Therefore, in this paper, we focus on the energy-efficiency in terms of software, and then analyze characteristics of the power consumption of context-aware Apps which are the biggest consumers on mobile devices. We define two main factors related to the power consumption: sorts of context and types of App usage. The first factor means that a sort of context used by context-aware Apps can affect the amount of consumed power, so this is an App-dependent factor. The second one is a user-dependent factor, and a type of ways that a user actually uses a context-aware App. Our analysis presented in this paper will be the cornerstone for energy-efficient context-awareness in mobile environments.

The paper is structured as follows. Section 2 provides our analysis about characteristics of the power consumption of context-aware Apps. Experimental results to prove the significance of our analysis are presented in Section 3. Section 4 concludes this paper with proposals for future works.

## 2. Analysis of Characteristics of the Power Consumption

We analyzed characteristics of the power consumption of context-aware Apps in order to define related factors. When a user uses context-aware Apps, they mostly consume power in two phases. In the first phase, they collect and process raw data from various embedded sensors, and then generate context. The second phase is to provide App services based on the inferred context. Therefore, we focused on consuming power in these two phases. Figure 1 shows two categories of factors influential in consuming energy of context-aware Apps: sorts of context and characteristics of App usage.

**Figure 1.** Two main factors related to the power consumption of context-aware Apps.



### 2.1. An App-Dependent Factor: Sorts of Context

In the first phase, a sort of context necessary for context-aware App services is an important factor. Depending on the type of context, the sort of embedded sensors to be used is determined, and the power consumed by each sensor is different. As shown in Figure 1, we referred a basic context model [23] which includes four types: spatial, temporal, behavioral, and personal context. Table 1 shows short descriptions and related sensors for each type of context. For example, in the case of an App which uses

the spatial context like in/out or home, it activates GPS or Wi-Fi to gather sensing data. Generally, GPS and Wi-Fi sensors are known as the biggest energy consumers or energy-drain components [24]. Therefore, we can analyze that Apps using context related to GPS or Wi-Fi may consume relatively more energy than ones activating other sorts of sensors.

This factor is App-dependent. Based on this analysis, it is possible to reduce power consumed by context-aware Apps by controlling activations of sensors which require relatively more power when they generate context.

**Table 1.** A simple four categories of mobile context and related sensors.

Type of context	Description (example)	Sensor
Spatial context	Location and location-related information of the mobile user (the latitude/longitude coordinates, the street address, the home, a friend's place, <i>etc.</i> )	GPS, Wi-Fi, Time
Temporal context	Time and date information (time, date, season, <i>etc.</i> )	Time, Light
Behavioral context	Physical movements of the mobile user (walking, running, staying quietly, <i>etc.</i> )	Accelerometer, Compass, Gyroscope
Personal context	Data inputted by the mobile user (updated calendar events, the phone contact lists, <i>etc.</i> )	-

## 2.2. A User-Dependent Factor: Types of App Usage

According to the real-time context inferred in the first phase, context-aware Apps provide appropriate services for a user. In this second phase, the way a user use Apps affects the power consumption. Even if a user uses the same App, the amount of consumed power will be different if it is actually used in different ways. Focusing on this point, we defined the following four properties about the way of using Apps. Here, we distinguished the terms “execution” and “activation”. “Execution” refers to the working state of Apps on smartphones (from their starts until their complete stops). “Activation” of Apps means the state of actuating some embedded sensors to get raw data during an execution.

- **Periodicity:** this factor considers whether an App is executed with regular intervals. If a user executes an App regularly (with reasonable intervals, not exact), its periodicity is “Periodic”. And we defined that the usage of an App is “Aperiodic” when it is used irregularly.
- **Frequency of execution:** this factor considers whether an App is used frequently or infrequently. For example, if an App is used 10 times ( $>\alpha$ ) a day, then we can say that it is used in a “Frequent” way. In contrast, if a user runs an App just 2 times ( $\leq\alpha$ ) a day, it is “Infrequent” usage.
- **Execution time (duration):** this factor is whether an App is executed for long time or short time. That is, this is about the temporal length of each execution of Apps. We classified characteristics of this property into two cases based on a certain threshold (time unit  $\beta$ ), “Long” execution and “Short” execution.
- **Frequency of activation:** this factor means whether an App frequently runs sensors to collect and to generate context during its execution. After an App is executed, if it actuates related sensors many times ( $>\gamma$ ) until it is completely stopped, its frequency of activation is “intensive”. Otherwise, the value of this property is “Not intensive”.

Periodicity is a property about the interval between executions, and Frequency of execution means the number of executions during the certain period (time). And Frequency of activation is the number of activations while an App is being executed.

Based on these four features, we categorized nine types of the way of App usage (b) ~ (j), as shown in Table 2.

**Table 2.** Types of App usage.

Properties of the way of using Apps				Types	
Periodicity	Frequency of execution	Execution time (duration)	Frequency of activation		
Periodic	Infrequent	Long	intensive	-	(a)
			Not intensive		
		Short	(both)		
	Frequent	Long	intensive	<b>PFLT</b>	(b)
			Not intensive	<b>PFLN</b>	(c)
		Short	(both)	<b>PFS</b>	(d)
Aperiodic	Infrequent	Long	intensive	<b>AILT</b>	(e)
			Not intensive	<b>AILN</b>	(f)
		Short	(both)	<b>AIS</b>	(g)
	Frequent	Long	intensive	<b>AFLT</b>	(h)
			Not intensive	<b>AFLN</b>	(i)
		Short	(both)	<b>AFS</b>	(j)

- (a) Ignorable cases: we can ignore four cases (PILT, PILN, PIST, and PISN) due to an assumption that the usage of an App is frequent when it is executed with reasonable intervals. Therefore, we ignored the cases of “Periodic” and “Infrequent” usage.
- (b) **PFLT**: in the way of this type, an App is executed multiple times ( $>\alpha$ ) (F) with periodic intervals (P) for a day, and its each execution lasts for a long time ( $>\beta$ ) (L), and it intensively collects and generates context during each execution (T).
- (c) **PFLN**: when a user uses an App in the way of this type, it is periodically (P) executed many times ( $>\alpha$ ) (F) and each execution has long duration ( $>\beta$ ) (L), but context collection and generation by related sensors is occurred infrequently (N).
- (d) **PFS**: an App is periodically (P) and frequently ( $>\alpha$ ) executed (F), but each execution has short duration ( $\leq\beta$ ) (S). In this case, there is no significant influence of the number of activation because the execution time is short.
- (e) **AILT**: a user aperiodically (A) executes an App a few times ( $\leq\alpha$ ) (I), but he/she uses it for a long time ( $>\beta$ ) (L) and it activates sensors frequently (T).
- (f) **AILN**: an App is aperiodically (A) executed a few times ( $\leq\alpha$ ) (I) during a long time ( $>\beta$ ) (L) and it rarely collects/generates context (N).
- (g) **AIS**: an App is executed a few times ( $\leq\alpha$ ) (I) with aperiodic intervals (A) for a day and it is stopped after a short time ( $\leq\beta$ ) (S). The influence of the frequency of activation on power consumption can be ignored, since the execution time is short.

- (h) **AFLT**: a user executes an App many times ( $>\alpha$ ) (F) aperiodically (A). The App is executed for long time ( $>\beta$ ) (L) and often run sensors to get/generate context (T).
- (i) **AFLN**: an App is executed multiple times ( $>\alpha$ ) (F) with irregular intervals (A) for a day and each execution lasts for a long time ( $>\beta$ ) (L), but context collection and generation is occurred less times (N).
- (j) **AFS**: in the way of this type, an App is executed multiple times ( $>\alpha$ ) (F) aperiodically (A) during a short time ( $\leq\beta$ ) (S). In this case, we can ignore the effect of the number of activation.

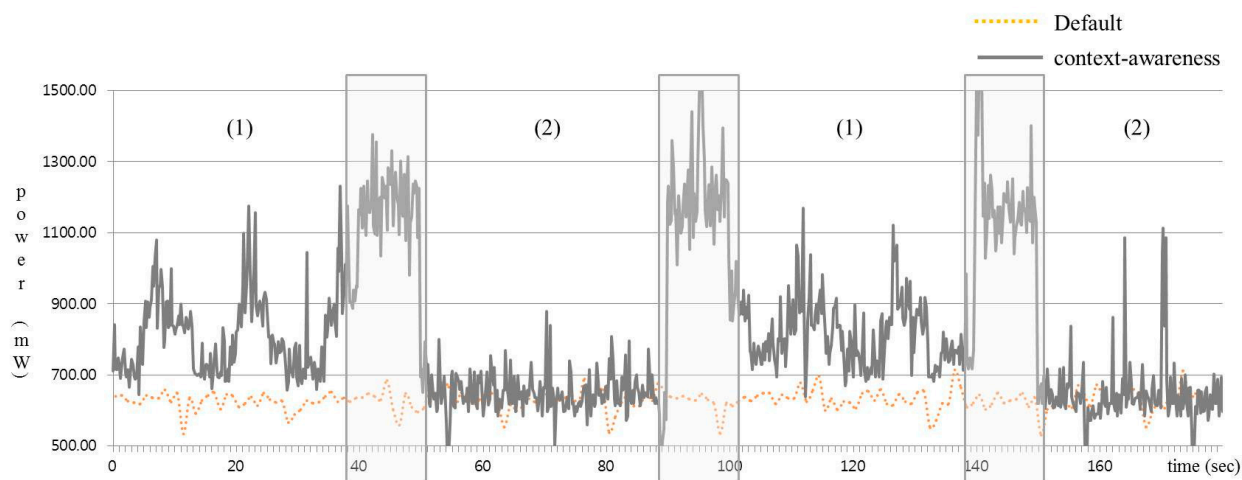
### 3. Experiment and Discussion

The goal of this paper is to discover a room for saving energy consumed by context-aware Apps of mobile devices. For this goal, we analyzed characteristics of the power consumption of context-aware Apps in Section 2. In this Section 3, some experimental results show reasonability of the analysis as well as the possibility of our goal. The experiment environment is Nexus 4 equipped with Android 4.4.2. On the test smartphone, we did not start any Apps at all, except our test program. We kept the brightness of the screen to the minimum.

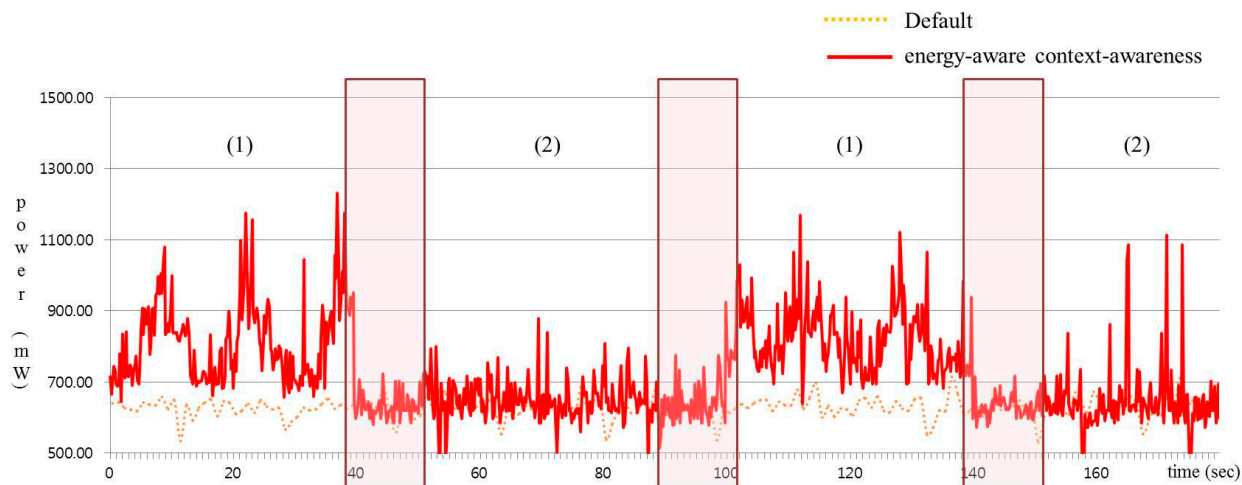
As mentioned in Section 2.1, an App-dependent factor “sorts of context” is significantly related to the power consumption. Our analysis and study will be a foundation for the development of low-power methods. As an example, Apps using spatial context generally use GPS and Wi-Fi sensors. Figure 2a shows a typical pattern of the power consumption for generating indoor or outdoor context which is a kind of spatial context. Generally, context-aware Apps activate a GPS sensor in outdoor (zone (1) in Figure 2a) and a Wi-Fi sensor in indoor (zone (2) in Figure 2a). However, if it is difficult to determine indoor and outdoor, then context-aware Apps consumes extremely more energy since they activate both of GPS and Wi-Fi (shaded zones in Figure 2a). In this case, we can apply some low-power methods to control the activation of sensors related to spatial context in order to reduce the power consumption. In Figure 2b, there is just slight difference of the power consumption in zones (1) and (2). However, as shown in the shaded zones of Figure 2b, this strategy can effectively reduce the amount of consumed power by limiting the context generation under situation hard to infer the real-time context. The result of our experiments was about 13% savings of the power consumption. Therefore, we can assure that it is possible for context-aware Apps to efficiently consume energy of mobile devices by energy-aware context-awareness methods derived from the first factor “sorts of context”.

The second experiment was to check the usefulness of a user-dependent factor “types of App usage”. For experiment, we selected typical five Apps (Family Safety [25], SeoulBus [26], Kingisa [27], Noom Walk [28], and DialApp), and measured the amount of power consumed while using each App in the way of each App usage type. Figure 3 shows the experimental result. The default case is when we kept lock screen of the test smartphone and did not execute any Apps at all. The second bar means the amount of consumed power per unit time when an App “Family Safety” is used in the PFLT way. That is, for a day, we used “Family Safety” many times with some temporal intervals, and kept each usage for a long time intentionally. This App continuously actuates a GPS sensor to trace the location of their family members. From this experimental result, we can draw some conclusions: an App consumes more power when an App is used such that the frequency of activation is relatively high. However, other factors (kinds of context used by an App, *etc.*) also have big influences on the power consumption.

**Figure 2.** The power consumption for context-awareness.

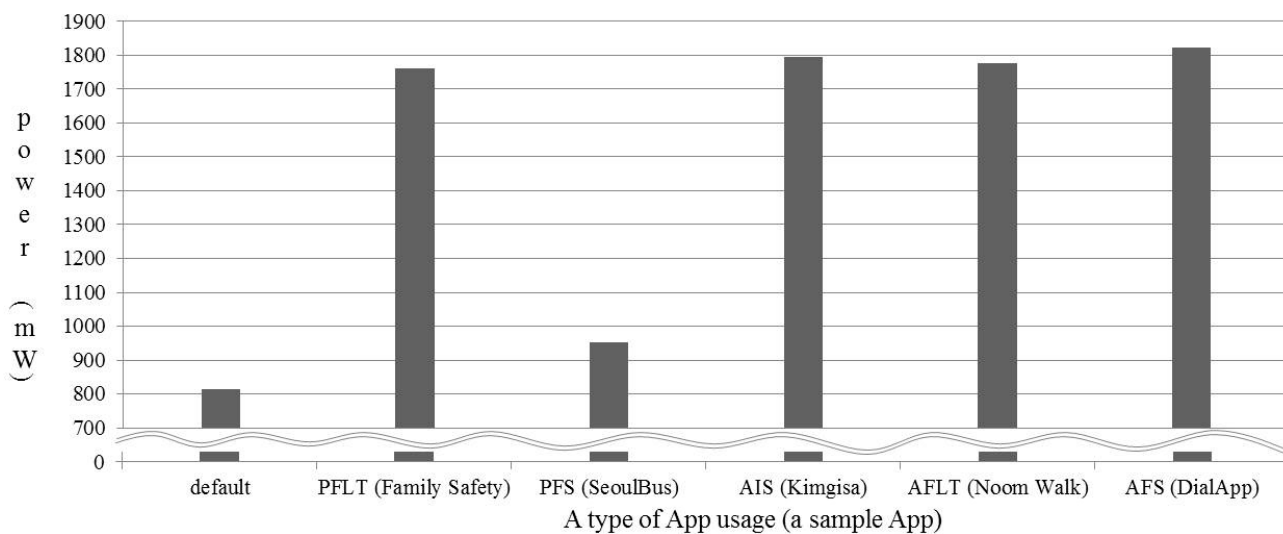


(a) the power consumption of context-awareness (indoor/outdoor)



(b) the power consumption of energy-aware context-awareness (indoor/outdoor)

**Figure 3.** The power consumption of five types of App usage.



Therefore, in order to reduce power consumed by context-aware Apps, we can propose some ideas of minimizing the number of generating context or reducing the frequency of activation of sensors, and so on. Based on our analysis, the following challenging issues or future works can be considered:

- A method to control the context generation: actually, the above first experiment is a pilot experiment for our idea. A possible low-power method can be to control the generation of context used by context-aware Apps. We can define context generation patterns or models of each “type of context”, and then control or limit the useless generation based on the model. For example, the first experiment applied a context generation control method to hold down the context generation in uncertain situations that indoor or outdoor, one of the “spatial context”, is unsure. This method can be expanded to other types of context.
- A method to control the frequency of activation: as another low-power method, context-aware Apps apply an algorithm for controlling their “frequency of activation”. According to our study in this paper, the frequency of activation of sensors is the most influential factor. Therefore, if an App is used in a way of “intensively frequent activation”, it can consider a method to effectively run sensors. For example, if the interval of activations is not reasonably long, it does not sense new sensing data and then can refer to the last one. Of course, this method should consider a type of context or characteristics of services provided by Apps.

#### 4. Conclusions

Context-aware services are emerging in mobile environments, and context-awareness is an essential technique. However, smartphones have some constraints for context-awareness, and limited power is a critical issue. Therefore, in this paper, we analyzed the characteristics of the power consumption of context-aware smartphone applications. There are two main factors which have a strong influence on the power consumption. A sort of context affects the amount of consumed power, since it determines the sort of embedded sensors to actuate. As a user-dependent factor, the way of using an App is also an influential factor. Our paper can provide an analysis for improving the energy-efficiency of context-aware Apps. With two experimental results, we showed that our analysis of characteristics of the power consumption for context-aware Apps can be useful for developing energy-saving methods. Based on our analysis, we plan to classify context-aware Apps, to analyze the style of their usage, to predict their power consumption, and to develop low-power methods. Before that, we are going to expand the analysis of power consumption and to compare the power consumed in all cases considering related factors.

#### Acknowledgments

This research was supported by the MSIP (Ministry of Science, ICT&Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (NIPA-2014-H0301-14-1017) supervised by the NIPA (National IT Industry Promotion Agency)

This paper was completed with Ajou university research fellowship of 2011 (S-2011-G0001-00079).



## Author Contributions

Meeyeon Lee, Deok-Ki Kim and Jung-Won Lee are responsible for the concept of the paper, the results presented and the writing. All authors have read and approved the final published manuscript.

## Conflicts of Interest

The authors declare no conflict of interest.

## References

1. Hoang, T.; Nguyen, T.; Luong, C.; Do, S.; Choi, D. Adaptive Cross-Device Gait Recognition Using a Mobile Accelerometer. *J. Inf. Process. Syst.* **2013**, *9*, 333–348.
2. Elgazzar, K.; Ejaz, A.; Hassanein, H.S. AppaaS: Provisioning of Context-aware Mobile Applications as a Service. In Proceedings of IEEE International Conference on Communications (ICC), Budapest, Hungary, 9–13 June 2013; pp. 2939–2943.
3. Cho, H.; Choi, M. Personal Mobile Album/Diary Application Development. *J. Converg.* **2014**, *5*, 32–37.
4. McNaull, J.; Augusto, J.C.; Mulvenna, M.; McCullagh, P. Flexible context aware interface for ambient assisted living. *Hum. Centric Comput. Inf. Sci.* **2014**, *4*, doi:10.1186/2192-1962-4-1.
5. Hong, S.; Chang, J. A New k-NN Query Processing Algorithm based on Multicasting-based Cell Expansion in Location-based Services. *J. Converg.* **2013**, *4*, 1–6.
6. Howard, N.; Cambrai, E. Intention awareness: improving upon situation awareness in human-centric environments. *Hum. Centric Comput. Inf. Sci.* **2013**, *3*, doi:10.1186/2192-1962-3-9.
7. Augusto, J.C.; Callaghan, V.; Cook, D.; Kameas, A.; Satoh, I. “Intelligent Environments: A manifesto.” *Hum. Centric Comput. Inf. Sci.* **2013**, *32*, doi:10.1186/2192-1962-3-12.
8. Oh, S. Using an Adaptive Search Tree to Predict User Location. *J. Inf. Process. Syst.* **2012**, *8*, 437–444.
9. Bisio, I.; Lavagetto, F.; Marchese, M. Context-Aware Smartphone Services. In *Pervasive Computing and Communications Design and Deployment: Technologies, Trends and Applications*, 1st ed.; Information Science Reference (IGI Global): Hershey, PA, USA, 2011; pp. 24–47.
10. Korpipaa, P.; Mantyjarvi, J.; Kela, J.; Keranen, H.; Malm, E.-J. Managing Context Information in Mobile Devices. *IEEE Pervasive Comput.* **2003**, *2*, 42–51.
11. Chon, Y.; Cha, H. Lifemap: A Smartphone-based Context Provider for Location-based Services. *IEEE Pervasive Comput.* **2011**, *10*, 58–67.
12. Noh, H.Y.; Lee, J.H.; Oh, S.W.; Hwang, K.S.; Cho, S.B. Exploiting Indoor Location and Mobile Information for Context-awareness Service. *Inf. Process. Manag.* **2012**, *48*, 1–12.
13. Yurur, O.; Labrador, M.; Moreno, W. Adaptive and Energy Efficient Context Representation Framework in Mobile Sensing. *IEEE Trans. Mob. Comput.* **2013**, *13*, 1681–1693.
14. Aldhino, A.; Kim, Y.J.; Rim, K.W. A Survey of Low-power Techniques for Liquid Crystal Display Systems with Light Emitting Diode Backlight Units. *IETE Tech. Rev.* **2011**, *28*, 351–361.
15. Woo, S.; Seo, W.; Kim, C.; Huh, J. User Input based Power Reduction Technique for Smartphone. *J. KIISE* **2013**, *40*, 45–51. (In Korean)

16. Liu, J.; Priyantha, B.; Hart, T.; Ramos, H.S.; Loureiro, A.A.F.; Wang, Q. Energy Efficient GPS Sensing with Cloud Offloading. In Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems (SenSys'12), Toronto, ON, Canada, 6–9 November 2012; pp. 85–98.
17. Bae, S-K. Power Consumption Analysis of Prominent Time Synchronization Protocols for Wireless Sensor Networks. *J. Inf. Process. Syst.* **2014**, *10*, 300–313. (In Korean)
18. La, C.-A.; Varga, L.-O.; Heusse, M.; Duda, A. Energy-Efficient Multi-Hop Broadcasting in Low Power and Lossy Networks. In Proceedings of the 17th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM' 14), Montreal, QC, Canada, 21–26 September 2014; pp. 41–50.
19. Khairy, A.; Ammar, H.H.; Bahgat, R. Smartphone Energizer: Extending Smartphone's Battery Life with Smart Offloading. In Proceedings of the 9th International Wireless Communications and Mobile Computing Conference (IWCMC'13), Cagliari and Sardinia, Italy, 1–5 July 2013; pp. 329–336.
20. Murmura, R.; Medsger, J.; Stavrou, A.; Voas, J.M. Mobile Application and Device Power Usage Measurements. In Proceedings of the 6th IEEE International Conference on Software Security and Reliability (SERE' 12), Gaithersburg, MD, USA, 20–22 June 2012; pp. 147–156.
21. Wang, C.; Yan, F.; Guo, Y.; Chen, X. Power Estimation for Mobile Applications with Profile-Driven Battery Traces. In Proceedings of the 2013 International Symposium on Low Power Electronics and Design (ISLPED' 13), Beijing, China, 4–6 September 2013; pp. 120–125.
22. Zhan, K.; Lung, C.-H.; Srivastava, P. A Green Analysis of Mobile Cloud Computing Applications. In Proceedings of the 29th Annual ACM Symposium on Applied Computing (SAC' 14), Gyeongju, Korea, 24–28 March 2014; pp. 357–362.
23. Tu, Y. xTalk—a Context-aware Mobile Application on the Nokia N95 8GB Smartphone. Master's Thesis, Technical University of Denmark, Denmark, 2008.
24. Pathak, A.; Jindal, A.; Hu, Y.C.; Midkiff, S.P. What is Keeping My Phone Awake?: Characterizing and Detecting No-sleep Energy Bugs in Smartphone Apps. In Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services (MobiSys'12), Low Wood Bay, UK, 25–29 June 2012; pp. 267–280.
25. zoemob (Family Safety). Available online: <https://www.zoemob.com/> (accessed on 15 August 2014).
26. SeoulBus. Available online: <http://seoulb.us/> (accessed on 15 August 2014).
27. Kingisa. Available online: <http://cafe.naver.com/kingisanavi> (accessed on 15 August 2014).
28. Noom (Noom Walk). Available online: <http://www.noom.com/> (accessed on 15 August 2014).