Energy Efficient Sensor Management Strategies in Mobile Sensing

Özgür Yürür

Department of Electrical Engineering
University of South Florida
Tampa, Florida, 33620
oyurur@mail.usf.edu

Wilfrido Moreno

Department of Electrical Engineering
University of South Florida
Tampa, Florida, 33620
wmoreno@usf.edu

Abstract — Mobile device based human-centric sensing and participatory sensing provide a vast context about the user state information. To capture and recognize any user state, then to classify it requires operating all existing sensors in a mobile device continuously. Nonetheless, constantly running sensors drain the mobile device's battery rapidly. Therefore, it is imperative to construct a framework to utilize sensors efficiently and try to recognize user states very accurately by consuming less power. In this paper, some strategies are proposed to correct the mentioned deficiency. It is aimed to detect and recognize user state transitions by first decreasing computational complexity during data processing stages, prompting time-varying efficient duty cycle schedules and setting adaptive sampling time periods to satisfy trade-off between accuracy vs. energy consumption.

Keywords: Energy Efficiency, Mobile Sensing, Sensor Management Strategies, Ubiquitous Sensing.

I. INTRODUCTION

Nowadays, mobile phones are powerful devices since they are not only used for their fundamental purposes like calling or texting but also used for browsing the Internet, recording voices, taking instant snapshots, tracking any geo-location and so on. These mobile phones manage to perform these kinds of rich features using their on-board sensors like the accelerometer, Bluetooth, camera, GPS, microphone, Wi-Fi, and etc. Therefore, they are introduced to electronics consumers as smart phones. Due to technological advances and consumer demands, it is likely that consumers will anticipate and expect future smart phones to have more sensors and more sensor-based applications.

By utilizing sensors, some meaningful information about user locations, routines and surroundings can be extracted in real-time, allowing some applications to adapt to constantly changing environmental conditions and user preferences. As an example of the user activity sensors, an online application can be used for socializing platforms to update current user locations so that user followers can track the user.

Continuously capturing user context and extracting information make the mobile phone very busy, which causes it to draw a great deal of current from the device battery. This consequence will be very detrimental when powering more than one sensor. As a result, the device battery would die very quickly. It has been reported [1] that today's mobile devices are not durable enough to use all sensors at the same time, giving an example of the Nokia N95 mobile phone with a new fully-charged battery. It experimentally examined that the phone would be totally drained within six hours if the GPS was switched on permanently even when not being actively used; whereas the phone supported telephone conversation up to ten hours. In summary, to be able to run context extracting applications, energy saver algorithms must be considered and applied.

The best energy saving algorithm would consist of powering a minimum set of sensors to recognize any user state transition. Assigning different operation duty cycles to different sensors is proposed in this method so that at any time, there might be some sensors running, while others might be shut down. For example, if a user stays in-doors, there would be no reason to power-up the GPS sensor while accelerometer and Wi-Fi might be open. A novel approach would also be to change the duty in cycles, which means tuning the percentage of Logic '1' statuses of active wave forms, which power a sensor. Suppose that user is reading a book

and as time passes by, the activity of the user will become less, except for sitting. In this case, active time for powering the accelerometer to recognize body movement can be shortened.

Besides adjustable duty cycles, adaptively changing sampling periods is another novel approach to the problem. A mixture mechanism for utilizing the combined pair assignments of different duty cycles and sampling periods would be a cure for consuming less energy for user state recognitions. Looking at the previous example (the user who reads a book), if the user suddenly starts to sleep, there is a great probability of user's body movement reaching almost zero until waking. Also supposing there are some limitations such as the upper or the lower level percentage of duty cycling, then when time passes by, due to non-activity (the same user state all the time), the duty cycle will go down below the lower level. However, the mechanism should not let this happen, and can then change the current sampling period and enlarge it by keeping the current duty cycle fixed. On the other hand, an opposite example would be reaching above the upper level, and this time sampling period would be shortened. Unfortunately, this mechanism cannot be adapted to every sensor since those sensors must run under some dedicated transmission protocols. For example; Bluetooth, GPS and Wi-Fi are inside of that category of sensors while the accelerometer, microphone, and camera are not.

Reducing power consumption for the system operation perspective does not mean that recognition of user state transitions is not concise. There is a challenging tradeoff between power consumption and accurate user context extraction. While providing less power consumption, the system has to be sure that the current user state is accurate and any state transition is detectable. Most importantly, due to the duty cycling approach, some user state(s) might be missed. The system also has to be sure that these missing user state transitions are estimated.

This paper addresses novel approaches to solve the complexity of the trade-off, and represents them in detail. Proposed methods are summarized as follows: First, after each sensor reading, before recognition of user states, a sensor-specific buffered-feedback decision maker is employed. For some sensors like the accelerometer or GPS, a threshold value should exist, which can be seen as an inertia value, for finding out any change of current user state. Threshold is used to compare newly captured samples with previous samples stored inside of the corresponding sensor-specific buffer. For other sensors like the microphone or Bluetooth, a specialized digital signal processing block is needed to anticipate the information of no status changes in user states without performing more complex processing algorithms. Secondly, to recognize any user state, a minimum set of sensors is selected. Each sensor has its own different operation methods which are based on a set of time-varying duty cycles and a set of time-varying sampling periods (fast, normal or slow sampling periods). Therefore, according to each selected sensor, the energy cost estimation is done before beginning the sensor operation. Then, the selected sensor begins to operate. Operation duration and frequencies can alter in time.

II. SUMMARY OF PRIOR WORKS

There are a fair number of works which have been proposed for mobile sensing to recognize user states accurately enough by trying to consume less power; however most of those works provide only partial answers to the tradeoff between data accuracy and less power consumption, and there has not been much work done for constructing a total framework. At this point, it is worth mentioning that Wang *et al*. [1] proposes a sensor management system, which is called Energy Efficient Mobile Sensing System (EEMSS). The system improves device battery life by powering a minimum set of sensors and applying sensor duty cycles. However, sensors have fixed duty cycles whenever they are active, and they are not adjustable to different user behaviors. The same authors in [2] study how sensor duty cycles can be optimized in order to minimize the expected user state estimation errors while maintaining an energy consumption budget. The hierarchical sensor management system is also studied by introducing "SeeMon" system in [3] which achieves energy efficiency and less computational complexity by only performing a continuous detection of context recognition when changes occur during the context monitoring. Moreover, to find a solution for the tradeoff, the advantage of a dynamic sensor selection scheme for accuracy power tradeoff in user state recognition is demonstrated in [4]. Rachuri *et al*. [5] also uses

different sampling period schemes for querying sensor data in continuous sensing modes in mobile systems to evaluate energy-accuracy tradeoffs. This solution makes an attempt to solve the problem, but at this time for localization applications the results can be found as follows: At first, "SenseLess", described in [6], is a system for saving energy consumption by sensing localization applications for mobile phones. Then, Constandache *et al.* [7] studies energy efficiency in mobile device based localization, and the authors show that humans can be profiled based on their mobility patterns and thus location can be predicted. The proposed "EnLoc" system achieves good localization accuracy with a realistic energy budget.

III. PROPOSED STRATEGIES FOR ENERGY EFFICIENCY

Mobile phones today have a number of sensors which are capable of actualizing interesting applications. However, updating new features into these devices brings about some limitations on their battery life. Therefore, sensors which are constantly employed by any application can cause high power consumption per time unit compared to what they do while a telephone conversation is being made. In [6], the approximate effect of present-day mobile phone sensors on device battery life is investigated by experimentally finding out their average power consumptions when they are individually turned on. According to the results, the least power consuming sensor is the accelerometer. The Bluetooth, microphone, GPS, Wi-Fi and video camera follow accelerometer respectively. Therefore, the best battery-care approach would be that user state recognition based applications should start their operations by using the accelerometer as a default sensor.

In this paper, there are two model approaches proposed for improving the tradeoff between power consumption and accuracy of user state recognitions. The first approach is called a context monitoring mechanism, which basically considers the achievement of energy efficiency in a computational manner; and the second one is called an adaptively sampling and duty cycling method, which aims to obtain energy efficiency in a physical manner while sensors are being operated.

A. Context Monitoring Mechanism

A major challenge in extracting user contextual information is based on monitoring the context continuously. Constantly context monitoring in a sensor-rich mobile phone imposes heavy workloads which cause some limitations in computing and battery power. Therefore, there should be a mechanism to organize required processes for analyzing the context in such way that redundant repetitions of the same contextual information can be avoided.

Once a transition occurs in a new user state, there is no necessity of recognizing and notifying the same context (existing of same conditions) redundantly again and again as long as the user state remains unchanged. A mechanism can be considered to elaborate computational complexity during the processing of a sensor data; as a result, a decision on the continuity of user states can be made at an early age of the total processing pipeline by achieving significant computational overhead save.

As illustrated in Fig.1, detection of user contextual information is recognized by employing an intelligent computational method. It is aimed at performing an inexpensive strategy without completely exploiting a raw sensor data to make the decision on the continuity of current user state.

The conventional user state recognition method tries to process and analyze raw sensor data, exploit context information, and compare it with a classification algorithm based on decision tree logic. In this method, almost all raw sensor data is carried through a processing pipeline, and ends with a user state decision. In the proposed approach, an intelligently pipelined context monitoring mechanism is introduced to reduce the high cost for required operation sequences. The main objective of the mechanism is to notify user states of a running application only when a user state transition occurs.

In Fig.1, the sensor produces a discrete raw data periodically. The data enters into a processing pipeline; output of the pipeline is to inform possible change of user states or not since the data is found to be identical to previous ones which were buffered.

The pipeline starts with a preprocessing structure. This structure basically filters out the required information from the raw sensor data. The information varies from one sensor to another. For example; the accelerometer and microphone give away only one part of the required information whereas Wi-Fi and GPS return a package that includes different kinds of information. Therefore, there might exist more than one piece of monitored information sent by the sensors. Different amounts of contextual information forces it to have the same number of information buffers. Each buffer has the same kind of contextual data which belongs to previous samplings. Buffers are designed to gain flexibility and speed on decision making, and also, most importantly, to help in decreasing the number of redundant computational operations which can yield to stopping the pipeline before feature extraction algorithms are applied. For example; let us imagine that the accelerometer converts an analog signal and returns related discrete digital data a hundred times in a second. Since any user state cannot be changed drastically that sensor misses transitions, there is no point of processing all algorithms at each sensor reading, and then send detected user state information to running application all the time. At that point, a tradeoff might come up for buffer size that resolves the accuracy of the decision on deciding current user state.

Buffers consist of comparison variables which are obtained from statistical analysis on the previously stored contextual information. Comparison variables can be standard deviation, variance or any other result of applied statistical tools. At the middle stage of the processing pipeline, decision maker classifications are applied among current and previous contextual information to interrogate the existence of future processes such as feature extraction. The feature extraction block is the place where new and different contextual information is exploited to bring about a possible user state transition. For instance; if an environment is detected as silence before, and if newly obtained data indicates there is a changing situation in environmental context, the feature extraction block analyzes the data and sends out the resulting outcome to the user state detector block to decide on which user state will be selected. Would it be 'loud', 'music' or 'speech'?

The user state detector block recognizes user state transitions, and informs the applications. Another important property belonging to this block is to assign a required sensor set for the recognition of current user states and possible user state transitions.

The duty cycle and adaptive sampling adjuster block keep tracking the length of time intervals in which sensor readings do not yield that cause it to miss any user state transition. Therefore, depending on how long time is spent on a user state, duty cycles and adaptive sampling periods can be adjusted to secure energy efficiency.

Lastly, the application is notified when a user state transition occurs. It also queries the mechanism any time a user defined event is wanted to be queried to check whether it might be happening or not.

B. *Adaptively Sampling and Duty Cycling*

Sensors on a mobile phone can be categorized into two classes depending on their operation styles. The first class includes the accelerometer and microphone. These sensors are turned on once to operate continuously; they need an external command to finish their operations and to become turned off. They both require an optimal minimum sampling period to capture meaningful contextual user information. The corresponding sampling period to each sensor can be adaptively changed in case some power considerations are taken. The second class of sensors includes GPS, Wi-Fi and Bluetooth. When these sensors are turned on, they automatically turn into the idle position after the required sensing operation is done. Also, there are some dedicated transmission protocols that these sensors must follow to accomplish their operations. Therefore, they might be unable to run them under different sampling periods. Although Bluetooth is employed in the second class while scanning nearby users, it may be also considered a member of the first class as well while its spectrum is being sensed and analyzed with different sampling periods.

1) *Sensor Operation Structure*

The energy cost of any sensor does not only depend on instant energy consumption per operation, but also on the duration of the operation itself. In Fig.2, sensor operation structures for both classes are illustrated. This structure starts with an

initialization block, and that block deals with waking the sensor up and then waiting for an acknowledged response which informs that the sensor is ready. For example; the study in [1] states that for GPS functionality, it requires at least 10 seconds to successfully synchronize with satellites. For other sensors, a shorter time period would be sufficient to power them up and to set their initial system requirements before sampling operations begin. The second block is called processing. This block is dedicated to providing efficiency in energy consumption. In this block, the sensor starts to capture user contextual information and continue on

this operation repeatedly. The third block ends the active duty of the sensor, and terminates it. After all these three blocks, the sensor shuts down until a new duty is assigned.

The processing block is the place where the duty cycle length and sampling period are adjusted dynamically for a sensor. Energy consumption is reduced by carefully assigning a pair of the duty cycle and sampling period. However, these assignments are going to cause a tradeoff between reduced energy consumption and accurate sensing. If successive sampling intervals are too long, there
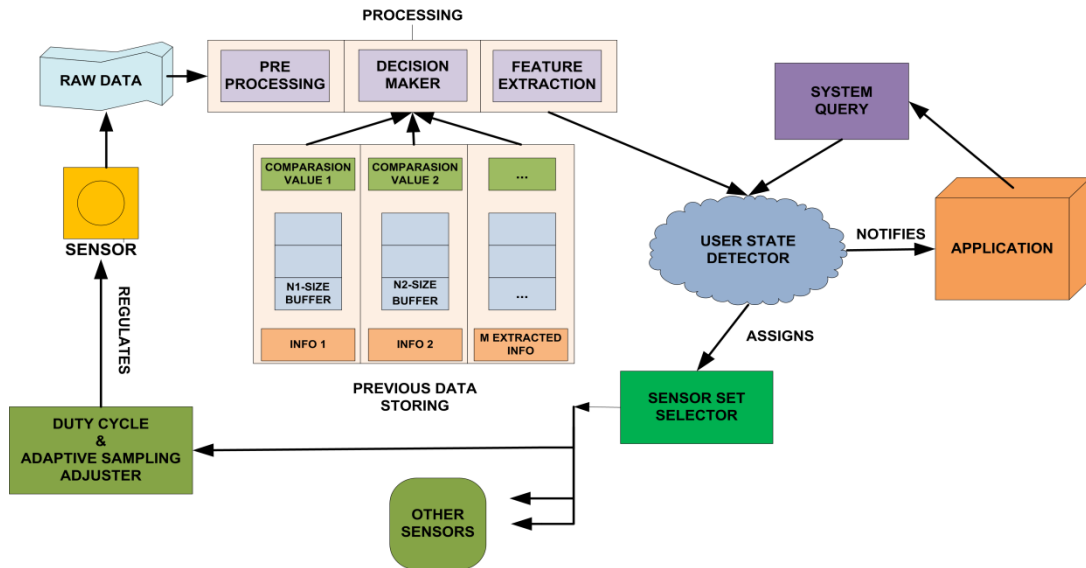

Figure 1.   Context Monitoring Mechanism

would not be sufficient samplings to present real conditions, and eventually it would cause it to sense user contextual information incorrectly from the sensed data. On the other hand, in the case that the intervals are too short, energy saving would not be satisfied. The same approach can also be applied to sensor sleeping time intervals as well. A longer sleeping time interval would reduce energy consumption; nonetheless, detection latency will be increased so that false detections could occur.

2)   Preliminaries and Construction of the Proposed Approach

Assuming that an application needs to employ a sensor, and β_{total}, t_{total} and t_c are given for the application as the total energy budget for accomplishing all sensing operations, the total active

time for the sensor and the corresponding time to drive a sensor per a cycle respectively. ϑ_{sample} and ϑ_{idle} are also given as constant default sensor properties for energy consumptions to sense a contextual data per time unit and to run with no operation while being idle per time unit respectively.

A novel approach is applied to the sensor, which enables it to change the duty cycle of active sensor prompting signal and sampling periods adaptively during sampling operations being executed. Therefore, a set of duty cycles and a set of sampling frequencies need to be introduced to the system.

$$DC(i) = [\ 0.2,\ 0.4,\ 0.5,\ 0.6,\ 0.8\ ] \qquad (1)$$

DC(i) is a required set for duty cycles. It is limited to 20% for the lowest level because below this level would bring high risk for recognizing user states accurately enough. On the other hand, at above 80% would help to recognize user states much more accurately has and avoid having energy wasted unnecessarily.

$$Fs(j) = [\ fs_{slow2},\ fs_{slow1},\ fs_{normal},\ fs_{fast1},\ fs_{fast2}\ ] \quad (2)$$

Different sampling frequencies are introduced into the system inside of a set called $Fs(j)$.

With the combined existence of $DC(i)$, $Fs(j)$, and t_c, the number of occurrences of sampling operations inside of an active cycle can be calculated as follows:

$$N(\ i, j\ ) = (\ DC^{T}\ x\ Fs\ )\ t_c \quad (3)$$

$N(i,j)$ is a [*length of DC(i) x length of Fs(j)*] matrix which gives various numbers of sample operation when a specific i and a specific j are selected together by the system. As illustrated in Fig.2, t_{run} can be extracted as:

$$t_{run} = t_{total} - (t_{init} + t_{term}) \quad (4)$$

Due to the fact that duration of t_{init} and t_{term} are approximately constant all the time for a sensor, the value of t_{run} yields to have the number of running active cycles during the processing block is active.

$$N_{run} = t_{run}\ /\ t_c \quad (5)$$

With the knowledge of how many times active cycle runs during processing and how many times sampling occurs inside of each active cycle, total energy consumption, θ_{run}, can be estimated.

$$\Theta_{run} \approx \sum_{n=1}^{N_{run}} [\ (\ \sum_{k=1}^{N(i=selected,j=selected)} \delta(k)\vartheta_{sample}) \\ + (1 - DC(i = selected))t_c\vartheta_{idle}] \quad (6)$$
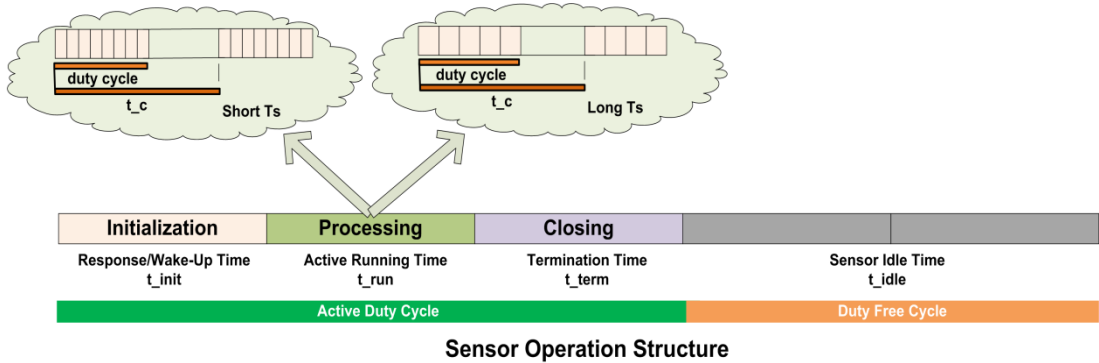

Figure 2. Adaptively Sampling and Duty Cycling

Note that idle times between two consecutive samplings are ignored; and for the calculation of energy consumption during idle times, continuous time is used; whereas, for the calculation of energy consumption at sampling times, discrete time is used.

Since the proposed approach adjusts a combination pair of the duty cycle and sampling frequency adaptively as each active cycle runs, the number of samples inside an active cycle is also changed adaptively. Thus, energy consumption at active task differs at each cycle. Plus, energy consumption of idle task also differs because of various duty cycle selections. As a result, total energy consumption will depend on aggregation of total wasted energies at every cycle.

The proposed method aims to consume less energy than given β_{total} as an ultimate goal. Assuming that θ_{init} and θ_{term} are energy

consumptions while sensor is initializing and terminating, the required inequality becomes:

$$\theta_{init} + \theta_{term} \approx (\, t_{run} + t_{term}) \, \vartheta_{idle} \qquad (7)$$

$$\beta_{total} \geq \theta_{init} + \theta_{run} + \theta_{term} \qquad (8)$$

The efficiency, ζ, gained after less energy consumption from anticipated energy budget, β_{total}, is calculated as:

$$\zeta = \frac{\beta_{total} - (\Theta_{init} + \Theta_{run} + \Theta_{term})}{\beta_{total}} 100\% \qquad (9)$$

IV. FUTURE WORK

Integration of the proposed method related to adaptively sampling and duty cycling, implementation of possible algorithms and simulation of them are considered as a future work for this study.

V. CONCLUSION

Novel approaches are introduced in this paper, which attempt to control mobile phone sensors in such a way that correct user state recognitions are still obtained while reducing energy consumption. The paper includes important information on how the system framework should be constructed and modeled.

REFERENCES

[1] 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 MobiSys '09: Proceedings of the 7th international conference on Mobile systems, applications, and services, New York, NY, USA, 2009, pp. 179–192.

[2] Y. Wang, B. Krishnamachari, Q. Zhao, and M. Annavaram, "Markovoptimal sensing policy for user state estimation in mobile devices," in IPSN '10: Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks. New York, NY, USA: ACM, 2010, pp. 268–278.

[3] 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 MobiSys '08: Proceeding of the 6th international conference on Mobile systems, applications, and services. New York, NY, USA: ACM, 2008, pp. 267–280.

[4] P. Zappi, C. Lombriser, T. Stiefmeier, E. Farella, D. Roggen, L. Benini, and G. Trster, "Activity recognition from on-body sensors: Accuracypower trade-off by dynamic sensor selection," in EUROPEAN CONFERENCE ON WIRELESS SENSOR NETWORKS. Springer-Verlag, 2008, pp. 17–33.

[5] K. K. Rachuri, M. Musolesi, and C. Mascolo, "Energy-accuracy trade-offs in querying sensor data for continuous sensing mobile systems."

[6] F. Ben Abdesslem, A. Phillips, and T. Henderson, "Less is more: energy efficient mobile sensing with senseless," in MobiHeld '09: Proceedings of the 1st ACM workshop on Networking, systems, and applications for mobile handhelds. New York, NY, USA: ACM, 2009, pp. 61–62.

[7] I. Constandache, S. Gaonkar, M. Sayler, R. Choudhury, and L. Cox, "Enloc: Energy-efficient localization for mobile phones," in INFOCOM 2009, IEEE, April 2009, pp. 2716 –2720.