A Middleware Solution for Optimal Sensor Management of IoT Applications on LTE Devices

Satyajit Padhy, Hsin-Yu Chang, Ting-Fang Hou, Jerry Chou, Chung-Ta King, and Cheng-Hsin Hsu(✉)

Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan
chsu@cs.nthu.edu.tw

Abstract. After many devices that have adopted LTE technology, it is optimistic to presume that 5G technology will have to address the huge traffic of data and volume of heterogeneous devices in future. Existing context-aware Internet of Things (IoT) applications directly control sensors on LTE devices in an uncoordinated and non-optimized manner, which leads to redundant sensor activations and energy wastage on resource-constrained IoT devices. Optimal and coordinated sensor usage dictates a comprehensive middleware solution to bring together the information from all IoT applications/sensors and intelligently select the best set of sensors to activate. In this paper, we design, implement, and evaluate a sensor management middleware for LTE devices that controls the tradeoff between energy consumption of sensors and accuracy of inferred contexts. The core task of this middleware is to minimize total energy consumption while making sure that the accuracy requested by IoT applications are met. Trace-driven simulations are conducted to demonstrate the merits of the proposed middleware and algorithms. The simulation results indicate that the proposed algorithms clearly outperform the current solution.

Keywords: IoT applications · Sensor management · LTE device · Context-aware

1 Introduction

Increasingly more Internet of Things (IoT) applications (apps) on LTE devices leverage the rich set of sensors to infer their contexts for enhancing user experiences. As we are progressing towards the IoT [8], a number of context aware IoT applications are being produced to take advantage of sensors available on LTE devices. In the future, multiple IoT applications running at the same time on an LTE device may request a multitude of overlapping contexts, e.g., location and time. For example, location awareness [10] can help to introduce some resource allocation techniques which will help to reduce delay by predicting channel quality. A context aware adaptive system [1] is required in the middleware layer to

J.-H. Lee and S. Pack (Eds.): QShine 2016, LNICST 199, pp. 283–292, 2017.
DOI: 10.1007/978-3-319-60717-7_28

address the heterogeneous data being generated from IoT applications. While a context may be answered by different sets of sensors depending on the requested accuracy and availability of sensors, uncoordinated and non-optimized use of the sensors by the multiple apps may turn on redundant sensors, leading to wastage of energy.

Choosing the *best* set of sensors to activate in order to satisfy the needs of various context-aware apps is very challenging. This is because there exists a tradeoff between context inference accuracy and energy consumption of sensors. On top of that, context-aware apps impose diverse accuracy requirements and LTE devices have different remaining battery levels at different time. Therefore, efficiently determining the set of sensors to activate *dictates* a comprehensive mobile middleware solution, which brings together various information from apps and sensors. In this paper, we propose a sensor management middleware, which sits between the context-aware apps and sensors. The middleware achieves *coordinated* and *optimized* uses of sensors, and provides efficient sensor management service to the context-aware apps.

The core of the middleware is the *sensor management* algorithm, which is repeatedly invoked to adapt to system dynamics. In this paper, the sensor management algorithm will optimally chose the best set of sensors for various context requests from multiple IoT applications on an LTE device. We develop two mathematical formulations of the sensor management problems: (i) energy optimization, which strives to find the set of sensors that consumes the least energy while satisfying the sensing requirements, and (ii) accuracy optimization, which strives to maximize the overall accuracy under an energy budget. Since low latency is one of the most important criteria in 5G technology, we develop two heuristic, real-time sensor management algorithms for resource-constrained mobile devices.

The rest of this paper is organized as follows. We survey the literature in Sect. 2. Section 3 describes the proposed middleware and proposed sensor management algorithms. Sect. 4 gives the trace-driven simulation results. Section 5 concludes the paper.

2 Related Work

Mobile context sensing has been studied in the literature. However there has not been much work done for context and sensor management related to 5G IoT environments. Taranto et al. [10] has proposed methodologies about how location aware context can be useful for 5G architecture. They briefly describe how location awareness can be leveraged across different layers of protocol stack on 5G architecture. Perera et al. [8] have surveyed various context aware computing methodologies that have been addressed in context aware IoT applications. They state that a large number of solutions exist in terms of system, middleware and application; however none of them addresses our core issue. Most existing studies on context-aware LTE IoT (smartphone) apps [2–4,6,11] consider location sensing. For example, Ma et al. [6] propose a system to predict the future locations of a mobile user based on his/her previous locations. Their prediction algorithm employs sensor readings from GSM and WiFi for coarse localization, which is more energy efficient than using GPS sensors. Different from our proposed OSM middleware, these studies [2–4,6,11] only consider location sensing, and thus their solutions are inapplicable to our problem. Contexts other than location have also been recently investigated [5,9,13]. For example, Yan et al. [13] employ accelerometers to classify the mobile user actions, e.g., stand, walk, and sit. None of the studies [5,9,13] consider the inter-dependency among inference algorithms of different contexts: each context is inferred independently.

3 Proposed System

3.1 System Architecture

Our proposed middleware sits between apps and the hardware. Many context-aware apps run on LTE devices, which may need different contexts at diverse accuracy and frequency. We collectively call a pair of accuracy and frequency as *request*. Apps may *register* or *unregister* requests through an Application Programming Interface (API) at any time. Each set of sensors is referred to as a *combination* in this paper. For example, a context `IsDriving` may be inferred by a combination of the GPS and the accelerometer. Moreover, a context may be inferred by various combinations, which renders the decisions even harder. For instance, `IsDriving` may also be inferred using the microphone. As illustrated in Fig. 1, the middleware consists of an API and four software components: (i) request manager, (ii) resource manager, (iii) context analyzer, and (iv) system model. The **request manager** keeps track of all registered requests and apps with a queue. It also checks if the callback function invocation fails, and automatically unregisters all the requests from any failed (exited) apps.

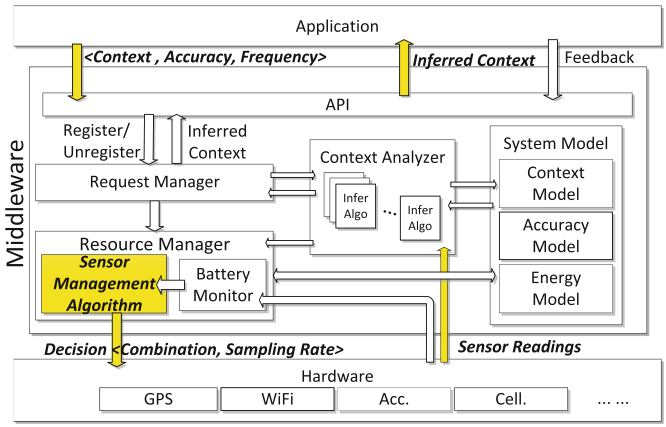

Fig. 1. The proposed middleware. Italic font indicates the focused components of our work.

The **resource manager** focuses on resource conservation and consists of two components: the battery monitor and sensor management algorithm. The sensor management algorithm takes the aggregated requests and system models as inputs, and generates decisions that activate the combinations of sensors and specify their sampling rates. The sensor management algorithms can either: (i) maximize the overall accuracy under a given energy budget or (ii) minimize the total energy consumption while achieving target accuracy levels which are inputs from apps or users. The **context analyzer** analyzes the sensor readings to infer contexts by hosting various inference algorithms for different combinations and contexts. The **System model** contains three parts: (i) context model, (ii) accuracy model, and (iii) energy model. The context model stores the relationship among contexts, inference algorithms, and sensor combinations, e.g., the action inference algorithm uses the accelerometer and WiFi to classify the user actions, such as walk, run, and still. The accuracy model captures the accuracy of the contexts inferred by the inference algorithms. Different metrics, such as *precision* and *recall* can be used to quantify the inference accuracy. The energy model captures the energy consumption of each sensor at different sampling rates.

We let R be the total number of requested contexts and S be the total number of sensors. We define a request as $<y_r, f_r>$, where r $(1 \leq r \leq R)$ is the requested context, y_r is the target accuracy, and f_r is the desired frequency. We let C be the total number of potential sensor combinations. We employ a boolean matrix $\mathbf{M}$ to capture the relation between combinations and sensors[1]. In particular, we let $m_{c,s} = 1$ $(1 \leq c \leq C,\ 1 \leq s \leq S)$ if combination c contains sensor s, and $m_{c,s} = 0$ otherwise. We collectively call all $a_{c,r}$ as $\mathbf{A}$. Last, we use e_s to denote the energy consumption of sensor s, where $1 \leq s \leq S$, in the next management window T. We write a decision as $<x_s, p_s>$, where x_s indicates whether the sensor s $(1 \leq s \leq S)$ should be activated, and p_s represents the sampling rate. Next, we present the two sensor management problems.

Problem 1 (Energy Minimization: EM). *Given requested contexts r $(1 \leq r \leq R)$ and combinations c $(1 \leq c \leq C)$, the EM problem selects a subset of combinations to achieve the minimum energy consumption while satisfying all the accuracy requirements y_r $(1 \leq r \leq R)$. Upon the combination subset is chosen, the decision is set based on the relation between combinations and sensors $(\mathbf{M})$. The EM problem is a NP-complete problem.*

Problem 2 (Accuracy Maximization: AM). *Given requested contexts r $(1 \leq r \leq R)$, combinations c $(1 \leq c \leq C)$, and an energy budget E, the AM problem selects a subset of combinations to maximize the achieved accuracy without exceeding the energy budget. Upon the combination subset is chosen, the decision is set based on the relation between combinations and sensors $(\mathbf{M})$. The AM problem is a NP-complete problem.*

[1] Throughout this paper, we use bold font to denote vectors or matrices.

3.2 Efficient Energy Minimization Algorithm (EEMA)

The EEMA algorithm maintains a set $\hat{\mathbf{R}}$, $\mathbf{X}$ of unmet requests and the chosen sensors so far. We define utility of a combination as a fraction of *profit* and *cost*. The profit is the number of unmet requests that can be satisfied by the combination, and the cost is the additional energy consumption, if the combination is chosen. The utility $g_c(\mathbf{A}, \mathbf{M}, \mathbf{X}, \hat{\mathbf{R}})$ of a combination c ($1 \leq c \leq C$) is written as:

$$g_c(\mathbf{A}, \mathbf{M}, \mathbf{X}, \hat{\mathbf{R}}) = \frac{p_c(\mathbf{A}, \hat{\mathbf{R}})}{w_c(\mathbf{M}, \mathbf{X})} = \frac{\frac{\sum_{1 \leq r \leq R, r \in \hat{\mathbf{R}}} \mathbf{1}_{[a_{c,r} \geq y_r]}}{\sum_{1 \leq r \leq R} \mathbf{1}_{[a_{c,r} \geq y_r]}}}{\sum_{1 \leq s \leq S, s \notin \mathbf{X}} m_{c,s} e_s}, \tag{1}$$

where $\mathbf{1}$ is the indicator function, $\mathbf{A}$ is the accuracy model, and $\hat{\mathbf{R}}$ is the set of unmet requests and where $\mathbf{M}$ is the boolean matrix of relation between combinations and sensors and $\mathbf{X}$ keeps track of the chosen sensors so far. We note that the denominator of Eq. (1), $w_c(\mathbf{M}, \mathbf{X})$, could be zero because some sensors may be always on for basic LTE device features. Figure 2 gives the pseudocode of our EEMA algorithm. The loop in lines 4–5 computes the latest $g_c(\mathbf{A}, \mathbf{M}, \mathbf{X}, \hat{\mathbf{R}})$ using Eq. (1) for all combinations. Line 6 picks the combination c^* with the highest utility. Lines 7 and 8 update the current decision and unmet requests. It is not hard to see that the time complexity of EEMA is $O(RC(S+R))$. from the loops starting from lines 3 and 4, respectively. S and R come from computing $w_c(\mathbf{M}, \mathbf{X})$ and $p_c(\mathbf{A}, \hat{\mathbf{R}})$, respectively. Lines 6, 7, and 8 dominate. Hence, the time complexity is $O(RC(S+R))$.

1: // Input: $\mathbf{A}$, $\mathbf{M}$, $\hat{\mathbf{R}}$; Output: $\mathbf{X}$
2: **let** $\mathbf{X} = \varnothing$
3: **while** $\hat{\mathbf{R}} \neq \varnothing$, $\max_{1 \leq c \leq C} g_c(\mathbf{A}, \mathbf{M}, \mathbf{X}, \hat{\mathbf{R}}) > 0$ **do**
4:     **for** each $c = 1, 2, \ldots, C$ **do**
5:         **compute** $g_c(\mathbf{A}, \mathbf{M}, \mathbf{X}, \hat{\mathbf{R}})$ using Eq. (1)
6:     **select** $c^* = \operatorname{argmax}_{c=1,2,\ldots,C} g_c(\mathbf{A}, \mathbf{M}, \mathbf{X}, \hat{\mathbf{R}})$
7:     $\mathbf{X} \leftarrow \mathbf{X} \cup \{s | m_{c^*,s} = 1\}$
8:     $\hat{\mathbf{R}} \leftarrow \hat{\mathbf{R}} - \{r | y_r \leq a_{c^*,r}\}$

Fig. 2. Efficient Energy Minimization Algorithm (EEMA).

3.3 Efficient Accuracy Maximization Algorithm (EAMA)

The EAMA algorithm maintains a set of $\hat{\mathbf{R}}$ of unmet requests, a list of available combinations $\mathbf{W}$ (i.e., those have not been selected), the energy consumption $e_{\mathbf{X}}$ and achieved accuracy $\hat{\mathbf{Y}}_{\mathbf{X}}$ with the current decision $\mathbf{X}$. Its goal is to find a decision $\mathbf{X}$ with the highest average accuracy without exceeding the energy budget E. The cost function $w_c(\mathbf{M}, \mathbf{X})$ is the same as the one used in the EEMA algorithm. The utility function $g'_c(\mathbf{A}, \mathbf{M}, \mathbf{X}, \hat{\mathbf{Y}})$ is written as:

$$g'_c(\mathbf{A}, \mathbf{M}, \mathbf{X}, \hat{\mathbf{Y}}) = \frac{\sum_{1 \leq r \leq R} a_{c,r} \mathbf{1}_{[a_{c,r} \geq \max(y_r, \hat{y}_r(\mathbf{X}))]}}{\sum_{1 \leq s \leq S, s \notin \mathbf{X}} m_{c,s} e_s}. \tag{2}$$

where $\mathbf{1}$ is the indicator function, and $\hat{\mathbf{Y}}(\mathbf{X}) = \{\hat{y}_r(x)|r = 1, 2, \ldots, R\}$ is the achieved accuracy with decision $\mathbf{X}$. Figure 3 gives the pseudocode of the EAMA algorithm. The loop in lines 4–5 computes the latest $g'_c(\mathbf{A}, \mathbf{M}, \mathbf{X}, \hat{\mathbf{Y}})$ using Eq. (2) for all combinations. Line 6 picks the combination c^* with the highest utility, and line 7 updates the available combinations. The if-clause between lines 8–13 checks if activating the sensors of combination c^* would lead to energy consumption within the energy budget. If yes, lines 9, 10 and the loop starting from line 11 update decision $\mathbf{X}$, total energy consumption $e_{\mathbf{X}}$, and the achieved accuracy $\hat{\mathbf{Y}}$, respectively. It can be derived that the time complexity of EAMA is $O(C^2(S + R))$.

1: // Input: $\mathbf{A}$, $\mathbf{M}$; Output: $\mathbf{X}$
2: **let** $e_{\mathbf{X}} = 0$, $\mathbf{W} = \{1, 2, \ldots, C\}$, $\hat{\mathbf{Y}} = 0$
3: **while** $\mathbf{W} \neq \varnothing$ **do**
4:    **for** each $c = 1, 2, \ldots, C$ **do**
5:       **compute** $g'_c(\mathbf{A}, \mathbf{M}, \mathbf{X}, \hat{\mathbf{Y}})$ with Eq. (2)
6:    **select** $c^* = \text{argmax}_{c \in \mathbf{W}}\, g'_c(\mathbf{A}, \mathbf{M}, \mathbf{X}, \hat{\mathbf{Y}})$
7:    $\mathbf{W} \leftarrow \mathbf{W} - \{c^*\}$
8:    **if** $e_{\mathbf{X}} + w_{c^*}(\mathbf{M}, \mathbf{X}) < E$ **then**
9:       $\mathbf{X} \leftarrow \mathbf{X} \cup \{s|m_{c^*,s} = 1\}$
10:       $e_{\mathbf{X}} = \sum_{s=1}^{S} e_s x_s$
11:       **for** each $r = 1, 2, \ldots, R$ **do**
12:          **if** $\hat{y}_r \leq a_{c^*,r}$ **then**
13:             $\hat{y}_r = a_{c^*,r}$

Fig. 3. Efficient Accuracy Maximization Algorithm (EAMA).

4 Trace Driven Simulations

4.1 Setup

We have developed a Java-based event-driven simulator to evaluate the proposed middleware for IoT context aware applications on LTE devices. We have also implemented the proposed sensor management algorithms: the EEMA and EAMA for efficient management. For comparisons, we have also implemented an algorithm called Per-app-Optimized (Per-app) algorithm, which emulates the state-of-the-art sensor management in LTE devices. The Per-app algorithm goes through all the requests, and for each request, it selects the combination achieving the highest precision. This is the same as having individual apps decide how to use sensors without considering overlapping sensors. Each app requests for a context randomly selected from the 6 contexts listed in Table 1. The same table also gives the precision reported in the literature [7,11–13]. We conduct the simulations on a PC with an Intel 3.4 GHz CPU. We consider both the EM and AM problems. For the EM problem, we let y_r be the accuracy requirement of individual requests. More specifically, each request is associated with a random precision uniformly distributed in y_r with value ranging between 0.3 and 0.9. An IoT app may make several context requests to the middleware. For the AM problem, we consider the energy budget $E = \{500, 700, 900, 1100, 1300\}$ mJ, with a sampling rate of 1/300 Hz. E is the energy limit in each management window. We report sample results from $E = 1000$ mJ, if not otherwise specified. We use $T = 1$ min as management window size. The mapping between combinations and sensors are chosen randomly by Bernoulli trail which basically decides whether a sensor should be activated or deactivated. We adopt three performance metrics: (i) energy consumption in mJ, (ii) mean precision in %, and (iii) success rate in %. The success rate refers to the ratio of satisfied context requests.

Table 1. The combinations, contexts, and sensors used in our simulations

	Context precision (%)						Sensor activation in Boolean					
Combination	IsSitting	IsStanding	IsWalking	IsRunning	InMeeting	IsDriving	Acc.	Blue.	WiFi	Mic.	GPS	Cell.
YAN [13]	95	91	83.8	0	73.86	74	1	0	1	0	1	0
CenceMe [7]	68	78	94	74	68	74	1	1	0	1	1	0
EEMSS [11]	89.44	0	78.2	90	0	63.86	1	1	1	0	1	0
EEMSS2 [11]	99.44	0	88.2	100	0	73.86	1	1	1	1	1	0
SAMMPLE [12]	0	0	0	0	68	0	1	0	0	0	1	0
SAMMPLE2 [12]	0	0	0	0	57	0	1	0	0	0	0	0
OTHER1	50	59	66	70	96	91	0	0	1	0	0	1
OTHER2	35	54	56	60	86	76	1	0	0	0	0	1

4.2 Results

We first validate the correctness of Per-app, EMA, and EEMA algorithms. We find that all requests from apps are satisfied by the resulting decisions. Figure 4 shows sensor activations with EAMA algorithm when the energy budget is set to 1000 mJ. The accelerometer is the least activated sensor, whereas GPS and BlueTooth are requested more frequently from various IoT applications. Next, we report the energy consumption achieved by individual algorithms in Fig. 5(a). We observe that EEMA consumes the least energy when we have a fixed energy

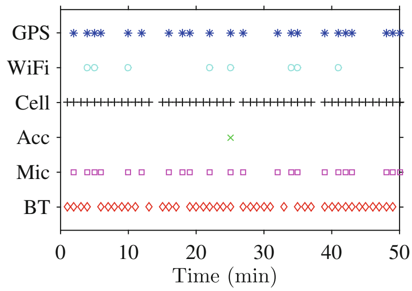

Fig. 4. Sensor activation by EEMA for $E = 1000$ mJ.

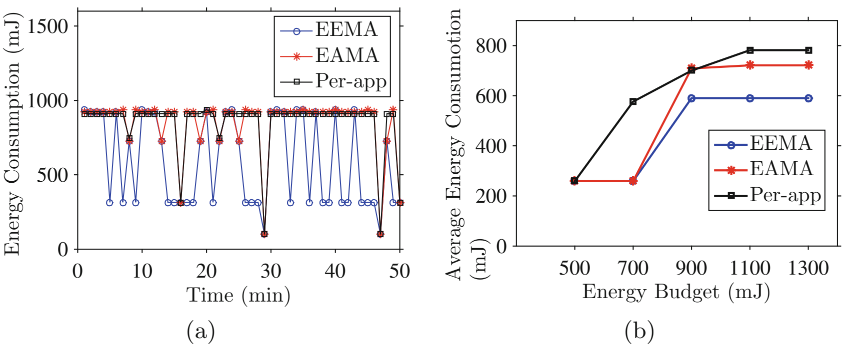

Fig. 5. Energy consumption with: (a) sample results from $E = 1000\,\mathrm{mJ}$ and (b) aggregated results under diverse E.

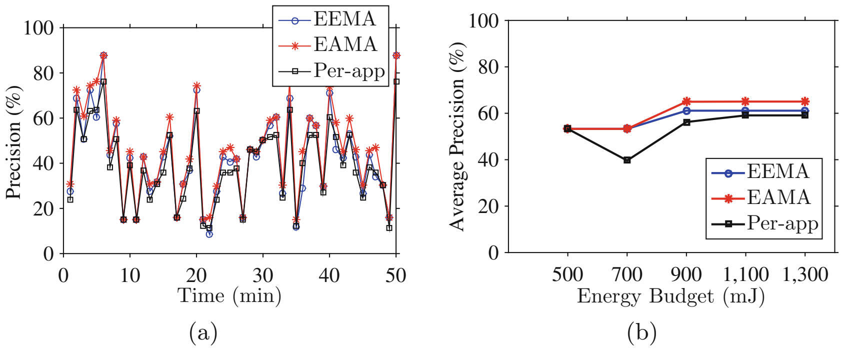

Fig. 6. Average precision with: (a) sample results from $E = 1000\,\mathrm{mJ}$ and (b) aggregated results under diverse E.

budget of ($E = 1000\,\mathrm{mJ}$) and outperforms the per-app algorithm. We then plot the aggregated results under different energy values E in Fig. 5(b). We see a significant saving in energy consumption in EEMA compared to per-app algorithm. The energy consumption of EAMA and per-app is non-decreasing as the energy budget increases, however EEMA shows better result than both of them. We justify the accuracy maximization problem by showing the precision and success rate by focusing on EAMA algorithm. In Fig. 6, we clearly see that the precision of EAMA is better compared to per-app and EEMA. Compared to the sample results in Fig. 6(a), the same observation is even more clear in the aggregated results in Fig. 6(b) with varying energy budgets. High success rate suggests the correctness of our accuracy model by stating the ratio of correctly inferred contexts out of all the requested contexts. In Fig. 7, we see that the success rate is higher for EAMA algorithms which suggests that we achieve higher overall accuracy under a specific energy budget.

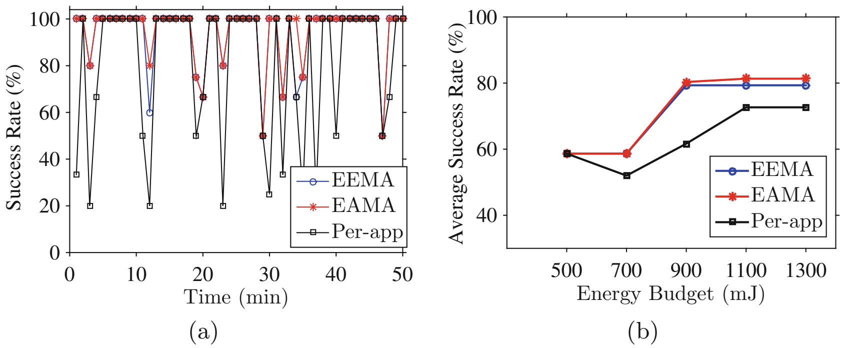

Fig. 7. Success rate with: (a) sample results from $E = 1000\,\text{mJ}$ and (b) aggregated results under diverse E.

5 Conclusion

Context-aware IoT applications are getting increasingly more popular. Multiple IoT apps may run at the same time on an LTE device and request for several overlapping contexts at a subsequent higher rate. In this paper, we developed a novel middleware solution to support efficient context inference from IoT applications in terms of energy consumption and accuracy. Instead of solely intending to reach optimal energy consumption for independent contexts, the proposed middleware selectively activates certain sensors while taking overlapping context requirements from multiple context-aware applications into consideration. We also rigorously studied the sensor management problem, which is the core issue in this middleware. We presented two optimization problem formulations: energy- and accuracy-optimization. We then proposed two heuristic algorithms to address these problems: EEMA and EAMA. Our extensive trace-driven simulations show the merits of our proposed middleware solution and algorithms.

Acknowledgement. This study was conducted under the Advanced Communication Technology Research and Laboratory Development project of the Institute for Information Industry, which is subsidized by the Ministry of Economic Affairs of the Republic of China. This work was also partially supported by Ministry of Science and Technology (MOST) of Taiwan under the grant number MOST104-3115-E-007-004.

References

1. Barbero, C., Zovo, P.D., Gobbi, B.: A flexible context aware reasoning approach for IoT applications. In: Proceedings of IEEE International Conference on Mobile Data Management (MDM 2011), Lulea, Sweden (2011)
2. Chon, Y., Talipov, E., Shin, H., Cha., H.: Mobility prediction-based smartphone energy optimization for everyday location monitoring. In: Proceedings of ACM Conference on Embedded Networked Sensor Systems (SenSys 2011), Seattle, WA (2011)

3. Kim, D., Kim, Y., Estrin, D., Srivastava, M.: SensLoc: sensing everyday places and paths using less energy. In: Proceedings of ACM Conference on Embedded Networked Sensor Systems (SenSys 2010), Zurich, Switzerland (2010)
4. Lin, K., Kansal, A., Lymberopoulos, D., Zhao, F.: Energy-accuracy trade-off for continuous mobile device location. In: Proceedings of International Conference on Mobile systems, Applications, and Services (MobiSys 2010), San Francisco, CA (2010)
5. Lu, H., Yang, J., Liu, Z., Lane, N., Choudhury, T., Campbell, A.: The Jigsaw continuous sensing engine for mobile phone applications. In: Proceedings of ACM Conference on Embedded Networked Sensor Systems (SenSys 2010), Zurich, Switzerland (2010)
6. Ma, Y., Hankins, R., Racz, D.: iLoc: a framework for incremental location-state acquisition and prediction based on mobile sensors. In: Proceedings of ACM Conference on Information and Knowledge Management (CIKM 2009), Hong Kong, China (2009)
7. Miluzzo, E., Lane, N., Fodor, K., Peterson, R., Lu, H., Eisenman, S., Zheng, X., Campbell, A.: Sensing meets mobile social networks: the design, implementation and evaluation of the CenceMe application. In: Proceedings of ACM Conference on Embedded Networked Sensor Systems (SenSys 2008), Raleigh, NC (2008)
8. Perera, C., Zaslavsky, A., Christen, P., Georgakopoulos, D.: Context aware computing for the Internet of Things: a survey. IEEE Commun. Surv. Tutor. **16**(1), 414–454 (2013)
9. Schirmer, M., Höpfner, H.: SENST*: approaches for reducing the energy consumption of smartphone-based context recognition. In: Proceedings of International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT 2011), Karlsruhe, Germany (2011)
10. Taranto, R., Muppirisetty, L., Raulefs, R., Slockand, D., Svensson, T., Wymeersch, H.: Location-aware communications for 5G networks: how location information can improve scalability, latency, and robustness of 5G. IEEE Signal Process. Mag. **31**(6), 102–112 (2014)
11. Wang, Y., Lin, J., Annavaram, M., Jacobson, Q., Hong, J., Krishnamachari, B., Sadeh, N.: A framework of energy efficient mobile sensing for automatic user state recognition. In: Proceedings of International Conference on Mobile Systems, Applications, and Services (MobiSys 2009), Krakw, Poland (2009)
12. Yan, Z., Jeung, H., Chakraborty, D., Misra, A., Aberer, K.: SAMMPLE: detecting semantic indoor activities in practical settings using locomotive signatures. In: Proceedings of International Symposium on Wearable Computers (ISWC 2012), Newcastle, UK (2012)
13. Yan, Z., Subbaraju, V., Chakraborty, D., Misra, A., Aberer, K.: Energy-efficient continuous activity recognition on mobile phones: an activity-adaptive approach. In: Proceedings of International Symposium on Wearable Computers (ISWC 2012), Newcastle, UK (2012)