Smart usage of context information for the analysis, design and generation of power-aware policies for mobile sensing apps

Doctoral seminar 2016

Rafael Pérez Torres

Dr. César Torres Huitzil
Dr. Hiram Galeana Zapién
LTI Cinvestav

Cinvestav

Agenda

Research background
Motivation

- The popularity of mobile devices is a result of advances in their computation, **sensing**, and communication dimensions [1].
  - The sensing facilities improve interaction with user, turning them into *omni-sensors* able to *know* about its surrounding environment.
  - Smartphones have become *context-aware* devices, gaining understanding about user's activity and environment.
  - **Context** is the set of environmental states and settings that either determines an application's behavior or in which an application event occurs and is interesting to the user [2].
- However, battery is not evolving at the same pace of advances in other smartphone's characteristics [3], growing only 5-10% each year [4, 5].
  - The energy constraint becomes critical when continuous access to sensors is needed, which is a core requirement of **mobile sensing applications**.

Research background

Overall problem identification

Overall issue

Conceptually, the main problem pursued by this research work is to design a location provider aware of the energy limitations of the mobile device. Nonetheless, as the location is a reflection of the mobility patterns described by people, a set of more specific problems could be addressed.

Types of mobility patterns

There are two types of mobility patterns considered by this research:

- Fine grain mobility patterns.
- Coarse grain mobility patterns.

Problem definition
Problem statement

Problem 1: Mobility pattern identification

Given a set of values $\mathcal{V} = v_1, v_2, \ldots, v_n$ obtained from sensor $\mathcal{S}$ in time range $[t_1, t_2]$, identify coarse-grain or fine-grain mobility information:

$$\text{PatternIdentifier}(\mathcal{V}) \rightarrow p_S \in \text{Patterns}$$

where $\text{Patterns} = \{\text{static}, \text{walking}, \text{running}, \text{vehicle}\}$ for fine-grain location information, while for coarse-grain information refers to the arrival and departure to-from stay points described by user's mobility.
Additionally, the sensor $\mathcal{S}$ for fine-grain information is `accelerometer`, while sensor $\mathcal{S}$ for coarse-grain information is GPS.

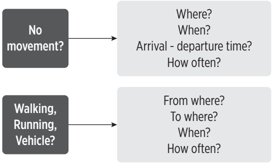


Figure: Context information related to mobility patterns

Problem definition
Problem statement

Problem 2: Policy generation

Given a set of detected mobility patterns $\mathcal{P} = \{p_{S_1}, p_{S_2}, \ldots, p_{S_n}\}$ in historical sensors data, accuracy requirements of mobile app a, and physical constraints status c of the mobile device, define a policy that selects the proper set of sensors $\mathcal{S}_{new}$ and its associated configuration $\mathcal{S}_{new_{conf}}$ while meeting application requirements and reducing energy consumption.

$$\text{PolicyGeneration}(\mathcal{P}, a, c) \longrightarrow \mathcal{S}_{\text{new}}, \mathcal{S}_{\text{new}_{\text{conf}}}$$

The $\mathcal{S}_{\text{new}_{\text{conf}}}$ configuration is referred as the *adaptive duty cycle* of associated sensor.

Interaction between problems

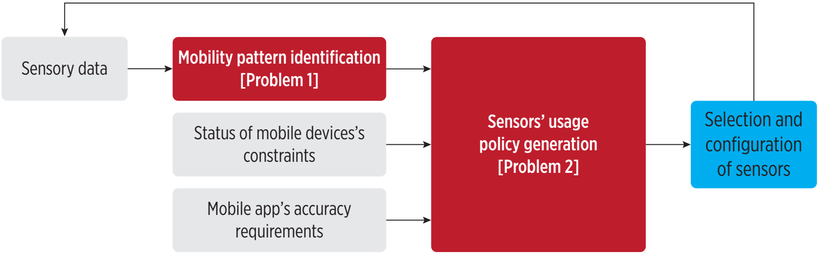


Figure: Interaction between problems

Hypothesis

Hypothesis

Dynamic policies driven by events detected in user mobility could reduce energy consumption of the mobile device when performing continuous location tracking. User mobility could be expressed by means of a spatial-time model learned from short and long time windows of context information extracted from GPS and accelerometer sensors data.

- An intelligent policy is a special rule that defines how sensors should be selected and configured to reduce energy consumption and achieve the mobile sensing app's requirements. It is intelligent in terms of self-adaptness to changes detected in context information across time.
- This research work is aimed at employing GPS and inertial sensors data (accelerometer) for inferring context information in terms of mobility patterns. This context information will then be exploited to adapt sensors' operation and produce power savings.

Objectives

Main objective

To reduce energy consumption in the mobile sensing apps, which perform continuous sensor readings, through self-adapting power-aware policies generated from context information obtained from sensors data.

Particular objectives

- To identify mobility patterns from context information obtained from an inertial sensor (accelerometer) and location providers (GPS).
- To generate an accurate representation of mobility patterns, which in conjunction with accuracy mobile app requirements and mobile device constraints, allows to create power-aware GPS sensing policies.
- To reduce energy consumption in location-based mobile sensing apps through a middleware that implements policies fed with mobility patterns learned from sensors data. The middleware also eases the development of LBS, isolating the complexity of energy efficient sensors management.

Expected contributions

Contributions

- A mechanism for detecting mobility patterns from the data read by sensors of mobile devices (GPS and accelerometer).
- A mechanism for generating policies for accessing sensors. The produced policies will allow to perform an intelligent usage of smartphone's sensing facilities in continuous sensor sampling, reducing the energy consumption.
- A middleware implementing the previous power-aware mechanisms for easing the development of location based services.

Methodology

Methodology steps

1. **Familiarization with state-of-art power-aware sensing related techniques**
2. **Formal definition and selection of mobility patterns to be identified**
3. **Research on pattern recognition algorithms focused on mobility patterns identification**
4. **Design of the Pattern Identification Element (PIE)**
5. **Research on (and proposition of) adaptive policies for energy efficient usage of sensors**
6. **Design of the Policy Generation Element (PGE)**
7. Development of a middleware involving the PIE and PGE for the Android platform
8. Experimentation in terms of accuracy and energy efficiency

Related work

Previous work

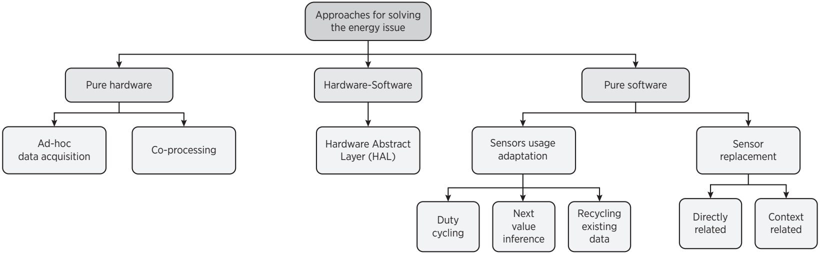


Figure: Taxonomy of related work solutions

Proposed solution
Problem's scenario

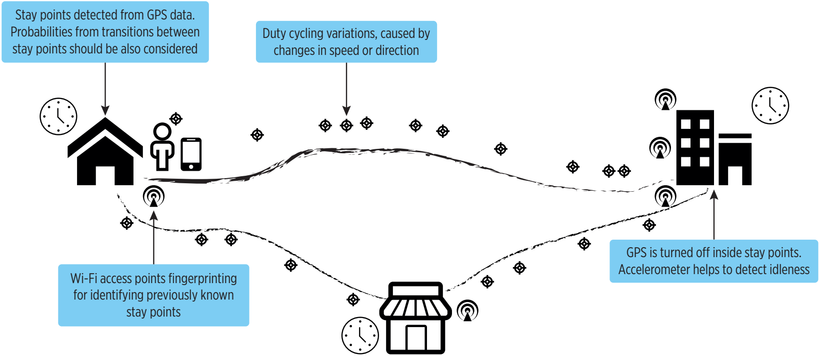


Figure: Basic problem's scenario

Proposed solution
Operation

- Overall problem divided in:
  - Detection and learning of stay points (The anchor points that user visits frequently).
  - Tracking of user commuting between stay points (Transition probability).

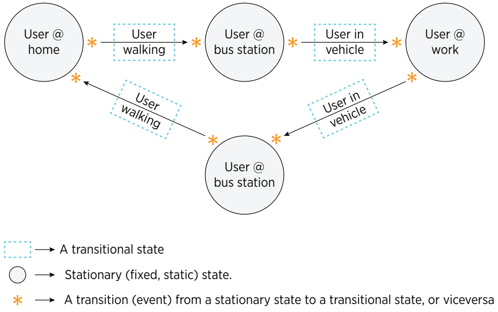


Figure: Information modeled by the expanded spatial-time model

Proposed solution
Fundamental characteristics

- Pure software approach, implementing all of its variants.
- Event-driven oriented, fully on-device.

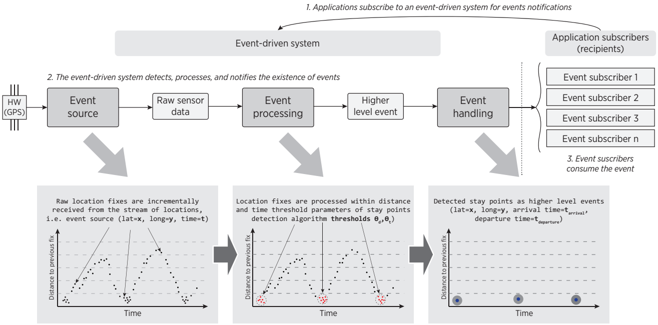


Figure: Event driven system components

- Top level mobility states-events: `on trajectory`, `on stay point`.
- Low level mobility states-events: `transportation mode changed`, `arriving to stay point`, `leaving stay point`.

Proposed solution
Fundamental characteristics

- Inspired on Cognitive Dynamic Systems [6].
- Time plays a key role.

Cognitive Dynamic System features

- Perception-action cycle (Observe and react for modifying environment or system operation).
- Memory (Learn from perception and action).
- Attention (Adapt resource allocating-management in a goal oriented scenario).
- Intelligence (Adapt system operation across time).

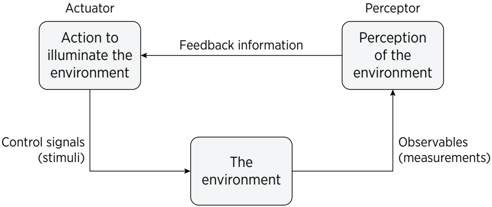


Figure: A generic Cognitive Dynamic System

Proposed solution
Overview

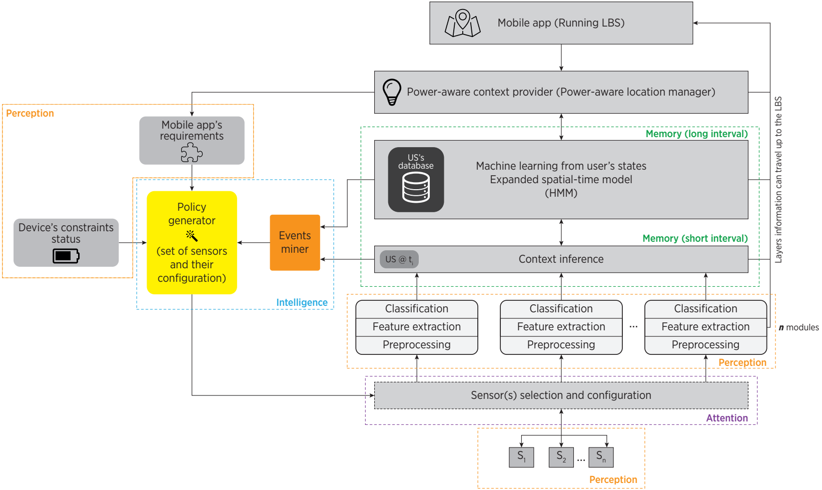


Figure: The layers of proposed solution

Proposed solution

Event-driven implementation

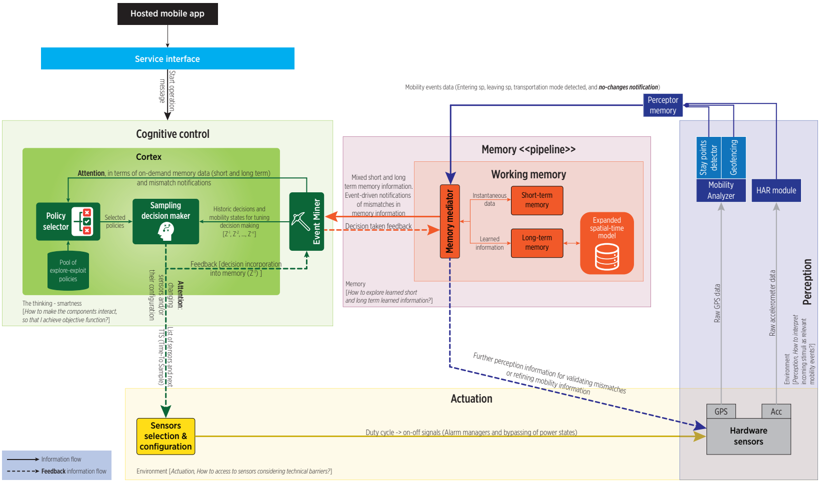


Figure: Cognitive components of platform and their interaction

Experimental results

Several experiments have been carried out during the development of system solution. Such experiments include:

- Exploring the feasibility of on-device mobility analysis.
- Validating energy saving of on-device approach vs. MCC oriented approach.
- Validating the cognitive capabilities of the platform and exploring their potential energy savings.
  - Using policies based on fixed sampling periods, aimed at different mobility modes `onTrajectory, onStayPoint`.

Experimental results

Feasibility of on-device mobility analysis

- Feasibility is understood as the ability for executing the middleware in the mobile device[1] under different stress levels without a premature finalization caused by CPU and memory consumption usage issues.

Table: Summary of results of first experiment (SP = stay point).

Sampling period (seconds)	Event-driven algorithm	Obtained GPS fixes	Average GPS fixes per SP	Running time (minutes)
30 seconds	Buffered	3,876	218.3	6,752
	Sigma	5,199	244.4	8,243
60 seconds	Buffered	5,307	155.6	14,877
	Sigma	3,054	126.3	8,428
90 seconds	Buffered	2,573	115.2	7,694
	Sigma	2,447	108.1	7,522
120 seconds	Buffered	1,708	77.4	8,460
	Sigma	1,993	82.2	10,214
150 seconds	Buffered	1,417	53.8	10,433
	Sigma	1,651	51.1	10,349

- Execution was performed free of memory and computing (timing) issues.

[1] Google Nexus 6, 2.7 GHz quad-core processor, 3 GB RAM, 3220 mAh battery, Android 6.

Experimental results

Energy performance (solution vs. MCC oriented approach)

- Purpose of comparing the energy consumption of the proposed middleware with respect to an MCC oriented solution that offloads location data processing.
- The MCC oriented solution subscribed for location updates to the middleware, but translated each location update into a string representation (116 bytes, in average) sent to an application server using cellular data network.

Sampling period (seconds)	Processing strategy	Obtained GPS fixes	GPS-on time (minutes)	Average acquisition time per fix (seconds)	Running time (minutes)	Data sent (bytes)	Data received (bytes)
30	On-device	12,341	1,614	7.84	7,790	-	-
	MCC oriented	9,324	770	4.98	5,402	1,084,901	18,796
60	On-device	10,816	1,219	6.76	12,028	-	-
	MCC oriented	7,205	764	6.45	7,907	838,640	14,696
90	On-device	7,868	1,178	8.91	13,075	-	-
	MCC oriented	5,624	546	5.84	8,946	653,833	12,223
120	On-device	5,189	809	9.26	11,289	-	-
	MCC oriented	4,332	387	5.43	8,931	504,012	8,838
150	On-device	5,576	933	9.94	14,998	-	-
	MCC oriented	4,564	452	6.06	11,619	530,764	10,309

Table: Summary of results of second experiment.

Experimental results

Energy performance (solution vs. MCC oriented approach)

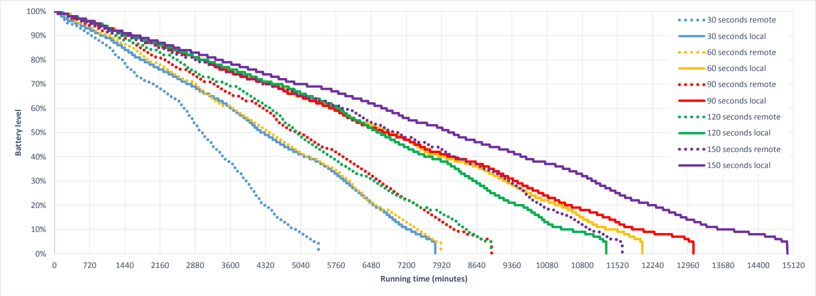


Figure: Energy performance comparison of on-device vs. MCC oriented sample apps using different GPS sampling periods. Battery lifetime increase in factors within the range 1.26**x** to 1.52**x**.

Preliminary results

Early adaptive sampling implementation

- Cognitive capabilities of the platform and early experimentation for ensuring power saving capabilities have also been performed.

Space information		Time information	
Total fixes	4,719	Total time (minutes)	8,848.56
Fixes inside stay points (94.3%)	4,451	Total stay points time (96.4%)	8,526.45
Fixes on trajectory (5.7%)	268	Total trajectory time (3.6%)	322.11
Detected stay points	6	Semantic time duration	Tuesday night - Monday night
Total trajectories	28		(approximately 6 days)

Table: Space and time summary of mobility information

- The employment of stay time information could be employed for reducing sensors sampling:

#	Semantic	Visit count	Stay time (minutes)	Absolute weight	Relative weight
1	Park	3	254	2.88%	2.99%
2	Home	8	5,767	65.18%	67.64%
3	Cinvestav	5	2,090	23.62%	24.51%
4	Fast food	6	78	0.89%	0.92%
5	Bob's home	6	260	2.94%	3.05%
6	Coffee shop	1	75	0.85%	0.89%

Table: Stay points weights and visits information

Preliminary results

Early adaptive sampling implementation

- A simple sampling policy created from two fixed sampling periods was implemented.
- The sampling periods are switched depending on changes in user mobility.
- Such simple sampling policy scheme allowed to reduce energy consumption.

Figure: Energy performance of fixed sampling period vs. simple sampling policies. Battery lifetime increase of 1.41**x** (42.8 hours)

Future work
Specific activities

Several tasks have to be completed in short-term for defining policies towards adaptive sampling.

- Build a software library for exploring - interpreting the mobility information collected by platform.
- Test platform under mobility cases for identifying mismatches in mobility information.
- Produce sampling policies for mobility tracking and mismatches handling (logarithmic, exponential, doubling).
- Evaluate the energy consumption and spatial-time accuracy of defined policies.

Detailed schedule

		2014	2015			2016		
	Work: ● Done, ● In progress, ○ To be done	3rd	1st	2nd	3rd	1st	2nd	3rd
	Step I							
1	State-of-art reading	●	●					
2	State-of-art works categorization		●	●				
3	Documentation of information found (committee request)			●				
	Step II							
4	Development of a mobile app for accelerometer and location data collection			●	●			
5	Analysis of data				●			
6	Formal definition of mobility pattern				●			
7	Selection of mobility patterns				●	●		
	Step III							
8	Research on classification algorithms for mobility patterns				●	●		
9	Definition of metrics for evaluating algorithms					●		
10	Implementation of algorithms in mobile platform					●	●	
11	Selection of best algorithms according to metrics						●	
	Step IV							
12	Definition and modeling of parameters needed by the PIE				●	●	●	●
13	Building of the PIE				●	●	●	●

Table: Schedule of activities (each column represents a four months period)

Detailed schedule

Work: ● Done, • In progress, ○ To be done

		2016			2017			2018
		1st	2nd	3rd	1st	2nd	3rd	1st
	Step V							
14	Formal definition of policy			●				
15	Research and evaluation of techniques for generation and adaption of policies			•	•			
16	Design and execution of experiments applied to use cases			•	•			
17	Selection of policies				•			
	Step VI							
18	Definition and modeling of PGE parameters					•		
19	Building of the PGE					•		
	Step VII							
20	Analysis of components into software abstractions					●		
21	Research on Android API for specialized components					●		
22	Development of middleware					•		
	Step VIII							
23	Definition of experiments aimed at accuracy and energy consumption metrics						○	
24	Development of experimental sample mobile apps						○	
25	Experiments execution						○	○
26	Final results analysis							○

Table: Schedule of activities (each column represents a four months period)

Detailed schedule

		2014	2015			2016			2017			2018	
Work: ● Done, ● In progress, ○ To be done		3rd	1st	2nd	3rd	1st	2nd	3rd	1st	2nd	3rd	1st	2nd
	Required tasks												
A	Related courses	●	●	●									
B	Research articles submission				●		●					○	
C	Predoctoral exam preparation									○			
D	Thesis writing	●			●			●			○	○	○

Table: Schedule of required activities

Conclusions

As conclusions, the current talk has provided:

- The refinement of background and scientific context of this research work.
- The improvements of the cognitive and event-driven proposed solution.
- Experimental results that validate the feasibility of on-device mobility analysis, and draw the path for incorporating true adaptive sampling, which is directed by cognitive-oriented mobility data processing.

Journal articles published along the development of thesis work:

- Pérez-Torres, R., Torres-Huitzil, C., & Galeana-Zapién, H. (2016). Power management techniques in smartphone-based mobility sensing systems: A survey. *Pervasive and Mobile Computing*, 1-21. https://doi.org/10/bds7 [7]
- Pérez-Torres, R., Torres-Huitzil, C., & Galeana-Zapién, H. (2016). Full On-Device Stay Points Detection in Smartphones for Location-Based Mobile Applications. *Sensors*, 16(10), 1693. https://doi.org/10/brst [8]

Thank you for your attention!

We make our world significant by the courage of our questions and by the depth of our answers.

Carl Sagan

Cinvestav

References I

[1] Nayeem Islam and Roy Want.
Smartphones: Past, Present, and Future.
IEEE Pervasive Computing, 13(4):89–92, 2014.

[2] Guanling Chen and David Kotz.
A Survey of Context-Aware Mobile Computing Research.
Technical report, 2000.

[3] Mikkel Kjaergaard.
Location-based services on mobile phones: Minimizing power consumption.
IEEE Pervasive Computing, 11:67–73, 2012.

[4] Xiao Ma, Yong Cui, and Ivan Stojmenovic.
Energy efficiency on location based applications in mobile cloud computing: A survey.
In *Procedia Computer Science*, volume 10, pages 577–584, 2012.

[5] Eric C. Evarts.
Lithium batteries: To the limits of lithium.
Nature, 526(7575):S93–S95, oct 2015.

[6] Simon Haykin.
Cognitive Dynamic Systems.
Proceedings of the IEEE, 94(11):1910–1911, nov 2006.

[7] Rafael Pérez-Torres, César Torres-Huitzil, and Hiram Galeana-Zapién.
Power management techniques in smartphone-based mobility sensing systems: A survey.
Pervasive and Mobile Computing, pages 1–21, feb 2016.

[8] Rafael Pérez-Torres, César Torres-Huitzil, and Hiram Galeana-Zapién.
Full On-Device Stay Points Detection in Smartphones for Location-Based Mobile Applications.
Sensors, 16(10):1693, 2016.

[9] Parthasarathy Ranganathan.
Recipe for efficiency.
Communications of the ACM, 53(4):60, 2010.

References II

[10] J.R. Lorch and A.J. Smith.
Software strategies for portable computer energy management.
IEEE Personal Communications, 5(3):60–73, jun 1998.

[11] Luca Benini, Alessandro Bogliolo, and Giovanni De Micheli.
A survey of design techniques for system-level dynamic power management.
IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 8(3):299–316, 2000.

[12] Alfredo J. Perez, Miguel a. Labrador, and Sean J. Barbeau.
G-Sense: A scalable architecture for global sensing and monitoring.
IEEE Network, 24(August):57–64, 2010.

[13] Rafael Perez-Torres and Cesar Torres-Huitzil.
A power-aware middleware for location & context aware mobile apps with cloud computing interaction.
In *Proceedings of the 2012 World Congress on Information and Communication Technologies, WICT 2012*, pages 691–696, 2012.

[14] Fehmi Ben Abdesslem, Andrew Phillips, and Tristan Henderson.
Less is more: energy-efficient mobile sensing with senseless.
In *Proceedings of the 1st ACM workshop on Networking, systems, and applications for mobile handhelds.*, pages 61–62, 2009.

[15] Lei Zhang, Jiangchuan Liu, Hongbo Jiang, and Yong Guan.
SensTrack: Energy-efficient location tracking with smartphone sensors.
IEEE Sensors Journal, 13(10):3775–3784, 2013.

[16] Yemao Man and Edith C H Ngai.
Energy-efficient automatic location-triggered applications on smartphones.
Computer Communications, 50:29–40, 2014.

[17] Ionut Constandache, Shravan Gaonkar, Matt Sayler, Romit Roy Choudhury, and Landon Cox.
EnLoc: Energy-efficient localization for mobile phones.
In *Proceedings - IEEE INFOCOM*, number 4, pages 2716–2720, 2009.

[18] Mikkel Baun Kjaergaard, Jakob Langdal, Torben Godsk, and Thomas Toftkjær.
EnTracked : Energy-Efficient Robust Position Tracking for Mobile Devices.
In *Proceedings of the 7th international conference on Mobile systems, applications, and services*, pages 221–234, 2009.

References III

[19] M. a. Álvarez De La Concepción, L. M. Soria Morillo, L. Gonzalez-Abril, and J. a. Ortega Ramírez.
Discrete techniques applied to low-energy mobile human activity recognition. A new approach.
Expert Systems with Applications, 41:6138–6146, 2014.

[20] Luis Morillo, Luis Gonzalez-Abril, Juan Ramirez, and Miguel de la Concepcion.
Low Energy Physical Activity Recognition System on Smartphones.
Sensors, 15(3):5163–5196, 2015.

[21] Sînziana Mazilu, Ulf Blanke, Alberto Calatroni, and Gerhard Tröster.
Low-power ambient sensing in smartphones for continuous semantic localization.
In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 8309 LNCS, pages 166–181, Ireland, 2013. Springer.

[22] Vijay Srinivasan and Thomas Phan.
An accurate two-tier classifier for efficient duty-cycling of smartphone activity recognition systems.
In *Proceedings of the Third International Workshop on Sensing Applications on Mobile Phones - PhoneSense '12*, pages 1–5, 2012.

[23] Sara Khalifa, Mahbub Hassan, and Aruna Seneviratne.
Pervasive self-powered human activity recognition without the accelerometer.
In *2015 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 79–86. IEEE, mar 2015.

[24] Donnie H Kim, Younghun Kim, Deborah Estrin, and Mani B Srivastava.
SensLoc: Sensing Everyday Places and Paths Using Less Energy.
In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pages 43–56, New York, NY, USA, 2010. ACM.

[25] Jeongyeup Paek, Kyu-Han Kim, Jatinder P. Singh, and Ramesh Govindan.
Energy-efficient positioning for smartphones using Cell-ID sequence matching.
In *Proceedings of the 9th international conference on Mobile systems, applications, and services - MobiSys '11*, page 293, New York, New York, USA, 2011. ACM Press.

[26] Jeongyeup Paek, Joongheon Kim, and Ramesh Govindan.
Energy-efficient rate-adaptive GPS-based positioning for smartphones.
In *Proceedings of the 8th international conference on Mobile systems, applications, and services - MobiSys '10*, volume 223-224, page 299, New York, New York, USA, 2010. ACM Press.

References IV

[27] Kaisen Lin, Aman Kansal, Dimitrios Lymberopoulos, and Feng Zhao.
Energy-accuracy trade-off for continuous mobile device location.
In *Proceedings of the 8th international conference on Mobile systems, applications, and services - MobiSys '10*, page 285, 2010.

[28] Yohan Chon, Elmurod Talipov, Hyojeong Shin, and Hojung Cha.
Mobility prediction-based smartphone energy optimization for everyday location monitoring.
In *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems - SenSys '11*, page 82, 2011.

[29] Hong Lu, Jun Yang, Zhigang Liu, Nicholas D. Lane, Tanzeem Choudhury, and Andrew T. Campbell.
The Jigsaw continuous sensing engine for mobile phone applications.
In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems - SenSys '10*, page 71, 2010.

[30] Brad K. Donohoo, Chris Ohlsen, Sudeep Pasricha, Yi Xiang, and Charles Anderson.
Context-aware energy enhancements for smart mobile devices.
IEEE Transactions on Mobile Computing, 13(8):1720–1732, 2014.

[31] Yi Wang, Jialiu Lin, Murali Annavaram, Quinn a Jacobson, Jason Hong, Bhaskar Krishnamachari, and Norman 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*, pages 179–192, 2009.

[32] Yiming Ma, Rich Hankins, and David Racz.
iLoc: a framework for incremental location-state acquisition and prediction based on mobile sensors.
In *Proceeding of the 18th ACM conference on Information and knowledge management - CIKM '09*, page 1367, New York, New York, USA, 2009.
ACM Press.

[33] Ozgur Yurur, Miguel Labrador, and Wilfrido Moreno.
Adaptive and energy efficient context representation framework in mobile sensing.
IEEE Transactions on Mobile Computing, 13(8):1681–1693, 2014.

[34] Yohan Chon, Yungeun Kim, Hyojeong Shin, and Hojung Cha.
Adaptive duty cycling for place-centric mobility monitoring using zero-cost information in smartphone.
IEEE Transactions on Mobile Computing, 13(8):1694–1706, 2014.

Related work

Pure hardware approach

Pure hardware approach

- The fundamental idea is the selection of power-aware hardware elements for providing physical data to upper layers, as well as the definition of mechanisms to adapt the hardware input parameters, like DVFS[2].
- Such mechanisms define control points for manipulating hardware (also known as power modes [9, 10, 11]).
- The hardware components obey a static behavior defined by power modes, whose control points are exported to upper layers of the mobile platform.

Variants

- Ad-hoc data acquisition
- Co-processing

[2] DVFS, Dynamic Voltage and Frequency Scaling

Related work

Hardware-software approach

- It is aimed at defining system-wide policies for deciding when to turn sensors on and off, or when to switch hardware components to a different power mode.
- It abstracts fine-grain operation parameters into a coarse-grain set, easing the hardware usage for upper platform layers.
- It is also able to detect changes in the workload of hardware components.
- Because of the coupled interaction with hardware, solutions are produced as HAL's, or low-level hardware middlewares.

Related work
Pure software approach

Pure software approach

- *Spending power to save power* [9].
- It employs context information obtained from sensor data, for achieving activity awareness and making informed decisions towards dynamic power-aware sensors management.
- The lower layers know how to turn circuits on and off, but are unable to define when; whereas higher software layers can dynamically adapt to changes in user context, delegating how to do it to the lower layers.
- Typically, pure software approach solutions are implemented through a layered middleware with the classification and machine learning modules embedded on it.

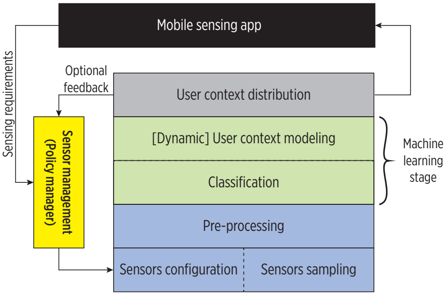
Related work

Pure software approach

Variants

- Sensors usage adaptation
  - Duty cycling
  - Next value inference
  - Recycling existing data
- Sensors replacement
  - Directly related
  - Context related

Generic pure software approach middleware



Related work

Characteristics of pure software approach solutions

Distinctive characteristics of pure software approach solutions

- **Optimization oriented (OO)**: Optimization orientation focused on minimizing energy consumption and/or the error in activity tracking.
- **Online learning (OL)**: Online learning from context information, enabling predictive features thanks to observance over long-time windows of sensory data.
- **User state oriented (US)**: Modeling of an enriched version of the context information, user state (US), for achieving full activity-awareness and ease the adaptation over the sensing dimension.

State-of-art solutions

Name	Variants	Machine learning technique	Sensors involved	Complexity	OL	OO	US
G-Sense [12]	User behavior learning (DC)	SDR	GPS	low			
Perez-Torres [13]	User behavior learning (DC)	SDR	GPS	low			
SenseLess [14]	User behavior learning (DC), Sensor replacement (CR, DR)	SDR	WPS, GPS, ACC	low			
SensTrack [15]	User behavior learning (DC), Sensor replacement (CR, DR)	SDR	ACC, orientation sensor, GPS, WPS	low			
Man and Ngai [16]	User behavior learning (DC, VI), Sensor replacement (CR)	SDR	ACC, magnetic field sensor, GPS	low			
EnLoc [17]	User behavior learning (DC, VI), Sensor replacement (DR)	SDR, Mobility Tree	WPS, GPS, cellular ID	medium		✓	
EnTracked [18]	User behavior learning (DC), Sensor replacement (CR)	SDR	ACC, GPS	medium		✓	

Table: Pure software solutions. (OL: Online Learning from user data, OO: Optimization Oriented solution, US: User State context insight)

State-of-art solutions

Name	Variants	Machine learning technique	Sensors involved	Complexity	OL	OO	US
Alvarez, Morillo [19, 20]	–	Ameva algorithm	ACC	medium			
Mazilu [21]	Sensor replacement (CR)	DT	Temperature, humidity, pressure	medium			
Srinivasan [22]	User behavior learning (DC)	DT	ACC	medium			
Khalifa [23]	Sensor replacement (CR)	KNN	Model of ACC-based harvesting device	medium			

Table: Pure software solutions. (OL: Online Learning from user data, OO: Optimization Oriented solution, US: User State context insight)

State-of-art solutions

Name	Variants	Machine learning technique	Sensors involved	Complexity	OL	OO	US
SensLoc [24]	User behavior learning (DC, RD), Sensor replacement (CR)	SDR	Wi-Fi fingerprinting, GPS, ACC	medium	✓		
CAPS [25]	User behavior learning (DC, RD), Sensor replacement (CR)	SDR	GPS, cellular ID	medium	✓		
RAPS [26]	User behavior learning (DC, RD), Sensor replacement (CR, DR)	SDR	WPS, GPS, ACC, Bluetooth, cellular ID	medium	✓		
A-Loc [27]	User behavior learning (DC, RD), Sensor replacement (CR, DR)	HMM, Bayesian estimation framework	GPS, WPS, Bluetooth, cellular ID	medium	✓	✓	
SmartDC [28]	User behavior learning (DC, RD), Sensor replacement (CR, DR)	HMM and LZ predictor	GPS, WPS, Wi-Fi and cellular ID fingerprinting	medium	✓	✓	

Table: Pure software solutions. (OL: Online Learning from user data, OO: Optimization Oriented solution, US: User State context insight)

State-of-art solutions

Name	Variants	Machine learning technique	Sensors involved	Complexity	OL	OO	US
Jigsaw [29]	User behavior learning (DC), Sensor replacement (CR)	Microphone: NB with Gaussian Mixture Model (GMM). ACC: DT. GPS: MDP.	ACC, Microphone, GPS	high		✓	✓
Donohoo [30]	User behavior learning (DC)	Several. KNN and NN selected as best.	ACC, GPS, WPS, cellular ID, light, device data, mobile app requirements	high			✓
EEMSS [31]	User behavior learning (DC), Sensor replacement (CR, DR)	GPS and ACC: SDR. Microphone: SSCH algorithm.	ACC, microphone, GPS	high			✓
iLoc [32]	User behavior learning (RD), Sensor replacement (CR)	HMM	Wi-Fi & GSM fingerprinting	high	✓		✓
Yurur [33]	User behavior learning (DC, RD)	HMM	ACC	high	✓		✓
FreeTrack [34]	User behavior learning (DC, RD), Sensor replacement (CR, DR)	HMM	GPS, Wi-Fi, cellular ID, battery status	high	✓		✓

Table: Pure software solutions. (OL: Online Learning from user data, OO: Optimization Oriented solution, US: User State context insight)

Additional experiments results

Accuracy evaluation of detected stay points

Table: Spatial and time accuracy observed in results of first experiment (SP = stay point).

Sampling period (seconds)	Event-driven algorithm	Detected SP's	Average SP stay time difference (seconds)	Average SP distance difference (meters)
30	Buffered	16 (out of 19)†	64.13	13.35
	Sigma	19 (out of 19)	68.78	16.01
60	Buffered	29 (out of 29)	98.24	14.97
	Sigma	21 (out of 29)†	82.35	19.42
90	Buffered	20 (out of 20)	104.95	20.6
	Sigma	20 (out of 20)	211.68	20.68
120	Buffered	19 (out of 21)†	63.7	35.56
	Sigma	21 (out of 21)	59.7	34.05
150	Buffered	24 (out of 29)†	116.4	59.11
	Sigma	28 (out of 29)‡	115.6	50.81

† Due to battery depletion, ‡ Actual SP miss

Experiment 2, factors of energy gains:

- 30 seconds gains 1.442162452
- 60 seconds gains 1.521175776
- 90 seconds gains 1.46152413
- 120 seconds gains 1.264046265
- 150 seconds gains 1.290736703
- Min 1.264046265 Max 1.521175776

Preliminary results

Early adaptive sampling implementation

- Cognitive capabilities of the platform and early experimentation for ensuring power saving capabilities have also been performed.

Space information		Time information	
Total fixes	6,652	Total time	6275.15 minutes
Fixes inside stay points	6380 (95.9%)	Total stay points time	5995.08 minutes (95.53%)
Fixes on trajectory	272 (4.1%)	Total trajectory time	280 minutes (4.47%)
Detected stay points	6		
Total trajectories	26		

Table: Space and time summary of mobility information

#	Semantic	Visit count	Stay time (minutes)	Absolute weight	Relative weight
1	Park	3	252	4.02%	4.20%
2	Home	8	3,825	60.97%	63.82%
3	Cinvestav	4	1,549	24.69%	25.85%
4	Fast food	4	72	1.15%	1.21%
5	Bob's home	7	220	3.50%	3.67%
6	Coffee shop	1	75	1.20%	1.25%

Table: Stay points weight and visits information