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

Rafael Pérez Torres

Dr. César Torres Huitzil

Dr. Hiram Galeana Zapién

Doctoral seminar, 2017

Cinvestav
Tamaulipas

Agenda

Introduction

State of the art

Theoretical framework

Solution

Implementation

Preliminary results

Future work

Research background

Motivation

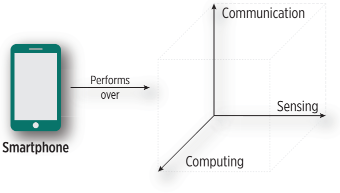


Figure: The advances in the communication, computing and sensing dimensions of mobile devices contribute to their acceptance by society [1].

Motivation

- The sensing dimension enables *context-awareness* in mobile devices, such as the smartphone.
- Battery advances are slower than those of other smartphone components [2], growing 5-10% yearly [3, 4], a critical issue for the **mobile sensing applications**.
- Scientific efforts have been done for achieving the energy efficiency of the GPS location provider.
- The understanding of mobility could augment the location-awareness of the smartphone for many purposes, such as energy savings and the development of Mobility Based Services (MBSs).

Research background

Problem statement

- The understanding of mobility is possible at different spatial-temporal scales:

Fine-grain mobility patterns identification

- They refer to the transportation mode employed by user when moving between stay points.
- Given a set of values $\mathcal{V} = v_{acc\ 1}, v_{acc\ 2}, \ldots, v_{acc\ n}$ obtained from accelerometer in the time interval $[t_1, t_2]$, identify fine-grain mobility information:

$$\textbf{FineGrainMobilityIdentifier}(\mathcal{V}) \rightarrow p_S \in \{\text{static, walking, biking, vehicle}\}$$

with each $v_{acc\ i} \in \mathcal{V}$ composed as $\langle acc_x, acc_y, acc_z, t \rangle$.

Coarse-grain mobility patterns identification

- They refer to motion at a large spatial scale related to user visiting stay points.
- Given a set of values $\mathcal{V} = v_{gps\ 1}, v_{gps\ 2}, \ldots, v_{gps\ n}$ obtained from GPS location provider in time interval $[t_1, t_2]$, identify coarse-grain mobility information:

$$\textbf{CoarseGrainMobilityIdentifier}(\mathcal{V}) \rightarrow p_S \in \{\text{new stay point, arrival, departure}\}$$

with each $v_{gps\ i} \in \mathcal{V}$ composed as $\langle lat, lon, t \rangle$.

Research background

Problem statement

Sensors sampling adaptation

- Given a set of coarse and fine-grain mobility patterns $\mathcal{P} = \{p_{s_1}, p_{s_2}, \ldots, p_{s_n}\}$, and accuracy requirements of mobile app $req_{accuracy}$, implement a sampling policy for the adaptive duty cycling of sensors while reducing energy consumption:

$$\textbf{PolicyGeneration}(\mathcal{P}, req_{accuracy}) \longrightarrow \mathcal{S}_{conf}$$

where $\mathcal{S}_{conf} \rightarrow s, \mathcal{T}_{real}$ represents the sampling $\mathcal{T}_{real}$ that must be implemented for sensor s. The $req_{accuracy}$ refers to the granularity of GPS sampling.

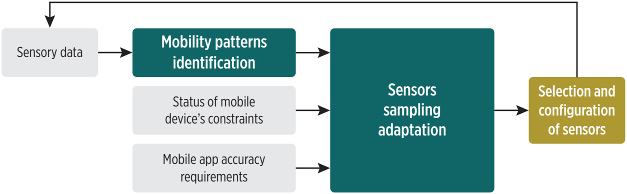

Sensory data
Mobility patterns identification
Status of mobile device's constraints
Mobile app accuracy requirements
Sensors sampling adaptation
Selection and configuration of sensors

Figure: Interaction between problems.

Research background

Hypothesis

Hypothesis

- The energy consumption of continuous and extended location tracking could be reduced by means of a cognitive dynamic system that learns an expanded spatial-time model from mobility events detected from sensors data and that employs such model in a cognitive controller for dynamically adapting GPS sampling rate through sampling policies tailored to current mobility state.

Research background

Objectives

Main objective

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

Particular objectives

- To detect mobility patterns from context information obtained from an inertial sensor (accelerometer) and location provider (GPS).
- To generate an accurate representation of detected patterns for summarizing user mobility.
- To dynamically adapt GPS sampling rate by means of a cognitive controller that employs the learned mobility representation and accuracy requirements for implementing power-aware sampling policies.
- To ease the development of mobile sensing applications that require user location tracking, i.e., LBSs and MBSs, isolating the complexity of sensors access and the associated efficient energy management.

Research background

Methodology

Methodology

1. **Revision of state of the art power-aware sensing techniques.**
2. **Formal definition and selection of mobility patterns to be identified.**
3. **Research on algorithms for detecting mobility patterns.**
4. **Design of the *Mobility Events Detector*.**
5. **Design of adaptive policies for energy efficient usage of sensors.**
6. **Design of the Cognitive Controller.**
7. **Development of a middleware involving the *Mobility Events Detector* and the Cognitive Controller for the Android platform.**
8. Experimentation in terms of spatial-time accuracy and energy efficiency.

State of the art

Taxonomy of solutions

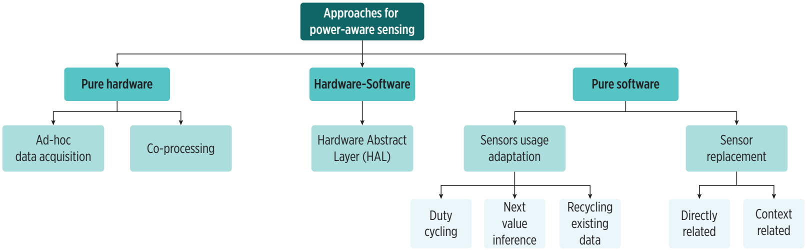


Figure: Taxonomy of solutions for power-aware sensing in mobility sensing systems.

State of the art

Remarks

Remarks

- The exploitation of coarse and fine-grain mobility information for modeling and characterizing user mobility has been barely explored.
- Although some of the proposed solutions employ a duty cycling strategy, it is fixed and obeys to instant mobility information, neglecting the temporal evolution of user mobility.
- A spatial-time accurate and energy efficient adaptive sampling could be produced by a cognitive approach that understands long-term mobility from fine and coarse-grain mobility events.
- The cognitive approach goes beyond typical pattern recognition and classic control strategies that follow a static configuration, as it evolves (in the learning and action tasks) across time.
- The smartphone itself could augment not only its location but also its mobility awareness (per-user basis).

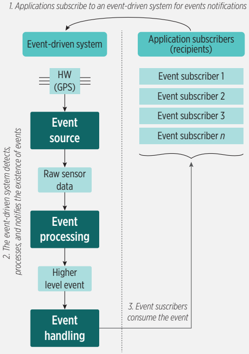
Theoretical framework

Stay points, Event Driven Systems (EDSs)

Stay point

- A stay point refers to a geographical zone (of size $\delta_{distance}$) where the user remains for an amount of time (δ_{time}).
- It is a virtual location defined by latitude (*lat*), longitude (*lon*), arrival time (*at*) and departure time (*dt*).

Event Driven Systems (EDSs)



1. Applications subscribe to an event-driven system for events notifications

Event-driven system

Application subscribers (recipients)

HW (GPS)

Event source

Raw sensor data

Event processing

Higher level event

Event handling

Event subscriber 1

Event subscriber 2

Event subscriber 3

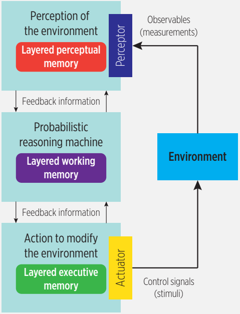
Event subscriber *n*

2. The event-driven system detects, processes, and notifies the existence of events

3. Event suscribers consume the event

Figure: The architecture of an EDS.

Cognitive Dynamic Systems (CDSs)



Perception of the environment

Layered perceptual memory

Perceptor

Observables (measurements)

Feedback information

Probabilistic reasoning machine

Layered working memory

Environment

Feedback information

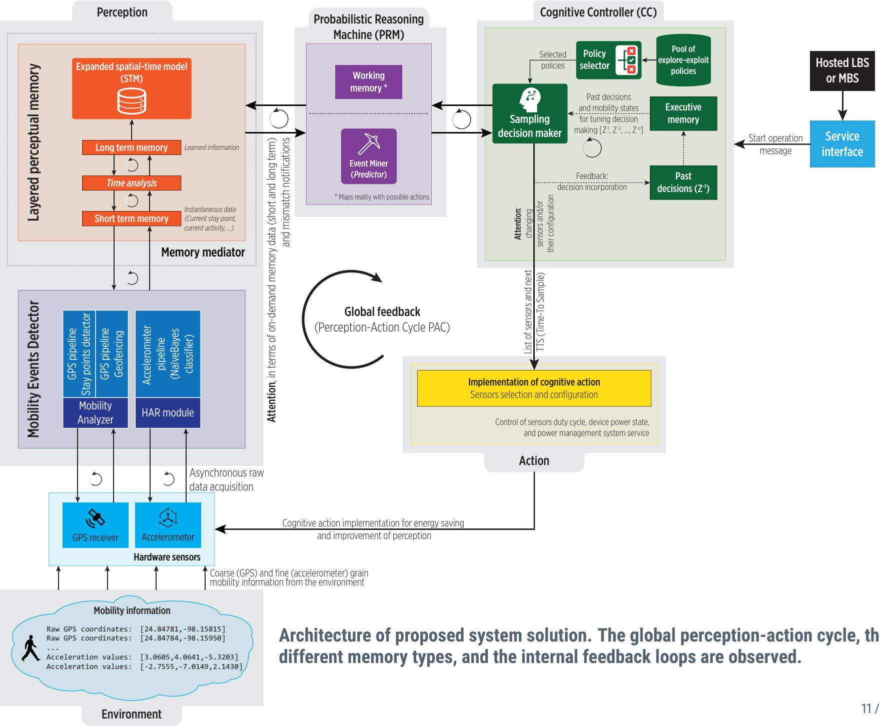
Action to modify the environment

Layered executive memory

Actuator

Control signals (stimuli)

Figure: The generic architecture of a CDS.



Perception

Layered perceptual memory

Expanded spatial-time model (STM)

Long term memory

Learned information

Time analysis

Short term memory

Instantaneous data (Current stay point, current activity, ...)

Memory mediator

Mobility Events Detector

GPS pipeline Stay points detector

GPS pipeline Geofencing

Mobility Analyzer

Accelerometer pipeline (NaiveBayes classifier)

HAR module

Asynchronous raw data acquisition

GPS receiver

Accelerometer

Hardware sensors

Coarse (GPS) and fine (accelerometer) grain mobility information from the environment

Mobility information

```
Raw GPS coordinates:  [24.84781,-98.15815]
Raw GPS coordinates:  [24.84784,-98.15950]
...
Acceleration values:  [3.0605,4.0641,-5.3203]
Acceleration values:  [-2.7555,-7.0149,2.1430]
```

Environment

Attention, in terms of on-demand memory data (short and long term) and mismatch notifications

Probabilistic Reasoning Machine (PRM)

Working memory *

Event Miner (*Predictor*)

* Maps reality with possible actions

Cognitive Controller (CC)

Selected policies

Policy selector

Pool of explore-exploit policies

Sampling decision maker

Past decisions and mobility states for tuning decision making [Z^{-1}, Z^{-2}, ..., Z^{-n}]

Executive memory

Feedback: decision incorporation

Past decisions (Z^{-1})

Attention: changing sensors and/or their configuration

Hosted LBS or MBS

Service interface

Start operation message

List of sensors and next TTS (Time-To Sample)

Global feedback (Perception-Action Cycle PAC)

Implementation of cognitive action

Sensors selection and configuration

Control of sensors duty cycle, device power state, and power management system service

Action

Cognitive action implementation for energy saving and improvement of perception

Architecture of proposed system solution. The global perception-action cycle, the different memory types, and the internal feedback loops are observed.

Perception components

Mobility Events Detector

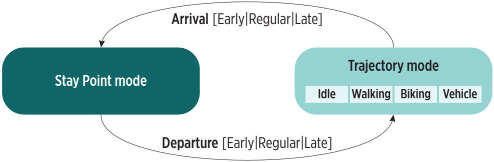


Figure: Individual's mobility as a sequence of high level states and associated events detected from raw sensor data [5, 6].

Mobility Events Detector

Aimed at identifying:

- Coarse-grain mobility events.
- Fine-grain mobility events.

Perception components

Mobility Events Detector: *Stay Points Detector* module

Stay Points Detector module

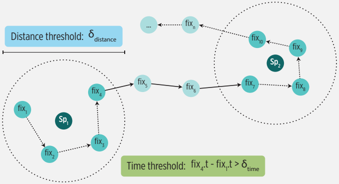
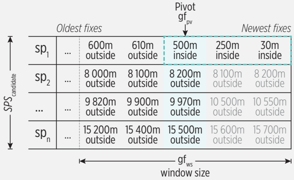


Figure: A conceptual representation of the stay points detection algorithm behavior.

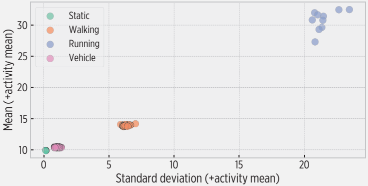
Geofencing module



sp_1	...	600m outside	610m outside	500m inside	250m inside	30m inside
sp_2	...	8 000m outside	8 100m outside	8 200m outside	8 100m outside	8 200m outside
...	...	9 820m outside	9 900m outside	9 970m outside	10 500m outside	10 550m outside
sp_n	...	15 200m outside	15 400m outside	15 500m outside	15 600m outside	15 700m outside

Figure: A conceptual representation of the window-based geofencing operation.

HAR module



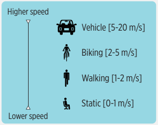


Figure: Distribution of mean and standard deviation features employed by the NaïveBayes classifier of the HAR module.

Layered perceptual memory

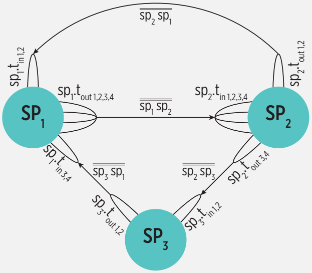
Short and long-term memory information

Layered perceptual memory

- Short-term memory information: current (observed) mobility status.
- Long-term memory information: the Expanded Spatial-Time model (STM).

Expanded Spatial-Time model

- The highest level of mobility information held by the system.



$\overline{sp_1\,sp_2}$ A summarized trajectory between sp_1 and sp_2

$\overline{sp_2\,sp_1}$ A summarized trajectory between sp_2 and sp_1

$\overline{sp_2\,sp_3}$ A summarized trajectory between sp_2 and sp_3

$\overline{sp_3\,sp_1}$ A summarized trajectory between sp_3 and sp_1

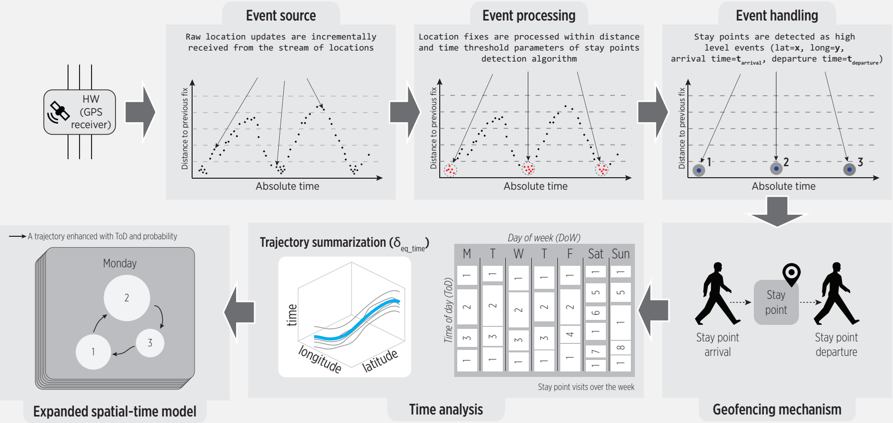
Figure: A conceptual representation of the STM's structure.

Layered perceptual memory

Expanded Spatial-Time model (STM)

Generation of the STM

- Incrementally built with the coarse-grain mobility events detected by the *Mobility Events Detector*.



Event source

Raw location updates are incrementally received from the stream of locations

Event processing

Location fixes are processed within distance and time threshold parameters of stay points detection algorithm

Event handling

Stay points are detected as high level events (lat=**x**, long=**y**, arrival time=$\mathbf{t}_{arrival}$, departure time=$\mathbf{t}_{departure}$)

Geofencing mechanism

Time analysis

Expanded spatial-time model

Figure: A conceptual representation of the steps for generating the STM from raw sensors data.

Probabilistic Reasoning Machine (PRM)

Working memory

PRM features

- It gives a meaning to the observed mobility information with respect of the STM information.
- It produces an estimation of future mobility state that links perceptual and working memory.

Interpretation

- The *Event Miner* traverses the **STM** for identifying whether learned information is:
  - Consistent, or
  - Inconsistent (mismatch)

  with respect of observed mobility information.

Estimation

- The *Event Miner* looks in the **STM** for a link (if any) with learned mobility information for generating spatial-time estimations:
  - Get next departure time.
  - Get next arrival time.

Cognitive controller (CC)

Description

Goals

- To reduce the energy consumption of location tracking by relying on PRM's estimations.
- To reduce the system uncertainty about current user mobility.

Possible cognitive actions

- **Exploitation policies**: When system uncertainty is low for saving energy purposes.
- **Exploration policies**: When system uncertainty is high for recovering for accuracy loss.

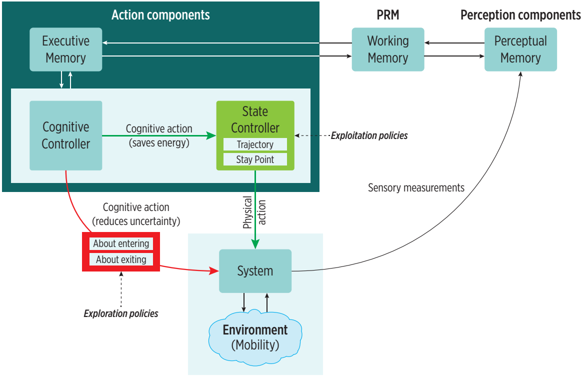


Figure: A generic architecture for a cognitive controller.

Cognitive controller

Policies tailored for user mobility

Stay point mode

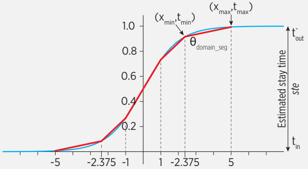


Figure: Approximation of the sigmoid through straight segments.

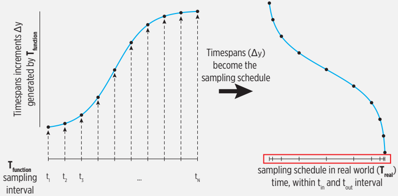
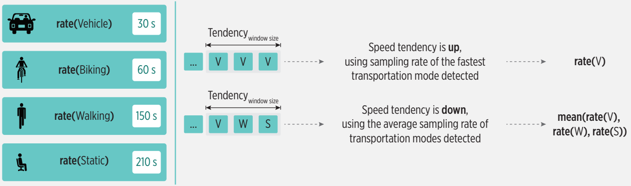

Timespans increments Δy generated by $T_{function}$
$T_{function}$ sampling interval
t_1
t_2
t_3
...
t_N
Timespans (Δy) become the sampling schedule
sampling schedule in real world (T_{real}) time, within t_{in} and t_{out} interval

Figure: A snapshot of the process for producing a sigmoid sampling.

Trajectory mode



rate(Vehicle)	30 s
rate(Biking)	60 s
rate(Walking)	150 s
rate(Static)	210 s

Speed tendency is **up**, using sampling rate of the fastest transportation mode detected → **rate**(V)

Speed tendency is **down**, using the average sampling rate of transportation modes detected → **mean(rate**(V), **rate**(W), **rate**(S))

Figure: GPS sampling adaptation during trajectory based on the speed tendency of detected transportation modes.

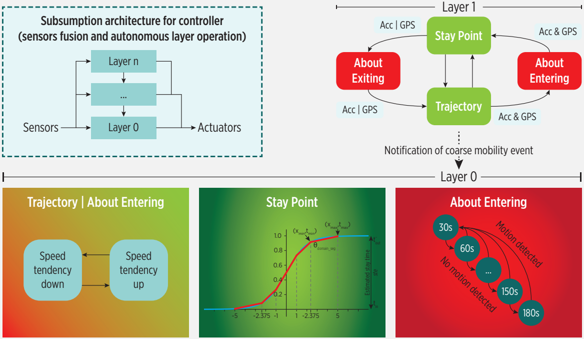
Cognitive controller

Sampling Decision Maker module

Sampling Decision Maker module

- It filters from the *pool of exploration-exploitation policies* those apt for the mobility state detected by PRM.
- It updates its *Executive Memory* with the selected cognitive action for feedback in further executions.

Implementation of cognitive action



Exploration-exploitation policy

Exploitation policy

Exploration policy

Figure: The subsumption architecture of the controller and the reactions for different coarse grain mobility events.

Implementation

Android device implementation

Smartphone device

The Google Nexus 6 smartphone with Android OS 7 was employed.

- Quad-core 2.7 GHz Krait 450 mobile processor (Qualcomm Snapdragon 805 chipset).
- 3 GB RAM.
- Qualcomm Snapdragon 805 chipset.
- 3220 mAh battery.

Technical barriers

1. Asynchronous access to sensors, affected by sensing infrastructure (GPS satellite signal intermittency).
2. Out-of-the-box energy saving mechanisms:
   - Application specific: **App StandBy**.
   - System wide: **Doze mode**.

Workarounds

1. Grace periods for sampling (`Timer` + `TimerTasks`).
2. `Alarms`, `WakeLocks`, foreground services.

Experimentation

Materials and methods

Ground-truth mobility information

- High frequency (1 *Hz*) location data collected employing a Qstarz BT-Q1000eX GPS logger.
- 6 trajectories collected by the same user, mostly using a vehicle as transportation mode.

Stay Points Detector	**Time threshold** (δ_{time}):	45 *min*
	Distance threshold ($\delta_{distance}$):	500 *m*
Geofencing	**Radio distance** ($gf_{distance}$):	250 *m*
	Window size (gf_{ws}):	3
Sampling period:	1 second	

Table: Input parameters for the discovery of ground truth mobility information.

Trajectory	Duration (days)	Inside SP time (minutes)	In traj. time (minutes)	Total SPs	Total visits	Individual SP weight	
Trajectory 1	5.38	7,442.10 (96.06%)	305.07 (3.94%)	4	13	Home: 72.25% Bob's place: 0.94%	Civestav: 26.03% Store: 0.78%
Trajectory 2	6.05	8,348.42 (95.80%)	365.73 (4.20%)	6	21	Home: 69.67% Bob's place: 1.20% Store: 0.82%	Cinvestav: 26.51% Park: 0.99% Stadium: 0.82%
Trajectory 3	6.15	8,542.12 (96.53%)	307.45 (3.47%)	6	31	Home: 67.94% Park: 2.94% Fast food: 1.15%	Cinvestav: 24.50% Bob's place: 2.58% City center: 0.89%
Trajectory 4	7.22	10,024.63 (96.42%)	371.98 (3.58%)	2	16	Home: 66.50%	Cinvestav: 33.50%
Trajectory 5	7.35	10,026.22 (94.74%)	556.20 (5.26%)	4	19	Home: 67.29% Cinema: 2.41%	Cinvestav: 29.33% City center: 0.97%
Trajectory 6	34.31	47,599.62 (96.34%)	1,807.52 (3.66%)	11	146	Home: 65.31% Home 2: 2.81% Bob's place: 0.37% Store: 0.35% Workshop: 0.20% Workshop 2: 0.11%	Cinvestav: 29.36% Fast food: 0.71% Restaurant: 0.36% City center: 0.30% Store 2: 0.12%

Table: Mobility information of collected ground truth trajectories (SP=stay point).

Experimentation

Spatial-time accuracy of *Stay Points Detector* and *Geofencing* modules

Description

- Evaluation of the spatial-time accuracy of the *Stay Points Detector* module: centroid distances and latencies.
- Evaluation of the spatial-time accuracy of the *Geofencing* module: missed visits, and latencies.

Stay Points Detector	**Time threshold** (δ_{time}):	45 *min*
	Distance threshold ($\delta_{distance}$):	500 *m*
Sampling periods:	30, 60, 90, 120, 150, 180 seconds.	
Trajectories:	All ground truth trajectories.	

Table: Input parameters for the spatial-time accuracy of stay points experiment.

Geofencing	**Radio distance** ($gf_{distance}$):	250 *m*
	Window size (gf_{ws}):	3, 5, 7
Sampling periods:	30, 60, 90, 120, 150, 180 seconds.	
Trajectories:	All ground truth trajectories.	
STM	Preloaded with ground truth stay points.	

Table: Input parameters for the spatial-time accuracy of *Geofencing* experiment.

Results

- *Stay Points Detector* and *Geofencing* modules performed with spatial-time accuracy:
  - A maximum centroid distance of 22.52 *m* is identified when employing the 180 seconds sampling period.
  - The length of the visits missed by the *Geofencing* module are under 90 seconds. Latencies are in function of the window size parameter value.

Experimentation

Reaction to STM mobility mismatches: Description

Description

- This experiment was focused on evaluating the ability of the system for reacting to mismatches in mobility information, with a particular emphasis on mismatch departure events.
- For some trials the information of the STM was intentionally modified by giving it longer stay times than those in the actual visits.
- The missed visits, the delay for detecting the temporal mismatches, and the departure latency were evaluated.

Geofencing	**Radio distance** ($gf_{distance}$):	250 m
	Window size (gf_{ws}):	3
Cognitive Controller	**Sigmoid segments** (Θ_{domain_segs}):	$(-5, -2.375), (-2.375, -1), (-1, 1), (1, 2.375), (2.375, 5)$ $(-4, -2.375), (-2.375, -1), (-1, 1), (1, 2.375), (2.375, 5)$
	Time separations (Θ_{time_sep}):	$[90, 150, 180, 150, 90]$ seconds $[90, 120, 180, 120, 90]$ seconds $[60, 150, 180, 150, 60]$ seconds $[60, 120, 180, 120, 60]$ seconds
Trajectories	All ground truth trajectories.	
STM	Preloaded with ground truth stay points, with increased time (injured) in the proportions $[5, 10, 15, \ldots, 90, 95, 100]$ %	

Table: Input parameters for the reaction to mobility mismatches experiment.

Experimentation

Reaction to STM mobility mismatches: Results

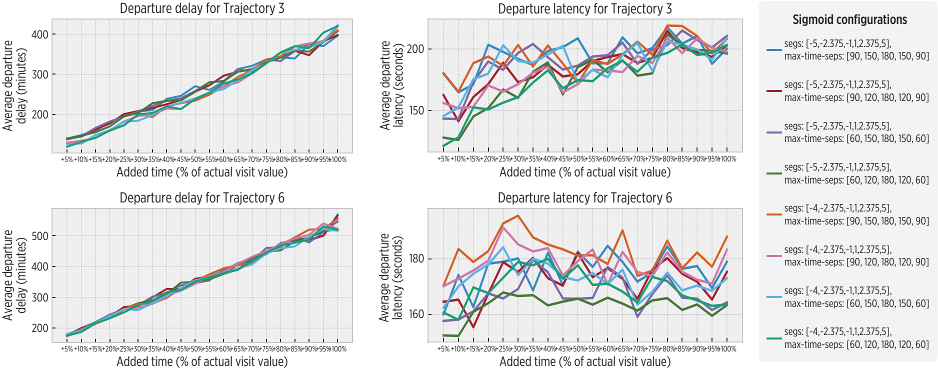


Figure: On the left, the evolution of the timespan between the expected and actual departures detected by the system when the STM is injured with additional time. Similar tendencies are observed, highlighting that the sigmoid sampling allows to identify that user is leaving the stay points before expected.
On the right, the variation of the latency values of departures detection. The observed values are under the theoretical 360 seconds maximum peak

Experimentation

Holistic evaluation: Description

Description

- This experiment was aimed at evaluating the overall spatial-time accuracy performance of the system with all of its component enabled, including:
  - The continuous learning of stay points in the STM for adapting GPS sampling rate.
  - The handling of mobility mismatches.

Geofencing	**Radio distance** ($gf_{distance}$):	250 m
	Window size (gf_{ws}):	3
Cognitive Controller	**Sigmoid segments** (Θ_{domain_segs}):	$(-5, -2.375), (-2.375, -1), (-1, 1), (1, 2.375), (2.375, 5)$ $(-4, -2.375), (-2.375, -1), (-1, 1), (1, 2.375), (2.375, 5)$
	Time separations (Θ_{time_sep}):	$[90, 150, 180, 150, 90]$ seconds $[90, 120, 180, 120, 90]$ seconds $[60, 150, 180, 150, 60]$ seconds $[60, 120, 180, 120, 60]$ seconds
	On trajectory sampling:	30 seconds
	Conservative sampling:	60 seconds
Trajectories	All ground truth trajectories.	
STM	Empty	

Table: Input parameters for the holistic evaluation experiment. The *conservative* sampling refers to the late departure mismatch reaction.

Experimentation

Holistic evaluation: Results

Sigmoid configurations

- segs: [-5,-2.375,-1,1,2.375,5], max-time-seps: [90, 150, 180, 150, 90]
- segs: [-5,-2.375,-1,1,2.375,5], max-time-seps: [60, 120, 180, 120, 60]
- segs: [-4,-2.375,-1,1,2.375,5], max-time-seps: [60, 150, 180, 150, 60]
- segs: [-5,-2.375,-1,1,2.375,5], max-time-seps: [90, 120, 180, 120, 90]
- segs: [-4,-2.375,-1,1,2.375,5], max-time-seps: [90, 150, 180, 150, 90]
- segs: [-4,-2.375,-1,1,2.375,5], max-time-seps: [60, 120, 180, 120, 60]
- segs: [-5,-2.375,-1,1,2.375,5], max-time-seps: [60, 150, 180, 150, 60]
- segs: [-4,-2.375,-1,1,2.375,5], max-time-seps: [90, 120, 180, 120, 90]
- 30 secs fixed sampling

Arrival latency

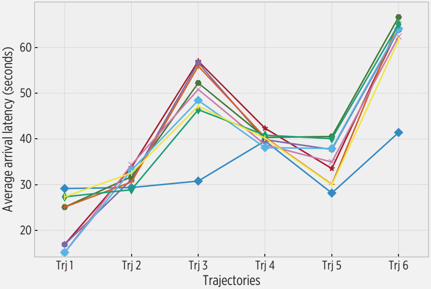


Figure: Arrival latency observed by the platform in experimental trials. The largest average value is below 65 seconds, explained by the fact that 2 location updates must be collected by the *Geofencing* module before identifying an arrival event.

Departure latency

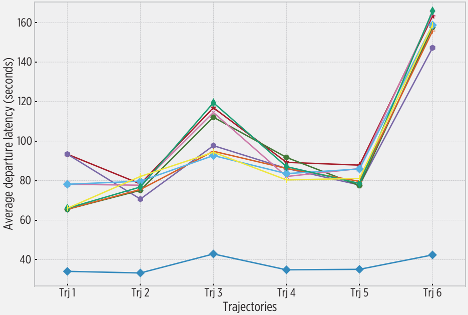


Figure: Departure latency observed by the platform across performed experimental trials. The latencies are within 65 and 165 seconds, which is aligned with the different values specified to the CC for its sigmoid-driven sampling.

Experimentation

Holistic evaluation: Results

Sigmoid configurations

- ★ segs: [-5,-2.375,-1,1,2.375,5], max-time-seps: [90, 150, 180, 150, 90]
- + segs: [-5,-2.375,-1,1,2.375,5], max-time-seps: [60, 120, 180, 120, 60]
- ◆ segs: [-4,-2.375,-1,1,2.375,5], max-time-seps: [60, 150, 180, 150, 60]
- ⬢ segs: [-5,-2.375,-1,1,2.375,5], max-time-seps: [90, 120, 180, 120, 90]
- × segs: [-4,-2.375,-1,1,2.375,5], max-time-seps: [90, 150, 180, 150, 90]
- + segs: [-4,-2.375,-1,1,2.375,5], max-time-seps: [60, 120, 180, 120, 60]
- ⬢ segs: [-5,-2.375,-1,1,2.375,5], max-time-seps: [60, 150, 180, 150, 60]
- ◆ segs: [-4,-2.375,-1,1,2.375,5], max-time-seps: [90, 120, 180, 120, 90]
- ◆ 30 secs fixed sampling

Trajectory distance difference

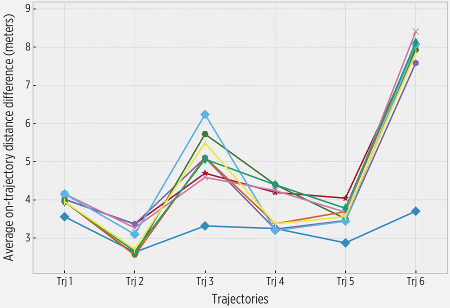


Figure: The average distance of equivalent trajectory segments during experimental trials. The values are enclosed within 2.5 *m* and 8.5 *m*, with the 30 seconds sampling obtaining the lowest value in each trial.

Overall reduction of location update requests

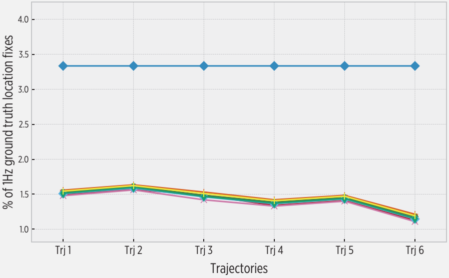


Figure: The proportion of location update requests employed by each experimental trial with respect to the corresponding 1 *Hz* ground truth trajectory. All of the parameter combinations outperform the 30 seconds sampling period, which provides a rough estimation of the energy savings that the system could achieve in on-device implementations.

Experimentation

Energy consumption of fixed-sampling periods: Description

Description

- This experiment evaluated the impact of sampling adaptations based on fixed sampling rates on the energy consumed by the proposed CDS.
- All system components were enabled, with the exception of the sigmoid sampling of the CC.
- The CC followed fixed sampling periods depending on the current mobility state recognized by the system.
- The experiment also demonstrated the mobility awareness that the system provides to the smartphone (STM).

Stay Points Detector	**Time threshold** (δ_{time}):	45 *min*
	Distance threshold ($\delta_{distance}$):	500 *m*
Geofencing	**Radio distance** ($gf_{distance}$):	250 *m*
	Window size (gf_{ws}):	3
HAR module	**Individual window length**:	5 seconds
	Meta-window size (HAR_{mws}):	5
Cognitive Controller	**Smartphone 1**:	On trajectory sampling periods: 30 seconds On stay point sampling period: 30 seconds
	Smartphone 2:	On trajectory sampling period: 30 seconds On stay point sampling period: one in the set $\{60, 90, 120, 150, 180\}$ seconds

Table: Input parameters for the energy consumption of fixed-sampling periods experiment.

Experimentation

Energy consumption of fixed-sampling periods: Results

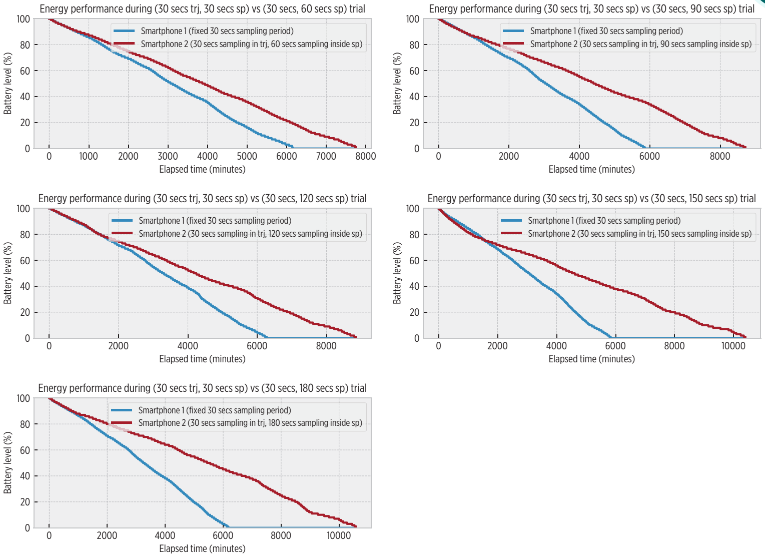


Figure: Energy performance of a fixed 30 seconds sampling versus a basic sampling adaptation consisting in a 30 seconds sampling in trajectory mode and a slower sampling rate during stay point mode. The separation between the lines in each plot starts after the system learns the stay points with the largest weight in user mobility (home and work places).

Experimentation

Energy consumption of fixed-sampling periods: Results

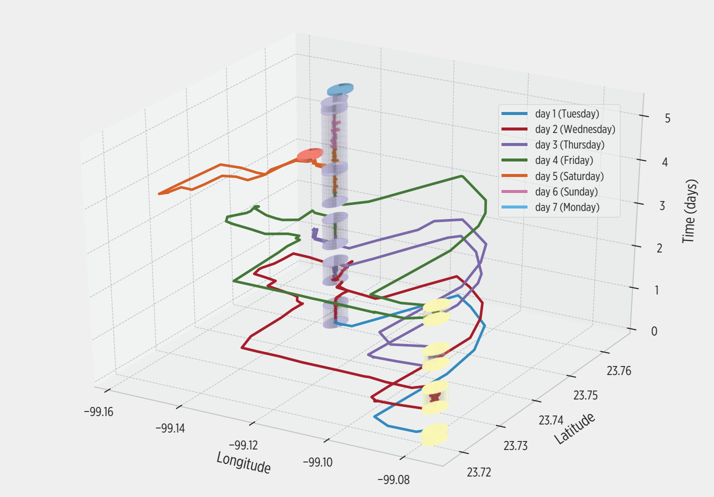


Figure: The information autonomously learned by the STM during the trial corresponding to the 30 seconds in trajectory and 60 seconds in stay point sampling scheme. The height of cylinders corresponds with the stay time during each stay point visit.

Experimentation

Energy overhead of cognitive system

Description

- Evaluation of the energy overhead of the proposed CDS.
- A sample application was developed for collecting HAR + GPS data with a 30 seconds sampling rate.
- One smartphone unit instructed the sampling rate without using any cognitive feature.
- Another smartphone unit used the cognitive features with the exception of the cognitive controller.

Stay Points Detector	**Time threshold** (δ_{time}):	45 *min*
	Distance threshold ($\delta_{distance}$):	500 *m*
Geofencing	**Radio distance** ($gf_{distance}$):	250 *m*
	Window size (gf_{ws}):	3
HAR module	**Individual window length**:	5 seconds
	Meta-window size (HAR_{mws}):	5
Cognitive Controller	Disabled	

Table: Input parameters for the energy overhead measurement experiment (*Geofencing* and *Stay Points Detector* enabled only in one smartphone).

Results

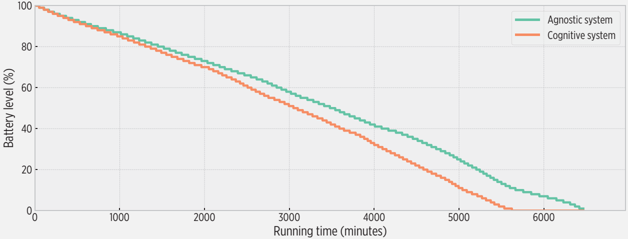


Figure: Energy overhead of the cognitive system. The running time difference is about 14 hours.

Future work

Pending tasks

		2017					2018							
#	Activity	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug
	Solution refinement activities													
1	Incorporation of the HAR module in the CC	●	●											
2	Incorporation of the accuracy requirement in the CC		●	●										
	On-device implementation													
3	Implementation of sigmoid-driven sampling				●									
4	Watchdog mechanisms for *Geofencing* and *Sampling Decision Maker* modules				●									
5	Refinement of the list of candidate stay points employed by the *Geofencing* module				●	●								
	Experimentation													
6	Experiments with larger and heterogeneous mobility						●	●	●					
7	Completion of evaluation framework					●	●	●						
8	Comparison with other solutions									○	○			
	Research work activities													
9	Thesis writing-review								○	○	○	○	○	
10	Thesis defense													○

Table: Schedule of pending activities of the research work for the last year of the doctoral program.

Conclusions

Conclusions

- The advances performed during the last year of work have been presented.
- The cognitive features of proposed system allow to identify, learn and exploit mobility information in a novel way.
- The experiments demonstrate the system's ability to identify the coarse-grain mobility events for building the STM:
  - Centroid distances of observed stay points are under $22.5m$, the arrival and departure latencies are within the active sampling rate.
  - The length of the visits missed by the *Geofencing* are short, no longer than 4 minutes.
- The STM information has proved its usefulness for the CC to reduce energy consumption with a minor impact on spatial-time accuracy:
  - Only $33 - 49$ % of the location updates of a fixed $30s$ sampling rate are employed in simulations.
  - On-device trials show a battery life increase of $26 — 76h$, with respect of a fixed $30s$ sampling rate.
- The hypothesis is demonstrated as shown by the implementation of the CDS and the associated experimentation.
- The EDS approach eases the design and implementation of the system, as it suits the characteristics of mobile platforms and mobility analysis.
- The developed middleware isolates sensors and power management complexity for long-term LBSs and MBSs.
- Advances on the research work have been submitted for publication (three journal articles [7, 8, 9]).

Thank you for your attention!

> Consider again that dot [Earth]. That’s here. That’s home. That’s us.
>
> *Carl Sagan*

Cinvestav
Tamaulipas

References I

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

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

[3] 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.

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

[5] Laura Alessandretti, Piotr Sapiezynski, Sune Lehmann, and Andrea Baronchelli.
Multi-scale spatio-temporal analysis of human mobility.
PLOS ONE, 12(2):e0171686, feb 2017.

[6] Xiang-Wen Wang, Xiao-Pu Han, and Bing-Hong Wang.
Correlations and Scaling Laws in Human Mobility.
PLoS ONE, 9(1):e84954, jan 2014.

[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 Perez-Torres, Cesar Torres-Huitzil, and Hiram Galeana-Zapien.
Full on-device stay points detection in smartphones for location-based mobile applications.
Sensors (Switzerland), 16(10), 2016.

[9] Rafael Perez-Torres, Cesar Torres-Huitzil, and Hiram Galeana-Zapien.
An on-device cognitive dynamic systems inspired sensing framework for the IoT.
IEEE Communications Magazine, forthcoming, 2018.

Research background

Contributions

Contributions

- An on-device mobility patterns detector that works with streams of raw data collected by smartphone's sensors (GPS and accelerometer).
- An on-device mobility analyzer that incrementally builds a model of user mobility from the detected mobility patterns.
- A cognitive controller inspired on CDSs that, based on the mobility information learned, dynamically adapts GPS sampling rate through power-aware policies.
- A middleware with the previous modules embedded for easing the development of LBSs and MBSs for the Android mobile platform.

State of the art

Taxonomy of solutions

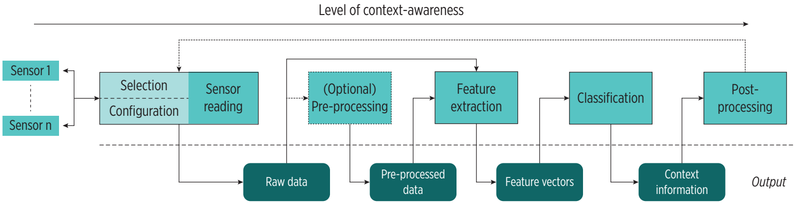


Figure: Stages of mobile sensing apps.

Theoretical framework

Stay points

Calculation

- A GPS position fix p is defined by latitude (lat), longitude (lon), and timestamp (t).
- A stay point (**sp**) is calculated from a set of consecutive GPS fixes $\mathbf{P} = \{p_m, p_{m+1}, \ldots, p_n\}$, and $\delta_{distance}$ and δ_{time} thresholds.
- Its composition must observe:

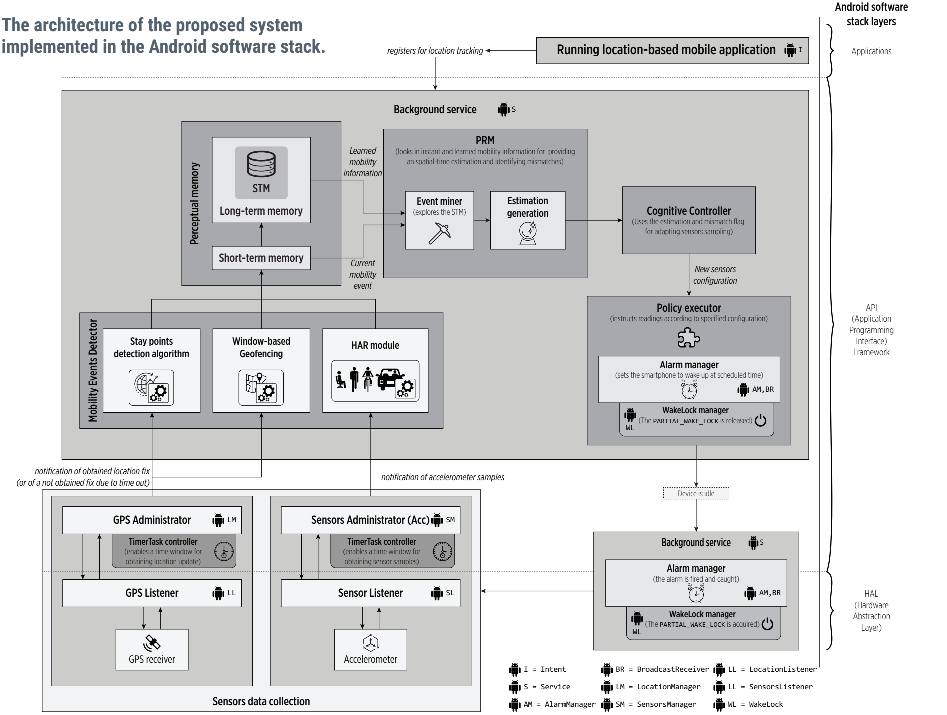
$$|p_n.t - p_m.t| \geq \delta_{time} \qquad \text{distance}(p_m, p_i) \leq \delta_{distance}, \forall m < i \leq n$$

- Its centroid coordinates are the arithmetical means:

$$\mathbf{sp}.lat = \frac{\sum_{i=m}^{n} p_i.lat}{|\mathbf{P}|} \qquad \mathbf{sp}.lon = \frac{\sum_{i=m}^{n} p_i.lon}{|\mathbf{P}|}$$

- The at and dt components are set to $p_m.t$ and $p_n.t$, respectively.

The architecture of the proposed system implemented in the Android software stack.


Android software stack layers
Applications
API (Application Programming Interface) Framework
HAL (Hardware Abstraction Layer)
Running location-based mobile application I
registers for location tracking
Background service S
Perceptual memory
STM
Long-term memory
Short-term memory
Learned mobility information
Current mobility event
PRM
(looks in instant and learned mobility information for providing an spatial-time estimation and identifying mismatches)
Event miner
(explores the STM)
Estimation generation
Cognitive Controller
(Uses the estimation and mismatch flag for adapting sensors sampling)
New sensors configuration
Policy executor
(instructs readings according to specified configuration)
Alarm manager
(sets the smartphone to wake up at scheduled time)
AM,BR
WakeLock manager
(The PARTIAL_WAKE_LOCK is released)
WL
Mobility Events Detector
Stay points detection algorithm
Window-based Geofencing
HAR module
notification of obtained location fix (or of a not obtained fix due to time out)
notification of accelerometer samples
Device is idle
Background service S
Alarm manager
(the alarm is fired and caught)
AM,BR
WakeLock manager
(The PARTIAL_WAKE_LOCK is acquired)
WL
GPS Administrator LM
TimerTask controller
(enables a time window for obtaining location update)
GPS Listener LL
GPS receiver
Sensors Administrator (Acc) SM
TimerTask controller
(enables a time window for obtaining sensor samples)
Sensor Listener SL
Accelerometer
Sensors data collection
I = Intent
S = Service
AM = AlarmManager
BR = BroadcastReceiver
LM = LocationManager
SM = SensorsManager
LL = LocationListener
LL = SensorsListener
WL = WakeLock

Experimentation

Materials and methods

Attributes evaluated

- Spatial-time accuracy.
- Energy consumption.

Desktop version

- It features different modules of the proposed system (except from HAR module).
- It allows the quick evaluation of system performance through different parameter combinations.
- It includes a logic *trajectory file* reader that simulates user motion.

On-device trials

- Two Nexus 6 smartphone units.
- The smartphones were always carried together with the GPS logger device by a campus student.
- No mobile apps other than the developed middleware were executed by the smartphones.
- The smartphones were not employed for communication tasks (texting, calls).

Experimentation

Stay Points Detector module spatial-time accuracy: Results

Trajectory	Sampling period (seconds)	Live stay points identified	Average centroid distance (meters)	Average arrival latency (seconds)	Average departure latency(seconds)
Trajectory 1	30	12 of 12	1.50	2.67	24.92
	60	12 of 12	3.43	***-12.33***	17.42
	90	12 of 12	4.04	15.17	32.42
	120	12 of 12	6.66	***-7.33***	52.42
	150	12 of 12	9.00	25.17	79.92
	180	12 of 12	10.88	22.67	77.42
Trajectory 2	30	16 of 16	1.59	7.38	13.62
	60	16 of 16	4.72	20.50	34.25
	90	16 of 16	4.42	37.38	26.75
	120	16 of 16	12.51	16.75	83.00
	150	16 of 16	15.04	***-0.12***	96.12
	180	16 of 16	15.58	65.50	71.75
Trajectory 3	30	19 of 19	3.39	2.26	12.16
	60	19 of 19	3.96	11.74	12.16
	90	19 of 19	8.05	40.16	29.53
	120	19 of 19	11.16	37.00	56.37
	150	19 of 19	18.06	48.05	59.53
	180	19 of 19	**22.52**	87.53	81.63
Trajectory 4	30	13 of 13	0.49	16.15	2.46
	60	13 of 13	1.05	41.54	34.77
	90	13 of 13	2.41	32.31	55.54
	120	13 of 13	4.00	32.31	71.69
	150	13 of 13	4.78	41.54	50.92
	180	13 of 13	5.19	73.85	62.46
Trajectory 5	30	18 of 18	1.49	0.17	13.61
	60	18 of 18	2.31	13.50	21.94
	90	18 of 18	5.12	***-3.17***	23.61
	120	18 of 18	14.46	10.17	***-44.72***
	150	18 of 18	13.58	30.17	***-39.72***
	180	18 of 18	14.39	26.83	***-71.39***
Trajectory 6	30	75 of 75	1.89	2.89	6.89
	60	**76 of 75**	3.54	8.49	24.89
	90	**76 of 75**	4.67	7.29	59.69
	120	**76 of 75**	6.71	29.29	77.69
	150	75 of 75	9.33	42.89	113.29
	180	**76 of 75**	10.29	51.69	108.89

Table: Spatial-time differences in detected stay points per sampling period (ST=stay time). The negative values in the ST difference and the arrival and departure latencies are caused by the combined effect of user mobility and sparse sampling rate on the *StreamedZhen* algorithm, which generates stay points in subtly different coordinates with different time information.

Experimentation

Geofencing module spatial-time accuracy: Results

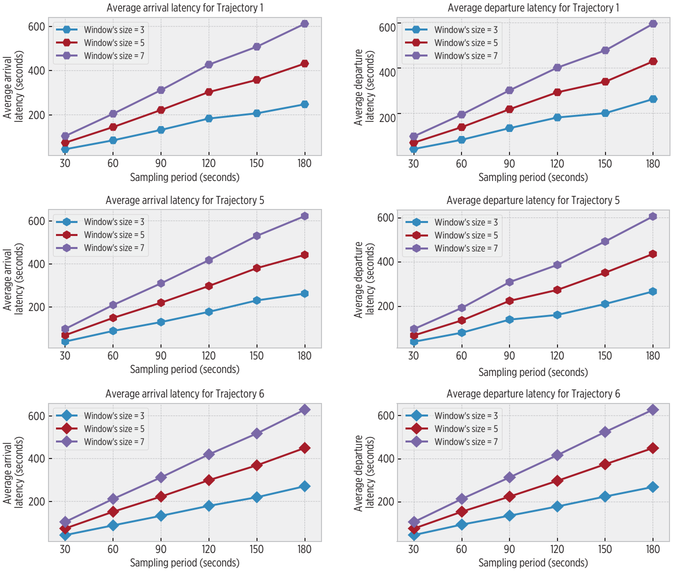


Figure: Arrival (left) and departure (right) latencies obtained by the Geofencing module for each combination of sampling period and window length values. There is a tendency on the results as the shortest window's sizes produce the shortest arrival latency values.

Preliminary experiments

Geofencing module spatial-time accuracy: Results

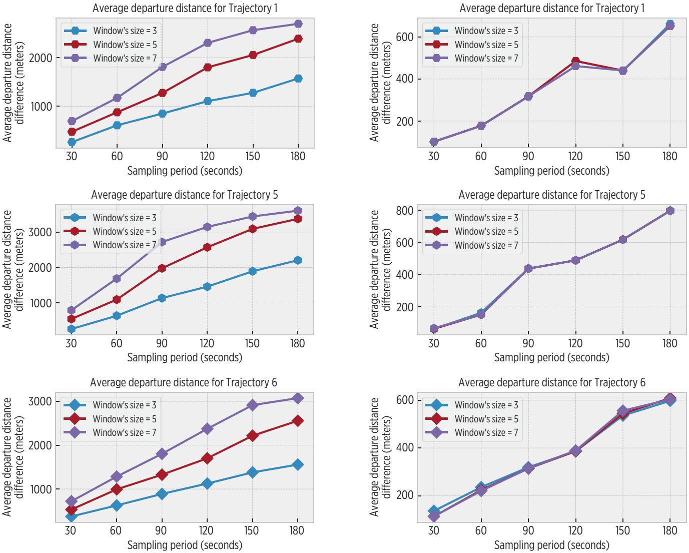


Figure: Departure distance difference obtained by the *Geofencing* module for each combination of sampling period and window size values. On the left, the latency of the window-based geofencing is accounted, while in the right it is ignored. The high speed with which user leaves from stay points allows to identify a tendency on the results, as departures are detected earlier by shorter window sizes.

Experimentation

Holistic evaluation: Results

Sigmoid configurations

- ★ segs: [-5,-2.375,-1,1,2.375,5], max-time-seps: [90, 150, 180, 150, 90]
- + segs: [-5,-2.375,-1,1,2.375,5], max-time-seps: [60, 120, 180, 120, 60]
- ◆ segs: [-4,-2.375,-1,1,2.375,5], max-time-seps: [60, 150, 180, 150, 60]
- ⬢ segs: [-5,-2.375,-1,1,2.375,5], max-time-seps: [90, 120, 180, 120, 90]
- × segs: [-4,-2.375,-1,1,2.375,5], max-time-seps: [90, 150, 180, 150, 90]
- + segs: [-4,-2.375,-1,1,2.375,5], max-time-seps: [60, 120, 180, 120, 60]
- ⬢ segs: [-5,-2.375,-1,1,2.375,5], max-time-seps: [60, 150, 180, 150, 60]
- ◆ segs: [-4,-2.375,-1,1,2.375,5], max-time-seps: [90, 120, 180, 120, 90]
- ◆ 30 secs fixed sampling

Arrival distance difference

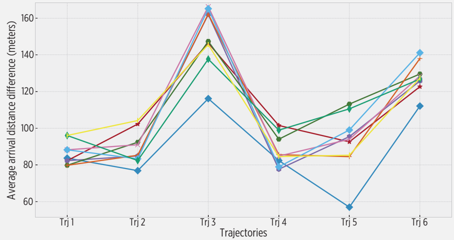


Figure: Arrival distance difference detected by the system throughout the different experimental trials. The values are shorter than for departure distance due to the decreasing speed that user describes when arriving to a stay point.

Departure distance difference

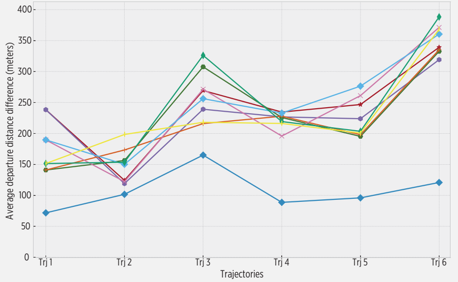


Figure: Departure distance difference detected by the system throughout the different experimental trials. The larger distance differences are caused by the high speed with which user leaves the stay points (mostly using a vehicle as transportation mode).

Preliminary experiments

Energy saving expectations of on-device stay points detection

Description

- This experiment explored whether a smartphone could detect stay points by itself, given its energy and computing constraints, and the energy savings of such implementation compared against typical solutions.
- Typical solutions implement a Mobile Cloud Computing (MCC) approach on which the smartphone only collects and offloads the processing to external servers.

Stay Points Detector	**Time threshold** (δ_{time}):	45 *min*
	Distance threshold ($\delta_{distance}$):	500 *m*
Sampling periods:	30, 60, 90, 120, 150 seconds	

Table: Input parameters for the energy saving expectations of on-device stay points detection experiment.

Results

Sampling period (seconds)	Processing strategy	Location update requests	GPS-on time (minutes)	Average acquisition time per fix (seconds)	Total experiment time (minutes)	Data sent (bytes)	Data received (bytes)
30	On-device	12,365	1,614	7.83	7,790	-	-
	MCC	9,324	770	4.95	5,402	1,084,901	18,796
60	On-device	10,851	1,219	6.74	12,028	-	-
	MCC	7,210	764	6.35	7,907	838,640	14,696
90	On-device	7,935	1,178	8.90	13,075	-	-
	MCC	5,626	546	5.82	8,946	653,833	12,223
120	On-device	5,246	809	9.25	11,289	-	-
	MCC	4,333	387	5.35	8,931	504,012	8,838
150	On-device	5,635	933	9.93	14,998	-	-
	MCC	4,566	452	5.93	11,619	530,764	10,309

Table: Summary of experimental results of on-device vs. MCC approach comparison.

Experimentation

Energy saving expectations of on-device stay points detection

Description

- This experiment explored whether a smartphone could detect stay points by itself, and the energy savings of such implementation with respect of typical Mobile Cloud Computing (MCC) based solutions.

Stay Points Detector	**Time threshold** (δ_{time}):	45 *min*
	Distance threshold ($\delta_{distance}$):	500 *m*
Sampling periods:	30, 60, 90, 120, 150 seconds	

Table: Input parameters for the energy saving expectations of on-device stay points detection experiment.

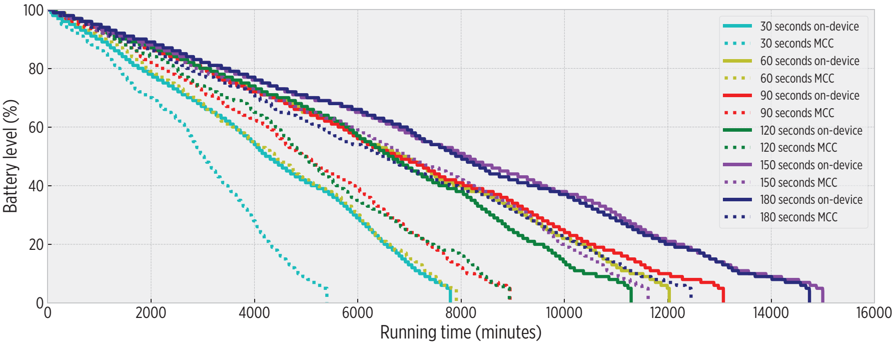


Figure: Energy performance comparison of on-device vs. MCC sample apps using different GPS sampling periods. Each of the on-device trials last longer than its corresponding remote implementation.

Experimentation

Comparison with other solutions

Work	Purpose	Mobility type	Involved sensors	Trajectory tracking	Place learning
SenseLess	Location tracking	Walking, static	Accelerometer, GPS	Yes	No
SmartDC	Place tracking	Not specified	Cellular id, Wi-Fi, GPS	No	Yes
Proposed system	Location and place tracking	Static, walking, biking, vehicle	Accelerometer, GPS	Yes	Yes

Table: Features comparison of proposed system and representative existing solutions.

Preliminary results

Comparison with other solutions

Work	**Place learning technique**	**Sampling adaptation technique**	**Energy metric**	**Energy result**
SenseLess	None	Simple Decision Rule (static, motion)	Energy profiling of sensors	41% consumption (of a fixed 10*s* GPS sampling)
SmartDC	Wi-Fi fingerprinting	Markov Decision Process (time series mobility predictor and energy budget).	Energy profiling of sensors	81% less than a periodic sensing scheme.
Proposed system	GPS Stream-based SPs detection, Geofencing, time-aware visits analysis.	Cognitive controller (sigmoid sampling)	Battery lifetime, amount of location updates	Simulation: 14%, 42% location updates (of fixed 10*s*, 30*s* GPS sampling). On device deployment: 53 hours battery life increase, 77% location updates (of a fixed 30*s* GPS sampling),

Table: Results comparison of proposed system and representative existing solutions.