Matador: Mobile Task Detector for Context-Aware Crowd-Sensing Campaigns

Iacopo Carreras and Daniele Miorandi
CREATE-NET
Via alla Cascata 56/D, Trento, Italy
name.surname@create-net.org

Andrei Tamilin
U-Hopper
Via alla Cascata 56/D, Trento, Italy
andrei.tamilin@u-hopper.com

Emmanuel R Ssebaggala, Nicola Conci
Univ. of Trento
Via Sommarive 5, I-38123 Povo, Trento
emmanuel.ssebaggala@studenti.unitn.it, conci@disi.unitn.it

***Abstract*—Ubiquity of internet-connected media- and sensor-equipped portable devices is enabling a new class of applications which exploit the power of crowds to perform sensing tasks in the real world. Such paradigm is referred as crowd-sensing, and lies at the intersection of crowd-sourcing and participatory sensing. This has a wide range of potential applications such as direct involvement of citizens into public decision making. In this work we present *Matador*, a framework to embed context-awareness in the presentation and execution of crowd-sensing tasks. This allows to present the right tasks, to the right users in the right circumstances, and to preserve normal device functioning. We present the design and prototype implementation of the platform, including an energy-efficient context sampling algorithm. We validate the proposed approach through a numerical study and a small pilot, and demonstrate the ability of the proposed system to efficiently deliver crowd-sensing tasks, while minimizing the consumption of mobile device resources.**

***Keywords*-Mobile, crowd-sensing, context-aware systems, localization, energy-efficiency**

I. Introduction

The ubiquitous availability of internet-connected media- and sensor-equipped portable devices is enabling a new class of applications which exploit the power of crowds to perform sensing tasks in the real world. Such a paradigm is typically referred as *crowd-sensing* [1], and lies at the intersection of crowd-sourcing and participatory sensing [2]. Crowd-sourcing allows to issue tasks to users requiring human intervention, while participatory sensing exploits smartphone sensing capabilities to record, analyze, and discover patterns that are important in people's lives. In a complementary manner, crowd-sensing allows issuing sensing tasks to users without necessarily requiring direct human intervention, potentially exploiting the sensing capabilities of their mobile devices.

Crowd-sensing has a wide range of potential applications, including direct involvement of citizens in public decision making, such as urban planning and quality assessment campaigns of public services. In this case, it provides a mean to inquire directly from citizens (or indirectly from citizen-created information sources) about their opinions, emotional tonalities regarding certain arguments, and problems, as well as to seamlessly involve citizens in decision making. This differs from a pure bottom-up interaction, which involves citizens-to-administration communication, resulting in a participatory civic reporting system where citizens with mobile phones can eventually submit thematic multimedia reports on civic issues observed in the neighbourhood. Conversely, the top-down interaction modality enables administration-to-citizens communication, resulting in a mobile civic crowd-sensing system, in which the administration can launch surveys or, more generally, tasks to inquire citizens.

In this work, we present *Matador*, a crowd-sensing framework which combines the power of crowdsourcing with the instantaneity and situation-awareness of mobile technologies. With respect to state of the art solutions [3], [4], [5], the system allows to specify the context in which a given sensing task should be executed by the user or by the user's device. The resulting system is able to deliver the right tasks to the right people in the right circumstances. The notion of context-awareness in the delivery and execution of tasks characterizes our approach and allows to (i) maximize conditions for user participation by presenting only tasks relevant to the user, with minimal user intervention (ii) minimize the consumption of mobile device resources, specifically the battery, thus preserving normal operation. In this work, we present the Matador system, together with the framework proposed for modelling context-aware mobile crowd-sensing tasks. We introduce our algorithmic solution designed to dynamically adapt crowd-sensing to user context and preserve the battery lifetime of users' smartphone. Finally, we validate our approach with a small scale field study, showcasing the potential of the proposed solution.

The reminder of this paper is organized as follows. In Sec. II we define the reference scenario and propose a framework for modelling context-aware crowd-sensing problems. Sec. III present the algorithm that we designed in order to dynamically adapt the context sampling to users mobility so to preserve energy. In Sec. IV, we describe the system design and its experimental validation. Finally, Sec. V outlines the

The work of Andrei Tamilin has been supported by the EC within the framework of the CITI-SENSE project, EU-ENV.2012.6.5-1

main conclusions of this work and future research directions.

II. Context-aware Crowd-Sensing

Our reference scenario targets a user carrying a smartphone with the *Matador* crowd-sensing mobile application installed. The application runs in the background and periodically synchronizes its list of tasks with those available on the server. The type of actions associated with a task can vary depending on the application scenario: an action can be a request for a multimedia content (e.g., a request to take a photo, record a video or sound clip), participation in a questionnaire (e.g., answer a question, express a free-text opinion). Each task is further characterized by its *context*, which specifies when such a task should be triggered to the user. The context can be defined along multiple dimensions, such as geographical (e.g., within a circular area, along a street), temporal (e.g., in given dates, during given hours), demographics (e.g., age, gender), user activity (e.g., movement speed, no active calls), etc.

Tasks can either be implicitly performed by the smartphone after receiving user authorization (e.g., a GPS logging campaign), or explicitly by requiring user intervention (e.g., taking a photo of a given situation).

A. Problem Formulation

A *mobile crowd-sensing task* t is a a tuple $t = \langle c^t, a^t \rangle$, where c^t is the *task context* scoping its applicability, and a^t is the *action* associated to it.

For the sake of clarity, hereafter we consider the simplest structure of a task context defined by a geographical dimension, represented by a circular area, and temporal dimension, represented by a time interval:

$$c^t = \langle lat^t, lon^t, rad^t, [start^t, end^t] \rangle \tag{1}$$

where lat^t, lon^t are latitude and longitude coordinates of the circle center, rad^t is its radius, and $start^t, end^t$ are start and end timestamps.

A crowd-sensing system typically manages multiple tasks represented by a *tasks list* $\mathbf{t} = \{t_j\}$ of n tasks t_j, where $t_j = \langle c^t_j, a^t_j \rangle$, $0 \leq j \leq n$.

A *user context* is an essential element of the mobile crowdsensing system: it determines if the user (or his device) is in conditions relevant to perform some tasks. Similarly to tasks, a user context can be structured along multiple dimensions, which can be determined by sensing capacities of the user device. Hereafter, we assume the simplest structure of a user context c^u defined by a geographical dimension, represented by the current estimate of user's location, and a temporal dimension, represented by the current time:

$$c^u = \langle lat^u, lon^u, acc^u, ts^u \rangle \tag{2}$$

where lat^u, lon^u are latitude and longitude coordinates of the user location, acc^u is the accuracy, with which the location has been obtained, and ts^u is the timestamp.

The list of chronologically ordered user context instances is a *user context history* $\mathbf{c}^u = \{c^u_i\}_{i \geq 0}$.

We define the *distance* $d^{u,t}$ between a user context c^u and a task context c^t as a function:

$$d^{u,t} = f_{dist}(c^u, c^t) \tag{3}$$

which depends on the dimensions constituting the contexts.

Given a user with the context c^u, the task t is said to be *detected* by the user if distance $d^{u,t} \leq \lambda$, where λ is a threshold that depends on a specific application scenario.

A user context sampling is the process of obtaining a user context by the mobile device. This process can be controlled by the following two parameters:

- sampling accuracy σ: this parameter adjusts the required accuracy when sampling the context. The higher its value, the more precise the sampled context is. As an example, when estimating the user location, different localization sensors can be used, each one characterized by a different location accuracy (e.g., GPS vs. Network-based);
- sampling rate ν: this parameter adjusts the time between any two consecutive context samplings. The higher its value, the more frequent the context is sampled.

In practice, the sampling rate can be directly manipulated in the mobile device, but there are no practical ways of explicitly controlling the sampling accuracy, e.g., location cannot be forced to be acquired with a given accuracy. A work around for controlling the accuracy is selecting the sensing method with an a priori known error rate; the resulting accuracy values are dependent on the method used, e.g., localization using cell towers is known to be roughly in hundreds of meters to kilometers. However, the estimated accuracy in the current position is known only after localization has occurred. For simplicity of the notation, hereafter we will interchangeably use σ for sampling method and the sampling accuracy value.

A *user context sampling* is a function that, given a user context history and a list of tasks, determines the appropriate conditions for the next user context sampling:

$$\langle \sigma, \nu \rangle = f_{sampling}(\mathbf{c}^u, \mathbf{t}) \tag{4}$$

Clearly, the variation of the aforementioned parameters of the user context sampling process is related to the amount of resources consumed by the mobile device, in particular the energy consumed by sensors involved in the sampling. In order to quantitatively describe the consumption of resources in the proposed crowd-sensing system, we adopt the following energy model: each of the sampling methods σ supported by the mobile device is associated with a *resource cost* ω required for its invocation for context sampling and is determined by the following function:

$$\omega = f_{cost}(\sigma) \tag{5}$$

In the present work we consider the simplest resource cost function determining the amount of energy consumed by the mobile device.

A *total cost* of the crowd-sensing system functioning is determined by costs of the individual user context samplings and is represented as the sum of their individual costs:

$$\Omega = \sum_{0 \leq i \leq |\mathbf{c}^u|} \omega_i \tag{6}$$

Given a task list $\mathbf{t}$, the goal of the mobile crowd-sensing system is to define the function $f_{sampling}$ that maximizes the number of tasks properly detected and presented to the user, and minimizes the total cost Ω of consumed resources.

III. Energy Efficient Context Sampling

One of the key challenges in implementing the sampling function $f_{sampling}(\mathbf{c}^u, \mathbf{t})$ lies in the minimization of the battery consumption in order to preserve the normal operation of users' mobile device. In this work, we focus on the energy consumption due to the user localization, which is considered to be critical for any location-based service. In fact, when obtaining the user location, battery consumption is a trade off for the accuracy of the location estimation. This depends on the type of sensor applied for the localization [6], and the frequency at which a given sensor is sampled. We consider the two most common smartphone localization methods: GPS-based and cellular network-based. The typical GPS antenna of a modern phone gives an accurate location estimation, with an error value in the range of a few meters. At the same time, GPS is also known to be energy intensive and to lead to significant battery drains [7], [8], [6]. Conversely, the cellular network method estimates the location with an accuracy in the order of hundreds of meters (or in the worst case several kilometers), but with negligible battery consumption [9], [6].

A. The adaptive sampling algorithm

The intuitive idea behind the adaptive energy-efficient user context sampling is to dynamically adapt (i) the way the context is sampled, choosing between GPS and network localization and (ii) the time between two consecutive context samples. This is regulated by the proximity to one or more tasks. As illustrated in Fig. 1, the aim is to utilize the cellular network localization method when approaching the closest task. In this case, the energy consumption should be quite limited. When the uncertainty on the user location, due to the coarse accuracy of the network localization, overlaps with the spatial validity of the closest task, we should switch to GPS localization. In this latter case, the location sampling time should vary over time on the basis on the approaching rate of the user to the closest task.

The overlap between a mobile user and the closest task is graphically illustrated in Fig. 2, and leads to the definition

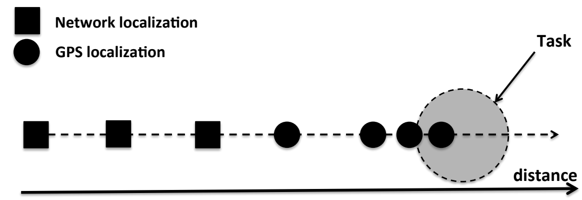


Figure 1. Adaptive user context sampling concept.

of the following distance function f_{dist}:

$$f_{dist} = \begin{cases} hvrs^*(c^u, c^t) - acc^u - rad^t & \text{if } ts^u \in [start^t, end^t] \\ \infty & \text{otherwise} \end{cases}$$

where $hvrs^*$ is the haversine formula for calculating the spatial distance between the latitude-longitude coordinates of the user and task location, acc^u is the accuracy on the user location estimation, rad^t is the radius of the closest task.

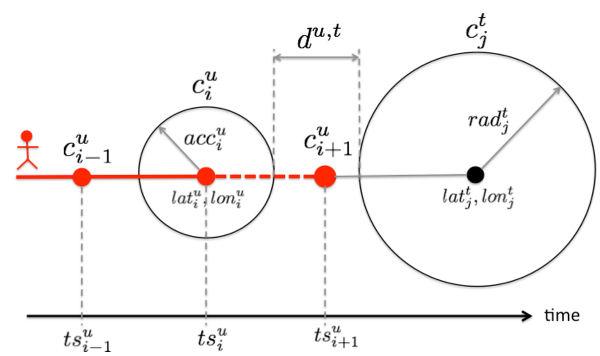


Figure 2. Adaptive user context sampling illustration.

The sampling function $f_{sampling}$ is implemented by the algorithm Alg. 1.

When the user distance $d^{u,t}$ from the closest task, computed by f_{dist}, becomes less than twice the user location accuracy the mobile application localization switches to GPS, instead of network localization. The calculation of such distance accounts for both the uncertainty on the user position, and the spatial validity of the closest task (Fig. 2). In the proposed approach, network localization is utilized as a rough probe for checking if there are tasks in range, while the precise probing of near tasks is performed using GPS. In order to generate the next sampling time (ts_{i+1}^u) the algorithm computes the average approaching speed $\bar{v}$ to the closest task based on the context history (line 9 of Alg. 1) and sets the value to the time anticipating the overlapping between user context and task context. It is worth noting that $\bar{v}$ differs from the user velocity in that it measures how fast the user is getting close to the nearest task, taking into account also his direction. As an example, if the user is moving around a certain task, although his velocity will

Algorithm 1 User context sampling function $f_{sampling}$

$\Delta t_{min} = 10s$; ▷ Minimum sampling interval
$\Delta t_{max} = 120s$; ▷ Maximum regular sampling
$\lambda = d^{u,t} = 1.5$ km; ▷ Task detection threshold

function NEXTCONTEXTSAMPLING($\mathbf{c}^u, \mathbf{t}$)
  $c_i^u \leftarrow getCurrentUserContext(\mathbf{c}^u)$;
  $c_{i-1}^u \leftarrow getPreceedingUserContext(\mathbf{c}^u)$;
Find the closest task t_j the user is heading to
  $t_j \leftarrow getClosestTask(c_i^u, \mathbf{t})$;
Approach rate to closest task t_j
  $v_i \leftarrow \frac{hvrs^*(c_i^u, c_j^t) - hvrs^*(c_{i-1}^u, c_j^t)}{ts_i - ts_{i-1}}$;
  $\bar{v} \leftarrow \frac{\sum_{j=i}^{j=i-3} v_i}{3}$ ▷ Average speed over 3 samples
  $d^{u,t} \leftarrow f_{dist}(c_i^u, c_j^t)$;
Selection of next sampling method

$$\sigma_{i+1} = \begin{cases} GPS & \text{if } d^{u,t} \leq \lambda \\ NETWORK & \text{otherwise} \end{cases}$$

$$\Delta t = \begin{cases} \frac{d^{u,t}}{\bar{v}} \bar{v} & \text{if } \Delta t_{min} < \frac{d^{u,t}}{\bar{v}} < \Delta t_{max} \\ \Delta t_{min} & \text{if } \frac{d^{u,t}}{\bar{v}} < \Delta t_{min} \\ \Delta t_{max} & \text{if } \frac{d^{u,t}}{\bar{v}} > \Delta t_{max} \end{cases}$$

Next sampling time (sec.)
  $ts_{i+1}^u \leftarrow ts_i^u + \Delta t$;
  return $\langle \sigma_{i+1}, ts_{i+1}^u \rangle$;
end function

never be null, the approach speed will be 0.
The sampling interval is parametrized in such a way so that it varies between a minimum value of 10s and a maximum regular interval of 120s. These bounds have been experimentally proved to produce optimal sensitivity to the change in user direction and speed.

Tasks with negative $d^{u,t}$ correspond to the case when the user context and the task context match. In this case, the tasks are marked as active, and will trigger the execution of the associated action.

B. Simulation Study

In order to give a quantitative characterization of the energy-efficiency of the presented adaptive sampling function, we conducted a series of programmed experiments to simulate the functioning of a mobile crowd-sensing system. The implemented simulation environment allows to (i) specify the route to be followed by the user, (ii) configure random variation of the user speed when moving along a route, (iii) place a set of tasks along the user's route for detection. Three simplifying assumptions are made: (1) a sampling method can be activated immediately without delays (in practice, depending on the environment conditions, getting a fix on GPS requires some time), (2) the accuracy value of a sampling method is constant through the simulation (we used $\sigma^{GPS} = 20$ meters and $\sigma^{NETWORK} = 1000$ meters), (3) average cost, in terms of battery consumption, for invoking a sampling is constant through the simulation. We compared the proposed context-aware sampling algorithm, with the case of a constant rate GPS sampling. In both cases, we measured the detection rate, and we compared the associated energy costs. Fig. 3 shows the results in the case of a user moving along a 30 Km route at a speed of 50 Km/h. It is possible to observe how the detection rate varies when the sampling rate is increased, and therefore the associated trade-off with the energy consumption. In particular, it is evident that the performance deteriorate rapidly for a sampling rate greater than *30* sec.. Setting the task detection rate to 80% leads to a required sampling rate of approximatively $60s$, and a total number of 36 GPS samples over a 30 km route. Under the same setting, the $Matador$ context-aware sampling algorithm provides similar performance at the cost of 12 GPS samples and 7 network samples. This leads to a significant saving in terms of energy consumption. In particular, neglecting the energy cost of Network samples [6], the proposed adaptive context sampling mechanism can lead to approximatively a 60% savings in terms of battery consumption.

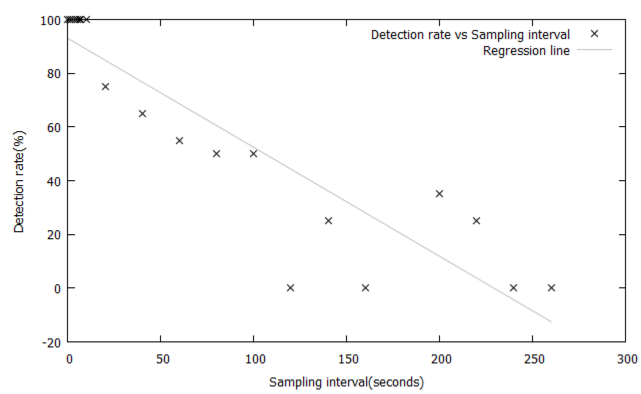


Figure 3. Task detection rate in the case of a constant GPS sampling, user moving along a path at a speed of 50 Km/h.

Fig. 4 provides a visual comparison of the constant GPS sampling with the *Matadador* context-aware sampling. When the user is far away from the closest task, only Network localization is used, with a considerable saving in terms of energy consumption. Instead, when approaching the task, GPS is used, but with a sampling rate which depends on the distance from the task.

Clearly, this simulation study represents an ideal case, but, at the same time, provides an indication on the potential savings that is possible to obtain by adapting the context sampling function to the specific crowd-sensing campaign being supported. In Sec. IV, we will provide the results of a small-scale pilot.

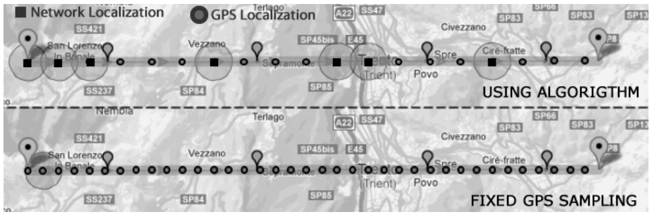


Figure 4. Visual comparison of the constant GPS sampling and the textitMatadador context-aware sampling.

IV. System Implementation and Experimentation

A. Prototype Implementation

The Matador context-aware mobile crowd-sensing system has been fully implemented into a working prototype consisting of a server-side web application and a smartphone mobile application.

The server-side part has been realized as a web application providing an interface to create tasks and monitor their execution. For each task, it is possible to configure: (i) its context, which in the current prototype implementation consists of the spatio/temporal region in which the the task should be triggered to users; (ii) the action requested to be performed by users. Currently, an action can be *implicit* or *explicit*. In the former case, no direct user intervention is required for triggering the action. In this case, the smartphone starts to collect data autonomously, and sends it back to the server for later processing. Examples of such data includes accelerometer data, anonymous GPS logging. Conversely, in the latter case, a direct human intervention is required in order to complete the task. In the current implementation, explicit tasks consists of a combination of questionnaires and multimedia reports (e.g., picture and videos). The server component receives users' responses for each task, computes response statistics and provides a dashboard for visualizing the analytics of each task. Fig. 5 depicts the user interface of the server-side component.

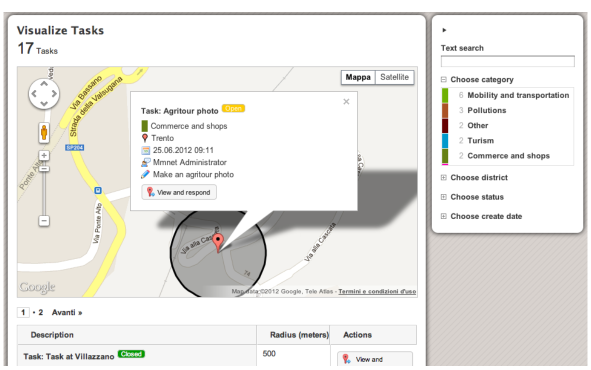


Figure 5. The *Matador* server-side user interface.

The server is accessible from the mobile through RESTful APIs, which are used to (i) periodically synchronize tasks and (ii) receive task responses from mobile users. A specific ontology has been created in order to define a common language between the server and mobile clients, and to properly represent and interpret tasks and tasks' responses. A typical *Matador* task is modelled as in the following XML sample:

```xml
<task id="taskID">
  <title>Task at Villazzano</title>
  <description>
    Take photo of the bus stop in Villazzano
  </description>
  <context>
    <space>
      <circle>
        <center>46.04552 11.13852</center>
        <radius units="m">50</radius>
      </circle>
    </space>
    <time>
      <validity format="DD.MM.YYYY">
        <from>01.07.2012</from>
        <to>31.07.2012</to>
      </validity>
    </time>
  </context>
  <action>
    <request id="requestID" type="photo">
      Please take a photo of the bus stop.
    </request>
  </action>
</task>
```

Listing 1. XML representation of a crowd-sensing task

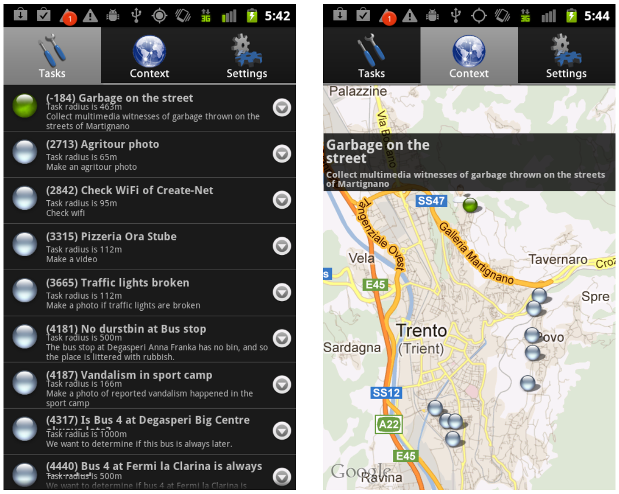
The mobile application has been implemented for Android-based smartphones and tested over a Nexus S. The application downloads tasks from the server and accordingly schedules the acquisition context following the proposed algorithm. In this initial implementation, tasks are synchronized at periodic time intervals (*120* sec.). When an active task is identified, a notification is presented to the user. The implemented multimedia capture component accesses the microphone and camera of the smartphone allowing the execution of multimedia actions assigned to tasks, e.g., taking a photo, recording a video, or recording a sound clip. Responses to tasks and acquired multimedia content are communicated to the server for elaboration.

Fig. 6 presents the user interface of the implemented mobile application.

B. Experimental Validation

In order to validate the proposed system, we have run a small field test, which consisted in a user driving along a route, and carrying a smartphone with the Matador mobile application installed. The application runs our adaptive sampling algorithm, and presents an alert to the user whenever a task is detected. In the experiment, the information related to the number of samples and detected tasks is saved for later processing.

The itinerary consisted of 400 Km of driving over a suburban road. 40 tasks were distributed along the path, with



(a) Tasks list. (b) Map of the closest tasks.

Figure 6. The *Matador* mobile application user interface.

a radius ranging from *250* to *500* meters. Distance between consecutive tasks was between 30 and 40 Km.. The speed of the vehicle varied between 25 Km/h and 130 Km/h.
Since the primary aim of the experimental validation was to evaluate the adaptive sampling mechanisms, we did not assume any specific constraint on the temporal context of tasks. Which means tasks were always active, independently from the specific time.

Tab. I shows the results of the experiment. The trip lasted for $4h$ and 20 mins, and 252 samples were taken, out of which 103 used GPS and 149 used network. There is a significant difference between the localization accuracy of Network and GPS: $2159m$ versus $5m$. Overall, a 76% detection rate was achieved. In particular, the algorithm proved to miss the detection of tasks in the case of rapid changes in the speed. This depends on the context memory used to estimate the current velocity and task approaching rate. In addition, GPS not always was able to provide the location within the predefined timeout ($20s$). Such delay, in the case of a task with a small radius, was enough to miss the detection.

Trip Duration	4h 20 mins
Total number of samples	252
Number of GPS samples	103
Number of network samples	149
Average Network Accuracy	2159 m
Average GPS Accuracy	5 m
Detection Rate	76%

Table I
EXPERIMENT RESULTS.

V. CONCLUSIONS

In this paper we presented *Matador*, a mobile context-aware crowd-sensing system which exploits user context in order to optimally deliver tasks to users, while preserving mobile device resources. We presented the system design, together with the framework we used to model and evaluate the developed algorithmic solutions. Our initial evaluation supports the proposed approach.
Current work is devoted to extending the dimensions utilized for characterizing the context, and implementing and evaluating a large-scale experimentation involving a larger user base.

REFERENCES

[1] R. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 32 –39, Nov 2011.

[2] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava, "Participatory sensing," in *Proc. of WSW*, 2006, pp. 117–134.

[3] L. S. Ravindranath, A. Thiagarajan, H. Balakrishnan, and S. Madden, "Code in the air: Simplifying sensing and coordination tasks on smartphones," in *Proc. of HotMobile*, 2012.

[4] M.-R. Ra, B. Liu, T. L. Porta, and R. Govindan, "Medusa: A Programming Framework for Crowd-Sensing Applications," in *Proc. of ACM MobiSys*, June 2012.

[5] M. Mun, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, R. West, and P. Boda, "Peir, the personal environmental impact report, as a platform for participatory sensing systems research," in *Proc. of ACM Mobisys*, 2009.

[6] K. Lin, A. Kansal, D. Lymberopoulos, and F. Zhao, "Energy-accuracy aware localization for mobile devices," in *Proc. of ACM Mobisys*, 2010.

[7] J. Paek, J. Kim, and R. Govindan, "Energy-efficient rate-adaptive gps-based positioning for smartphones," in *Proc. of ACM MobiSys*, Jun. 2010.

[8] C.-L. Wu, Y.-T. Huang, C.-L. Wu, H. hua Chu, P. Huang, and L.-J. Chen, "An adaptive duty-cycle scheme for gps scheduling in mobile location sensing applications," in *Proc. of PhoneSense*, Nov. 2011.

[9] K. Lin, A. Kansal, D. Lymberopoulos, and F. Zhao, "Energy-accuracy trade-off for continuous mobile device location," in *Proc. of ACM MobiSys*, Jun. 2010.