Crowd-Sensing: why Context Matters

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*—Crowd-sensing is becoming a popular computing and sensing paradigm for enclosing humans in the sensing loop. The underlying idea is that people, together with their mobile device, can act as mobile and pervasive sensors, gathering information about the surrounding environment and potentially providing direct input. In this work we focus on how to embed context-awareness in a crowd-sensing system in order preserve the battery of user's mobile device, while maximizing the user participation to crowd-sensing campaigns. We present the design and implementation of the Matador platform, and a preliminary evaluation obtained through a small-scale pilot study.**

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

I. Introduction

The increasing availability of internet-connected 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].

Participatory sensing is increasingly becoming a sensing paradigm to include people-in-the-loop for performing data gathering tasks. In a typical scenario, people are motivated by their personal pay-off to participate in collective data gathering, and the assurance to anonymously share the data they provide. As an example, PEIR [3] provides a way for users to measure their personal exposure to pollution, while contributing to the creation of a fine-grained map of the air quality in an urban setting.

In a similar fashion, crowd-sourcing exploits the wisdom of the crowds in order to perform specific tasks. This has been widely explored in the web, Amazon Mechanical Turk being the most prominent example.

At the intersection of Participatory sensing and Crowd-sourcing lies *mobile crowd-sensing* [1], which is a mobile sensing paradigm different from participatory sensing in that user involvement is minimal and sensing can occur autonomously. In such a setting, tasks are typically delivered to mobile users and the results aggregated and processed on a server [5], [6].

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

Crowd-sensing has a wide range of potential application scenarios, including direct involvement of citizens in public decision making, such as urban planning and quality assessment campaigns of public services. In this case, public administrators can combine a bottom-up approach, where citizens with mobile phones can eventually submit thematic multimedia reports on civic issues observed in the neighbourhood, with a top-down one, where the administration can directly inquiry citizens (Fig. 1). Additional examples of crowd-sensing application scenarios include the detection of rare events such as earthquakes [7], or the pervasive monitoring of pollution such as noise [8].

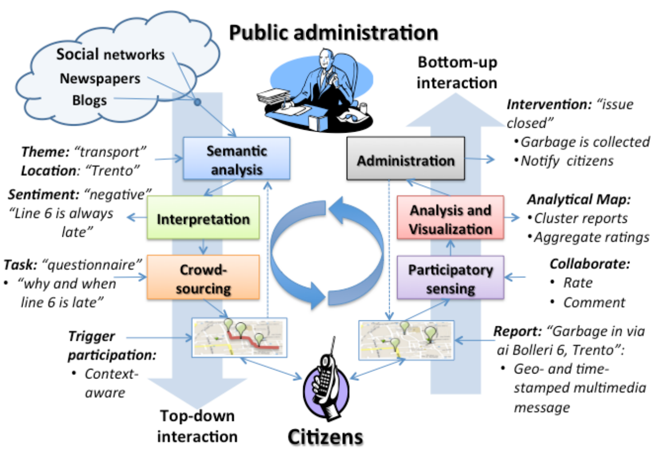


Figure 1. Crowd-sensing approach to public decision making.

In this work, we present *Matador*, a crowd-sensing framework which allows to specify the context in which a given sensing task should be executed by the user or by the user's device. With respect to existing solutions and platforms [5], [6], [3], the system allows to dynamically trade-off the resources necessary to run a given crowd-sensing campaign and optimally match them to the target user base. The resulting system is able to deliver the right tasks to the right people in the right circumstances. In this initial implementation, the proposed platform dynamically controls the context acquisition based on the active crowd-sensing campaigns, and trades off the localization accuracy with the energy consumption. We present the Matador system, together with its design and implementation, and some preliminary results,

978-1-4673-5077-8/13/$31.00 ©2013 IEEE

showcasing the potential of the proposed solution.

II. System Design and Implementation

Our reference scenario targets a user carrying a smartphone with the *Matador* crowd-sensing mobile application installed and running in the background. The application periodically synchronizes the list of tasks with those available on the backend 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, mood, habits).
Tasks can be performed either implicitly by the smartphone after receiving user authorization (e.g., a GPS logging campaign), or explicitly by also requiring user intervention (e.g., taking a photo of a given situation).

A. Design Goals and Architecture

The Matador system is composed by (i) a server component, which will be used to dynamically configure crowd-sensing campaigns, and easily process and visualize the collected data, and (ii) a mobile extension, which is a mobile application running on users device and triggering tasks whenever relevant for the user. The main design goals of the *Matador* system can be summarized as follows:

- minimal impact on normal device utilization: we expect that the use of the Matador mobile application will not affect the normal utilization of the mobile device, especially in terms of energy consumption.
- support for both explicit and implicit tasks: the Matador mobile application should to be able to (i) collect data without human intervention (e.g., accelerometer data) and (ii) collect data requiring also explicit user input (e.g., take a photo);
- ease of configuration: we expect the Matador platform to be used by public administrations. This means that there should be an easy way to create and configure crowd-sensing campaigns and to visualize results;
- ease of deployment: we expect the platform to be easily deployable, without requiring users to perform time-consuming configuration operations. Further, we expect the platform to be deployable over off-the-shelf components, without the need for dedicated hardware.

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

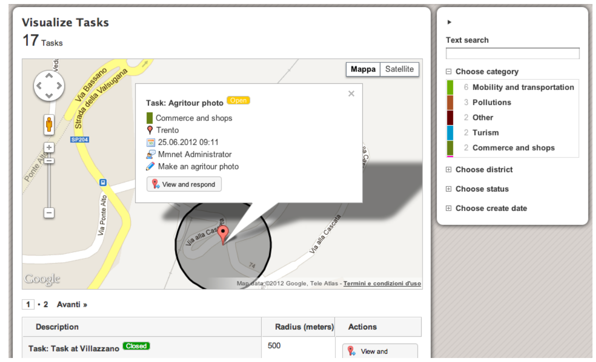


Figure 2. The *Matador* server-side user interface.

The server-side part is a web application (Fig. 2), and provides an interface to create, configure and dispatch tasks to mobile users. After their creation, tasks execution can be monitored in real-time. In this initial implementation of Matador, for each task it is possible to configure: (i) its context, which 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 user intervention is required for triggering the action, a part for having the mobile application running in the background. 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. The server component receives users' responses for each task, computes response statistics and provides a dashboard for visualizing the analytics of each task.

In addition to the user interface, a RESTful API has been implemented on the server to communicate with mobile clients to (i) synchronize tasks and (ii) receive task responses. 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.

The mobile application has been implemented for Android-based smartphones. The software architecture is depicted in Fig. 3, and is constituted by:

- the *Context Manager* (CM), which is in charge of acquiring user context, as well as of scheduling the next sampling time. In both operations, the CM trades off the energy consumption for the detection rate, taking into account the characteristics of the crowd-sensing campaign;
- the *Tasks Engine* (TE) is responsible for (i) the synchronization of tasks with the remote server, and (ii)

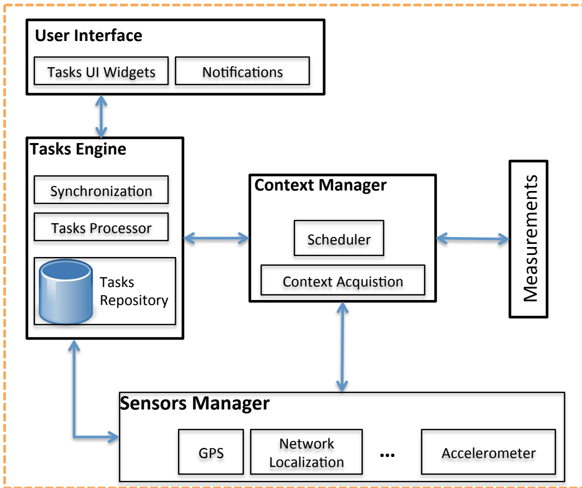
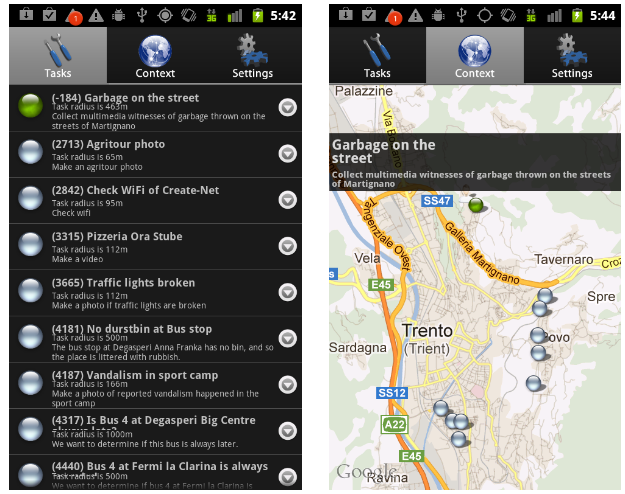


Figure 3. Matador mobile application software architecture.



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

Figure 4. The *Matador* mobile application user interface.

the proper task execution when notified by the CM. In the current implementation, tasks synchronization is performed at a regular and configurable time period. A specific protocol has been implemented in order to synchronize only the new tasks, or those that have been updated;

- the *User Interface Manager* (UIM), handles all users interaction aspects. In particular, when a task requires the user intervention, the UIM autonomously renders the appropriate GUI widget (e.g., questionnaire, request for photo, etc.) for collecting the user input;
- the *Sensors Manager* (SM) manages the access to the on-board sensors and supports specific policies to efficiently acquire sensors data. In addition, it provides the possibility of accessing and querying aggregate views of sensors data and to embed specific semantic (e.g., user running).

Fig. 4 presents the user interface of the implemented mobile application.

B. Energy Efficient Context Sampling

Acquiring contextual information over a mobile device typically involves the utilization of one or more on-board sensors. Examples include accelerometers, GPS, WiFi, network, etc.. Clearly any sensing operation consumes energy, and one of the key challenges for any crowd-sensing system relies in the minimization of the battery consumption, so to preserve the normal utilization of users' mobile device. In the current implementation of Matador, the focus has been on the optimization of the energy consumption due to the user localization. The idea is to trade off the the accuracy of the location estimation with the battery consumption by varying the type of sensor applied for the localization [9], and the frequency at which a given sensor is sampled.

We have then developed an adaptive context sampling algorithm [10], which dynamically adapts (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 present in the crowd-sensing campaign. As illustrated in Fig. 5, 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 sampling should adapt over time on the basis on the approaching rate of the user to the closest task.

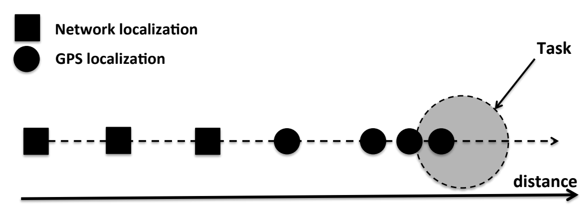


Figure 5. Adaptive user context sampling concept.

III. Preliminary Validation

In order to validate the proposed system and the adaptive sampling algorithm, we have run a small field test, which consisted in a car driving along a 20 Km route. The driver, carried a smartphone (a Nexus S) with the Matador mobile application installed. The application adapts the context sampling according to the illustrated algorithm, and saves the necessary information for later processing.

The itinerary consisted of a sub-urban road, featuring two roundabouts, a 10 Km high speed segment and various accelerations/decelerations. 10 tasks were randomly distributed along such path. Each task was characterized by a circular geographical validity, with a 150 m radius and the center along the path. The itinerary has been driven 10 times. A visual representation of an experiment and of its results is depicted in Fig. 6.

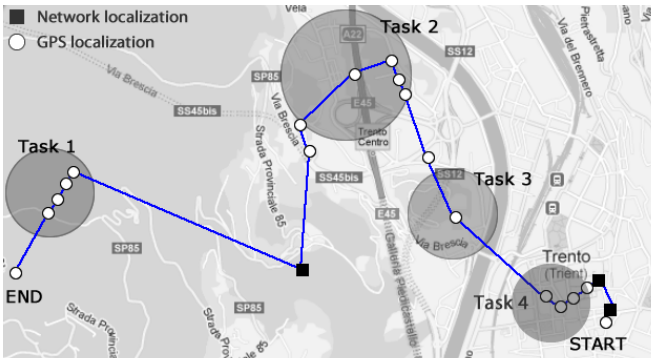


Figure 6. A visual representation of an experiment.

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. In total, the context was acquired 252 times, out of which 103 used GPS and 149 used Network for localization. As expected, there is a significant difference between the location accuracy through Network and GPS localization: 1855 m. versus 5 m.. Overall, a 65% detection rate was achieved. As intuitive, the algorithm missed the detection of tasks in the case of rapid changes in the speed, or very small tasks. In addition, GPS not always was able to provide the location within the predefined timeout ($20s$) or at the expected time. Such delay, in the case of a task with a small radius, was enough to miss the detection. The number of GPS samples is still higher than the number of Network samples because in the case of small tasks (250 m radius) and a car driving at 100 Km/h, the requested sampling granularity should be very high.

Total number of samples	250
Number of GPS samples	145
Number of network samples	55
Average Network Accuracy	1855 m.
Average GPS Accuracy	5 m.
Detection Rate	65%

Table I
EXPERIMENT RESULTS.

IV. 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 its implementation. Our preliminary evaluation supports the proposed approach and demonstrates how crowd-sensing systems can benefit from context-awareness. Current work is devoted to extending the dimensions utilized for characterizing the context, and implementing and evaluating 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.

[3] 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.

[4] I. Schweizer, R. Bartl, A. Schulz, F. Probst, and M. Muhlhauser, "Noisemap - real-time participatory noise maps," in *Proc. of Phonesense*, Nov. 2011.

[5] 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.

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

[7] M. Faulkner, M. Olson, R. Chandy, J. Krause, K. M. Chandy, and A. Krause, "The next big one: Detecting earthquakes and other rare events from community-based sensors," in *Proc. of IPSN 2011, April 12-14, 2011, Chicago, IL, USA*, Apr. 2011.

[8] R. K. Rana, C. T. Chou, S. S. Kanhere, N. Bulusu, and W. Hu, "Ear-phone: an end-to-end participatory urban noise mapping system," in *Proc. of IPSN*, Apr. 2010.

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

[10] A. Tamilin, I. Carreras, E. Ssebaggala, A. Opira, and N. Conci, "Context-aware mobile crowdsourcing," in *Proc. of UbiMi Workshop*, Pittsburgh, US, Sept. 2012.