Energy-efficient collaborative context sensing on mobile phones

Julien Eberle
LSIR, I&C, EPFL

***Abstract*—As a natural evolution of the current Location-Based services on smartphones, Context-Based services are about to emerge. But due to their limited capacities in terms of computational power and energy, these mobile devices require smart algorithms and strategies to get convincing results on context identification.**

This research proposal explores optimized enablers that were developed to support context sensing on smartphones like continuous location tracking or activity recognition through three selected papers. Finally, different sampling optimization approaches that I will investigate by leveraging collaboration between devices are presented along with my previous work on optimal mobile sensing.

***Index Terms*—mobile sensing, collaborative sensing, energy-efficiency**

I. INTRODUCTION

WITH the mass adoption of smartphones and other mobile devices, sensing capabilities became more affordable and accessible to a huge part of the population. Moreover, the connectivity of these devices has also expanded, enabling long-range as well as short-range high speed data transfer. The range of features sensed goes from location to acceleration, air pressure, temperature or to even more domain specific sensors like CO_2 or radioactivity, all conveying a certain notion of context to mobile applications and through internet connectivity to global services.

Proposal submitted to committee: July 6th, 2012; Candidacy exam date: July 13th, 2012; Candidacy exam committee: Prof. Michael C. Gastpar, Prof. Karl Aberer, Dr Olivier Dousse, Prof. Matthias Grossglauser.

This research plan has been approved:

Date: ____________________

Doctoral candidate: ____________________
(name and signature)

Thesis director: ____________________
(name and signature)

Thesis co-director: ____________________
(name and signature)

Doct. prog. director: ____________________
(R. Urbanke) (signature)

EDIC-ru/05.05.2009

Having such a companion that follows the user everywhere she goes and share most of her life, makes it ideal as a non-intrusive activity and context recognition device. This can enable the mobile device to go from the Location-based services like Facebook Places[1], Foursquare[2] or Google Latitude[3] to Context-based services like the ones provided by AegisMobility[4] or Overlay Media ltd.[5]. By the end of last year Gartner Inc. predicted a huge market of almost $100 billion for these kind of services in the near future[6]. But before this really takes off, a few challenges need to be addressed.

As for every embedded and mobile device the limitations of the platform in terms of energy available and limited computational power requires optimized sensor management and software. Minimizing the power consumption, while optimizing the accuracy of the measurements by the sensors has recently been a subject of intense research. Moreover ideas such as collaboration between the devices and fusion between the sensors have made their way up and can be seen as a way to further optimize mobile sensing and enable the context-based services era.

After introducing the background and some previous work on energy-efficient context sensing and collaborative sensing in the section II, the three selected papers are summarized in section III.

1) The first paper [1] addresses the scaling issue when generalizing activity inference models to large groups of users with very different profiles, by merging training data from multiple users through Community Similarity Network (CSN).
2) The second paper [2] presents Kobe, an implementation of a feature classification system using the cloud to compute the most effective classification parameters according to predefined configurations (connectivity, CPU idle, phone model). Several different classifiers were tested on this framework, like transportation mode, acceleration classification or face recognition. It also offers

[1] http://www.facebook.com/places/
[2] http://foursquare.com
[3] http://latitude.google.com
[4] http://www.aegismobility.com/home/context-based-services
[5] http://www.overlaymedia.com
[6] http://www.gartner.com/it/page.jsp?id=1827614

the possibility to offload part of the computation to the cloud, as long as it meet the constraints specified by the user on energy budget and latency.

3) The third paper [3] demonstrates an energy-efficient system, EnTracked$_T$, to track the trajectories of mobiles phones by combining GPS sampling and dead reckoning. The sensor management strategies have been evaluated in a simulation and a real world data collection and detailed energy consumption measurements have been made to assess the impact of each sensor in different situations like the transportation modes.

Finally the section IV describes my current research and future directions for my thesis.

II. Related work

Research on sensing on smartphones aims at several goals. On one hand researchers tried to extract as much information about the context of the device as possible. For example tuning machine learning algorithms or adding external sensors were shown to improve the accuracy of activity recognition on smartphones and wearable sensors [4]–[6]. Another important part of context is the location and tracking the phone everywhere was made possible by using dead-reckoning or radio fingerprinting for example [7]–[9]. Finally the social aspect was also addressed, linking people around to their avatar in social networks [10].

On the other hand the energy efficiency and optimization for low performance devices were the main drivers and many researchers revisited the continuous sensing problem. Reduction of the energy consumption for continuous location tracking was addressed by some strategies to down-sample the GPS like sensors substitution, piggybacking or trajectory estimation [3], [11]. Lin et al. [12] took another approach by evaluating first the accuracy needed in a location based search scenario and selecting then the most adapted sensor modality. Activity recognition framework were also optimized to use more sparsely the sensors with higher cost and adapt their algorithm to the limitations of the mobile platform [2], [13]–[15].

Only recently researchers have tried to leverage the collaboration between mobile devices to further improve the context identification and energy efficiency. Sensor data sharing between the clients allows them to reduce their own sensing needs by removing duplicates, like in Sheng et al. approach [16] or to improve the robustness of the classifiers build on multi-users training data, like in Lane et al. work[1]. In their *Darwin Phone* Miluzzo et al. [17] shared the result of the inference instead of the raw sensor data, thus reducing the amount of data transfered while leveraging collaboration to better sense the context.

III. Survey of the Selected Papers

These three papers were chosen to best cover the fields mentioned in the previous section.

A. Collaboration

This section is based on Lane et al. - *Enabling large-scale human activity inference on smartphones using community similarity networks (csn)* [1]

1) Contribution: Lane et al. first show on two different datasets that training one single inference model leads to very poor classification accuracy when applied to certain users with properties deviating from the average. The classification accuracy shows a wide-spread distribution within a large population. More generally these kind of issues arise when deploying at large scale the inference models for context identification on an heterogeneous population.

The solution they proposed, illustrated on Figure 1 is to collect the training data over a well-diversified population and build a similarity network based on some attributes of these users. Then the inference models are trained individually for each user including all the training data, weighted according to the similarity measures. Unlike previous improvement to activity classification, CSN doesn't isolate the users but takes advantage of collaborative data gathering.

2) Implementation: The system is implemented as an Android application that collects accelerometer, microphone and GPS information as training data, but also to feed into the local inference model. The application implements also a mechanism to obtain some feedback from the user: either assessing the current inference or explicitly labeling training data. The computation of the similarity networks and the models from these collected data are done on the Amazon's cloud infrastructure: Amazon Web Services. Finally the models are pushed back to each user, allowing them to locally do the classification in real-time.

3) Similarity network: From the point of view of one user, the similarity network is a list of weighted connections expressing a certain similarity measure with any other user. This paper describes three examples of such a similarity measure, namely *physical similarity*, *lifestyle similarity* and *sensor-data similarity*.

- The physical similarity is computed as the Mahalanobis distance between the vectors containing physical information such as the age, weight, height and different well-being scores.
- The lifestyle similarity is based on the mobility, diurnal patterns and the distribution of the time performing an activity. The mobility is expressed as an histogram of the GPS locations on a low granularity grid and the diurnal patterns is represented as an hourly histogram over one week containing information about non-stationarity.
- The sensor-data similarity is measured by taking a Locality Sensitive Hashing function of the feature vectors. These functions guarantee that similar input data have a high probability of yielding the same output. In this case a list of binary hashing function is used to build a histogram and the histograms are compared by taking the inner product.

4) Machine learning: The features chosen for classification were based on previous work and as they are not the main contribution of this work, the reader can refer to the original

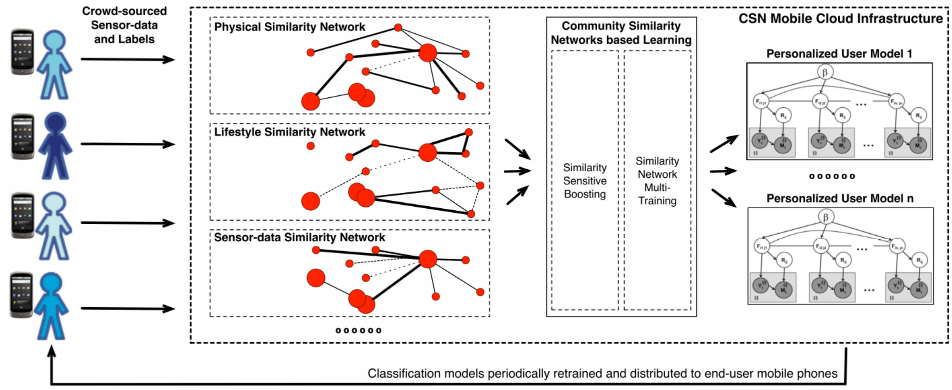

Fig. 1: Figure from [1] showing the architecture of the system using Community Similarity Networks.

paper for the complete list of features and references.

The classification is performed using a modified version of Boosting, using naive Bayes classifiers as the weak classifiers, for each user individually. To take into account the different level of similarity with the other users providing the training data, the weight in the first iteration of Boosting is equal to the similarity between the current user and the one who collected the sample. This approach was called Similarity-sensitive Boosting.

To merge the different classifiers generated with each similarity networks a multi-training is performed. At each iteration unlabeled data is given a label if a majority of the classifiers agreed on it. Then the classifiers are retrained using Similarity-sensitive Boosting. Multi-training helps propagating labels across data from different users, reducing the need of labeled data for training. The experiment shows that only 15 min of labeled data per user is needed to reach an accuracy of 74% instead of 36 min if the models were trained individually.

In addition the authors show in their evaluation that using CSN the model generated is more accurate (82%), compared to versions trained individually (65%) or without introducing the similarity in the Boosting method (52%).

5) Discussion: With this work, Lane et al. have demonstrated that collaboration between users helps achieving better results in term of accuracy of an inference model. But as a centralized approach, it may lead to scalability issues if deployed for a large population of users. Indeed all users' profiles will be stored and the similarity networks size will grow quadratically. This specific issue is discussed in the end of their paper and a clustering of similar users is proposed to reduce the size of the CSN. Moreover the profiles used for computing the similarity are often considered as private information and one could think about a way of expressing the similarity measures in a privacy preserving way.

Nevertheless, this system doesn't aim at being energy-efficient, which can be a huge obstacle in mass adoption of a mobile application. The next section presents another approach that links activity recognition to energy-efficiency.

B. Context/activity recognition

This section is based on Chu et al. - *Balancing Energy , Latency and Accuracy for Mobile Sensor Data Classification* [2]

1) Contribution: As an introduction Chu et al. highlight the shortcomings of Statistical Machine Learning when used on mobile clients. The energy consumption is not always taken into consideration while designing classification pipelines and the solutions including such considerations do it at the cost of latency or accuracy. Usually parameters are hand tuned, but this is always specific to the case studied and some external environment, such as the connectivity and the energy remaining.

Some adaptivity has also been tried by down-sampling or up-sampling the sensors according to the energy budget. But these policies don't take the whole pipeline into account and sometimes more expensive sensing have only a small marginal accuracy improvement. The option of offloading the computation to the cloud should also be carefully studied as balancing the cost of local computation with the cost of communication is not trivial.

The solution proposed by the authors is a system, namely Kobe, finding off-line a Pareto optimal accuracy/latency/energy trade-off, by searching the parameter space for all the predefined configurations and then adapting at runtime the local parameters on the mobile device.

2) Implementation: Kobe is implemented in C# on both the servers and the mobile phones; the later uses the Mobile .NET Compact Framework. The system interfaces with a SQL-like API to the different applications requiring sensor data classification. The modular design on the client side includes a sensor sampling module, a feature extraction module and a model computation module.

On the server side, several level of optimization are applied to explore various processing pipelines, expressed as ordering of the client modules and their respective parameters. The Pareto-optimal combinations are identified and from these, only the ones complying with the application constraints are kept and sent to the client.

As part of the optimization (described in the next section), the server needs to model the computing latency and the energy consumption. For this purpose an hybrid cost model is used. First an approximation is computed using regression and then only on few selected cases a complete emulation is run as this is much more time consuming. The accuracy is obtained by doing a cross-validation on the training set.

At runtime, the client chooses the most adapted pipeline and parameters, and changes it in case the environment changes. The common *ping* command tests the connectivity at each network-cell change on the phone and CPU-load is also periodically checked.

3) Optimization: The optimal configuration is computed once, at the development time. Therefore this computation can be in the order of several days if needed. The developers who want to query the Kobe system have to provide the training data. Searching the complete parameter space is done using Grid Search, in parallel on a cluster of servers, as described in the previous section.

The possible environments are a combination of the network speed (fast, medium or slow), the CPU load (heavy, moderated or light), and the execution mode of the module (on-device or in the cloud). Adding to that the parameters of the classifier and different execution profiles for each kind of mobile phone makes the complete search space.

Moreover, the classification pipeline is further optimized by using lazy evaluation and classifier substitution. When including a conjunction of classifiers the queries is evaluated in such an order that the average total cost is minimized by putting less likely used and expensive classification at the end. Binary classifiers are used as substitution when only one class has to be identified against the other classes or when watching for a class change.

Finally Kobe includes the possibility for tuning the parameters again after a period of usage by re-optimizing using current users' data as training data.

4) Evaluation: The authors propose five scenarii to evaluate their system, not only on the three optimized goals, but also with respect to the time needed for developers to port their algorithms to Kobe's API: Transportation Mode Inference, Image Recognition, Sound Classification, Accelerometer Classification and Face Recognition. In addition, they prototyped two applications using these classifiers, namely Offict Fit using accelerometer classification and sound classification for recommending better practices at work like not slouching, and Coktail Party that uses sound classification and face recognition for triggering reminders during a discussion with a colleague.

The comparison of Kobe against different baselines shows that on the above scenarii the gain in latency can be more than a factor 2, while returning more accurate results. Other strategies, like only changing the sampling frequency often suffer from underutilization as the classification pipeline is not adapted to the new input. This penalty is also visible when the configuration used was optimized for another kind of mobile phone. Optimizing the order when using several classifiers and the binary classifier substitution also yields substantial improvements in the latency, i.e. up to 66% for Offict Fit and 31% for Coktail Party.

5) Discussion: The framework proposed here considers the building of a context-based application as assembling pre-built blocks. This makes the optimization feasible using Grid Search and simplifies the work of developers. Moreover as most of the optimization work is done offline, there is no major scalability issue when introducing more classifiers or users. But this also limits the creativity and the possibilities when it comes to developing more specific applications.

For example certain tasks that need continuous sampling from the sensors, could achieve a significant energy consumption reduction with a more detailed sampling strategy, as we will see in the next section.

C. *Energy efficient system*

This section is based on Kjærgaard et al. - *Energy-efficient trajectory tracking for mobile devices* [3]

1) Contribution: Most of the down-sampling approaches for energy-saving rely on the fact that as long as the current model can predict the current value with a high confidence, there is no need to make or communicate any measurement. Depending on the goal of the application different models are used. For example if one needs to track the position of the user, the uncertainty or error is computed relatively to the last known position, but if one needs to track the trajectory the error is computed relatively to the expected path. EnTracked$_T$ which is an evolution of EnTracked, from the same authors, was built to do both position and trajectory tracking, while optimizing the energy consumption. The system can be used by other applications on-device through an API or from a server, requesting location updates.

2) Strategies: Three strategies are proposed to decide when the next GPS sampling should be done to minimize the trajectory or the position error. The system then continuously selects the most adapted strategy to the current accuracy/energy needs.

- Heading-aware: Unlike some previous works on inertial dead-reckoning, this strategy only considers the relative heading changes. Listening to the compass, it computes the accumulated orthogonal distance D_{orth} from the initial heading θ_{start} at time step t_k using the following formula:

$$D_{orth}(t_k) = \sum_{i=1}^{k} (t_i - t_{i-1}) s_{gps} sin(||\theta_{start} - \theta_i||)(1 + \sigma),$$

  where θ_i is the heading measured at time t_i, σ the average error of the compass and s_{gps} the estimated speed from the GPS. To also take into account the uncertainty about the speed a maximum time is set between two measurements: $\Delta t = (1 - u) \cdot E_{trajectory}/s_{gps}$, $E_{trajectory}$ being the error constraint. The parameter u was set experimentally to 0.01. If the user is changing the orientation of the phone in her pocket or in her hands, it will only increase the energy consumption, but not the error.
- Distance-aware: This strategy uses previous work done on EnTracked. The sensing schedule is formulated as a optimization problem, minimizing the energy consumption and it is described in a sum of recursive function, each of these function computing the energy consumption at a time step according to the previous states and the current decision. This problem was solved using dynamic programming.
- Movement-aware: The variance of the accelerometer values, combined with the speed measured by the GPS are used to determine if the user is still or moving by setting

fixed thresholds. Unfortunately this strategy is not usable while biking or driving as the accelerometer may not detect when the car (resp. the bike) starts to move. Like in the heading aware strategy, a maximum is set between two measurements, using the same formula with $u = 1$ and s_{gps} as the maximum speed of the target.

3) Trajectory simplification: To save storage space and communication bandwidth, the trajectories recorded by EnTracked$_T$ are simplified to a subset of GPS locations. A trajectory is defined as a spatio-temporal linear interpolation between those selected measurements. The error of a simplified trajectory compared to the complete one is defined as the maximum Euclidean distance between a GPS measured point on the complete trace and the interpolated one that has the same timestamp.

Two algorithms are considered for the simplification process:

- Douglas-Peucker: the algorithm considers the complete trajectory and starts with only the first and last points. Then it recursively adds the point with the maximum error on the segment, stopping when the whole trajectory is below the error threshold.
- Section-Heuristic from Lange et al.: This algorithm takes the point in the order they were captured and ignores the next point at time t as long as the current considered segment ending at $t + 1$ is below the error threshold. If a point is added, the considered segment starts at this point.

4) Evaluation: EnTracked$_T$ was implemented in Python for S60 on Nokia N97 and a simulation was also performed on data collected in different conditions, like biking, driving, walking, etc. The power profiles used in the system were taken from a previous work from the same authors, detailing the power consumption and operational delay of each sensor.

The evaluation compared the described EnTracked$_T$ system to the previous EnTracked and to a simple periodic scheduling algorithm. In addition versions without the heading-aware strategy and testing each of the trajectory simplification algorithms were considered.

The results showed a significant improvement in using the heading-aware strategy, decreasing the power consumption from 270 mW to 90 mW while walking and 290 mW to 170 mW while driving. Globally, for a given error threshold, EnTracked$_T$ consistently reports lower power consumption overhead, almost half of the others.

The Section Heuristic algorithm was consuming slightly more power than Douglas-Peucker's strategy when dealing with large amount of data, but yielding also a better accuracy.

5) Discussion: Most researches aiming at such real world deployment often underestimate the special cases where the system actually doesn't work optimally. Even if experiments have been conducted with real users, unforeseen scenarii can always happen. This usually means that before having the chance to be included in a commercial product, the developed system needs to be resilient to any kind of failure and degrade its performances gracefully in the worst case. Here for example, if the user enters a region without GPS signal coverage, like entering a building, no backup solution is proposed unlike some other similar work [11].

To provide a better resiliency and add other opportunities for optimization, we propose in the next chapter to investigate sensors fusion and devices collaboration.

IV. Research Proposal

The philosophy behind energy-efficiency in mobile sensing is pretty simple: sensing less to sense more [18]. Indeed, selecting the most useful measurements and turning off the sensors when no measurement is needed, allows the devices which have a limited sensing budget to collect information on longer periods and thus improving the overall utility of the measurements.

For our work, we are interested in sensing the context of the device and representing it in a model. The model should encapsulate the dependencies between the different modalities sensed and provide a way of judging the utility of a certain input measurement. From that, a sensing policy can be learned and optimized.

The two next sections present the work that has already been done and some future directions and concrete problem I will address in my research.

A. Previous work

In my previous work on finding an optimal mobile sensing strategy, namely OptiMoS [19], we explored a two-tier framework. The goal was to reduce the sampling needed as input from the sensors, while keeping the error of the model low. On the first tier, raw data from the sensors is segmented and the second tier takes care of the down-sampling. Both these operations are achieved near optimally using some heuristics. The models used were a support vectors regression and a simple linear regression, but the principles can be applied to almost any other interpolation model.

With the mass-adoption of smartphones and their increasing capabilities to sense their context, the opportunities for collaboration between the devices are now in a sufficient number to be worth taking into account for a real world application in our optimization problem.

By collaborating, mobile devices may reduce their own expenses. Indeed, as the information is shared between the different sensors (i.e. sensors fusion) or different devices, each device will need to make fewer measurements. This collaboration can be seen at different levels:

- centralized: Using its own global model, the centralized coordinator can select what each agent should sense to centrally optimize the model's accuracy. Similar methods as the one studied in OptiMoS can be used to obtain optimal or near optimal results. The main drawback of this approach is the cost of long-range communication and the need to track continuously the mobile devices.
- distributed: Using short-range communication, each device may get the measurements from its neighbors instead of sensing. The physical proximity between the devices can support our assumption that their measurements would have been similar, considering sensing a band-limited field. The optimization is done considering that

short-range communication is much less expensive than sensing.

In my last, not yet published, project I studied opportunistic strategies for collaborating mobile sensing nodes. Each node was optimizing locally the accuracy of its own model with a limited sensing budget. The proposed approach was to broadcast the measurements and some other information like the remaining budget to allow the neighboring nodes to share effectively and avoid duplicate sensing. Then the policy was defining the sensing behavior according to the neighboring nodes, their needs and the own need of the node. For this work I was using a very simple model, giving the same utility to all measurement. A real-world dataset containing GPS and Bluetooth measurement was used as input for simulating and evaluating the strategies.

It turned out that the problem was almost a next encounter prediction problem. Therefore very much linked to the mobility of the nodes. The choice of a particular mobility model or dataset could then change completely the optimal strategy.

B. Future work

For the next steps I am proposing here, I will continue investigating solutions for an optimal sensing strategy for mobile devices, thus minimizing the sampling while guaranteeing a low modeling error.

Formally this optimization can be expressed as follow:

$$argmin_{\pi}\left(\sum_{n\in\{n_0,n_N\}} error\left(\mathcal{M}^n(M_\pi)\right)\right)$$

$$\text{u. c. }\left(\sum_{m\in M_\pi} c(m)\leq B_n\right)\forall n\in\{n_1,n_N\},$$

where we optimize over the sensing policy π by minimizing the error (e.g. residual sum of square) of the model $\mathcal{M}^n$ at the node n knowing the set of measurements M_π taken with the policy π. In the constraints $c(m)$ represents the cost of the measurement m and B_n the budget of the node n, for all the N nodes.

First, I will exploit further the collaborative approach, formalizing the next encounter prediction problem and trying to prove the optimality of the strategies developed in the project presented above. I will use as reference for comparison the offline optimal strategy and perform a competitive analysis. Indeed, as an online algorithm, it may be possible to build an input while the algorithm is running and generate a sequence on which it is not optimal. In such cases, randomization could help guaranteeing a bounded competitiveness in average.

This first approach will suppose that the encounters are only depending on the few last time steps (Markov decision process of order n), but it is clear that in the real life they may follow some more regular patterns, but on the other hand restricting the evaluation of the strategy to a single dataset may prevent us to generalize the results. Therefore, a more synthetic way of representing the behavior of the node will be used. For this purpose, I will look into mobility models like Random Way-point, random walk on a grid or the more recent Heterogeneous Random Walk from Piorkowski et al. [20].

Finally, my last approach will be to apply reinforcement learning to learn the policy dynamically while the devices are used in their everyday environment. I will revise the simple base policy by including a more complete notion of utility that can be computed from the model itself. A good example of such a model that I will look into is presented in J. Kho et al. [21], where the variance in the prediction of a Gaussian Process regression is used as input to compute the mean Fisher information, expressing the "utility" of a measurement. This can be used to compute the reward function and make a system that will adapt itself to the behavior of its user.

REFERENCES

[1] N. D. Lane, Y. Xu, H. Lu, S. Hu, T. Choudhury, A. T. Campbell, and F. Zhao, "Enabling large-scale human activity inference on smartphones using community similarity networks (csn)," in *Proceedings of the 13th international conference on Ubiquitous computing - UbiComp '11*. New York, New York, USA: ACM Press, Sep. 2011, p. 355.

[2] D. Chu and N. D. Lane, "Balancing Energy , Latency and Accuracy for Mobile Sensor Data Classification," in *ACM Sensys11*, ser. SenSys '11, vol. 50, no. 6. ACM, 2011, pp. 54–67.

[3] M. B. Kjæ rgaard, S. Bhattacharya, H. Blunck, and P. Nurmi, "Energy-efficient trajectory tracking for mobile devices," in *MobiSys '11*, Jun. 2011, pp. 307–320.

[4] M. Keally, G. Xing, and A. Pyles, "PBN : Towards Practical Activity Recognition Using Smartphone-Based Body Sensor Networks," in *SenSys*, 2011, pp. 246–259.

[5] J. Lester, T. Choudhury, N. Kern, G. Borriello, and B. Hannaford, "A Hybrid Discriminative/GenerativeApproach for Modeling Human Activities ," pp. 766–772, Jul. 2005.

[6] M. Stikic, D. Larlus, S. Ebert, and B. Schiele, "Weakly Supervised Recognition of Daily Life Activities with Wearable Sensors." *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 12, Feb. 2011.

[7] O. Dousse, J. Eberle, and M. Mertens, "Place learning via direct WiFi fingerprint clustering," in *MDM '12 Proceedings of the 2012 IEEE 13th International Conference on Mobile Data Management - Industry track*, 2012.

[8] J. Seitz, T. Vaupel, J. Jahn, S. Meyer, J. Boronat, and J. Thielecke, "A Hidden Markov Model for urban navigation based on fingerprinting and pedestrian dead reckoning," *Information Fusion (FUSION), 2010 13th Conference on*, pp. 1–8, 2010.

[9] M. Kourogi, T. Ishikawa, and T. Kurata, "A method of pedestrian dead reckoning using action recognition," in *IEEE/ION Position, Location and Navigation Symposium*. IEEE, May 2010, pp. 85–89.

[10] A. Gupta, M. Miettinen, M. Nagy, N. Asokan, and A. Wetzel, "PeerSense: Who is near you?" in *2012 IEEE International Conference on Pervasive Computing and Communications Workshops*. IEEE, Mar. 2012, pp. 516–518.

[11] Z. Zhuang, K.-H. Kim, and J. P. Singh, "Improving energy efficiency of location sensing on smartphones," *Proceedings of the 8th international conference on Mobile systems applications and services MobiSys 10*, p. 315, 2010.

[12] K. Lin, "Energy-Accuracy Trade-off for Continuous Mobile Device Location," in *Mobisys'10*. ACM Press, 2010, pp. 285–298.

[13] K. K. Rachuri, C. Mascolo, and P. J. Rentfrow, "SociableSense : Exploring the Trade-offs of Adaptive Sampling and Computation Offloading for Social Sensing Categories and Subject Descriptors," in *MobiCom '11*, ser. MobiCom '11. ACM, 2011, pp. 73–84.

[14] G. Raffa, J. Lee, L. Nachman, and J. Song, "Don't slow me down: Bringing energy efficiency to continuous gesture recognition," in *Wearable Computers ISWC 2010 International Symposium on*. IEEE, 2010, pp. 1–8.

[15] H. Lu, J. Yang, Z. Liu, N. D. Lane, T. Choudhury, and A. T. Campbell, "The Jigsaw continuous sensing engine for mobile phone applications," pp. 71–84, Nov. 2010.

[16] X. Sheng and J. Tang, "Energy-Efficient Collaborative Sensing with Mobile Phones," in *INFOCOM*, 2012, pp. 1916–1924.

[17] E. Miluzzo, C. T. Cornelius, A. Ramaswamy, T. Choudhury, Z. Liu, and A. T. Campbell, "Darwin phones," in *Proceedings of the 8th international conference on Mobile systems, applications, and services - MobiSys '10*. New York, New York, USA: ACM Press, Jun. 2010, p. 5.

[18] F. Ben Abdesslem, A. Phillips, and T. Henderson, "Less is more," in *Proceedings of the 1st ACM workshop on Networking, systems, and applications for mobile handhelds - MobiHeld '09*. New York, New York, USA: ACM Press, Aug. 2009, p. 61.

[19] Z. Yan, J. Eberle, and K. Aberer, "OptiMoS: Optimal Sensing for Mobile Sensors," in *International Conference on Mobile Data Management (MDM)*, 2012.

[20] M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser, "A parsimonious model of mobile partitioned networks with clustering," in *2009 First International Communication Systems and Networks and Workshops*. IEEE, Jan. 2009, pp. 1–10.

[21] J. Kho, A. Rogers, and N. R. Jennings, "Decentralized control of adaptive sampling in wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 5, no. 3, pp. 1–35, May 2009.