Rao-Blackwellized Particle Filters for Recognizing Activities and Spatial Context from Wearable Sensors

Alvin Raj[1], Amarnag Subramanya[2], Dieter Fox[1], and Jeff Bilmes[2]

[1]University of Washington, Dept. of Computer Science & Engineering, Seattle, WA
[2]University of Washington, Dept. of Electrical Engineering, Seattle, WA

1 Introduction

Recent advances in wearable sensing and computing devices and in fast probabilistic inference techniques make possible the fine-grained estimation of a person's activities over extended periods of time [6]. Such technologies enable applications ranging from context aware computing to support for cognitively impaired people to monitoring of activities of daily living.

The focus of our work is on providing accurate information about a person's activities and environmental context in everyday environments based on wearable sensors and GPS devices. More specifically, we wish to estimate a person's motion type (such as walking, running, going upstairs/downstairs, or driving a vehicle) and whether a person is outdoors, inside a building, or in a vehicle. These activity estimates are combined with GPS information so as to estimate the trajectory of the person along with information about which buildings the person enters. To do this, our approach assumes that the bounding boxes of buildings are known (extracted from satellite images).

Another emphasis of our work is on performing activity recognition based on a minimum number of sensor devices. There are in fact a variety of systems that utilize multiple sensors and measurements taken all over the body [5, 9]. Our approach, by contrast, attempts to produce as accurate as possible activity recognition requiring only one sensing device mounted only at one location on the body. Our reasoning for reducing the total number of sensors is threefold: 1) it can be unwieldy for the person wearing the sensors to have many such sensors and battery packs mounted all over the body, 2) we wish to minimize overall system cost, and 3) we wish to extend operational time between battery replacement/recharge.

In this paper, we show how Rao-Blackwellized particle filters can be applied to efficiently estimate joint posteriors over a person's activity and spatial context. Extensive experiments demonstrate that, by performing such joint inference, our system is able to generate more consistent estimates for a per-

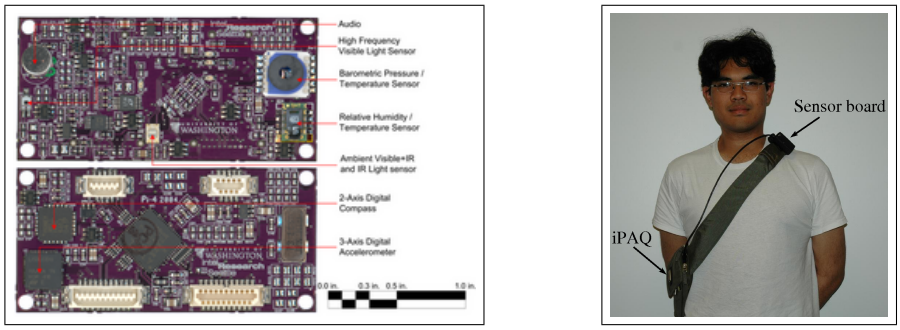

Fig. 1. (left) Sensor board and (right) complete sensing package worn by a user.

son's motion trajectory and activities. Our approach consistently outperforms a model that estimates a person's activities and locations independently.

This paper is organized as follows. In Section 2, we give a brief overview of our sensor system. Section 3 describes our activity model including all modeling assumptions, inference, and learning algorithms. We discuss related work in Section 4. Experiments are described in Section 5, followed by a discussion and conclusions.

2 Wearable Sensor System

Our customized wearable sensor system consists of a multi-sensor board, a Holux GPS unit with SIRF-III chipset, and an iPAQ PDA for data storage. The multi-sensor board shown in Fig. 1 is extremely compact, low-cost, and uses standard electronic components [6]. It weighs only 121g (about a quarter pound) *including battery and processing hardware*. Sensors include a 3-axis accelerometer, microphones for recording speech and ambient sound, photo-transistors for measuring light conditions, and temperature and barometric pressure sensors. The overall cost per sensor board is approximately USD 400. The time-stamped data collected on this device is transfered via a USB connection to an iPAQ handheld computer. GPS data is transfered from the receiver via Bluetooth to the PDA. The overall system is able to operate for more than 8 hours.

3 Activity Model

3.1 Overview

The complete dynamic Bayesian network for our activity model is shown in Fig. 2, representing the probabilistic relationships between GPS measurements (g_k, h_k), sensor-board measurements m_k, the person's location l_k, her motion velocity v_k, the type of motion s_k she is performing, and the environment e_k she is in. We now describe the individual components starting at the sensor level of the model.

GPS measurements are separated into longitude / latitude information, g_k, and *horizontal dillusion of precision* (hdop), h_k. hdop provides information

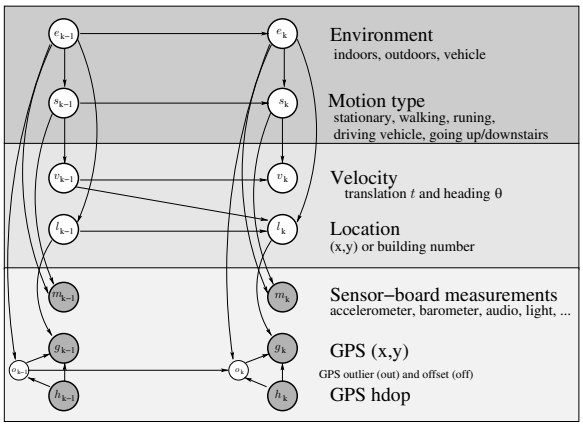

Fig. 2. Dynamic Bayesian network for joint inference.

about the accuracy of the location information, which depends mostly on the visibility and position of satellites. The node $o_k = (\text{out}_k, \text{off}_k)$ explicitly models GPS outliers and systematic GPS offset. Outliers typically occur when the person is inside a building or under trees. Unfortunately, outliers are not always indicated by a high hdop value. GPS offset is due to systematic bias in the estimates provided by the GPS unit. In our experience, this bias can be up to 10m, depending on the locations of satellites and atmospheric changes.

A GPS measurement g_k depends on the person's location, l_k, the hdop value, h_k, and the GPS outlier and offset values, o_k. The likelihood is given by

$$p(g_k \mid l_k, h_k, o_k) = \begin{array}{ll} \mathcal{N}(g_k; l_k - \text{off}_k, \sigma^2_{h_k}) & \text{if } \text{out}_k = 0 \\ \varepsilon\mathcal{N}(g_k; g_k, \sigma^2_{h_k}) & \text{if } \text{out}_k = 1 \end{array} \quad (1)$$

That is, if the measurement is not an outlier, then the likelihood is given by a Gaussian centered at the person's location l_k (shifted by an offset off_k). The variance of the Gaussian is a linear function of the hdop value h_k. In order to keep a consistent likelihood ratio between outliers and non-outliers, we set the outlier likelihood relative to the Gaussian likelihood of non-outlier measurements (using $\varepsilon = 0.8$ in our experiments). The probability of a measurement being an outlier depends on the previous outlier state, the hdop value, and the environmental state (we set $\text{out}_k = 1$ if the person is inside a building or has exited a building very recently). The offset value off_k is modeled as a process with small Gaussian drift.

Sensor-board measurements m_k consist of 3D acceleration, barometric pressure, temperature, visible and IR light intensity, and raw audio. We use the boosted classifiers introduced by [6] to extract probability estimates for the person's instantaneous environment and motion state. These classifier outputs specify the observation m_k, which depends on the current environment and motion state (see [10] for more details).

Location $l_k = (x_k, y_k)^T$ of the person is estimated in longitude / latitude coordinates if the person is outside or driving a vehicle. If the person is inside

a building, we only estimate *which* building the person is in, not the actual location inside the building. The location at time k depends on the person's previous location l_{k-1} and motion v_{k-1}, and on the current environment. If $e_k = \texttt{inside}$, then $p(l_k)$ is non-zero only if l_k is inside the bounding box of a building. We extract the bounding boxes of buildings from satellite images.

Velocity represents the motion between locations at consecutive points in time. We adopt a piecewise linear motion model on polar velocity coordinates $v_k = (t_k, \theta_k)^T$, namely translational speed, t_k, and heading direction, θ_k. We assume that the heading at time k only depends on the previous heading and translation speed, where the size in rotation (heading change) depends on the speed. We model this relationship via a speed dependent Gaussian variance $\sigma^2_{t_{k-1}}$ with

$$p(\theta_k|\theta_{k-1}, t_{k-1}) \sim \mathcal{N}(\theta_k; \theta_{k-1}, \sigma^2_{t_{k-1}}). \tag{2}$$

The translational speed t_k depends on the previous speed t_{k-1} and the current motion state s_k using the following product model:

$$p(t_k \mid t_{k-1}, s_k) \propto \mathcal{N}(t_k; t_{k-1}, \sigma_a^2) \sum_{i=1}^{I} \alpha_{s_k[i]} \mathcal{N}(t_k; \mu_{s_k[i]}, \sigma^2_{s_k[i]}). \tag{3}$$

The first factor is a Gaussian centered at the previous speed, where σ_a^2 represents acceleration. The dependency on the motion state, s_k, is implemented by the second factor, a mixture of I Gaussians, where $\alpha_{s_k[i]}$ represents the weight of the i-th mixture component, given state s_k (similar to [7]). For instance, if the motion state is `walking`, then most weight is on the component with a mean at typical walking speed. In the `driving` mode, the mixture components are more spread out, with significant weight on higher velocities.

Motion states represent different types of motion a person can be involved in. In our current system, these states include $S = \{$`stationary`, `walking`, `running`, `going up/down stairs`, `driving vehicle`$\}$. The motion state s_k depends on the previous motion state s_{k-1} and the current environment e_k.

Environment captures the person's spatial context, which is $E = \{$`indoors`, `outdoors`,`vehicle`$\}$. The edge between e_k and s_k allows the system to model both soft and hard constraints between the motion state and the environment. For example, whenever the environment is in the `indoors` or `outdoors` state, we *a priori* preclude `driving` from being a possible value of the motion type (*i.e.*, it has zero probability). Moreover, other "soft constraints" are imposed by the fact that the two nodes are related probabilistically, and the probabilities are learned automatically (see Section 3.3).

3.2 Inference

During inference, our system estimates a joint posterior distribution over the complete state space. Unfortunately, exact inference is not tractable in our model due to its combination of discrete and continuous hidden states. In [10]

we show how to perform efficient inference using a discretization of the state space along with an adaptive pruning strategy. Here, we describe how Rao-Blackwellized particle filters (RBPF) can be applied for efficient inference in such a model. We omit a comprehensive derivation of our algorithm, its correctness can be shown similar to the derivations given in [3, 2, 7].

Just like regular particle filters, RBPFs represent posteriors over a state space by temporal sets of weighted samples: $S_k = \{s_k^{(i)}, w_k^{(i)} \mid 1 \leq i \leq N\}$. A particle filter updates such sample sets according to a sampling procedure often referred to as sequential importance sampling with re-sampling (SISR, see also [11]). RBPFs derive their efficiency from a factorization of the state space, where posteriors over one part of the state space are represented by samples, and posteriors over the remaining parts are estimated exactly, conditioned on each sample. We rely on the following factorization:

$$\begin{aligned} &p(e_k, s_k, l_{1:k}, v_{1:k}, o_{1:k} \mid m_{1:k}, g_{1:k}) \\ &= p(e_k, s_k \mid l_{1:k}, v_{1:k}, o_{1:k}, m_{1:k}, g_{1:k})\, p(l_{1:k}, v_{1:k}, o_{1:k} \mid m_{1:k}, g_{1:k}). \end{aligned} \tag{4}$$

Our RBPF algorithm samples the variables in the second factor of (4), and computes exact posteriors over the variables in the first factor. Accordingly, each particle $s_k^{(i)}$ has the form

$$s_k^{(i)} = \left\langle p_k^{(i)}(e_k, s_k), l_{1:k}^{(i)}, v_{1:k}^{(i)}, o_{1:k}^{(i)} \right\rangle,$$

where $l_{1:k}^{(i)}, v_{1:k}^{(i)}, o_{1:k}^{(i)}$ are sampled values, and $p_k^{(i)}(e_k, s_k)$ is a distribution over the current environment and motion state corresponding to these values.

Table 1 summarizes our RBPF algorithm for iteratively updating sample sets over time. The algorithm accepts as input a sample set S_{k-1} along with the most recent sensor board measurement m_k, the most recent GPS measurement g_k and the most recent hdop h_k. Each iteration of the loop starting in Step 2 generates a new particle. In Step 3, the distribution over e_k and s_k is predicted based on the particle's previous distribution over these variables. This prediction is performed by marginalization over the previous time step:

$$\widehat{p}_k^{(i)}(e_k, s_k) = \sum_{e_{k-1}, s_{k-1}} p(s_k \mid e_k, s_{k-1})\; p(e_k \mid e_{k-1})\; p(e_{k-1}, s_{k-1}) \tag{5}$$

In Step 4, the algorithm generates a sample from this predictive distribution, which is used in Steps 6–8 to sample the particle's motion and GPS outlier values. Step 5 updates the location based on the previous location and motion, and the current environment. The function f distinguishes between locations inside and outside buildings. If $\tilde{e}_k^{(i)} = \texttt{indoors}$ and $l_{k-1}^{(i)}$ was in a building bounding box, then $f(l_{k-1}^{(i)}, v_{k-1}^{(i)}, \tilde{e}_k^{(i)}) = l_{k-1}^{(i)}$, otherwise $l_k^{(i)}$ is computed by shifting $l_{k-1}^{(i)}$ according to the motion $v_{k-1}^{(i)}$. f additionally models a motion away from the building if the previous location was inside a bounding box.

Once $l_k^{(i)}, v_k^{(i)}$, and $o_k^{(i)}$ are sampled, the particle's distribution over environment and motion state is updated in Step 9 using the following equation:

Inputs:

Previous sample set: $S_{k-1} = \{s_{k-1}^{(i)}, w_{k-1}^{(i)} \mid 1 \leq i \leq N\}$

Observations: g_k, m_k, h_k

1. $S_k = \emptyset$ *// Initialize*
2. **for** $i = 1, \ldots, N$ **do** *// Generate samples*

// Predictive distribution over environment and motion state

3. Compute $\widehat{p}_k^{(i)}(e_k, s_k)$ using (5) with prior $p_{k-1}^{(i)}(e_{k-1}, s_{k-1})$
4. Sample $(\tilde{e}_k^{(i)}, \tilde{s}_k^{(i)}) \sim \widehat{p}_k^{(i)}(e_k, s_k)$

// Update location using previous location, motion, and env.

5. $l_k^{(i)} = f(l_{k-1}^{(i)}, v_{k-1}^{(i)}, \tilde{e}_k^{(i)})$

// Sample motion and GPS outlier conditioned on $(\tilde{e}_k^{(i)}, \tilde{s}_k^{(i)})$

6. Sample $\theta_k^{(i)} \sim p(\theta_k^{(i)} \mid \theta_{k-1}^{(i)}, t_k^{(i)})$ *// Heading, see (2)*
7. Sample $t_k^{(i)} \sim p(t_k^{(i)} \mid t_{k-1}^{(i)}, \tilde{s}_k^{(i)})$ *// Translation velocity, see (3)*
8. Sample $o_k^{(i)} \sim p(o_k^{(i)} \mid o_{k-1}^{(i)}, h_k, \tilde{e}_k^{(i)})$ *// Outlier*

// Posterior distribution over environment and motion state

9. Compute $p(e_k^{(i)}, s_k^{(i)})$ using (6) based on $l_k^{(i)}, v_k^{(i)}, o_k^{(i)}$ and $\widehat{p}(e_k^{(i)}, s_k^{(i)})$.

// Update particle weight

10. Calculate $w_k^{(i)}$ using normalization factor of Step 9 and GPS likelihood (1).
11. **endfor**
12. Multiply / discard samples in S_k based on normalized weights w_k
13. **return** S_k

Table 1. RBPF for joint inference over environment, motion state, and location.

$$p_k^{(i)}(e_k, s_k) \propto \widehat{p}_k^{(i)}(e_k, s_k)\; p(l_k^{(i)}|e_k)\; p(o_k^{(i)}|e_k, h_k)\; p(v_k^{(i)}|s_k)\; p(m_k|e_k, s_k) \quad (6)$$

Since the sampling steps 5–8 have not considered the most recent observations, each particle still needs to receive an importance weight, which is given by the normalization factor computed in (6), times the likelihood of the GPS measurement defined in (1). Finally, in Step 12, the particles are re-sampled.

3.3 Parameter Learning

The parameters of our model are learned using labeled training data. To learn the mapping of raw sensor board measurements to binary classifiers, we use a technique introduced by Lester and colleagues [6]. This approach extracts approximately 650 features from short temporal windows of sensor data and then uses boosting to learn sequences of decision stumps that are combined to form binary classifiers [10]. The observation model for these classifiers is then trained along with the parameters related to e_k and s_k using standard maximum likelihood training based on the labeled data. The translational velocity model (3) is learned using EM to get a mixture of Gaussians for each motion state. The only parameters set manually are those related to GPS noise and outlier detections.

4 Related Work

Recently, estimating activities from wearable sensors has received significant attention especially in the ubiquitous computing and artificial intelligence communities. Bao and Intille [1] use multiple accelerometers placed on a person's body to estimate activities such as standing, walking, or running. Kern and colleagues [5] and Lukowicz *et al.* [9] added a microphone to a similar set of accelerometers in order to extract additional context information. These techniques rely on Gaussian observation models and dimensionality reduction techniques such as PCA and LDA to generate observation models from the low-level sensor data or features extracted thereof. These approaches feed the sensor data or features into static classifiers [1, 4], a bank of temporally independent HMMs [6], or multi-state HMMs [5] in order to perform temporal smoothing. None of these approaches estimates a user's spatial context.

To learn low-level sensor-board classifiers we rely on the approach introduced by Lester *et al.* [6], who showed how to apply boosting in the context of sensor-based activity recognition. In contrast to the discrete inference system used in [10], the RBPF algorithm described in this paper produces more accurate location traces and provides more flexibility in handling GPS outliers. In [10], we also showed how virtual evidence can be used to learn activity models from sparsely labeled data.

Using location for activity recognition has been the focus of other work. For instance, Liao and colleagues [7] showed how to learn a person's outdoor transportation routines from GPS data. More recently, the same authors presented a technique for jointly determining a person's activities and her significant places [8]. However, these approaches are very limited in their accuracy due to the fact that they only rely on location information.

5 Experiments

Our system was evaluated by an outside team as part of the DARPA ASSIST program. Our goal in this program is to develop techniques that can automatically generate reports that summarize and visualize relevant information collected by a soldier during a mission. The current focus of our research is on providing an accurate trace of where the person went, which buildings she entered, and how she moved between places.

The accuracy of our inference system was tested on a set of sequences collected via the ASSIST program (see Fig. 3). The environmental states were indoors, outdoors, and vehicle, and the activities were stopped, walk, run, drive, and going up/downstairs. In each test run, a soldier and one of our team members followed an exactly specified activity sequence by moving between marked waypoints. The resulting 28 traces provided about 2 hours of fully labeled training and test data. In order to test the accuracy of the system, we divided the data into 4 sets, each containing 7 randomly selected traces (sampling without replacement). We then performed four runs, during which, each of the 4 sets was used for testing, while the remaining 3 sets were used for training. Our RBPF algorithm used 2,000 particles for inference,


Fig. 3. Experimental setup: (left) Part of the evaluation area with waypoints. The subjects followed fully scripted traversals through the area. (right) A soldier and one of our team members wore a sensor system. Ground truth annotations were provided by four additional observers equipped with stop watches and audio recorders.

State	stopped	walk	run	up	down	drive
No GPS	71.6	80.2	80.8	58.9	60.2	80.0
RBPF	65.3	79.1	74.8	36.3	36.3	93.6

Env.	outdoor	indoor	vehicle
No GPS	94.1	87.1	88.0
RBPF	94.4	85.9	93.6

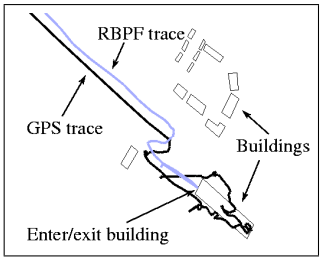

Fig. 4. Accuracy (number of correct frames/total number of frames): (left) Percentage accuracies in detecting motion state and environment. Results are given for an HMM that ignores GPS and for our RBPF. (right) Raw GPS trace (gray) and trace estimated by our RBPF (black). The RBPF trace is aligned such that it enters and exits the building at the correct time and location. Overall, the spatial consistency of RBPF traces is 92.8% vs. 33.8% for raw GPS.

which was performed in real-time on an Intel 3.2 GHz desktop PC with about 1 GB of RAM. To extract an activity and location sequence from our RBPF, we used the history of the most likely particle at the end of each test run.

The table in Fig. 4 compares the accuracy of our system to the accuracy of the Viterbi sequences extracted by a hidden Markov model that ignores information provided by the GPS sensor. As can be seen, the performance of the RBPF is not better than that of the hidden Markov Model. This however does not indicate that, GPS information is not useful for improving activity recognition performance (see [10]). The above trend may be due to the way parameters are learnt in the two approaches. Whilst, in the case of a hidden Markov model, all parameters are jointly trained, the same is not true in the case of the RBPF.

To assess the impact of our joint inference on the accuracy of location traces, we proceeded as follows. Whenever the person was inside a building, we determined how often the raw GPS trace and the trace estimated by our RBPF was inside the bounding box of that building. Averaged over all test traces, our RBPF algorithm improved this accuracy from 8.7% for raw GPS to 85.9%. One example trace is shown in the right panel in Fig. 4. As can be seen, our RBPF is able to correctly align the location trace using information about the buildings.

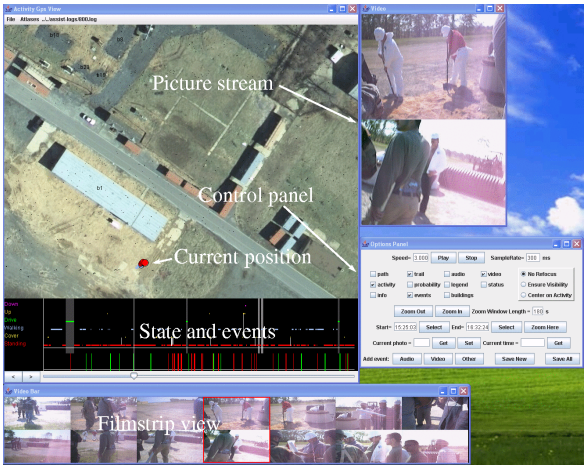

Fig. 5. User interface: The person's path is overlayed on a satellite image. A stream of pictures taken every second can be displayed along with audio recording and information about the person's activity and environmental context. Automatically detected faces and audio events are used to mark interesting events.

Fig. 5 shows the user interface of our system. The interface provides movie player style replay capabilities, including recorded pictures and audio, estimated activity states, location trace, and events extracted from the data.

6 Discussion

We presented an approach for estimating a person's low-level activities and spatial context using data collected by a small wearable sensor device. Our approach uses a dynamic Bayesian network to model the dependencies between the different parts of the system. It performs efficient inference over the joint state space using Rao-Blackwellized particle filters. Our system was evaluated as part of a DARPA project demonstration. The results show that our system achieves significant improvements in generating spatially consistent activity and location traces.

While these results are extremely encouraging, they only present the first step toward fully recognizing a person's context. Our next goal is the development of systems that can automatically generate high-level summaries of long-term activities such as vacation trip diaries, activity summaries for family members of elderly people, or after action reporting of soldier missions. To achieve this goal, we are investigating hierarchical reasoning techniques and integration of additional information provided by cameras and speech recognition. Finally, we aim to combine data collected by multiple people and to detect patterns in long-term data.

We believe that our findings are very relevant for the robotics community since the extraction of high-level context information from various streams of continuous sensor data is a fundamental problem in robotics. For instance, a similar technique and sensor suite could be applied to determine the navigability of outdoor terrain traversed by a robot.

Acknowledgments

The authors would like to thank Tanzeem Choudhury, Jonathan Lester, and Gaetano Borriello for useful discussions and for making their feature learning code available. Additional thanks go to the Intel Research Lab in Seattle for providing the sensor boards used in this research, to Hanna Pasula for developing the user interface and to the NIST evaluation team for their extraordinary effort in preparing and running the evaluation. This work has partly been supported by DARPA's ASSIST and CALO Programmes (contract numbers: NBCH-C-05-0137, SRI subcontract 27-000968), and by the NSF Human and Social Dynamics (HSD) program under contract number IIS-0433637.

References

1. L. Bao and S Intille. Activity recognition from user-annotated acceleration data. In *Proc. of the International Conference on Pervasive Computing and Communications*, 2004.
2. H.H. Bui, S. Venkatesh, and G. West. Policy recognition in the abstract hidden markov model. *Journal of Artificial Intelligence Research (JAIR)*, 17, 2002.
3. A. Doucet, J.F.G. de Freitas, K. Murphy, and S. Russell. Rao-Blackwellised particle filtering for dynamic Bayesian networks. In *Proc. of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2000.
4. J. Ho and S. Intille. Using context-aware computing to reduce the perceived burden of interruptions from mobile devices. In *Proc. of the Conference on Human Factors in Computing Systems (CHI)*, 2005.
5. N. Kern, B. Schiele, and A. Schmidt. Recognizing context for annotating a live life recording. *Personal and Ubiquituous Computingd*, 2005.
6. J. Lester, T. Choudhury, N. Kern, G. Borriello, and B. Hannaford. A hybrid discriminative-generative approach for modeling human activities. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2005.
7. L. Liao, D. Fox, and H. Kautz. Learning and inferring transportation routines. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 2004.
8. L. Liao, D. Fox, and H. Kautz. Hierarchical conditional random fields for GPS-based activity recognition. In S. Thrun, H. Durrant-Whyte, and R. Brooks, editors, *Robotics Research: The Eleventh International Symposium*, Springer Tracts in Advanced Robotics (STAR). Springer Verlag, 2006.
9. P. Lukowicz, J. Ward, H. Junker, M. Stäger, G. Tröster, A. Atrash, and T. Starner. Recognizing workshop activity using body worn microphones and accelerometers. In *Proc. of Pervasive Computing*, 2004.
10. A. Subramanya, A. Raj, J. Bilmes, and D. Fox. Recognizing activities and spatial context from wearable sensors. In *Proc. of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2006.
11. S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, Cambridge, MA, September 2005. ISBN 0-262-20162-3.