Wireless Networks
DOI 10.1007/s11276-006-5064-1

Indoor location tracking using RSSI readings from a single Wi-Fi access point

G. V. Zàruba · M. Huber · F. A. Kamangar · I. Chlamtac

Published online: 8 June 2006

Abstract This paper describes research towards a system for locating wireless nodes in a home environment requiring merely a single access point. The only sensor reading used for the location estimation is the received signal strength indication (RSSI) as given by an RF interface, e.g., Wi-Fi. Wireless signal strength maps for the positioning filter are obtained by a two-step parametric and measurement driven ray-tracing approach to account for absorption and reflection characteristics of various obstacles. Location estimates are then computed using Bayesian filtering on sample sets derived by Monte Carlo sampling. We outline the research leading to the system and provide location performance metrics using trace-driven simulations and real-life experiments. Our results and real-life walk-troughs indicate that RSSI readings from a *single access point* in an indoor environment are sufficient to derive good location estimates of users with sub-room precision.

Keywords Location estimation · System design · Wireless experiments · Simulation · Monte Carlo sampling

G. V. Zàruba (✉) · M. Huber · F. A. Kamangar
Department of Computer Science and Engineering,
The University of Texas at Arlington
e-mail: zaruba@cse.uta.edu

M. Huber
e-mail: huber@cse.uta.edu

F. A. Kamangar
e-mail: kamangar@cse.uta.edu

I. Chlamtac
Create-Net and University of Trento,
email: chlamtac@create-net.it

1. Introduction and motivation

The globally pervasive computing environments [29] of the future with their large number of heterogeneous, mobile computing nodes pose many challenges but also offer unprecedented opportunities. In such environments, sensors and other computational nodes are omnipresent while people may carry a large numbers of small, connected devices that permit communication and computation at any point in time. However, to harvest the power of such computing environments it is necessary that computing nodes are context-aware, i.e., that they are able to adjust their operation to the particular context in which they currently operate. An important part of context-awareness is location-awareness [3,13], where nodes are enabled to obtain estimates of their physical location, thus tailoring the physical environment and its software representation to the user. Overall, in pervasive computing the more information is available about the local environment of users and computing entities the better the applications can adapt to the user. Thus, we argue that estimating location via various sensor readings is an important enabler of pervasive computing.

Location aware computing has a bright future in the fields of personal navigation, personal security, prompt healthcare and entertainment. Furthermore, information on the physical location of mobile nodes can greatly help in urban search and rescue missions, as well as enable geographical routing in ad hoc multi-hop networks. The determination of physical location is sometimes referred to as location estimation, location identification, localization, positioning or geolocation identification. Recent projects and interest in intelligent home environments such as [5] require the environment to know the whereabouts of the inhabitants in order to adapt the environment to them. In this paper we restrict ourselves to indoor localization, more precisely to in-home localization

(without loosing the generality to apply our results to any indoor environment).

Due to the fact that networked mobile users are enabled with wireless radio communication network interfaces (such as Wi-Fi), protocols that provide location estimation based on the received signal strength indication (RSSI) of wireless access points are becoming more-and-more popular, precise, and sophisticated. The main benefit of RSSI measurement based systems is that they do not require any additional sensor/actuator infrastructure but use already available communication parameters and downloadable wireless maps for the position determination.

The work we are presenting relies only on received signal strength measurements from wireless radio access points to determine the location of users. The major contribution of our paper is to show that good results can be achieved with readings from a ***single access point.*** This is in contrast to previous works where at least three access points were required to localize users inside office buildings restricted to the corridors of these buildings.

1.1. Localization techniques

The best-known location determination system is the Global Positioning System (GPS) [8]. A GPS receiver can estimate its location by measuring the propagation time of GHz range radio signals from several satellites to the receiver. Although, after the recent liberalization of GPS the precision of the obtained GPS position is quite accurate (down to a meter), GPS receivers need to have line-of-sight to at least three or four satellites in the sky. This restriction of GPS prohibits/restricts its use in dense urban environments, indoors, and in areas with tall and dense vegetation where line of sight to the required number of satellites is not available. Moreover GPS localization is not reliable in a pervasive computing context, where small wearable and implanted terminals might be carried in a way that restricts line-of-sight to the GPS satellites.

Although GPS is the widest spread (and global) positioning system, there are several other approaches available for location estimation and even more have been proposed in the literature. The types of sensors used to obtain the location information vary from ultrasonic through photonic to radio signal strength measurement sensors. Most of the available and proposed location identification systems rely on proprietary and deployment-expensive infrastructure and protocols. Wide area cellular systems use either mobile terminal integrated GPS receivers, or triangulate their position based on received signal strengths from several known-location base stations [18,25]. Since our work is focused towards the indoor environment we will not revisit outdoor positioning systems. In the indoor environment, infrared and laser transmitter/receiver systems [1,28], ultrasonic sensor/actuator systems [21], computer vision systems [7,16], physical contact [20] and close proximity radio identification (RFID) sensor [27] based localization systems have been proposed and built to track mobile users. RSSI based location systems [4,17,22,24,31] are becoming more-and-more precise and sophisticated. Their main benefit is that they do not require any additional sensor/actuator infrastructure. Yet, most of the RSSI approaches need readings from at least 3 access points at each location to provide sufficiently accurate estimates.

In general, RSSI based positioning includes two phases: (i) the training phase where the wireless map of the environment is determined using field measurements and (ii) the positioning phase where position estimation is performed based on the wireless map. Note that the training phase is an offline process and as such only needs to be redone if there have been major changes to the wireless propagation environment (e.g., relocation of access points). Let us now revisit some of the relevant previous approaches to RSSI based location estimation.

In [18] the authors apply an extended Kalman filter [8] to RSSI measurements of cellular base-stations. Filtering out the noise (assumed to be Gaussian noise) they calculate intra-cell position, movement pattern, and velocity vectors in order to determine the most probable next cell crossing. Although [18] considers relatively macro-term outdoor movements, it was the first major work (to our best knowledge) to apply statistical methods to RSSI measurements to obtain position information.

RSSI based measurement techniques can be broadly divided into deterministic and probabilistic techniques. Both deterministic and probabilistic techniques require a long and human labor-intensive training phase and provide less precision than probabilistic techniques.

Deterministic techniques include [2] and [24] where the location area is subdivided into smaller cells and readings are taken in these cells from several known access points (the training phase). In the positioning phase the most likely cell is selected, i.e., the cell that best fits the current measurement.

Probabilistic positioning techniques include [4,17,22,24,31] where a probability distribution of the user's location is defined over the area of the movement. The goal of the positioning is to reach a single mode for this distribution, which is the most likely location of the tracked user. In [22] the authors establish and train a Bayesian belief network with a preset number of discretized location possibilities (cells). The Bayesian network is established with the a-priori probability distribution of a user being at a given location and by the conditional probabilities with which a given RSSI is measured at the location. By inverting the Bayesian network, they derive the conditional probabilities (and thus the a-posteriori distribution over locations) of a user being at the different cells given the current RSSI reading. The results of this approach show

a very coarse location determination with a large computational and memory overhead. In [31], the authors use a similar Bayesian model except that inversion calculations are not made for all base-stations, but for only the strongest subset of them, thus reducing the computational burden. Both [22] and [31] apply their systems to hallways of office environments and show relatively reliable positioning with coarse and predefined resolutions. The model in [24] is a generalized version of [4] that applies machine learning techniques to the Bayesian network to increase precision (which in turn increases the computational burden). In [17] the authors take the Bayesian approach a step further by including directions of users in their model, hence sampling the RSSI measurements not only for different locations but also for different orientation of the receiving antennas (and user "obstacles").

Our work can be categorized as a probabilistic approach. The main tool and theme throughout this paper will be Bayesian filtering using Monte Carlo sampling (introduced in [7]), where the probability distribution of the location of users is captured, followed, and calculated by sampling. This method can use an arbitrary a-priori distribution converging (or "collapsing") to a single mode of the sampled distribution. Furthermore, Monte Carlo sampling is not only less computationally expensive than evaluating Bayesian networks but we believe that it can also be easily deployed in the case where the reference points (the access points) are mobile themselves. Our approach can be seen as a wireless counter-part to the sonar and computer vision location systems introduced in [7]. Furthermore, we establish a framework to enable users to change their orientation, and thus face an arbitrary direction, similarly to [17]. Section 4.2 shows simulation results that demonstrate the applicability of such approach.

1.2. Wireless maps—the training phase

To estimate location from received signal strength readings it is necessary to have a spatial statistical representation of the received signal strengths from the surrounding access points. All the above mentioned RSSI based indoor localization approaches rely on a long "training" phase where the entire target area is measured with some spatial precision. This precision will in turn have a great impact on the precision of the location estimate.

However, such data collection/measurement requires significant human labor (and the target area needs to be re-measured every time a new access point is introduced or in general whenever the wireless propagation properties may change). Often, it would be preferable to be able to use a simple model of the environment to determine a model for the signal strength's distribution. A number of such modeling techniques have been devised for managed wireless networks in office environments. In [11] the authors evaluate ray-tracing techniques that are used to derive indoor propagation models while in [12] a statistical approach is introduced that builds a wireless map based on statistical properties of rooms and the area. The authors in [12] claim that the statistical approach has an error of less than 3 dB compared to measured values. The ray-tracing approaches provide an even better model but come with a larger computational complexity. [1] In this paper we are devising a framework based on RSSI measurement samples infused into ray-tracing to determine wireless signal strength maps.

1.3. Sequential Monte Carlo sampling

Probabilistic approaches to mobile node localization from RSSI measurements rely on the precise estimation of a posterior probability distribution, $p(s_t \mid d_1, \ldots, d_t)$, of the likelihood of the node's state (location), s_t, given a history of the received measurements, $d_1, \ldots, d_t$. The goal here is to derive a representation that explicitly deals with missing information about the motion of the mobile node and the uncertainty present in the measurement data. The main problem when using such probabilistic representations and the corresponding estimation algorithms is that they can be prohibitively complex if no simple, parametric representation of the uncertainty is available. As a result, most existing approaches to localization using RSSI measurements rely either on the discretization of space into a small number of regions of interest [2 ,12,24,31] or an unrealistic model of the uncertainty of the measurements [18,24]. An example of the latter can be seen in Kalman filter-based approaches where an assumption has to be made that the probability distribution for the locations as well as for the measurement error model are Gaussian. However, in the presence of highly ambiguous measurements such as RSSI readings in indoor environments, these distributions are generally multi-modal, indicating the existence of multiple possible positions for the node and for multiple locations that match a particular set of RSSI readings. Since Kalman filters can estimate only the mean of the posterior distribution, in a situation where a user could with high probability be at one of two locations that are distant from each other, the Kalman filter's estimate may be the Euclidian mean of those two location, thus absolutely wrong.

In recent years, Monte Carlo sampling-based techniques for the estimation of probability distributions have been developed [6,7,10] and applied to different problems including visual target tracking [14] and mobile robot localization using laser range finder or sonar data [7,26]. In these

[1] If access points are static, the radio-wave ray-tracing is an offline process, i.e., it can be run prior to the localization, thus its computational complexity is not a major issue.

simulation-based techniques, empirical probability distributions, $p_N(s)$, are represented by a set of N weighted, random samples, $\{(s^{(i)}, w^{(i)}) \mid i \in [1, N]\}$ as $p_N(s) = \sum_{i=1}^{N} w^{(i)}\delta_{s^{(i)}}(s)$, where $w^{(i)}$ is the weight of sample $s^{(i)}$ and $\delta_{s^{(i)}}(s) = \delta_{s^{(i)}}(s - s^{(i)})$ denotes the Dirac delta function. This distribution approximates the actual probability distribution, $p(s)$, as $\int_{s1}^{s2} p(s)ds \approx \int_{s1}^{s2} p_N(s)ds = \sum_{s^{(i)}\in[s1,s2]} w^{(i)}$.

As a result, Monte Carlo sampling-based techniques can be used to represent arbitrary probability distributions as long as the number of samples is sufficiently high. Calculations of posterior distributions $p(s \mid d)$ in these techniques (also referred to as bootstrap or particle filters) are performed by re-sampling operations on the samples representing the prior distribution $p(s)$. The computational complexity of these Monte Carlo techniques is therefore determined directly by the number of samples used to represent the distribution and computation time can be traded off against precision in a straightforward manner by modifying the number of samples used. As a result, even nodes with minimum computation power should be able to successfully run the resulting algorithms, albeit with reduced precision.

The rest of the paper is organized as follows: Section 2 presents our approach to wireless map calculation. Section 3 describes our approach to filtering RSSI readings to obtain location estimates. Section 4 presents simulation results on our system and provides real-life measurements and comparisons. Finally, we conclude and introduce future research directions.

2. Wireless map calculation

This section outlines a five-step approach (including two-steps of ray-tracing) to calculate wireless signal strength maps. In the first step a scaled floor-plan with all the walls, doors, and windows (and other major obstacles) of the environment is entered. In the second step, a small number of measurement points are defined and entered in the floor-plan, and signal-strength measurements are taken at the same physical locations. Step three witnesses the execution of the first ray-tracing step to provide a parametric description of signal strengths at the measurement points (e.g., how many different obstacles do radio wave rays pass through and/or reflect off until they reach the measurement points). In step four the parametric representation of the signal at the measurement points is approximated using the real measurement values. Step five serves to calculate the wireless map using ray-tracing again, but this time with the inserted transmission and reflection properties of the obstacles obtained in the previous step. In the next subsections all these five steps are described in more detail.

2.1. Step one: Floor-plan

To obtain a wireless map, the first step is to define the environment. This requires measuring the wall, window and door lengths of rooms as well as identifying major obstacles that could absorb or reflect the radio signal. In our experiments we used a 150 square meter floor-plan house, which was already equipped with a Wi-Fi access point. We have defined four different obstacle types: brick walls, interior walls, windows and doors. We have used the xfig format [30] due to its simple design and parametric format to define the floor-plan. The map is depicted in Fig. 1, where black lines around the perimeter are brick walls (obstacle type O_1), grey lines around the perimeter are windows (O_2), grey lines inside the house are interior walls (O_3) and dark lines inside the house are doors (O_4). Furthermore, the grey square indicates the location of the access point (in Bedroom 2) while the scattered small discs represent the measurement points to be used in the second step. The floor-plan could be enhanced by adding furniture locations and other objects. This would

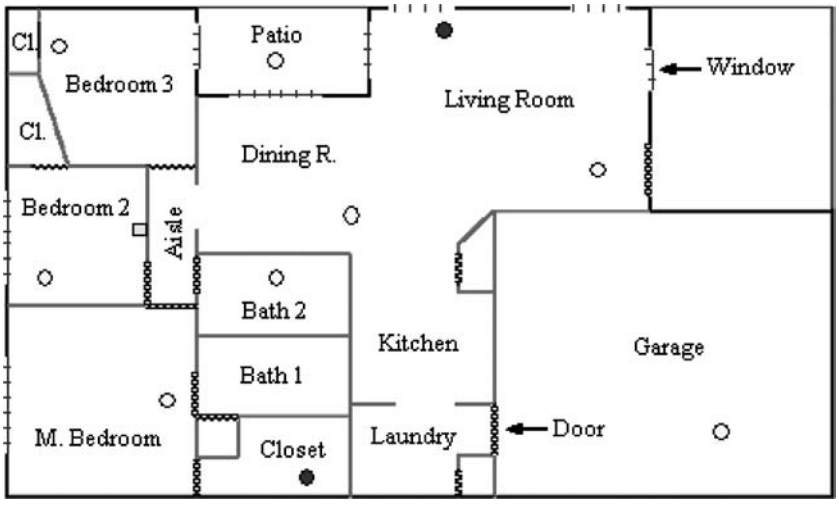

Fig. 1 Floor-plan of a house with measurement points

increase the complexity of the fourth step because the transmission and reflection parameters for these objects must also be calculated.

2.2. Step two: Measurement and measurement points

In the second step the user is required to define measurement points well-spread across the floor-plan (as represented by the scattered discs in Fig. 1). The number of measurement points to be defined depends on the number of obstacles in the floor-plan. To obtain good results, at least twice as many measurement points are needed as there are obstacle types (as explained in the next subsection). In our example we have defined 10 measurement points (8 due to the 4 different obstacle types and an additional 2 for increased precision). Note that the more measurement points we define the better the precision of the final map will be. An average signal strength reading for each of the measurement points needs to be taken and recorded. Let us denote the measurement points by $M_1 \ldots M_m$, where m is the total number of measurement points. Furthermore, let us denote the power readings taken at these measurement points by $W_{M_1} \ldots W_{M_m}$.

In our example, several signal strength measurements at each of the measurement points were taken using an Orinoco Wi-Fi interface equipped, Linux managed laptop. These measurements were averaged for each measurement point and stored for later use. Additionally, we have performed measurements to obtain no-obstacle one-meter distance signal strength measurements for the access-point—wireless network adapter pair to be used as the calibration for step four; we will denote this measurement by P_0.

2.3. Step three: Parametric ray-tracing

In this step we are running a "parametric" ray-tracing algorithm to find the signal strength of the access point at all the measurement points as a function of the transmission and reflection parameters of the obstacles. Wi-Fi works at the 2.4 GHz frequency range, where the radio signal coming off the access point can be approximated with the sum of single straight-line propagation rays emanating in all directions around the omni-directional antenna of the access point [11]. If a radio energy beam hits an obstacle, some portion of the energy of the beam is assumed to be reflected (bounce) and some portion of the energy is assumed to be transmitted (let through) by the obstacle. We denote the reflection and transmission coefficients of the ith obstacle type by R_{O_i} and T_{O_i} respectively, where $\left(R_{O_i} + T_{O_i}\right) < 1$. (Thus, the obstacle is assumed to absorb $1 - \left(R_{O_i} + T_{O_i}\right)$ portion of the energy beam going through it). For example, if obstacle-1 is in the way of a traced radio ray, then it is assumed to reflect R_{O_1} portion of the energy of the ray, while letting T_{O_1} portion of the same incoming energy propagate through.

Rays are generated starting from the access point in all directions but to keep the process computationally feasible, the direction of the rays has to be discretized between $(0, 2\pi)$ with a $\Delta\alpha$ precision. Thus, the resulting $2\pi/\Delta\alpha$ rays need to be traced one-by-one sequentially. A ray crossing an obstacle will cease to exist but will spawn two other rays, one going on in the same direction (if the perimeter of the target area is not reached) and the other bouncing back from the obstacle. Each of these newly spawned rays will inherit properties from its parent, such as the distance traveled, the number and type of obstacles it has already been reflected and transmitted by. Each ray remains in the system until the total distance traveled by it reaches a pre-defined threshold. If a ray crosses the area of a measurement point, then its current parameters (distance traveled, number and type of different transmissions and reflections) are stored for that measurement point. Let us denote (i) the different types of obstacles by $O_1 \ldots O_b$; (ii) the number of reflections from these obstacles by $r_1 \ldots r_b$ (e.g., r_i corresponds to the number of reflections from obstacle-type i, i.e., O_i); (iii) the number of transmissions going through obstacles by $t_1 \ldots t_b$ (e.g., t_i corresponds to the number of transmissions by obstacle-type i, i.e., O_i); (iv) and the distance of travel by d. The power-footprint of a crossing ray at the measurement point is:

$$\frac{P_0}{d^2} \prod_{i=1}^{b} R_{O_i}^{r_i} * \prod_{i=1}^{b} T_{O_i}^{t_i}.$$

Furthermore, let us denote the number of rays going through the measurement points by $N_{M_1} \ldots N_{M_m}$ respectively. Thus, for each of these points we have a polynomial defining the received signal strength:

$$P_{M_j} = \sum_{k=1}^{N_{M_j}} \left(\frac{P_0}{{}^k d^2} \prod_{i=1}^{b} R_{O_i}^{{}^k r_i} * \prod_{i=1}^{b} T_{O_i}^{{}^k t_i} \right) \tag{1}$$

In Eq. (1), $R_{O_1} \ldots R_{O_b}$ and $T_{O_1} \ldots T_{O_b}$ are unknowns to be determined, and upper-left indices represent ray indexes for the kth ray (i.e., ${}^k r_i$ is the number of reflections the kth ray had encountered on obstacle-type (i). Thus by the end of the third step for each measurement point we have a polynomial representation of the total signal strength.

2.4. Step four: Obstacle parameter determination

The signal strength measurements obtained in the second step $\left(W_{M_1} \ldots W_{M_m}\right)$ can now be used as estimates for $P_{M_1} \ldots P_{M_m}$ (i.e., $P_{M_1} - W_{M_1}, \ldots, P_{M_m} - W_{M_m}$ should all be close to zero). By looking at Eq. (1) we note that we have

Fig. 2 Calculated wireless map

$2b$ unknowns. In step-four we will estimate the values of these unknowns thus finding values for the reflection and transmission parameters of all obstacle types. In order to be able to solve the above polynomials for the $2b$ unknowns, we need at least $m \geq 2b$ equations, i.e., $m \geq 2b$ measurement points. Due to the coarse modeling of the environment, we have to also include an error term e_i into the measurement. Then, for all i, where $m \geq i \geq 1; W_{M_i} = P_{M_i} + e_i$. We can now calculate a least square error estimate for all unknowns by minimizing the following function F:

$$F\left(R_{O_1} \ldots R_{O_b}, T_{O_1} \ldots T_{O_b}\right) = \sum_{j=1}^{m} e_j^2 = \sum_{j=1}^{m} \left(W_{M_j} - P_{M_j}\right)^2$$

Minimization of function F is a complex problem due to the number of variables and the order of the individual polynomials. Thus, to minimize F we employ a heuristic optimization technique, namely simulated annealing (SA) [15,19]. With simulated annealing, sub-optimal values for $R_{O_1} \ldots R_{O_b}$, and $T_{O_1} \ldots T_{O_b}$ can be obtained (thus finding estimates on how what proportion of the energy reaching the different obstacle types is reflected or transmitted). Our SA algorithm takes the polynomial description file generated in the third step, the measured values for the measurement points, and P_0 (obtained in the second step) as the input and provides the estimated values for the reflection and transmission parameters for all defined obstacles.

2.5. Step five: Ray-tracing for wireless power maps

In step five, the actual wireless power map is obtained using the previously calculated power reflection and transmission coefficients for the defined obstacles. The first task is to define the angular precision of the ray-tracing by selecting an appropriate $\Delta\alpha$ as well as defining the resolution of the wireless power map by discretizing the target area into consecutive Δs side-sized square cells. The selection of Δs has little impact on the complexity of computation of the ray-tracing or the localization algorithms. However, Δs has an impact on the precision of the location estimate. The ray-tracing process is similar to that of the third step with the exception that radio rays now leave a scalar power level footprint in all cells they travel through. Figure 2 shows a visualization of the wireless power map for our sample floor-plan, with $\Delta s = 0.3$ m, where a darker cell indicates higher signal strength (the darkness of cells changes logarithmically in the power level, since a dB scale is used to represent the values). A comparison and discussion on the simulated and measured power levels can be found in Section 4.1.

3. Monte Carlo sampling-based bayesian-filtering

This section describes the basic approach to mobile node localization from RSSI measurements and introduces four estimation models that have been implemented and tested. Our goal is to obtain an estimate of the posterior probability distribution, $p(s_t \mid d_1, \ldots, d_t)$, of potential states (locations), s_t, using the RSSI measurements, $d_1, \ldots, d_t$, and the wireless power map introduced in Section 2.

The calculation of the distribution of the user, given the sequence of RSSI readings is performed recursively using a Bayes filter:

$$p(s_t \mid d_1, \ldots, d_t) = \frac{p(d_t \mid s_t, d_1, \ldots, d_{t-1}) \cdot p(s_t \mid d_1, \ldots, d_{t-1})}{p(d_t \mid d_1, \ldots, d_{t-1})}$$

Assuming that the Markov assumption holds, i.e., $p(s_t \mid s_{t-1}, \ldots, s_0, d_{t-1}, \ldots, d_1) = p(s_t \mid s_{t-1})$, the filtering equation can be transformed into the recursive form:

$$p(s_t \mid d_1, \ldots, d_t) = \frac{p(d_t|s_t) \cdot \int p(s_t|s_{t-1}) \cdot p(s_{t-1}|d_1, \ldots, d_{t-1}) ds_{t-1}}{p(d_t|d_1, \ldots, d_{t-1})},$$

where $p(d_t \mid d_1, \ldots, d_{t-1})$ is a normalization constant. In the case of the localization of a mobile node from RSSI measurements, the Markov assumption requires that the state contains all available information that could assist in predicting the next state and thus an estimate of the non-random motion parameters of the node is required as part of the state description. Starting with an initial, prior probability distribution, $p(s_0)$,[2] a system model, $p(s_t \mid s_{t-1})$, representing the motion of the mobile node, and the measurement model, $p(d \mid s)$, it is then possible to derive new estimates of the probability distribution over time, integrating one new

[2] A usual initial distribution is a uniform distribution, which means that the user is equally likely to be located at any part of the house in our application.

measurement at a time. Each recursive update of the filter can be broken into two stages:

Prediction: Use the ***system model*** to predict the state distribution based on previous readings:

$$p(s_t|d_1,\ldots,d_{t-1}) = \int p(s_t|s_{t-1}) \cdot p(s_{t-1}|d_1,\ldots,d_{t-1}) ds_{t-1}$$

Update: Use the ***measurement model*** to update the estimate:

$$p(s_t|d_1,\ldots,d_t) = \frac{p(d_t|s_t)}{p(d_t|d_1,\ldots,d_{t-1})} p(s_t|d_1,\ldots,d_{t-1})$$

To address the complexity of the integration step and the problem of representing and updating a probability function defined on a continuous state space (which therefore has an infinite number of states), the approach presented here uses a sequential Monte Carlo filter to perform Bayesian filtering on a sample representation. As introduced in Section 1.3, a distribution is represented by a set of weighted random samples and all filtering steps are performed using Monte Carlo sampling operations. In particular, the initial sample distribution, $p_N(s_0)$, is represented by a set of uniformly distributed samples with equal weights, $\{(s_0^{(i)}, w_0^{(i)}) \mid i \in [1, N], w_0^{(i)} = 1/N\}$, and the filtering steps are performed as follows:

Prediction: For each sample, $(s_{t-1}^{(i)}, w_{t-1}^{(i)})$, in the sample set, randomly generate a replacement sample according to the system model, $p(s_t \mid s_{t-1})$. This result in a new set of samples corresponding to $p(s_t|d_1,\ldots,d_{t-1})$:

$$\left\{\left(\tilde{s}_t^{(i)}, w_t^{(i)}\right) \mid i \in [1, N], w_t^{(i)} = 1/N\right\}$$

Update: For each sample, $(\tilde{s}_t^{(i)}, w_t^{(i)})$, set the importance weight to the measurement probability of the actual measurement, $\tilde{w}_t^{(i)} = p(d_t \mid \tilde{s}_t^{(i)})$. Normalize the weights such that $\sum_{i\in[1,N]} \eta \cdot \tilde{w}_t^{(i)} = 1.0$ and draw N random samples for the sample set $\{(\tilde{s}_t^{(i)}, \eta \cdot \tilde{w}_t^{(i)}) | i \in [1, N]\}$ according to the normalized weight distribution. Set the weights of the new samples to $1/N$, resulting in a new set of samples $\{(s_t^{(i)}, w_t^{(i)}) \mid i \in [1, N], w_t^{(i)} = 1/N\}$ corresponding to the posterior distribution $p(s_t \mid d_1, \ldots, d_t)$.

To apply the filter to the problem of mobile node localization from RSSI measurements, a system model and a measurement model have to be provided. The latter is here given by the wireless power map introduced in Section 2 and the assumption that actual RSSI readings will diverge from the map according to a Gaussian probability distribution with a standard deviation of 3 dB. For the system model, four different motion models have been implemented and tested in an example environment with a single wireless access point and no additional location information. The first three models assume that the receiving antenna of the entity to be located is always facing the access point (so the similarly calculated wireless map can be used without modification), while the fourth model extends the framework to deal with users that may change their rotational direction relative to the access point. The following sections introduce the individual movement models and their implications for the Monte Carlo filter before they are evaluated in Section 4.

3.1. Simple sequential Monte Carlo filter

The simplest localization algorithm tested here uses a system model that assumes that at every point in time, the node moves with a random velocity drawn from a normal distribution with a mean of 0 m/s and a standard deviation of 1 m/s, $v_t \sim N(0, 1)$. This model corresponds to a random displacement of the mobile node at every point in time. No information about the environment is included in this model, and as a consequence, the filter permits the estimates to move along arbitrary paths (including ones that move through the walls). Since this motion model also does not consider any past motions (i.e., it does not estimate the speed and/or acceleration of the user), the state of the localization filter can be represented as a vector of the x and y location, $s = (xy)^T$, while the measurements d correspond to RSSI measurements in dB.

3.2. Simple sequential Monte Carlo filter with boundary information

To model the effects of boundaries and limit the simulated sample trajectories to physically possible ones, the second filter uses information about the location of the walls to modify the Gaussian velocity model by limiting available choices to velocities that do not lead to collisions with permanent obstacles (such as windows or walls). Figure 3 shows an example for this motion model, which corresponds to a random displacement limited by the walls. The figure shows the probability density of moving to a new location within one time step assuming that the mobile node is located at the center of the distribution. Dark regions correspond to high probabilities while light colors represent low displacement probabilities (dark lines indicate walls).

Similarly to the simplest system model of Section 3.1, this filter system model uses a two-dimensional state vector, $s = (xy)^T$, and measurements, d, corresponding to the RSSI readings.

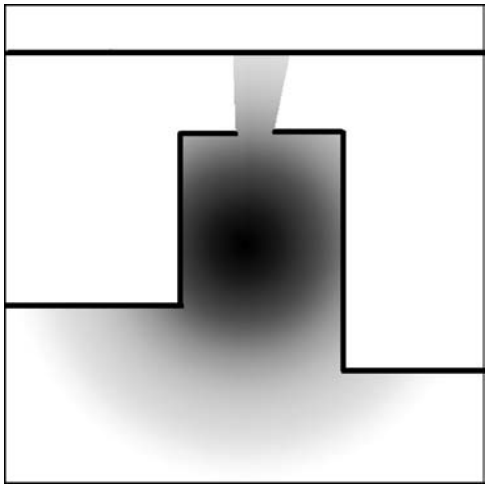

Fig. 3 Probabilistic displacement model with wall limitations

3.3. Sequential Monte Carlo filter with internal motion model

Due to the inertia of a physical node moving in the environment, the random displacement model used in Section 3.1 might not be sufficiently realistic and a more complex model was implemented as a third filter. This movement model assumes that nodes tend to move at a constant velocity, i.e., have a zero mean acceleration. Thus the model relies on the previous velocity with a current random acceleration, where the acceleration is drawn randomly from a normal distribution with mean 0 m/s^2 and a standard deviation of 1 m/s^2, $a \sim N(0\ \text{m/s}^2, 1\ \text{m/s}^2)$. To permit the use of this model, the state used in this filter system model has to include an estimate of the current velocity of the mobile node and is thus a four-dimensional vector consisting of the position and velocity of the mobile node, $s = (x\ y\ v_x\ v_y)^T$. To address walls, particles that cross walls in the prediction step of the filtering update are assigned a weight of 0 and are thus discarded during the re-sampling step at the end of the update step.

3.4. Sequential Monte Carlo filter with internal motion and rotation model

In the previous three system models an implicit assumption was made that the user is always facing the access point. Due to movement trajectories that do not go towards the access point this assumption is likely not to hold unless the user exhibits the discipline to indeed always face the access point. A more appropriate user (and thus filter system) model would let the user face toward her current velocity vector, thus toward the direction she is moving. While the previous model extensions did not induce any change to the measurement model, allowing users to turn away from the access point has a definite impact also on the latter.

Looking at the filter system model, an absolute direction needs to be added that is calculated based on the current velocity vector of the user, thus $s = (x\ y\ v_x v_y\ \phi)^T$. Similarly to the last two models, particles crossing permanent obstacles will have their weight set to 0, thus eliminating them from the set of valid particles in the measurement step.

The measurement model also needs to be updated by including the absolute direction ϕ of the user. For this a rotation model needs to be established that determines how much the difference is between a an RSSI reading taken facing the access point and a reading facing another direction (denoted by ϕ). To obtain such an approximate model we have performed several experiments in the environment at various locations. In each of these experiments we have continuously taken measurements while turning around the axis of the receiver's antenna. The average relative reading of these measurements is depicted by the solid curve in Fig. 4. (In our experimental setup the Orinoco card sticks out from the left-hand side of the laptop, thus turning by 90 degrees will align the receiver antenna with the access point—resulting in higher readings.) It can be observed that the measured relative gain (not surprisingly) somewhat resembles a sin wave. The dashed line of Fig. 4 shows an empirical interpolated sin function $S(\phi) = 7.5 \sin(\phi) + 1.2$ which is used in the measurement model as a compensation for the user's direction.

4. Performance studies

The goals of our evaluation studies are three-fold: (i) to validate our ray-tracing based wireless power map with real-life measurements (Section 4.1); (ii) to simulate a scenario where a mobile user needs to be tracked using our previously described filtering methods (Section 4.2); and (iii) to validate the wireless map generation combined with Monte Carlo filtering by a real walk-through of the sample house (Section 4.3).

4.1. Wireless map calculation studies

Our ray-tracing map generation was run with an angular precision of $\Delta\alpha = 2\pi/360$ radians (i.e., with a 1 degree precision) and square cells of size 0.3 m $\times$ 0.3 m covering the floor-plan of the sample house. We have taken 50 measurements for each of our 10 measurement points in step 2 of our wireless power map construction technique and averaged the results (for each measurement point). The wireless power map was obtained based on the above assumptions.

To evaluate our calculated power map we have randomly selected 20 cells (same 0.3 m cell side size) from the floor plan. At each of these cells we performed another 50 power

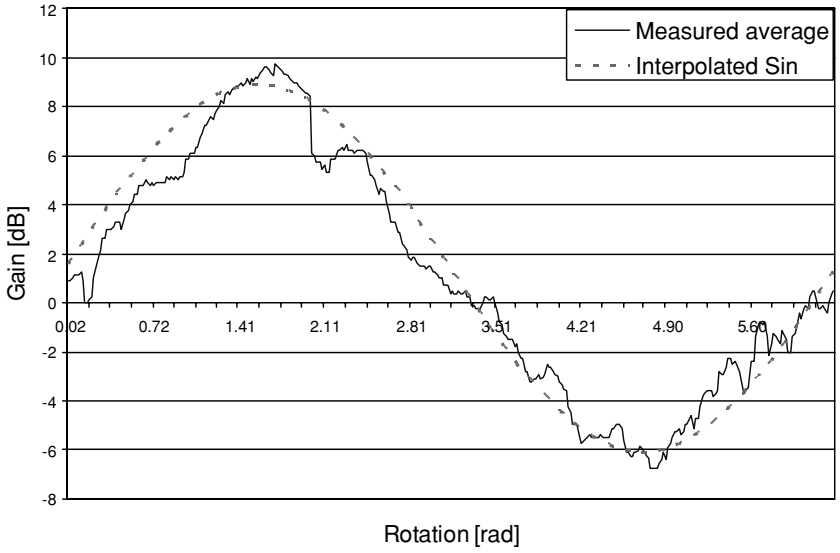

Fig. 4 Relative antenna gain vs. rotation (away from access point)

Table 1 Ray-tracing calculated and measured signal strength

Random cell	1	2	3	4	5	6	7	8	9	10
Calculated	45.3	37.8	29.5	37.0	47.3	41.4	54.1	38.7	64.9	59.5
Measured	43.8	39.6	28.3	37.2	46.4	40.9	52.0	36.3	62.7	60.0
Square error	2.5	3.3	1.5	0.0	0.8	0.2	4.5	5.5	4.9	0.2
Random cell	11	12	13	14	15	16	17	18	19	20
Calculated	48.7	24.6	30.1	36.1	23.9	16.8	27.4	25.4	36.4	43.1
Measured	46.9	24.2	32.1	33.2	23.2	15.9	28.2	23.1	39.0	41.8
Square error	3.3	0.2	3.8	8.3	0.5	0.8	0.6	5.6	6.5	1.7

level measurements. For each cell, the average and the standard deviation of the 50 readings were calculated. The mean measured and calculated values for the 20 random cells are shown in Table 1. Our measurements show that the standard deviation of our readings (although being somewhat location dependent), are averaging to around 2 dB, thus we can safely assume a 3 dB standard deviation for the measurements in the oncoming sections. To obtain a single scalar number describing the precision of our ray-tracing method, we have calculated the square-root of the mean square difference between the calculated and measured values to be 1.65 dB. Thus, we can conclude that our ray-tracing approach provides a sufficiently good estimate of the wireless propagation behavior of the sample house.

4.2. Location estimation simulation studies

To evaluate the particle filtering-based location estimation approach, we have created a discrete event simulation program in C++ that mimics the movement of a user inside the sample house as the environment. Power level measurement samples are drawn from the signal map according to the location of the user assuming a zero-mean Gaussian-noise model with a standard deviation of 3 dB (see remarks in the previous section). Power reading samples are taken every half-second and fed into the Monte Carlo filtering process. Although particles are expected to be distributed around the user they do not provide a single point estimate. To obtain the most likely position of the user we select the particle that has the most other particles in its radius-R surrounding. The movement path of the user can be defined in a text file by describing to what position in the house she should move and with what velocity. We have defined a 190 second long movement path using pedestrian speed (1 m/s) visiting several rooms of the house as depicted in Fig. 5.

To observe the behavior of the particles we have also created a graphical interface where the user, the particles, and the location estimate of the user are displayed after each sampling step. Figure 6 shows a short sequence of particle behavior.[3] The user and the single point estimate are denoted with the letters *U* and *E* respectively, while the particles are represented with red (gray) dots or dot-clouds.

We have implemented four different system update models as described in Section 3. We have run simulations for all four models to evaluate their location estimates' precision

[3] Due to the obvious limitations of a static media like paper the moving image of particles cannot be nicely represented. The authors will make the animation of the filter behavior available at their web-site..

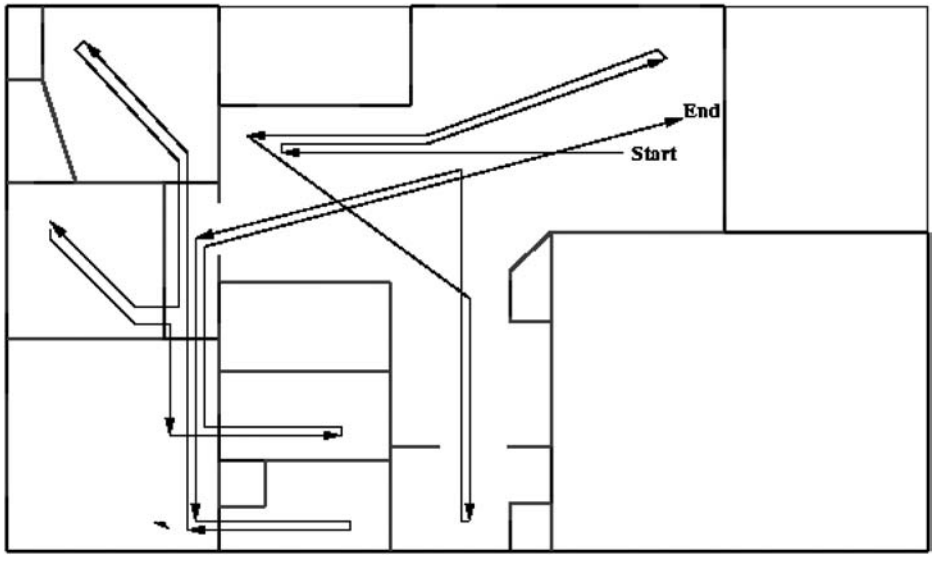

Fig. 5 Movement path of user in the sample house

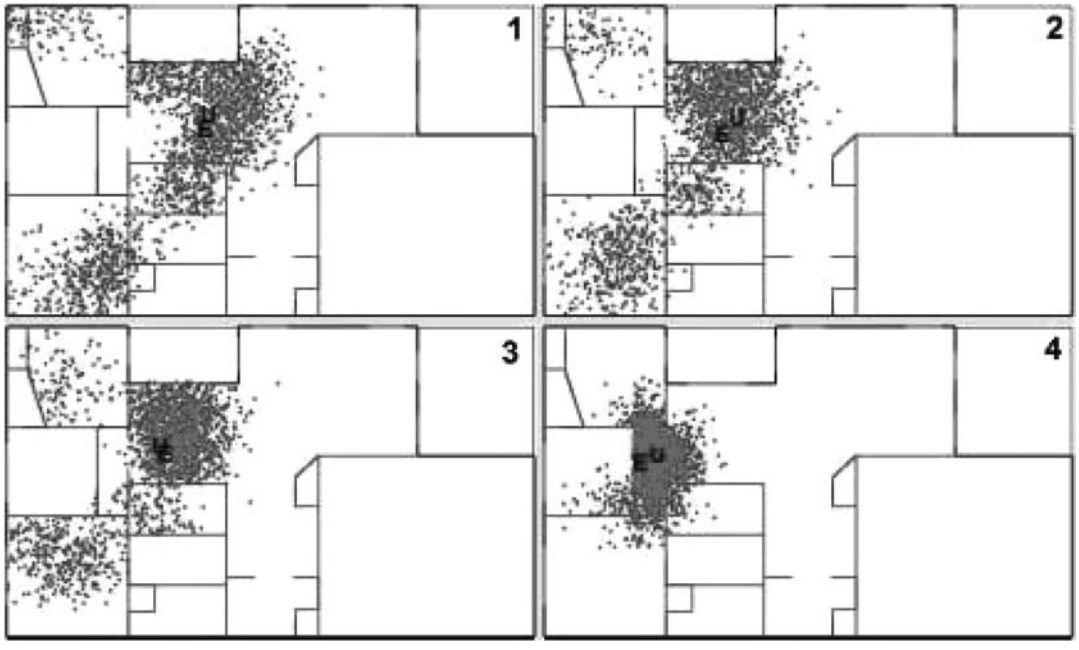

Fig. 6 Particle filtering in progress (U = user; E = single location estimate)

as a function of the number of particles. We measured the estimates' precision by calculating the average Euclidian distance between the user and the location estimate during the simulations. Figure 7 shows the impact of the number of particles on the estimates' precision for the zero-mean velocity, the wall and door considering normal variant, the 4-dimensional particle-filtering model, and the internal rotational model. As we can observe, the wall considering and the 4D particle models' estimates come on average as close as 1 m to the location of the user.

To investigate how the location estimates' precision varies over time for a fixed particle number, the deviations between the actual and estimated location were recorded along the path of the user. Figures 8–11 show the corresponding graphs for the wall ignoring, the wall considering, the 4-D, and the rotational particle models, respectively. In addition, Fig. 12 depicts the rotational error (in $(0, \pi]$) of the user during the simulation; the mean rotational error was 49 degrees with a standard deviation of 41 degrees.

Although, as we can observe, the simple zero-mean velocity Gaussian system model performs the worst it still provides great performance. The wall considering and the 4D particle filter models perform similarly, with the former having fewer but longer error bursts. All in all the latter two

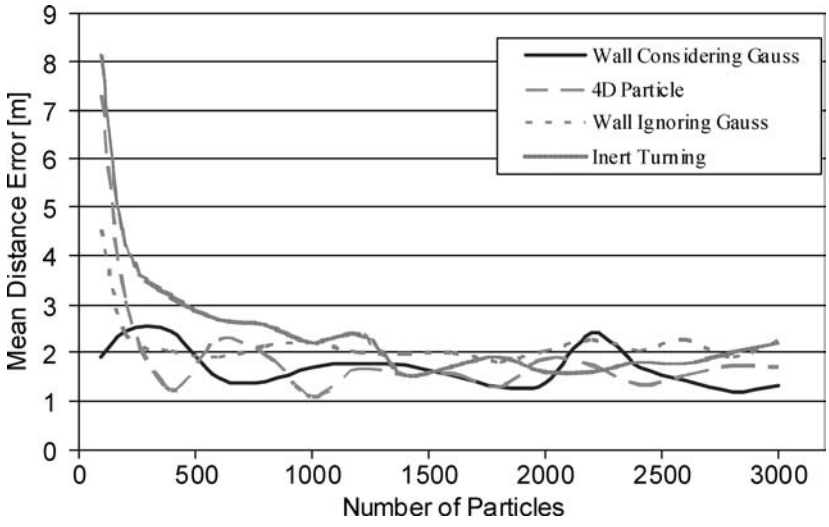

Fig. 7 Number of particles vs. precision of the location estimate

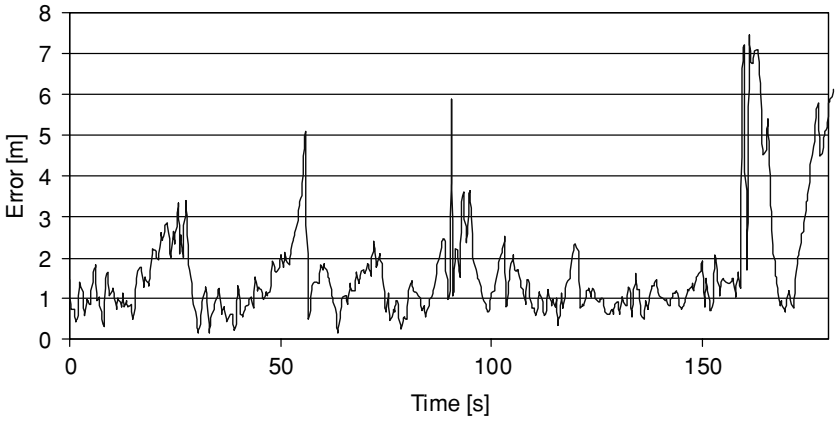

Fig. 8 Wall ignoring model error vs. simulation time

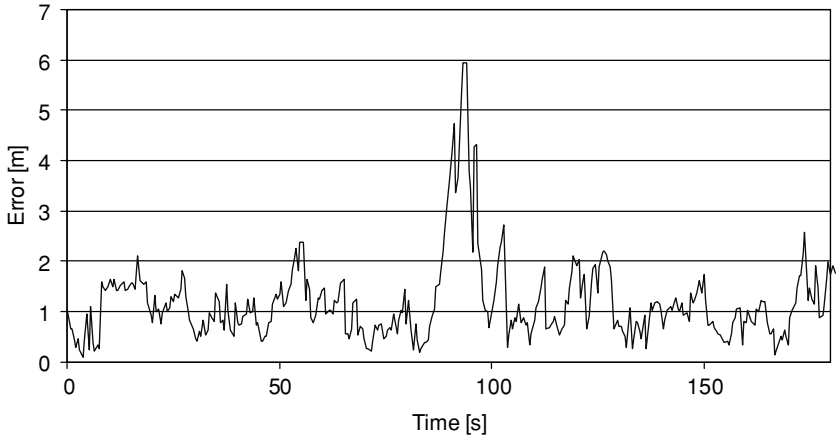

Fig. 9 Wall considering model error vs. simulation time

models rarely (less than 5% of the time and even then only for a short time) miss the room the user is actually located in, thus we argue that our location system can be deployed in a smart environment to locate users.

The internal rotation model draws the RSSI measurement from a normal distribution where the mean is the stored value in the power map corresponding to the location of the user plus $S(\phi)$, and the standard deviation is 3 dB. The figure tells us that the burstiness of errors is higher than for the other three models. Figures 11 and 12 show high error peaks at times where the simulated user has abruptly turned its direction, i.e., when there was a sudden large change in the velocity vectors and thus in ϕ. As the velocity vectors remain constant in the next epochs (and thus the change in ϕ

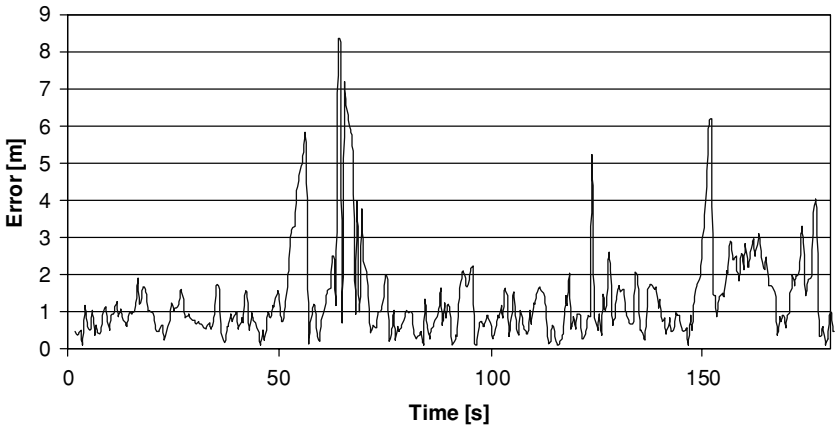

Fig. 10 4-D particle model error vs. simulation time

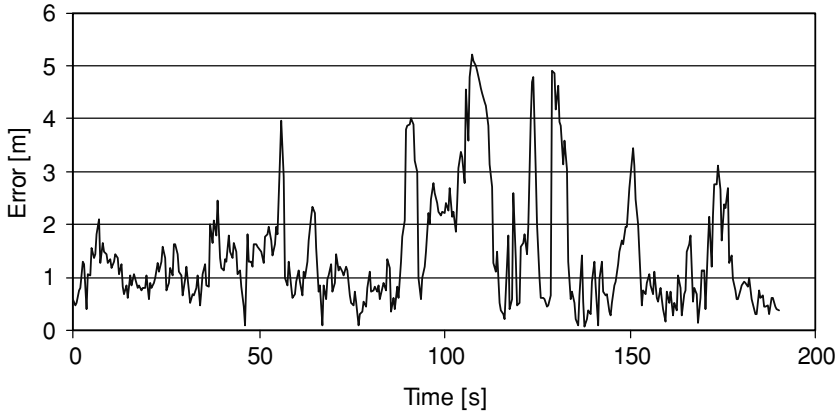

Fig. 11 Internal rotation model distance error vs. simulation time

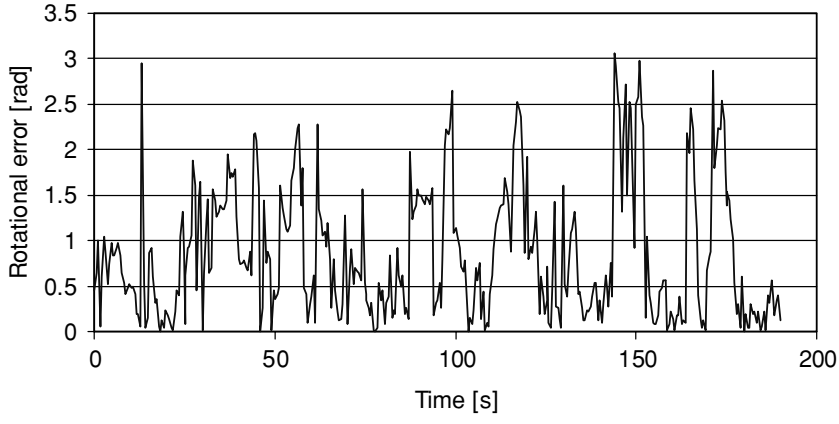

Fig. 12 Internal rotation model distance error vs. simulation time

becomes small), the error is again reduced to an acceptable level. Additional sensors (as outlined in Section 5.1) are expected to increase the precision of the estimate of the user's velocity and provide sub-meter location precisions.

4.3. Location estimation walk-trough studies

Our simulation program has been modified not to obtain the current RSSI reading from the wireless map but from the wireless interface (i.e., the Orinoco Wi-Fi card). Determining the error of the estimation is more problematic in a walk-through study, since the exact physical location of the user is hard to resolve. We have taken an approach where a simulated user (from the same simulation as before) indicates to the person performing the walk-through where she should be located. Thus, we can use the same movement paths for the walk-through as we used for the simulation. The error obtained during the walk-through thus will contain an error term that depends on the user's perception as to where she should be compared to her actual physical location. The

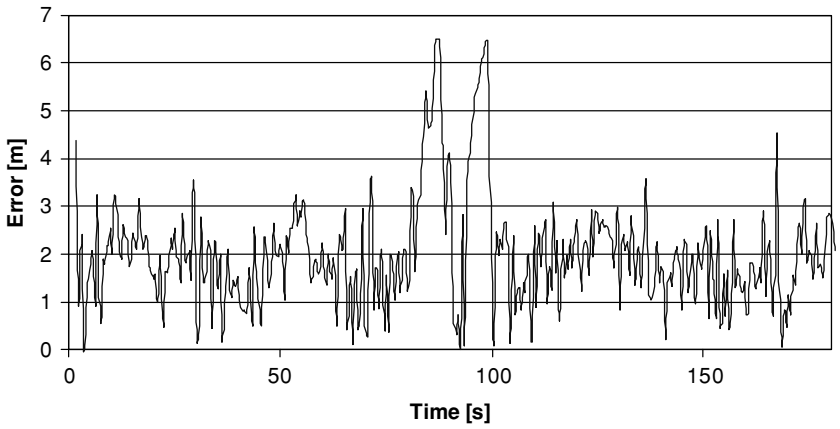

Fig. 13 Distance error vs. simulation time in a real walkthrough

system model used for this experiment resembled the model used in the 4-D experiments. Figure 13 shows the distance error obtained throughout the walk-through. At around 80 s when the user goes to the aisle of the house (where the power map is the most symmetric), the error becomes large for a longer time but settles down after the user has moved sufficiently far from the access point (away from the point of symmetry). A better (less symmetric) selection of the access point's location may make this problem less significant. The average error observed during the walk-trough was 2.1 m. Other than for the previously mentioned short period and for very short transients when changing rooms, the estimated location was always in the same room as the user, thus the estimate had a sub-room precision.

5. Conclusions

Location awareness for wireless, mobile computing devices has the potential of being an important enabler of pervasive or ubiquitous computing, intelligent environments, personal navigation, personal security, prompt healthcare, and entertainment. Similarly, information on the physical location of mobile nodes can greatly help in urban search and rescue missions, as well as enable geographical routing in ad hoc multi-hop networks. To facilitate location awareness for a vast number of devices and with minimal cost, it is important that localization can be performed using minimal infrastructure and with the signals generally available in today's networks.

In this paper we presented an approach for localization that uses only existing RSSI measurements and a map of the expected wireless power measurements in the environment to estimate the position of a mobile node equipped with a wireless network card. To facilitate this we also present a multi-step technique that permits the construction of the wireless power map with a significantly reduced requirement for actual measurements. As opposed to previous approaches, the techniques introduced here do not require an excessively human labor intensive model construction phase and can operate successfully with minimal infrastructure. To achieve the latter and to avoid the problems of discretization, prohibitive computational complexity, and unrealistic, uni-modal assumptions about the probabilistic location estimates, we use Monte Carlo filtering techniques to efficiently estimate the multi-modal distribution of mobile node locations.

To demonstrate the performance and applicability of the techniques, experiments were performed in a home environment with a single access point. First, a wireless power map was constructed using the presented ray-tracing technique and compared to the actual readings in the home. This comparison showed that the constructed map is within approximately 2 dB of the actual measurements. Using this map, three movement models were implemented in the Monte Carlo sampling-based location estimators and simulation studies were performed to evaluate the precision of the filters' estimates for the location of a user that moves within the home. These experiments showed that the filters were successful at estimating the mobile node's location even though the available RSSI data from a single access point is highly ambiguous. It has been demonstrated that even a simple movement model can produce results with an average precision around 1 m. To facilitate a more user friendly system model we have extended the framework to compensate for situations where the user turns away from the access point. Our results show that the error of the location estimate does not significantly increase in such situations.

We have also performed a real-life walk-through of the house used in our previous examples. Our results indicated that sub-room precision of the estimates is achievable even in situations where the divergence from the calculated power map is more than 3 dB. This illustrates that the presented localization technique, together with the wireless power mapping approach, have the potential to provide location awareness to existing wireless devices without the need for aug-

mentation of the existing infrastructure. *We consider showing that sub-room precision positioning is possible with RSSI readings from a single access point as our major contribution.*

5.1. Future work

To further improve the performance of the location estimation techniques, a number of possible extensions will be studied. In particular, more advanced motion models that take advantage of additional sensors and that *integrate* Kalman filtering techniques [8] within the Monte Carlo sampling framework will be investigated. We are also investigating the fusion of readings from other sensors in the model, which would facilitate taking readings not only of the location of the user but also of it derivatives (velocity, acceleration, jerk); particularly we are interested to fuse reading from inexpensive accelerometers into the model and expect to obtain sub-meter precision estimates. Another important aspect of future work is the study of different techniques for the extraction of a location estimate from the distribution. While the maximum probability region-based approach used in this paper led to good results, other techniques such as maximum-likelihood and clustering based on self-organizing maps will be researched to optimize the precision of the location estimates.

Acknowledgment We would like to express our gratitude to Vinay Seshadri for his help with the graphical user interface.

References

1. R. Azuma, Tracking requirements for augmented reality, Communications of the ACM 36(7) (July 1997).
2. P. Bahl and V.N. Padmanabhan, RADAR: An in-building RF-based user location and tracking system, in: *Proceedings of IEEE Infocom 2000*, Tel Aviv, Israel, (March 2000) Vol.2, pp.775–784.
3. P.J. Brown, J.D. Bovey and X. Chen, Context-aware applications: From the laboratory to the marketplace, IEEE Personal Communications Magazine 4 (Oct. 1997) 58–64.
4. P. Castro, P. Chiu, T. Kremenek and R. Muntz, A probabilistic room location service for wireless networked environments, *Ubiquitous Computing 2001* (Sept. 2001).
5. S.K. Das, D.J. Cook, A. Bhattacharya, E.O. Heierman III and T.-Y. Lin, The role of prediction algorithms in the mavHome smart home architecture, *IEEE Personal Communications Special Issue on Smart Homes* (2003) (To appear).
6. A. Doucet, N. de Freitas and N. Gordon (Eds.), *Sequential Monte Carlo Methods in Practice*, Springer-Verlag (2001).
7. D. Fox, W. Burgard, F. Dellaert and S. Thrun, Monte carlo localization: Efficient position estimation for mobile robots, in: *Proceedings of the National Conference on Artificial Intelligence*, Orlando, FL (1999).
8. A. Gelb (Ed.), *Applied Optimal Estimation*, MIT Press (1974).
9. I.A. Getting, The global positioning system, IEEE Spectrum 30 (Dec. 1993) 36–47.
10. N. Gordon, Bayesian methods for tracking, PhD thesis, University of London (1993).
11. M. Hassan-Ali and K. Pahlavan, Site-specific wideband and narrowband modeling for indoor radio channel using ray-tracing, in: *Proceedings of the PMIRC'98*, Boston, MA (1998).
12. M. Hassan-Ali and K. Pahlavan, A new statistical model for site-specific indoor radio propagation prediction based on geometric optics and geometric probability, IEEE Transactions on Wireless Communications 1(1) (Jan. 2002) 112–124.
13. J. Hightower and G. Borriello, Location systems for ubiquitous computing, IEEE Computer Magazine (Special Issue on Location Aware Computing) 34(8) (Aug. 2001) 57–66.
14. M. Isard and A. Blake, Contour tracking by stochastic propagation of conditional density, *European Conference on Computer Vision*, Cambridge, UK (1996) pp. 343–356.
15. S. Kirkpatrick, C.D. Gelatt Jr. and M.P. Vecchi, optimization by simulated annealing, Science 220 (4598) (May 1983) 671–680.
16. J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale and S. Shafer, Multi-camera, multi-person tracking for easy Living, in: *Proceedings of the 3rd IEEE International Workshop on Visual Surveillance*, Piscataway, NJ (2002) pp. 3–10.
17. A.M. Ladd, K. Bekris, A. Rudys, G. Marceau, L.E. Kavraki and D.S. Wallach, Robotics-based location sensing using wireless ethernet, in: *Proceedings of the 8th ACM MobiCom*, Atlanta, GA (Sept. 2002) pp. 227–238.
18. T. Liu, P. Bahl and I. Chlamtac, A hierarchical position-prediction algorithm for efficient Management of resources in cellular networks, in: *Proceedings of the IEEE GLOBECOM '97*, Phoenix, AZ, 2 (Nov. 1997) pp. 982–986.
19. N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller and E. Teller, Equation of state calculation by fast computing machines, Journal of Chemical Physics 45 (1953) 1087–1091.
20. R.J. Orr and G.D. Abowd, The smart floor: A mechanism for natural user identification and tracking, in: *Proceedings of the Conference on Human Factors in Computing Systems*, Hague, Netherlands (April 2000) pp. 1–6.
21. N.B. Priyantha, A. Chakraborty and H. Balakrishnan, The cricket location-support system, in: *Proceedings of the 6th ACM MobiCom*, Boston, MA (Aug. 2000) pp. 32–43.
22. T. Roos, P. Myllymaki, H. Tirri, P. Misikangas and J. Sievanen, A Probabilistic Approach to WLAN User Location Estimation, International Journal of Wireless Information Networks 9(3) (July 2002) 155–164.
23. S. Russel and P. Norvig, *Artificial Intelligence—A Modern Approach*, Prentice Hall, NJ (1995).
24. A. Smailagic, D.P. Siewiorek, J. Anhalt, D. Kogan and Y. Wang, Location sensing and privacy in a context-aware computing environment, IEEE Wireless Communications Magazine (Oct. 2002) 10–17.
25. S. Tekinay (Ed.), IEEE Communications Magazine, *Special Issue on Wireless Geolocation Systems and Services* (April 1998).
26. S. Thrun, Particle filters in robotics, in: *Proceedings of the 17th Annual Conference on Uncertainty in AI (UAI)*, Edmonton, Canada (Aug. 2002).
27. T. Tsukiyama, Global navigation system with RFID Tags, in: *Proceedings of the SPIE Mobile Robots Conference*, Newton, MA (Oct. 2001) pp. 256–264.
28. R. Want, A. Hopper, V. Falco and J. Gibbons, The active badge location system, ACM Transactions on Information Systems 10(1) (Jan. 1992) 91–102.

29. M. Weiser, The computer for the 21st century, Scientific American (Sept., 1991) 94–104.
30. http://www.xfig.org
31. M. Youssef, A. Agrawala, A.U. Shankar and S.H. Noh, A probabilistic clustering-based indoor location determination System, Tech. Report, University of Maryland CS-TR 4350 (March 2002) http://www.cs.umd.edu/Library/TRs/

Gergely V. Záruba is an Assistant Professor of Computer Science and Engineering at The University of Texas at Arlington (CSE@UTA). He has received the Ph.D. degree in Computer Science from The University of Texas at Dallas in 2001, and the M.S. degree in Computer Engineering from the Technical University of Budapest, Department of Telecommunications and Telematics, in 1997. Dr. Záruba's research interests include wireless networks, algorithms, and protocols, performance evaluation, current wireless and assistive technologies. He has served on many organizing and technical program committees for leading conferences and has guest edited journals. He is a member of the IEEE and its Communications Society.

Manfred Huber is an Assistant Professor of Computer Science and Engineering at The University of Texas at Arlington (CSE@UTA). He received his M.S. and Ph.D. degrees in Computer Science from the University of Massachusetts, Amherst in 1993 and 2000, respectively. He obtained his "Vordiplom" from the University of Karlsruhe, Germany in 1990. Dr. Huber is the co-director of the Robotics and of the Learning and Planning Laboratory at CSE@UTA. His research interests are in reinforcement learning, autonomous robots, cognitive systems, and adaptive human-computer interfaces. He is a member of the IEEE, the ACM, and the AAAI.

Farhad A. Kamangar is a Professor of Computer Science and Engineering at The University of Texas at Arlington (CSE@UTA). He has received the Ph.D. and M.S. degrees in Electrical Engineering from The University of Texas at Arlington in 1980 and 1977 respectively. He received his B.S. degree from the University of Teheran, Iran in 1975. Dr. Kamangar's research interests include image processing, robotics, signal processing, machine intelligence and computer graphics. He is a member of the IEEE and the ACM.

Imrich Chlamtac is the President of CREATE-NET and the Bruno Kessler Professor at the University of Trento, Italy and has held various honorary and chaired professorships in USA and Europe including the Distinguished Chair in Telecommunications Professorship at the University of Texas at Dallas, Sackler Professorship at Tel Aviv University and University Professorship at the Technical University of Budapest. In the past he was with Technion and UMass, Amherst, DEC Research. Dr. Imrich Chlamtac has made significant contribution to various networking technologies as scientist, educator and entrepreneur. Dr. Chlamtac is the recipient of multiple awards and recognitions including Fellow of the IEEE, Fellow of the ACM, Fulbright Scholar, the ACM Award for Outstanding Contributions to Research on Mobility and the IEEE Award for Outstanding Technical Contributions to Wireless Personal Communications. Dr. Chlamtac published close to four hundred refereed journal, book, and conference articles and is listed among ISI's Highly Cited Researchers in Computer Science. Dr. Chlamtac is the co-author of four books, including the first book on Local Area Networks (1980) and the Amazon.com best seller and IEEE Editor's Choice Wireless and Mobile Network Architectures, published by John Wiley and Sons (2000). Dr. Chlamtac has widely contributed to the scientific community as founder and Chair of ACM Sigmobile, founder and steering committee chair of some of the lead conferences in networking, including ACM Mobicom, IEEE/SPIE/ACM OptiComm, CreateNet Mobiquitous, CreateNet WiOpt, IEEE/CreateNet Broadnet, IEEE/CreateNet Tridentcom and IEEE/CreateNet Securecomm conferences. Dr. Chlamtac also serves as the founding Editor in Chief of the ACM/URSI/Springer Wireless Networks (WINET), the ACM/Springer Journal on Special Topics in Mobile Networks and Applications (MONET).