

Reducing the Calibration Effort for Probabilistic Indoor Location Estimation

Xiaoyong Chai, and Qiang Yang, *Senior Member, IEEE*

Abstract—WLAN location estimation based on 802.11 signal strength is becoming increasingly prevalent in today’s pervasive computing applications. Among the well-established location determination approaches, probabilistic techniques show good performance and thus become increasingly popular. For these techniques to achieve a high level of accuracy, however, a large number of training samples are usually required for calibration, which incurs a great amount of offline manual effort. In this paper, we aim to solve the problem by reducing both the sampling time and the number of locations sampled in constructing a radio map. We propose a novel learning algorithm that builds location-estimation systems based on a small fraction of the calibration data that traditional techniques require and a collection of user traces that can be cheaply obtained. When the number of sampled locations is reduced, an interpolation method is developed to effectively patch a radio map. Extensive experiments show that our proposed methods are effective in reducing the calibration effort. In particular, unlabeled user traces can be used to compensate for the effects of reducing calibration effort and can even improve the system performance. Consequently, manual effort can be reduced substantially while a high level of accuracy is still achieved.

Index Terms—Location estimation, 802.11 signal strength, Bayesian methods, interpolation, Hidden Markov Model, EM

I. INTRODUCTION

WITH the recent development in mobile computing devices and wireless techniques, location-aware systems are of growing interest and are becoming increasingly popular as well as practical. In building such systems, a fundamental issue is to know the locations of mobile devices in a wireless environment, where an important goal is to increase the accuracy of location estimation. In indoor settings, radio frequency (RF)-based techniques are particularly effective among the existing solutions because they provide ubiquitous coverage and use the inexpensive wireless LAN (WLAN) as the fundamental infrastructure. In recent years, a variety of systems have emerged [1] [2] [3] [4] [5] [6].

Most RF-based systems estimate locations by measuring the strength of the signals propagated from the access points (APs) in the environment. They usually work in two phases [6]: an *offline* training phase and an *online* location estimation phase. In the offline phase, a so-called *radio map* is built. In the online phase, the strength of received signals is used to lookup the radio map to estimate the location. A radio

map is a table of signal strength values received at selected locations from the APs in the area of interest. RF signals provide rich information on locations since the signal strength varies noticeably with the distance between the APs and the physical locations where a wireless device is located. However, location estimation is still a challenging problem because of the non-trivial ways in which signals propagate. A large number of samples are usually required to be collected offline for calibration in order to make the radio map robust to the noisy signals. To obtain the signals, a calibration process is very labor intensive. Let N_m be the number of consecutive signals to form one sample. Because the wireless signals are uncertain in nature and a single scan may probably miss some APs, we should collect several signals and use their mean as a single sample. Let N_s be the sampling time at each location. For a fixed N_m , specifying the sampling time N_s is equivalent to specifying the number of samples collected at each location. Moreover, let N_l be the number of selected (sampled) locations. The amount of calibration effort can thus be quantitatively measured as $N_m \times N_s \times N_l$. Suppose that in a small environment with 100 locations ($N_l = 100$) and 100 samples are collected at each location, one sample per second ($N_s = 100$). Typically several hours are required to collect such an amount of calibration data, let alone the tedious labeling process. The problem is more serious when the area of concern, such as a shopping mall, is very large and where spatially high-density calibration is needed. In this paper, we focus on how to significantly reduce the offline calibration effort while still achieving high accuracy in location estimation through machine learning techniques.

One way to reduce the manual effort is through reducing both N_s and N_l [7]. That is, reduce the sampling time at each location and reduce the number of locations to sample from. However, simply reducing N_s and N_l results in inaccurate radio maps and thus lowers their accuracy in location estimation. In our work, a radio map stores a signal-strength distribution at each sampled location, which statistically measures the strength values of signals that can be received at this location. When N_s and N_l are reduced, signal-strength distributions at the sampled locations are easily screwed and those at the skipped locations (i.e., locations not sampled) are missing. Experiments show that 26% of accuracy is lost when N_s and N_l are both reduced by two-thirds.

To make up for the loss of accuracy and to achieve further improvement, we propose a novel HMM-based algorithm (M^*) that makes use of user traces. While the calibration data can be viewed as labeled samples since the true positions from which these samples are taken are known (labeled), user traces

Xiaoyong Chai is with the Department of Computer Sciences, University of Wisconsin, 1210 W. Dayton St., Madison, WI 53706-1685 USA. Email: xchai@cs.wisc.edu

Qiang Yang is with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong. Email: qyang@cse.ust.hk

are sequences of signal strength recording a user's movement in the environment. These are unlabeled samples because the signal strength received during the movement is recorded without any position label. The most attractive property of user traces is that without the labeling process, sequences of samples can be collected easily and cheaply. Using a hidden Markov model to model user traces, our method provides a way to build a probabilistic location estimation system that requires only a small fraction of the calibration data. Trained from a limited number of labeled samples, the system can gradually improve its performance as more user traces are obtained. Experiments show that when all the calibration data are used, an accuracy of 85% within three meters is obtained using a Bayesian-estimation method. Using 60 unlabeled traces, the same accuracy is achieved requiring only 1/6 of the calibration data as before. Moreover, using 100 traces, the accuracy reaches 86% with only 1/9 of the calibration data. Therefore, the manual effort can be reduced substantially while higher accuracy can be achieved. Furthermore, the method M^* makes the implementation of location tracking straightforward, as we will see in Section IV.

Our HMM-based method requires an initial complete radio map to be known. When N_l is reduced, however, the constructed radio map is incomplete because signal-strength distributions are missing at those skipped locations, as we will see in Section II-C. In our work, a novel interpolation method (M^+) is developed as a pre-processing step of M^* . To patch an incomplete radio map, the method M^+ interpolates signal-strength distributions at one location from those at neighboring locations. Moreover, M^+ is also able to reinforce an inaccurate radio map and thus improve its performance in location estimation. Experiments show that using M^+ , we can achieve about 5% improvement in accuracy when all the locations are sampled and reduce the loss by about 10% when the number of sampled locations N_l is reduced by two-thirds. Comparison between M^+ and a kernel-based interpolation method [7] is made in Section V-C.3.

Our main contribution is to exploit unlabeled trace data and use an EM-based learning algorithm to supplement a limited number of calibrated data for accurate location estimation in a wireless environment. Using this method, a location-estimation system can be initialized from a limited number of sampled data and gradually improve its performance by using more and more unlabeled traces. As a consequence, offline manual effort can be reduced substantially. A by-product is that location tracking can be performed as a filtering process through the learned hidden Markov model. We evaluate our methods by conducting experiments in a real-world indoor wireless environment.

The rest of the paper is organized as follows. In Section II, we introduce the problem of location estimation based on 802.11 signal strength. In particular, we present a Bayesian-estimation method in detail. In Section III, we present our main contribution in this paper — an EM-based learning algorithm that explores unlabeled trace data. In Section IV, we study the problem of location tracking and present an HMM filtering method. Extensive experimental evaluation of our proposed methods is shown in Section V and we conclude the paper in

Section VI.

II. LOCATION ESTIMATION BASED ON 802.11 SIGNAL STRENGTH

A. Overview of Previous Work

In general, location estimation can be classified into two categories: *deterministic* techniques and *probabilistic* techniques. Deterministic techniques [1] [2] [8] use deterministic inference methods to estimate a user's location, such as Triangulation and K-nearest neighbor averaging (KNN). The RADAR system [1] [2], one of the pioneering and most comprehensive work using signal-strength measurements, is based on KNN to infer a user's location. It maintains a radio map with which each online signal-strength measurement is then compared. The coordinates of the best K location matches are averaged to give an estimation.

Probabilistic techniques [3] [4] [6] [9] [10] [11] form the second category. They are also called distribution-based techniques since they store the signal-strength distributions from the APs as the content of a radio map. In contrast to the first category, in the second category, probabilistic inference methods are used to estimate a user's location. In [6], locations in the area are pre-clustered into groups so as to reduce the computational cost of searching the radio map. In [9], correlation among consecutive samples from the APs is introduced to enhance the system performance. In [10], a perturbation technique is used to compensate small-scale variation in signal strength. Furthermore, in [3] and [4], spatial and motion constraints are applied in a postprocessing step to refine the estimation. The core to all these techniques is the use of Bayesian inference to compute the posterior probabilities over locations.

However, compared with the various techniques on location estimation, in previous literature, little attention has been paid to the issue of reducing the calibration effort. To the best of our knowledge, [7] [12] are among the only work that explicitly considers minimizing the calibration effort for an indoor 802.11 location estimation system. In [7], the authors experimentally studied the impact of (1) shortening the time spent at each calibration location and (2) skipping some of the locations altogether on the estimation accuracy. In the following, we refer to these methods as M_1 , reducing the sampling time at each location, and M_2 , reducing the number of locations sampled, respectively. Concerning the second situation where some locations are skipped during calibration, they represented locations as functions of signal strength and by formulating the problem as one of interpolation, they showed that a significant fraction of calibration locations can be skipped. In our work, we not only consider how to apply interpolation to patch an incomplete radio map with skipped calibration locations, but also consider how to use interpolation to further reinforce a complete radio map. Furthermore, to progressively reduce the manual effort, we propose a learning algorithm that can extract calibration information from unlabeled data to supplement a small amount of labeled calibration data. The similar idea of using unlabeled data to improve localization accuracy was also explored in [12]. In their work,

Bayesian network models, which embody extant knowledge about Wi-Fi signals as well as physical constraints of the environment, were constructed. The authors explored the ideas that prior knowledge may provide sufficient constraints to obviate the need to know the actual locations of the training data. As a result, they proposed a hierarchical Bayesian model whose parameters can be learned from unlabeled data. However, the low calibration requirement comes at the cost of resolution: predictive error distance is above ten feet. We observe that trace data record consecutive signal measurements during a user's movement. Such sequential data provide not only spatial but also temporal correlations that we can explore. Thus, we focus on unlabeled trace data in this work. The similar idea of using unlabeled trace data to improve localization accuracy was also used in [13]. By assuming piecewise linear Gaussian distributions over locations, they employed a version of Monte Carlo localization algorithm for tracking people. Unlabeled traces are used to tune a motion model so as to adapt it to individual persons, exploiting regularities when a person navigates the environment. However, directly refining the radio map was not considered in their work.

B. Noisy Characteristics of Wireless Channel

The IEEE 802.11b standard works over the radio frequencies in the 2.4 GHz band. The standard is widespread because the band is license-free at most places around the world. It is also attractive because the RF-based techniques are popular and inexpensive, providing much ubiquitous coverage and requiring little overhead.

A WLAN and a wireless device held by a user have different functionality: APs in the WLAN broadcast signals and the wireless device acts as a sensor which senses the location by analyzing the signals received. Although signal strength varies noticeably with the distance between APs and the wireless device, accurate location estimation using measurements of signal strength is still a difficult task due to the *noisy characteristics* of signal propagation. Subject to reflection, refraction, diffraction and absorption by structures and even human bodies, signal propagation suffers from severe multi-path fading effects in indoor environments [14]. A transmitted signal can reach the device through different paths, each having its own amplitude and phase. These different components combine and reproduce a distorted version of the original signal. Moreover, changes in environmental conditions, such as temperature or humidity, also affect the signals to a large extent.

Fig. 1 gives a typical example of a normalized histogram of signal strength received from an AP at a fixed location. Several hundreds of measurements were sampled to construct the histogram. It is clear from the figure that even at a fixed location, the signal strength received from the same AP varies with time. Furthermore, the number of APs covering a location also varies with time. As shown in Fig. 2, not only the number of APs changes over time, the group of APs detectable at the location also changes as well, as indicated by the numbers beside each point. For example, the fourth sample in Fig. 2 contains signals from AP1, AP3, AP5, AP6 and AP7, while the fifth sample contains signals from AP4 and AP6.

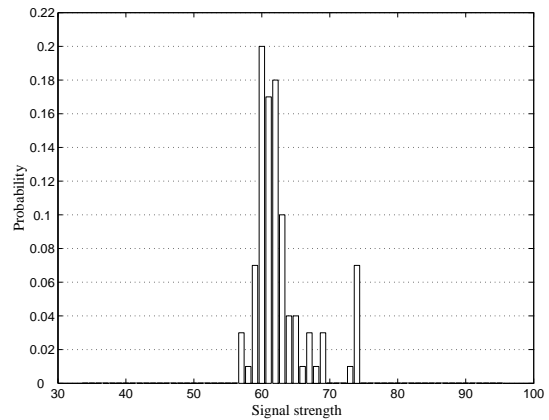


Fig. 1. Signal-strength distribution at a fixed location

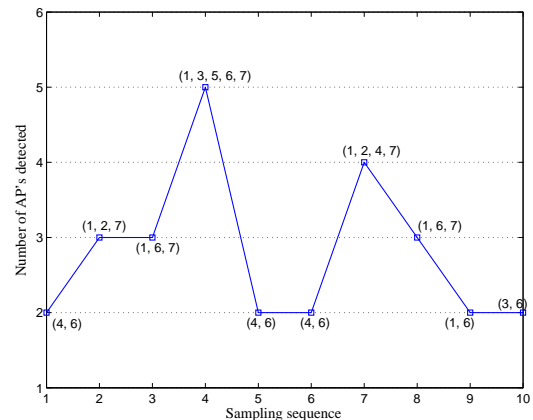


Fig. 2. Variation of AP coverage over a fixed location

C. Background: Probability-Based Location Estimation

Since our work lies in the category of probabilistic techniques, in this section we introduce the Bayesian framework of location estimation. In general, an estimation is represented as a probability distribution over all the locations in the area of interest. The Bayesian inference method is used to compute the distribution conditioning on the observed signal strength. The estimated location is the one with the maximum probability in the distribution.

Formally, we model the physical area of interest as a finite location-state space $\mathbb{L} = \{l_1, \dots, l_n\}$. The location-state space \mathbb{L} is defined as a set of physical locations with x- and y-coordinates:

$$\mathbb{L} = \{l_1 = (x_1, y_1), \dots, l_n = (x_n, y_n)\}.$$

An example, each location l can represent a grid cell in the hallways in an environment.

All possible signal-strength values are modelled as a finite observation space $\mathbb{O} = \{o_1, \dots, o_m\}$. An observation o in the observation space \mathbb{O} is a set of k signal-strength measurements received from k different access points where k is the number of APs which have the strongest signals. Normally, in an environment, signals from many APs are detectable somewhere, either located within the area of concern or located outside. A subset of k APs is selected in order to reduce the

computational cost. Thus, each observation o is represented as a vector of k pairs as follows:

$$o = \langle (b_1, ss_1), \dots, (b_k, ss_k) \rangle,$$

where b_i represents the i th AP scanned and ss_i is the signal strength received from b_i .

In the offline training phase, calibration data are collected at each location l_i . That is, signal-strength measurements are recorded at each location as observations. After the data are collected, we build a histogram of observation for each AP b_j at each location l_i . This is done by constructing the conditional probability $Pr(ss_j|b_j, l_i)$, which is the probability that AP b_j has the signal-strength measurement ss_j at location l_i . By making an independence assumption among signals from different APs, we multiply the probabilities of all APs to obtain the conditional probability of receiving a particular observation o at location l_i as follows:

$$Pr(o|l_i) = \prod_{j=1}^k Pr(ss_j|b_j, l_i), \quad (1)$$

which is exactly the content stored in a radio map. In the online phase, a posterior distribution over all the locations is computed using Bayes rule:

$$Pr(l_i|o^*) = \frac{Pr(o^*|l_i)Pr(l_i)}{\sum_{i=1}^n Pr(o^*|l_i)Pr(l_i)}, \quad (2)$$

where o^* is a new observation obtained. $Pr(l_i)$ encodes prior knowledge about where a user may be. $Pr(l_i)$ can be set as a uniform distribution, assuming every position is equally likely. The estimated location l^* is the one which obtains the maximum value of the posterior probability:

$$l^* = \arg \max_{l_i} Pr(l_i|o^*). \quad (3)$$

The advantage of the above Bayesian-estimation method is that it treats uncertainty in location sensing in a statistical manner. Noise in signal propagation is captured by these conditional probabilities. For example, the possibility of obtaining signal-strength measurement ss_j from AP b_j at location l_i is represented by $Pr(ss_j|b_j, l_i)$. Since location estimates are represented probabilistically (i.e., by the posterior distribution $Pr(l_i|o^*)$ in (2)), the Bayesian method preserves complete information contained in the signals and all this information is readily available for use, as we will see in Section III. Moreover, by making an independence assumption¹ among signals from different APs, the Bayesian method is robust to the situations where signals from some APs are missing or some APs are removed from the environment. In either case, the conditional probabilities $Pr(ss_j|b_j, l_i)$ corresponding to those absent APs can be just omitted in the computation of $Pr(o^*|l_i)$ and subsequently $Pr(l_i|o^*)$ in (1) and (2).

In our work, the performance of location estimation is measured using the notion of *accuracy*. Let $D_i = \{O_i, L_i\}$ be a data point where O_i is a signal-strength measurement and L_i is the corresponding grid label. Let D be a given error distance

threshold measured in meters between two physical locations. Suppose that given O_i , a location-estimation system predicts a location grid label L_j . If the distance between L_i and L_j is less than the error distance D , then it is called a correct prediction. Otherwise, it is considered as an error. For a test data set T_s consisting of N data points, if the location-estimation system makes C correct predictions, then the system is said to have an accuracy, within the error distance of D , of

$$Acc = \frac{C}{N}. \quad (4)$$

III. USING UNLABELED TRACES TO REDUCE THE OFFLINE CALIBRATION EFFORT

A. Overview

As discussed before, the amount of calibration effort is determined by the following factors:

- N_m : the number of signals that are averaged to form one sample. Because the wireless signals are uncertain in nature, we need to collect several signals and use their mean as a single sample;
- N_s is the sampling time spent at each location to collect signal samples. For a fixed N_m , specifying the sample time N_s is equivalent to specifying the number of samples collected at each location; and
- N_l is the number of locations to sample from.

Therefore, the offline calibration effort is $O(N_m \times N_s \times N_l)$.

In our work, we fix the number of signals collected in one second to be a constant 10, and vary the other two variables. Each sample consists of the average value from 10 signals received at a location. In particular, we wish to study how to enhance the location-estimation accuracy while allowing low calibration effort. A reduction in calibration effort then corresponds to reductions in both the number of samples taken at each location (N_s) and the number of locations to sample from (N_l). As we will show later experimentally, reducing N_l and N_s will result in a highly incomplete and inaccurate radio map. To patch up for the missing parts of the radio map and to improve its robustness, we will use interpolation and exploit the extra unlabeled user traces which are much easier to obtain.

More specifically, user traces are sequences of signal-strength measurements recording a user's movements in the environment. The main difference between calibration data and user traces lies in whether the true position where an observation is taken is known or not. Each sample of the calibration data has its location label, and therefore it is recorded as a pair (o, l) , where l is the location at which o is taken. On the other hand, a user trace has no location label assigned when recorded. It appears as a sequence of observed samples $\langle o^1, o^2, \dots, o^m \rangle$, where the superscript m is the time index. Therefore it cannot be used directly for training as the calibration data. While labeling signal samples with the correct locations is time-consuming, collecting them is relatively easy. This is especially true when samples are collected consecutively as a user is walking around in the environment. Then, an interesting question is how to extract useful information contained in user traces to improve a radio

¹Signal-strength measurements from different APs passed a χ^2 test of independence with a significance level of $\alpha = 0.05$.

map that is built from a limited amount of calibration data. In this paper, we propose a method in which we use a hidden Markov model to model user traces and apply an EM algorithm to improve an inaccurate radio map. We call this method M^* .

B. Modelling User Traces Using Hidden Markov Model (HMM)

We use an HMM to model user traces. HMM is a well-known technique in pattern recognition and has a wide range of applications [15] [16]. In pervasive computing, HMM and its variations have been successfully used in tracking and recognizing human activities [17]. An HMM is a stochastic finite state machine which models a Markov process with parameters. It is termed “hidden” since the internal states of the process are viewed as hidden and only the outputs of the states are observable. In modelling user traces, the underlying process is a user’s sequential changes in location, where the user’s locations are the hidden internal states and the signal-strength measurements are the observations.

For our purpose, an HMM for user-trace modelling is defined as a quintuple $\langle \mathbb{L}, \mathbb{O}, \lambda, A, \pi \rangle$:

- \mathbb{L} : a location-state space – $\mathbb{L} = \{l_1, l_2, \dots, l_n\}$;
- \mathbb{O} : an observation space – $\mathbb{O} = \{o_1, o_2, \dots, o_m\}$;
- λ : a radio map – $\lambda = \{Pr(o_j|l_i)\}$, where $o_j \in \mathbb{O}, l_i \in \mathbb{L}$;
- A : a location-state transition matrix – $A = Pr(l_j|l_i)$, where $l_i, l_j \in \mathbb{L}$;
- π : an initial location-state distribution – $\pi = Pr(l_i)$, where $l_i \in \mathbb{L}$.

The HMM is defined on a location-state space \mathbb{L} and an observation space \mathbb{O} , both of which are given in Section II-C. The radio map λ is a set of conditional probabilities which give the likelihood of obtaining signal-strength measurement $o_j \in \mathbb{O}$ at location $l_i \in \mathbb{L}$. The initial location-state distribution π encodes prior knowledge about where a user may be. Both λ and π are also given in Section II-C. The transition matrix A specifies how a user travels through the state space. While a user can freely navigate the environment, his movement is subject to certain constraints imposed by the environment. For example, he can only walk in hallways or rooms but cannot walk across rooms. Also, the user has limited mobility. That is, he can not move too quickly in an indoor environment, but can only move to nearby locations in consecutive time steps. All this prior information can be encoded into A by a proper $Pr(l_j|l_i)$. In an HMM, λ , A and π are the parameters adjustable. Let $\theta = (\lambda, A, \pi)$ denote an HMM’s model parameter. Given an observed user trace t and a model parameter θ , the well-known Viterbi algorithm [16] can be used to infer the most probable hidden state sequence in the HMM, which is a sequence of a user’s location changes. An illustration of user-trace modelling using HMM is shown in Fig. 3. In the figure, the non-shaded nodes are the hidden location states, and the shaded nodes are the observations.

C. Improving A Radio Map λ Using EM Algorithm

When the calibration data are insufficient, a radio map built from a small number of labeled samples is inaccurate.

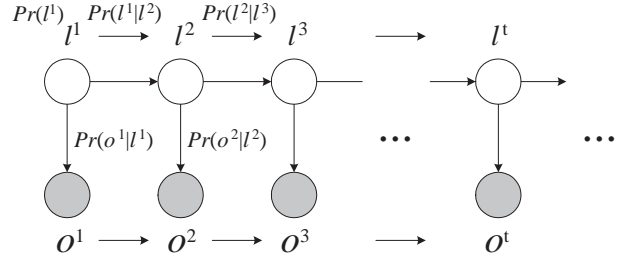


Fig. 3. An illustration of user-trace modelling using HMM

To reduce the calibration effort and in the meantime still achieve good performance, we apply an EM algorithm [18] to improve a radio map using unlabeled traces. Let λ^0 denote an initial radio map which is built from a limited amount of labeled calibration data. In the case that interpolation is used, λ^0 is the resulting interpolated radio map. Let A^0 denote an initial location-state transition matrix and π^0 denote an initial state distribution. Both A^0 and π^0 are set *a priori*. An HMM can then be initialized by the model parameter $\theta^0 = (\lambda^0, A^0, \pi^0)$. Given a set of unlabeled traces T , EM is used to adjust the model parameter $\theta = (\lambda, A, \pi)$ iteratively to find θ^* such that the likelihood $Pr(T|\theta^*)$ is maximized. That is, maximize the likelihood of the traces T generated by λ such that $Pr(T|\theta^*) \geq Pr(T|\theta)$. Here, $Pr(T|\theta)$ is calculated as follows:

$$\begin{aligned} Pr(T|\theta) &= \prod_{t \in T} Pr(t|\theta) = \prod_{t \in T} \sum_q Pr(t|q, \theta) Pr(q|\theta) \quad (5) \\ &= \prod_{t \in T} \sum_q \left(Pr(l^1|\theta) Pr(o^1|l^1, \theta) \right. \\ &\quad \left. \times \prod_{k=2}^{n_t} Pr(l^k|l^{k-1}, \theta) Pr(o^k|l^k, \theta) \right) \quad (6) \end{aligned}$$

In (5) and (6), $t = (o^1, o^2, \dots, o^{n_t})$ is a trace of length n_t and $q = (l^1, l^2, \dots, l^{n_t})$ is a possible location sequence with the same length as t . The likelihood of T (given θ) is the product of the likelihood of each individual trace t (given θ), which is a weighted summation over all possible hidden location-state sequences. As given in (5), $Pr(t|\theta) = \sum_q Pr(t|q, \theta) Pr(q|\theta)$, where $Pr(q|\theta)$ gives the probability of q being a user’s true sequence of location changes and $Pr(t|q, \theta)$ is the likelihood of observing the trace t when the user’s movement is q . In (6), $Pr(q|\theta)$ is expanded into $Pr(l^1|\theta) \times \prod_{k=2}^{n_t} Pr(l^k|l^{k-1}, \theta)$, which can be calculated from π and A , and $Pr(t|q, \theta) = Pr(o^1|l^1, \theta) \times \prod_{k=2}^{n_t} Pr(o^k|l^k, \theta)$ can be obtained from λ , where π , A and λ are the three components of the current model parameter θ . Furthermore, θ^* maximizing the likelihood $Pr(T|\theta^*)$ means that the parameter θ^* best explains the signal-strength measurement sequences in the traces.

The EM algorithm is an iterative process through two steps: an Expectation step (E-step) and a Maximization step (M-step). A standard method is to maximize a so-called *Q-function* during the iterations. The Q-function is defined as follows:

$$Q(\theta, \theta^k) = \sum_{t \in T} \sum_q \log Pr(t, q|\theta) Pr(t, q|\theta^k), \quad (7)$$

where θ^k is the parameter obtained after the k th iteration and θ is a free parameter. In the E-steps, the Q-function (7) is calculated; in the M-steps, maximization is taken over θ and the model parameter is updated from θ^k to θ^{k+1} :

$$\theta^{k+1} = \arg \max_{\theta} Q(\theta, \theta^k).$$

In particular, the M-step in the $(k+1)$ th iteration for updating the radio map $\lambda^{k+1} = \{Pr(o_j|l_i)^{(k+1)}\}$ is as follows:

$$Pr(o_j|l_i)^{(k+1)} = \frac{\sum_{t \in T} \sum_{s=1}^{t_n} Pr(t, l^s = l_i | \theta^k) \delta(o^s, o_j)}{\sum_{t \in T} \sum_{s=1}^{t_n} Pr(t, l^s = l_i | \theta^k)}, \quad (8)$$

where $\delta(x, y)$ is a function such that $\delta(x, y) = 1$ if $x = y$, otherwise $\delta(x, y) = 0$. During the iterations, a sequence of model parameters $\theta^0, \theta^1 \dots \theta^*$ is generated, where $\theta^0 = (\lambda^0, A^0, \pi^0)$ is the initial parameter and $\theta^* = (\lambda^*, A^*, \pi^*)$ is the converged parameter obtained when iterations terminate. The EM algorithm guarantees that $Pr(T|\theta^{k+1}) \geq Pr(T|\theta^k)$ and the parameter converges to θ^* when the likelihood does not change in consecutive iterations. Interested readers please refer to [18]. Therefore, starting from an initially inaccurate radio map λ^0 , EM tunes it to best explain the set of unlabeled traces. Meanwhile, these traces are implicitly used to improve the radio map and useful information in T is thus extracted and absorbed. When a new radio map λ^* is learned, it can be used to substitute the initial map λ^0 for location estimation. To avoid the bias towards unlabeled traces, we take an additional step. We use λ^* to label the traces to get a new set of labeled samples. This new set of samples, together with the original calibration data, produces a modified radio map λ' , which is then used in the online location-estimation phase. Fig. 4 illustrates the EM algorithm from another angle: an E-step labels the traces with a current radio map, and then an M-step updates a radio map based on the labeled traces.

Let N_T be the number of traces, l be the average trace length, and N_L be the number of locations. The time complexity of EM training is $N_{iter} \times O(N_T N_L^2 l)$, where N_{iter} is the number of iterations ($N_{iter} < 10$ in our experiments). The training time is quadratic in the number of locations, but it can be done offline.

D. Using Interpolation to Learn an Initial Radio Map

The HMM-based algorithm presented above requires an initial radio map λ_0 to be known. In this section, we discuss how to re-construct such an initial radio map when some locations are skipped, that is, when N_l is reduced.

When N_l is reduced, the constructed radio map is incomplete because signal-strength distributions are missing at those skipped locations. A consequence of this reduction is that the HMM-based method cannot be used anymore, since the method requires a complete radio map to be available as an initial distribution λ_0 . Thus we need to patch the incomplete radio map. Our approach to making up for the missing distributions is to apply a linear interpolation method, which we denote as M^+ . The idea is to construct them from available ones at sampled locations. Similar ideas of treating a location as a function of signal strength were used in [7] [19].

However, small-scale variations prevent us from sampling and interpolating at close-by locations, where calibration locations are at sub-meter range (at the order of wave-length) [10]. Thus, we only consider using interpolation when calibration locations are at least three meters apart.

Instead of sampling at each location in \mathbb{L} , calibration data are only collected at a subset of locations $\mathbb{L}_1 \in \mathbb{L}$. The rest locations \mathbb{L}_2 ($\mathbb{L}_2 = \mathbb{L} - \mathbb{L}_1$) are skipped. Signal-strength distributions at locations in \mathbb{L}_1 can be built in the same way as introduced in Section II-C. That is, construct conditional probabilities $Pr(o|l_i)$ from the samples collected at l_i ($l_i \in \mathbb{L}_1$). The missing distributions at the locations in \mathbb{L}_2 are then interpolated to complete the radio map. An illustration of the interpolation method is shown in Fig. 5, where l_a, l_b ($\in \mathbb{L}_1$) are the locations directly sampled and l_c ($\in \mathbb{L}_2$) is one of the locations skipped between l_a and l_b . In the figure, d_1 and d_2 are the distances from l_c to l_a and l_c to l_b , respectively. We interpolate the signal-strength distribution at l_c from those at l_a and l_b as follows ((9)):

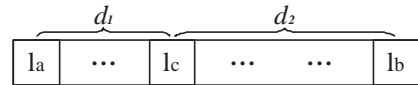


Fig. 5. An illustration of interpolation, where $Pr(o|l_c)$ is interpolated from $Pr(o|l_a)$ and $Pr(o|l_b)$

$$Pr(o_j|l_c) = \frac{d_2}{d} Pr(o_j|l_a) + \frac{d_1}{d} Pr(o_j|l_b), \quad o_j \in \mathbb{O} \quad (9)$$

where $d = d_1 + d_2$. The idea of (9) is to exploit the similarity between $Pr(o_j|l_c)$ and $Pr(o_j|l_a)$ to rebuild the signal-strength distribution at the skipped location l_c . The difference between the two distributions depends on the distance between l_c and l_a . This is intuitive since the closer the two locations are, the more similar the signals they received. In (9), such dependence is assumed to be linear and the coefficients $\frac{d_1}{d}$ and $\frac{d_2}{d}$ are used to normalize the interpolated distribution $Pr(o_j|l_c)$.

Compared to our linear interpolation method, more complex nonlinear relationships can also be assumed. For example, the work of [7] used a kernel-based method to capture the similarity between the signals from two neighboring locations. In Section V-C.3, we will compare this method with ours empirically.

Since the signal-strength distributions at those skipped locations in \mathbb{L}_2 are now interpolated, we also refer to these locations as *interpolated* locations.

IV. LOCATION TRACKING

A potential application of the HMM-based method M^* is object tracking. Tracking is the process of continuously estimating a people's trajectory as he moves. One feasible solution is to treat a sequence of sensor readings as a set of independent observations and then apply the basic location estimation technique repeatedly. In doing so, however, the sequential nature of the readings is lost. In particular, motion constraints confine possible location changes. (This is also our motivation of using HMM and the EM algorithm to improve

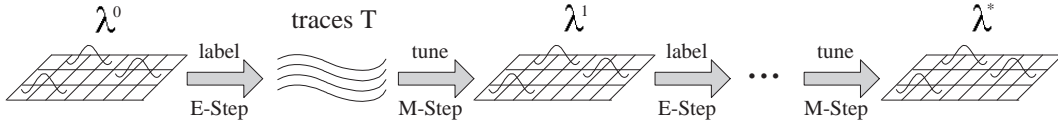


Fig. 4. An illustration of improving a radio map using the EM algorithm

a radio map.) A better solution is cast the tracking problem as a filtering process through which we can estimate the states of a dynamic system from noisy observations along with the time. The same idea was proposed in [4], where a different term, *sensor fusion*, was used.

Bayesian-filter techniques provide a powerful tool for tracking [11] [20]. Among various candidates, we adopted the HMM filter, also known as a grid-based method, for two reasons. First, the location state space is discrete and consists of a finite number of states. These are sufficient conditions for tractable computation. Second, our M^* method tunes not only the radio map λ but also the initial state distribution π and the location-state transition matrix A . These are exactly the model parameters of an HMM filter.

Given a sequence of observations $o^{1:t} = \langle o^1, o^2, \dots, o^t \rangle$ up to time t , $Pr(l^t | o^{1:t})$, the degree of belief in the location state l at time t , is recursively calculated. The probability density function $Pr(l^t | o^{1:t})$ is obtained recursively in two stages:

- **Prediction:** Suppose that $Pr(l^{t-1} | o^{1:t-1})$ is available, the prediction stage is to compute $Pr(l^t | o^{1:t-1})$ as follows:

$$\begin{aligned} Pr(l^t | o^{1:t-1}) &= \sum_{l^{t-1}} Pr(l^t | l^{t-1}, o^{1:t-1}) \times Pr(l^{t-1} | o^{1:t-1}) \\ &= \sum_{l^{t-1}} Pr(l^t | l^{t-1}) \times Pr(l^{t-1} | o^{1:t-1}) \quad (10) \end{aligned}$$

Note that $Pr(l^t | l^{t-1}, o^{1:t-1}) = Pr(l^t | l^{t-1})$ under the first-order Markovian assumption, and $Pr(l^t | l^{t-1})$ is given by the location-state transition probability matrix $\{Pr(l_j | l_i), l_i, l_j \in \mathbb{L}\}$.

- **Update:** At time step t , a new observation o^t is available, and it can be used to update the belief using Bayes' rule:

$$\begin{aligned} Pr(l^t | o^{1:t}) &= \frac{Pr(o^t | l^t) \times Pr(l^t | o^{1:t-1})}{Pr(o^t | o^{1:t-1})} \\ &= \frac{Pr(o^t | l^t) \times Pr(l^t | o^{1:t-1})}{\sum_{l^t} Pr(o^t | l^t) \times Pr(l^t | o^{1:t-1})} \quad (11) \end{aligned}$$

Intuitively, we first predict where a user is (at time step t) based on our knowledge on his previous position (at time step $t - 1$), and then update our prediction based on the observation newly obtained. Subsequently, we can estimate the user's location as the location point which maximizes $Pr(l^t | o^{1:t})$. Therefore, location tracking can operate using our method.

The complexity of filtering is $O(N_L^2)$ operations per time step. When the location-state space is too large to make real-time tracking feasible, fast approximation-based methods such as particle filtering can be used. This is beyond the scope of this paper.

V. EXPERIMENTS

In this section, we evaluate the performance of our proposed methods. First, we study the effects of reducing the sampling time N_s and the number of sampled locations N_l on accuracy. Then, we present the results on using interpolation (M^+) to improve a radio map. After that, we apply the EM algorithm and show the effectiveness of exploiting unlabeled traces to improve the overall location-estimation performance. Finally, the performance of tracking is investigated. In all experiments, we will use the accuracy measure as defined in (4).

A. Comparison Baselines

In this section, we present several baselines that we use as comparison targets for our proposed HMM-based method M^* . In experimental results we subsequently present, we will show that M^* outperforms these methods under various experimental conditions.

1) M_1 : Reducing the Sampling Time at Each Location:

One baseline method is to reduce the calibration effort by reducing the sampling time N_s at each sampled location. This method is simple and straightforward, and it has been proven to be quite effective [7]. Although it is not necessary to spend much time at each location during calibration, it normally requires tens or even hundreds of samples to stabilize signal-strength distributions and reduce the influence of noisy wireless channels. When the calibration data are scarce, only five or ten samples available at each location, the limited samples are not representative enough. This is because a signal-strength distribution constructed from a small number of samples is easily biased towards these samples, that is, these samples gain much higher probability values in the constructed distribution than they do in the actual distribution. As a consequence, when a measurement is obtained online, it can be easily rejected as an outlier only because it does not appear in the training data. Although there are existing techniques to smooth these distributions [21], they are still far from being satisfactory when the training data are extremely insufficient. Our experiments reveal that only 3% of accuracy is lost when N_s is reduced from 60 to 30 at each location, which is a good tradeoff since half the effort can be saved. However, accuracy decreases by 12% when N_s is further reduced to 10. Therefore, reducing N_s has its limitation in achieving significant calibration-effort reduction.

2) M_2 : Reducing the Number of Locations Sampled:

A second baseline method is to reduce the calibration effort by reducing the number of locations N_l at which we collect samples offline. Similar to the method M_1 , this method is also straightforward [7]. By the method M_2 , instead of sampling at each location in \mathbb{L} , we collect samples at a subset of

locations $\mathbb{L}_1 \in \mathbb{L}$ and skip the rest \mathbb{L}_2 ($\mathbb{L}_2 = \mathbb{L} - \mathbb{L}_1$). Signal-strength distributions at locations in \mathbb{L}_1 are built in the same way as introduced in Section II-C. However, signal-strength distributions at the locations in \mathbb{L}_2 are missing since we have no calibration samples directly collected at these locations. Therefore, the resulting radio map is incomplete. Let $r = |\mathbb{L}_1|/|\mathbb{L}|$ denote the sampling ratio, that is, the ratio of sampled locations to all the locations. With the decrease of r , the performance of the incomplete radio map deteriorates, as we will see in Section V-C.2.

3) M^+ : *Using Linear Interpolation to Reconstruct A Radio Map*: A third baseline method is M^+ , which applies linear interpolation as we described in Section III-D. This method can be used even if a radio map is complete, because then it can be used to reinforce it.

More specifically, let S_i denote the set of samples collected at location l_i . An implicit assumption made by the Bayesian-estimation method in constructing a radio map is that locations are independent from one another. That is, S_i is exclusively used to construct the conditional probability $Pr(o|l_i)$ at location l_i . However, signals received at different locations are correlated, especially at the neighboring ones. As in the example shown in Fig. 6, strength values of the received signals at location l_i tend to be between those at location l_{i-1} and those at location l_{i+1} from a statistical perspective. It is thus feasible to use S_{i-1} , S_{i+1} and other sample sets to assist S_i in constructing $Pr(o|l_i)$.

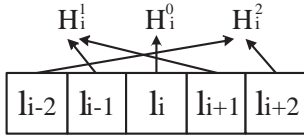


Fig. 6. An illustration of interpolation, where $Pr(o|l_c)$ is interpolated using S_{i-2} , S_{i-1} , S_i , S_{i+1} and S_{i+2}

Suppose that $Pr(o|l_i)$ is constructed from S_i only. We use H_i^0 to denote $Pr(o|l_i)$. As given in (9), we can interpolate another signal-strength distribution at location l_i from those at l_{i-1} and l_{i+1} or those at l_{i-2} and l_{i+2} . The resulting interpolated distributions, denoted as H_i^1 and H_i^2 , are given in (13) and (14) below. H_i^0 , H_i^1 and H_i^2 are three signal-strength distributions that are at the same location l_i and are obtained from different sample sets. They can then be combined to produce an integrated distribution H_i , as given in (15), to substitute H_i^0 . In the equation, $\omega_0 \sim \omega_2$ are the weights associated with these three distributions, where $\omega_0 + \omega_1 + \omega_2 = 1$. Let $W = (\omega_0, \omega_1, \omega_2)$ be a weight vector. Different W specifies different roles of H_i^0 , H_i^1 and H_i^2 in the integrated distribution H_i . Through interpolation, we make S_{i-2} , S_{i-1} , S_{i+1} and S_{i+2} help S_i construct $Pr(o|l_i)$. As we will see in the experiments, interpolation show good performance in reinforcing radio maps, especially when the calibration data are scarce.

$$H_i^0 = Pr(o|l_i), \quad (12)$$

$$H_i^1 = \frac{1}{2}Pr(o|l_{i-1}) + \frac{1}{2}Pr(o|l_{i+1}), \quad (13)$$

$$H_i^2 = \frac{1}{2}Pr(o|l_{i-2}) + \frac{1}{2}Pr(o|l_{i+2}), \quad (14)$$

$$H_i = \omega_0 H_i^0 + \omega_1 H_i^1 + \omega_2 H_i^2. \quad (15)$$

In summary, using interpolation, we can patch a radio map or reinforce one where only $r|\mathbb{L}|$ of the locations are sampled. Essential to the interpolation method introduced in Sections III-D and V-A.3 is the idea of exploiting useful information from local neighborhoods. However, since only the calibration data are used, improvement that can be achieved by method M^+ is still limited. This is especially true when method M_2 is used to reduce the calibration effort, where only a fraction of the locations are sampled directly. As will be shown in Section V-C.2, reducing two-thirds of N_l (i.e., the sampling ratio $r = 1/3$) still incurs a loss of 16% in accuracy even if interpolation is used.

B. Experimental Setup

Our experimental testbed was set up in the office area of the CS Department at Hong Kong University of Science and Technology. The building is equipped with an IEEE 802.11b wireless network in the 2.4 GHz frequency bandwidth. Cisco Aironet 340 access points are deployed. The layout of the floor is shown in Fig. 7. This area has a dimension of 64×50 meters. Experiments were carried out in the four hallways (HW1~HW4) and two rooms (Room1 and Room2) as labeled in the figure. The four hallways measure 19.5, 37.5, 46 and 21 in meters, respectively. To form the location-state space, we modelled the environment as a space of 99 locations, each representing a 1.5×1.5 meter grid cell. The sampling C++ program was run on an IBM 1.29GHz P4 laptop under Windows XP. The sampling rate was set to 10Hz. The wireless card used was Linksys Wireless-B Notebook Adapter supporting up to 11 Mbps transfer rate. We developed an API² to interface the sampling program. Calibration data and unlabeled user traces were collected by two persons across multiple days. Since a single scan may probably miss some APs, we took average over the signal-strength measurements of ten consecutive scans and took it as one sample. For the calibration data, one hundred samples were collected at each location, one sample per second. Thus, the length of the calibration time spent at each location is equal to the number of samples used from that location to construct the radio map. Traces were recorded when a person navigated the environment, walking through the hallways. On average, a trace contains 40 unlabeled samples.

C. Analysis of the Baseline Methods

1) *Testing Baseline Method M_1 : Reducing the Sampling Time*: Experiments were first carried out to study the effect of the varying length of sampling time on accuracy. The entire calibration data set was repeatedly divided into two subsets: a training set Tr and a testing set Ts . Tr comprises the first N_s collected samples at each location, and they were used to construct a radio map. Ts comprises the rest of the samples, and we evaluated the performance of the constructed radio map

²Available at <http://www.cs.wisc.edu/~xchai/Download/wlanapi.zip>.

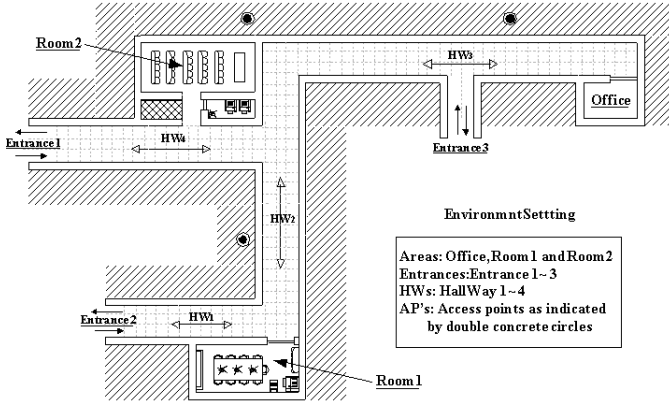


Fig. 7. The layout of the office area of CS Department of Hong Kong University of Science and Technology

by testing it on them. Starting from five samples per location (i.e., $N_s = 5$), we increased the number by five each step to reproduce the effect of gradually increasing the calibration time. The estimation accuracy was measured with an error distance of three meters. The error distance is the distance between the predicted and the actual location points. Results are shown in Fig. 8, with the number of training samples ranging from 5 to 60.

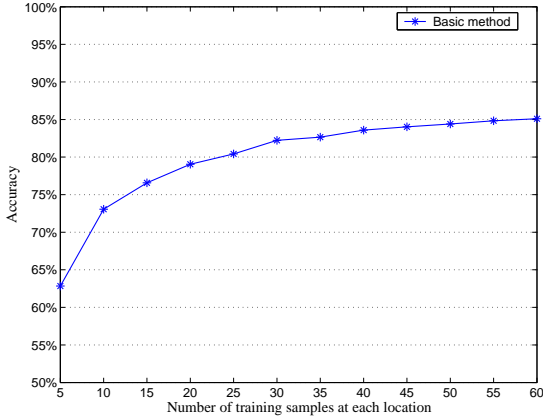


Fig. 8. Accuracy v.s. number of training samples at each location (M_1)

When the training samples are scarce, increasing the amount of calibration data has a significant influence on accuracy. As shown in the figure, the accuracy increases by 10.2% as the number of training samples increases from five to ten. Enhancement is less significant when more training samples are available. Overall, the discrepancy can be as large as 22.3%, accuracy ranging from 62.8% ($N_s = 5$) to 85.1% ($N_s = 60$). As we can see, reducing the sampling time can degrade the system performance significantly. Thus, method M_1 has limited ability in saving the calibration effort.

2) *Testing Baseline Method M_2 : Reducing the Number of Locations Sampled*: Another set of experiments was conducted to examine the effect of reducing the number of locations sampled (N_l). For this purpose, we set up the experiments as follows. Out of the total $|\mathbb{L}|$ (=99) locations in modelling the test environment, we selected $|\mathbb{L}_1|$ locations

by skipping every several locations between them. In the experiments, we varied $|\mathbb{L}_1|$ among three possible values: 50, 31 and 24. Correspondingly, the sampling ratio r ($= |\mathbb{L}_1|/|\mathbb{L}|$) is approximately 1/2, 1/3 or 1/4. Fig. 9 illustrates the case where $r = 1/3$: The 31 locations marked with black squares form \mathbb{L}_1 , and the other 68 skipped locations form \mathbb{L}_2 . In all the three cases, the calibration data at the locations in \mathbb{L}_1 were still used to construct the signal-strength distributions while the data at the locations in \mathbb{L}_2 were no longer used for training but for testing only. As a result, incomplete radio maps were constructed. Given a radio map, we then measured how the location-estimation accuracy is affected.

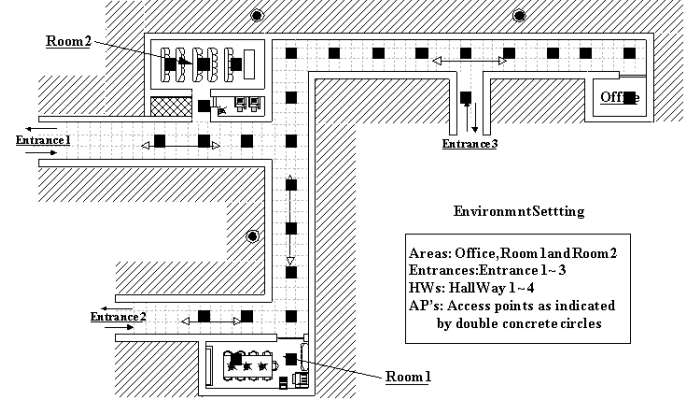


Fig. 9. Layout illustration of reducing the number of locations sampled, where dark dots are the sampled locations.

Fig. 10 shows the effect of reducing the number of locations sampled with varying sampling ratios. For illustration, the factor of reducing the sampling time N_s was also considered. In the figure, the dashed curve is the same as the one in Fig. 8, which is shown for comparison. The curve is denoted as “Basic” since no location is skipped and the radio maps are complete. The other three curves represent the situations where $r = 1/2$, $r = 1/3$ and $r = 1/4$. As we can see from the figure, the more locations that are skipped, the lower the accuracy that the resulting incomplete radio map has. Also, with the increase of the training samples at sampled locations, the accuracy of a radio map increases. This is intuitive because the more training samples that we have at each sampled location, the more accurate signal-strength distributions these locations have. However, the discrepancy between the performance of a complete radio map (i.e., the “Basic” one) and that of an incomplete radio map (e.g., the one with “ $r = 1/2$ ”) is large. For example, when $N_s = 60$, the accuracy of the latter is 18.5% lower than that of the former.

It is also interesting to compare the two methods, M_1 and M_2 , in terms of their effects on accuracy. From Fig. 8 and Fig. 10, we can see that reducing the sampling time is more effective than reducing the number of locations sampled. The accuracy decreases by 6% when the sampling time is reduced by 2/3 ($N_s = 20$ in Fig. 8), while the accuracy decreased by 16% when 2/3 of the locations are skipped ($r = 1/3$ in Fig. 10).

3) *Testing Baseline Method M^+ : Using Interpolation to Improve the Performance*: We also evaluated the method of

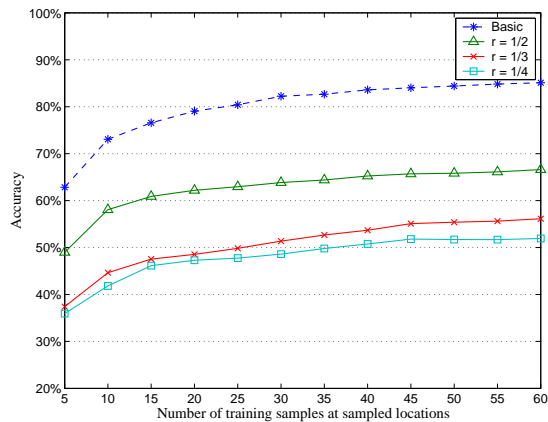


Fig. 10. Accuracy v.s. number of training samples at each sampled location (M_2)

interpolating a radio map. In the experiments, we varied both the sampling time N_s and the number of sampled locations N_l , as in Sections V-C.1 and V-C.2.

Fig. 11 shows the effect of using interpolation to reinforce complete radio maps. A radio map was first constructed with N_s training samples at each locations, where N_s varied from 5 to 60. After that, we reinforced it using the interpolation method introduced in Section V-A.3. In the experiments, three different weight vectors were tested: $W_1 = (0.5, 0.3, 0.2)$, $W_2 = (0.3, 0.3, 0.3)$ and $W_3 = (0.2, 0.4, 0.4)$, corresponding to the three solid curves in the figure. Again, the “Basic” curve in Fig. 8 is also shown for comparison, where interpolation was not used. As we can see, M^+ is useful to improve the performance of a radio map, especially when the calibration data are scarce: Improvement is above 5% when $N_s \leq 20$. To be more specific, the method is effective in making samples at neighboring locations help one another to adjust signal-strength distributions. As given in (15), the roles these samples play are determined by the weight vector. Overall, W_2 , which views H_i^0 , H_i^1 and H_i^2 equally important, achieves the best performance among the three ($W_1 \sim W_3$).

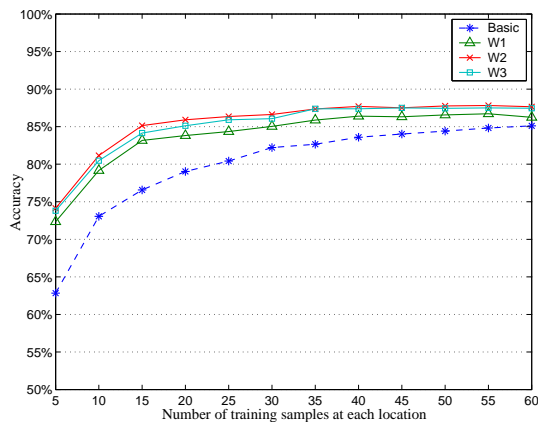


Fig. 11. Improvement achieved through using interpolation to reinforce radio maps

To evaluate M^+ in patching an incomplete radio map and thus improving its performance, we reduced the number

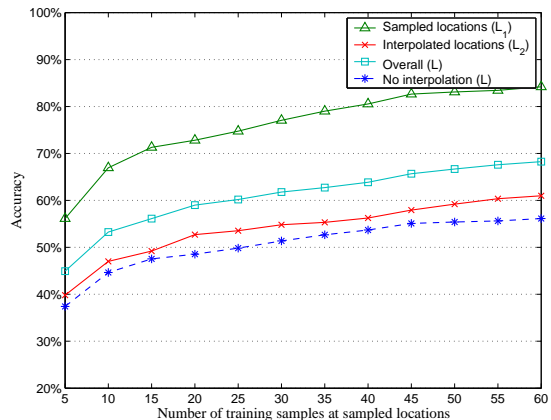


Fig. 12. Improvement achieved through using interpolation to patch radio maps ($r = 1/3$)

of sampled locations and interpolated the skipped locations afterwards. For this purpose, signal-strength distributions at the locations in \mathbb{L}_1 were constructed from the calibration data (with varying N_s); the distributions at those locations in \mathbb{L}_2 were built using M^+ . After an incomplete radio map was patched, we then measured how the location-estimation accuracy is affected at both the sampled locations and the interpolated locations.

Fig. 12 shows the effect of using interpolation to improve the radio maps with a fixed sampling ratio $r = 1/3$. For a fixed number of training samples, for example $N_s = 20$, three measurements were taken. The first one is the *sampling accuracy*. It is the accuracy of the signal-strength distributions at the sampled locations in \mathbb{L}_1 , which were built directly from the calibration data ($N_s = 20$). The second measurement is the *interpolation accuracy* of the distributions at the locations in \mathbb{L}_2 . These distributions were interpolated from the sampled ones with $N_s = 20$. The last one is the *overall accuracy* of the patched radio map. We obtained it by measuring the performance of the radio map over all the locations in \mathbb{L} . For comparison, the performance of incomplete radio maps with the same sampling ratio ($1/3$) is shown by the dashed curve in the figure, where M^+ was not used. As we can see from the figure, both the sampling accuracy on \mathbb{L}_1 and the interpolation accuracy on \mathbb{L}_2 increase as more calibration data are available. This is because as more training samples are obtained at the sampled locations, the sampled signal-strength distributions and subsequently the interpolated distributions are more accurate. In general, the interpolation accuracy is about 20% lower than the sampling accuracy, and thus the overall accuracy lies between them. Compared with the dashed curve where M^+ was not used, using M^+ does achieve much improvement: accuracy increases by 7.5% when $N_s = 5$ and by 12.1% when $N_s = 60$.

We also compared our interpolation method M^+ with the kernel-based interpolation method proposed in [7]. In their approach, the interpolation formulation fits a radial basis function which takes signal-strength measurements as input and outputs location coordinates. Calibration data are used to compute Gaussian kernel weights by least squares fitting. For

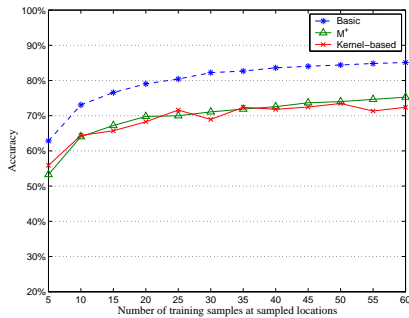


Fig. 13. Comparison of M^+ and kernel-based interpolation ($r = 1/2$)

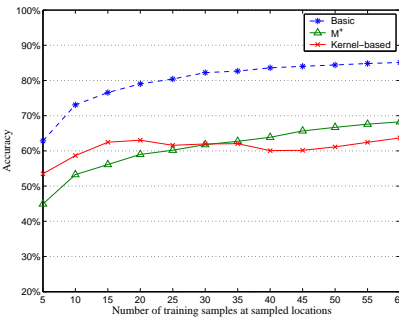


Fig. 14. Comparison of M^+ and kernel-based interpolation ($r = 1/3$)

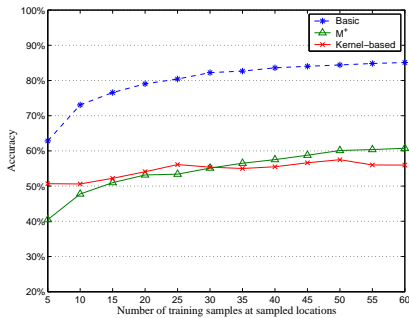


Fig. 15. Comparison of M^+ and kernel-based interpolation ($r = 1/4$)

comparison, we implemented their algorithm, with the number of signal-strength clusters at each sampled location set to three. The results of overall accuracy with varying sampling ratios ($r = 1/2, 1/3, 1/4$) are shown in Fig. 13 through Fig. 15. The two methods show similar accuracy when the sampling ratio is high ($r = 1/2$). The kernel-based method performs better than M^+ when the sample ratio is low and the calibration data are limited (less than 30 samples per sampled location in Fig. 14 and Fig. 15). As the calibration data increase, M^+ outperforms the kernel-based method. This may be explained by the probabilistic nature of M^+ method that quantitatively measures the likelihood of any possible signal-strength observation. Thus, the more calibration data M^+ uses, the more accurate the sampling distributions and the interpolated distributions. Moreover, instead of imposing a mapping between signal strength and location coordinates as the kernel-based method does, M^+ explicitly constructs signal-strength distributions at the interpolated locations. These signal-strength distributions are required by M^* on initialization. Thus we only study M^+ in the following of this paper. The “Basic” curve corresponding to the case where all locations are sampled ($r = 1$) are also shown in the figures. In addition, by comparing the M^+ curves in Fig. 13 through Fig. 15 with those in Fig. 10, we can see the effectiveness of M^+ : on average, accuracy increases by 7.21% when $r = 1/2$, by 10.2% when $r = 1/3$ and by 6.6% when $r = 1/4$.

It is also interesting to investigate the influence of r on the sampling accuracy and the interpolation accuracy. The experimental results are shown in Fig. 16 and Fig. 17. In general, the interpolation accuracy decreases as expected when less locations are directly sampled. The sampling accuracy also decreases when r is changed from $1/2$ to $1/3$. However, the accuracy then increases when r is further decreased to $1/4$, as shown in Fig. 16. This is possibly because when the sampling ratio is low ($1/4$), the sampled locations are sparse and thus the signal-strength distributions at these locations can be easily distinguished from one another. Therefore, the sampling accuracy increases instead. However, since the interpolation accuracy decreases more significantly when r is changed to $1/4$, the overall accuracy decreases, as shown in Fig. 13 through Fig. 15.

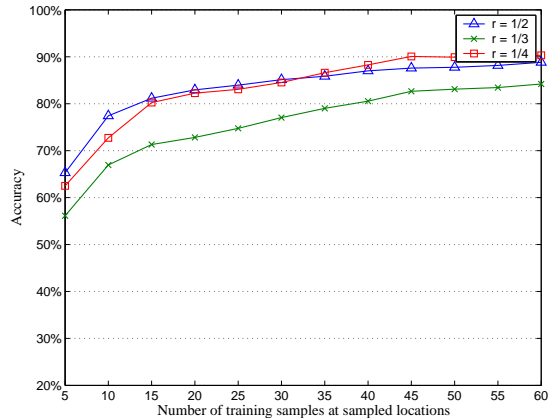


Fig. 16. Sampling accuracy of patched radio maps with varying sampling ratios r

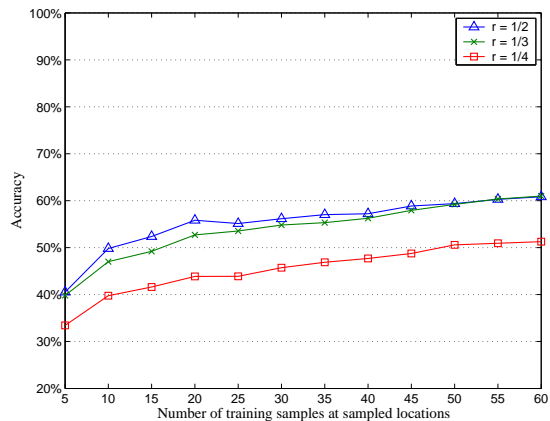


Fig. 17. Interpolation accuracy of patched radio maps with varying sampling ratios r

D. Testing Our HMM-based Method M^*

In this section, we measured the performance of using unlabeled traces in the HMM-based method M^* , which corresponds to our main contribution in the paper. Radio maps were first initialized using M^+ when there were skipped locations. Then we used M^* to improve them. The initial state distribution π^0 is set to a uniform distribution over all the locations. The initial location-state transition $A^0 = Pr(l_j|l_i)$ is also set to a uniform distribution but with the following constraints: $Pr(l_j|l_i) \neq 0$, if l_j is within 4.5 meters of

l_i ; otherwise, $Pr(l_j|l_i) = 0$. Such constraints reflect our assumption that a user can only move to a nearby location in consecutive time steps and he moves with a normal speed (i.e., within a radius of 4.5 meters in the next second). To investigate the utility of unlabeled traces, we also varied the number of traces used in training the HMM.

Fig. 18 shows the improvement in accuracy using unlabeled traces, where $N_l = 99$ and the sampling time is fixed at $N_s = 20$. When no learning is performed (the number of traces used is zero), the accuracy is about 79%. The accuracy goes up as the number of traces increases. Improvement is about 4% when 20 traces are used and 9% using 100 traces. At this point, the radio map tends to stabilize as the influence of using more traces is lessened.

Fig. 19 shows the effect of using unlabeled traces to reduce the sampling time. The dashed curve is the same as the one in Fig. 8. It is denoted as “Basic (0 Trs)” since only the calibration data but no traces are used. The other three curves show the performance of improved radio maps tuned by the EM algorithm using 20, 60 and 100 traces. The improvement is significant when the calibration data are extremely scarce. At the point where $N_s = 5$, an increase of 12.8% is achieved using 20 traces and 23.8% using 100 traces, compared with a 11.3% increase using the interpolation method M^+ . This shows that unlabeled user traces are informative and by extracting valuable information contained in these traces, M^* is much more effective than M^+ . Thus, using unlabeled traces, we can progressively reduce the sampling time and a high level of accuracy can still be achieved.

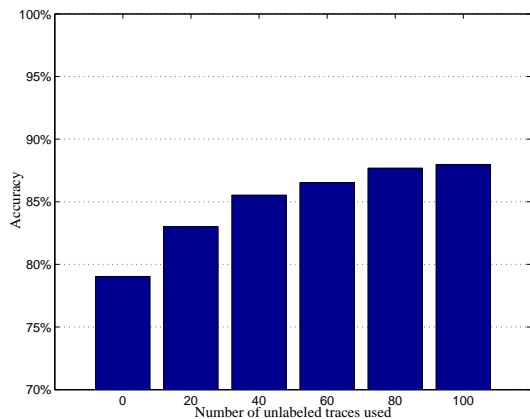


Fig. 18. Improvement achieved through using an increasing number of traces ($N_s = 20$, $N_l = 99$)

It is most interesting to see how much further improvement that M^* can achieve after M^+ is used. For this purpose, we conducted experiments to evaluate the learning algorithm when both N_s and N_l are reduced and M^+ is applied to improve the radio maps. In the first set of experiments, we fixed the sampling ratio to 1/3. The results are shown in Fig. 20, Fig. 21 and Fig. 22. Fig. 20 compares the overall accuracy of radio maps tuned by different numbers of traces. The overall accuracy on \mathbb{L} using M^+ from Fig. 12 is shown by the dashed curve for comparison. As we can see, the improvement is significant. When $N_s = 5$, we achieved an increase of 17.2%

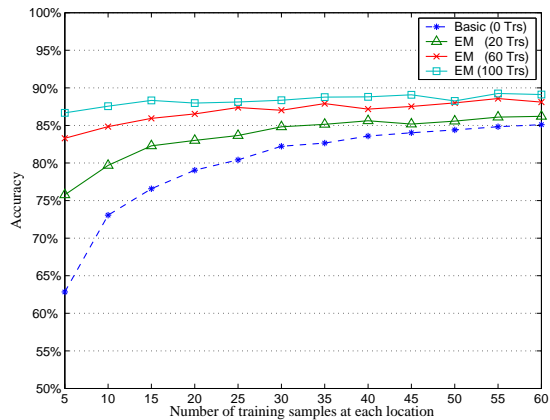


Fig. 19. Effect of using a varying number of traces to reduce the sampling time

using 20 traces and an increase of 33.2% using 100 traces. To be more illustrative, the improvement in sampling accuracy at the sampled locations and the improvement in interpolation accuracy at the interpolated locations are shown in Fig. 21 and Fig. 22, respectively. Unlabeled traces are particularly effective in adjusting the distributions at the interpolated locations.

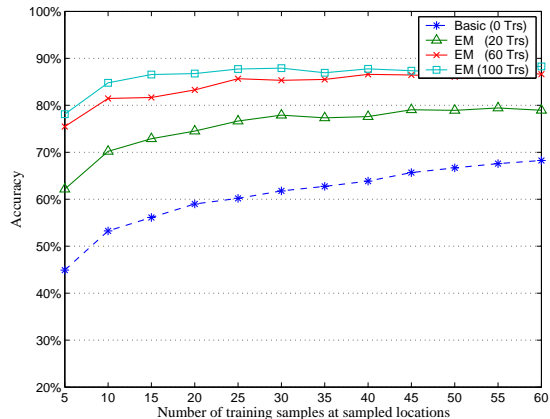


Fig. 20. Improvement in overall accuracy using M^+ and M^*

We also varied both the number of user traces and the sampling ratio r . The experimental results are shown in Fig. 23, where the sampling time N_s is fixed at 20. When no trace is used, the discrepancy between the performance of radio maps with different sampling ratios is significant: Accuracy decreases by 16.2% when r is reduced from 1/2 to 1/4. As the number of user traces increases, all the radio maps exhibit better performance and in the meantime, the difference between them is reduced: the accuracy differs only by 1.8% when 100 traces are used. This experiment shows from another angle that M^* is good at adjusting screwed signal-strength distributions towards correct ones.

E. Location Tracking Experiments

Another advantage of applying M^* is that it also tunes the initial state distribution π and the location-state transition matrix A , in addition to the radio map λ . These are exactly

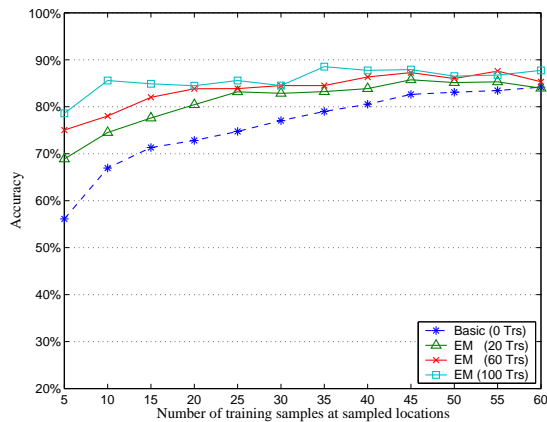


Fig. 21. Improvement in sampling accuracy using M^+ and M^*

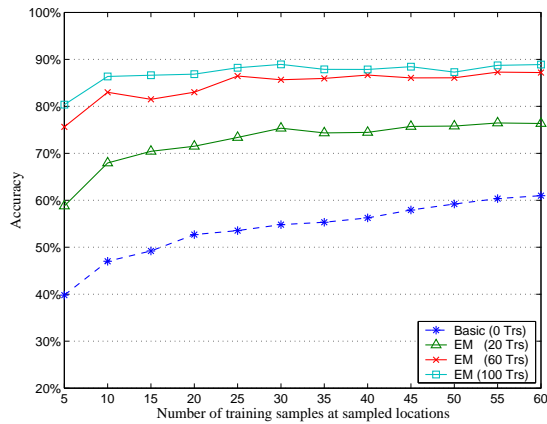


Fig. 22. Improvement in interpolation accuracy using M^+ and M^*

the model parameters that we need to implement an HMM filter. In the experiments, the performance of tracking was tested. The data set was again divided into two subsets: a training data set Tr and a testing data set Ts . Tr consists of the first N_s collected samples for training, where N_s ranges from 5 to 60 at an interval of 5. Ts comprises the rest of the samples, and they were used to produce unlabeled traces. A trace was generated as follows: we first planned a walk trip in the hallways, which consists of a sequence of location points; a sample at each of those locations was then randomly selected from Ts ; finally, those samples were concatenated to produce a trace. Consequently, the actual location labels of the samples in a trace are known for evaluation. In total we generated 200 traces, each trace containing 60 unlabeled samples on average. In the experiment, both the hand-coded HMM filter as used in [4], denoted as *Robotics Tracker*, and our tuned HMM filters using unlabeled traces, denoted as M^* tracker, were examined. Moreover, to evaluate the benefits of using temporal correlation of samples in tracking, we also tested M^* method which treats samples in a trace independently. Since it can be viewed as a static method, we refer to it as M^* Static. Tracking, on the contrary, is dynamic; it smooths location-state transitions along the time dimension. Results are shown in Fig. 24 and Fig. 25.

As can be seen, tracking using filters improves the perfor-

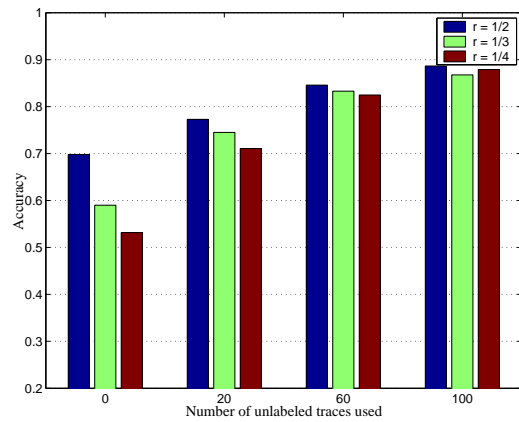


Fig. 23. Improvement in overall accuracy using an increasing number of traces with varying sampling ratios ($N_s = 20$)

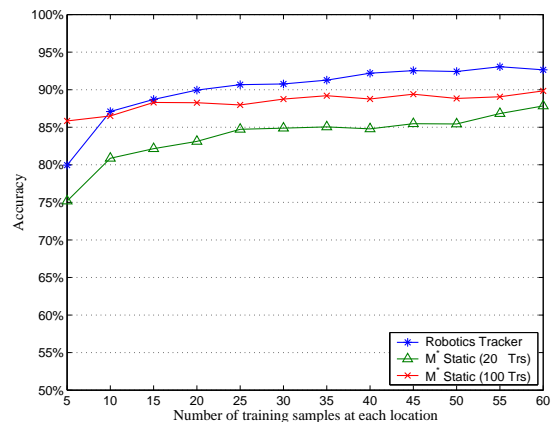


Fig. 24. Static location estimation vs. robotics tracking

mance of location estimation: Given a user's previous location estimate, motion constraints (i.e., location-state transition matrix) effectively reduce the number of possible locations a user may be at. In other words, history location information is propagated during tracking. Thus, the estimation accuracy is increased. Moreover, tuning HMM model parameters is worthwhile, especially when the calibration data are scarce.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we empirically study the effect of reducing the calibration effort on estimation accuracy by reducing both the sampling time and the number of locations sampled. An interpolation method is developed to exploit the limited labeled calibration data available to complete and improve a radio map. When additional user traces are available, our proposed EM-based learning algorithm can explore these unlabeled trace data to supplement the labeled calibration data to further improve location-estimation performance. Experiments show that both methods can adjust inaccurate radio maps into accurate ones, and furthermore, unlabeled traces can be effectively used to compensate the effects of reducing the calibration data. As a result, manual effort is reduced substantially while high accuracy is still achieved.

In the future, we plan to take complex spatial and temporal environment dynamics into consideration. For example, in

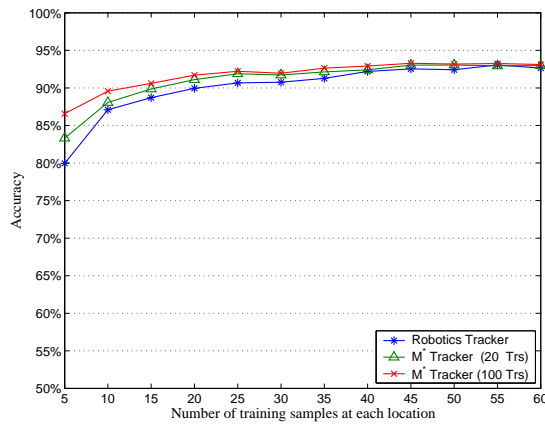


Fig. 25. Robotics tracking vs. M^* tracking

building a location-sensing system, a daytime radio map can be much different from a nighttime one. Instead of building radio maps for different periods of time, it is interesting to find a mapping between them and thus eliminate the need for tedious and repeated manual effort. It is also interesting to explore new methods that can handle small-scale variations better, especially in highly dynamic areas. Moreover, we wish to test the validity of our proposed methods in large-scale environments where reducing intensive calibration is of important concern. Another important direction is to speed up offline EM training. Clustering algorithms are promising to achieve the goal. Before running the EM algorithm, location points are clustered into groups based on their similarities in signal strength; as a result, the entire radio map is decomposed into a set of smaller ones, each corresponding to one cluster. Then the EM algorithm can be applied to each cluster separately, and each run updates a portion of the entire radio map. Besides reducing the training time, performing clustering also brings the benefit of locality: when high accuracy is demanded in some region, a local radio map can be tuned intensively.

ACKNOWLEDGMENT

The authors would like to thank the support of Hong Kong RGC Grant HKUST 6187/04E and Grant NECLC05/06.EG01 from NEC China Lab. They also thank the anonymous referees for their comments.

REFERENCES

- [1] P. Bahl, A. Balachandran, and V. Padmanabhan, "Enhancements to the RADAR user location and tracking system," Microsoft Research, Technical Report, February 2000.
- [2] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proceedings of IEEE INFOCOM2000*, 2000, pp. 775–784.
- [3] C. Gentile and L. K. Berndt, "Robust location using system dynamics and motion constraints," in *IEEE Conference on Communications*, June 2004.
- [4] A. Ladd, K. Bekris, G. Marceau, A. Rudys, L. Kavradi, and D. Wallach, "Robotics-based location sensing using wireless ethernet," in *Proceedings of MOBICOM2002*, Atlanta, Georgia, USA, September 2002.
- [5] 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*, vol. 9, no. 3, pp. 155–164, July 2002.

- [6] M. Youssef, A. Agrawala, and U. Shankar, "WLAN location determination via clustering and probability distributions," in *Proceedings of IEEE PerCom2003*, March 2003.
- [7] J. Krumm and J. C. Platt, "Minimizing calibration effort for an indoor 802.11 device location measurement system," Microsoft Research, Technical Report, 2003.
- [8] A. Smailagic, D. P. Siewiorek, J. Anhalt, D. Kogan, and Y. Wang, "Location sensing and privacy in a context aware computing environment," *Pervasive Computing*, 2001.
- [9] M. Youssef and A. Agrawala, "Handling samples correlation in the horus system," in *IEEE INFOCOM 2004*, 2004.
- [10] M. Youssef and A. Agrawala, "Small-scale compensation for WLAN location determination systems," in *IEEE Wireless Communications and Network Conference*, 2003.
- [11] D. Fox, J. Hightower, L. Liao, and D. Schulz, "Bayesian filtering for location estimation," *IEEE Pervasive Computing*, vol. 2, no. 3, pp. 24–33, 2002.
- [12] D. Madigan, E. Elnahrawy, R. Martin, W. Ju, P. Krishnan, and A. Krishnakumar, "Bayesian indoor positioning systems," in *Proceedings of IEEE INFOCOM 2005*, 2005.
- [13] M. Berna, B. Lisien, B. Sellner, G. Gordon, F. Pfening, and S. Thrun, "A learning algorithm for localizing people based on wireless signal strength that uses labeled and unlabeled data," in *IJCAI'03*, Acapulco, Mexico, August 2003.
- [14] H. Hashemi, "The indoor radio propagation channel," in *Proceedings of the IEEE*, vol. 81, no. 7, 1993, pp. 943–968.
- [15] H. Bunke and T. Caelli, *Hidden Markov Models - Applications in Computer Vision*. World Scientific Publishing Co, 2001.
- [16] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," in *Proc. IEEE*, vol. 77, no. 2, 1989, pp. 257–286.
- [17] S. Luhr, H. H. Bui, S. Venkatesh, and G. A. West, "Recognition of human activity through hierarchical stochastic learning," in *First IEEE International Conference on Pervasive Computing and Communications*, March 2003.
- [18] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via EM algorithm," *Journal of the Royal Statistical Society Series B*, vol. 39, pp. 1–38, 1977.
- [19] C. Fretzagias and M. Papadopoulou, "Cooperative location-sensing for wireless networks," in *Proceedings of IEEE PerCom2004*, March 2004.
- [20] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, 2002.
- [21] A. McCallum and K. Nigam, "A comparison of event models for naive Bayes text classification," in *Proc. of AAAI/ICML-98 Workshop on Learning for Text Categorization*, 1998, pp. 41–48.



Xiaoyong Chai received his MPhil degree in computer science from the Hong Kong University of Science and Technology in 2005. Currently, he is a PhD student in the Department of Computer Sciences at the University of Wisconsin, Madison. His research interests include data mining, pervasive computing, and artificial intelligence.



Qiang Yang is a faculty member in the Hong Kong University of Science and Technology's Department of Computer Science and Engineering. His research interests are AI planning, machine learning, case-based reasoning, and data mining. He received his PhD from the University of Maryland, College Park. He is a senior member of the IEEE and an associate editor for the IEEE TKDE and IEEE Intelligent Systems. Contact him at the Dept. of Computer Science and Engineering, Hong Kong Univ. of Science and Technology, Clearwater Bay, Kowloon, Hong Kong;

qyang@cse.ust.hk; <http://www.cse.ust.hk/~qyang>.