Learning Adaptive Temporal Radio Maps for Signal-Strength-Based Location Estimation

Jie Yin, Qiang Yang, *Senior Member, IEEE* and Lionel M. Ni, *Member, IEEE*

***Abstract*— In wireless networks, a client's locations can be estimated using signal strength received from signal transmitters. Static fingerprint-based techniques are commonly used for location estimation, in which a radio map is built by calibrating signal-strength values in the offline phase. These values, compiled into deterministic or probabilistic models, are used for online localization. However, the radio map can be outdated when signal-strength values change over time due to environmental dynamics, and repeated data calibration is infeasible or expensive. In this paper, we present a novel algorithm, known as LEMT (*L*ocation *E*stimation using *M*odel *T*rees), to reconstruct a radio map using real-time signal-strength readings received at the reference points. This algorithm can take into account real-time signal-strength values at each time point and make use of the dependency between the estimated locations and reference points. We show that this technique can effectively accommodate the variations of signal strength over different time periods without the need to rebuild the radio maps repeatedly. The effectiveness of LEMT is demonstrated using two real data sets collected from an 802.11b wireless network and a RFID-based network.**

***Index Terms*— Location Estimation, Temporal Radio Maps, Received Signal Strength, Reference Points**

I. INTRODUCTION

The advent of wireless technology and mobile computing devices has fostered growing commercial and research interest in developing various location-estimation systems. A central task in building such systems is to develop techniques for estimating the locations of mobile devices – and hence users – in wireless environments. In indoor settings, much effort has been focused on the development of Radio-Frequency (RF)-based location-estimation techniques using Received Signal Strength (RSS) measurements, by making use of popular infrastructures such as the IEEE 802.11b wireless local-area-networks (WLANs) [1] [6] [10] [22] and Radio Frequency Identification (RFID) based networks [11]. Being able to accomplish these tasks plays an important role in many location-aware applications that range from context-dependent content delivery to the monitoring of moving objects and people [4] [19].

RF-based location-estimation systems utilize signal strength received from signal transmitters, such as WLAN Access Points (APs) and RFID tags, to infer the locations of users. In theory, signal strength decays linearly with log distance and a simple triangulation method using signal strength from three or more than three signal transmitters could uniquely identify a user's locations. However, in practice, it is impossible to obtain an accurate signal propagation model because physical characteristics of an environment, such as walls, furniture and even human activities, add significant noise to RSS measurements. Therefore, techniques based on static location fingerprints are often adopted in indoor location-estimation systems.

Fingerprint-based techniques consist of two phases: an *offline training phase* and an *online localization phase* [1] [6] [10] [22]. In the offline phase, a *radio map* is built by tabulating RSS measurements received from signal transmitters at predefined locations in the area of interest. These values comprise a radio map of the physical region, which is compiled into a *deterministic* or *probabilistic* model for online localization. In the online localization phase, the real-time RSS samples received from signal transmitters are used to search the radio map to estimate a user's current location based on the learned model.

In the offline phase, a learned location-estimation model is essentially a mapping function between the signal space and the location space. Deterministic techniques build such a mapping by simply storing the average RSS values at a collection of known locations, and use the nearest neighbor method to locate a client. Probabilistic techniques, on the other hand, construct the mapping by storing the RSS distributions as the content of a radio map. The distributions are then used in a maximum likelihood calculation for localization. With sufficient training data, probabilistic methods are typically more accurate than their deterministic counterparts by directly handling the uncertainty of RSS measurements. However, a major limitation of both fingerprint-based methods is that the radio maps are static. Once learned in the offline phase, a static radio map is applied thereafter to estimate the locations in later time periods without adaptation. This simplistic assumption poses a serious problem to the effectiveness of location estimation. In dynamic indoor environments, radio signal propagation suffers from time-correlated fading effects, which typically consist of two components: the long term fading caused by the shadowing effect of the building or natural features, and the short term fading caused by rapid scattering around a moving device. As a result, RSS samples measured in the online phase may significantly deviate from those stored in the radio map. Therefore, using static fingerprint-based techniques for location estimation can be grossly inaccurate and thus requires repeated data gathering to maintain predictive accuracy.

To take into account dynamic environmental changes, several adaptive algorithms have been proposed in recent years [7] [9] [11]. Haeberlen et al. [7] adapt the static radio map by calibrating new RSS samples at a few known locations and fitting a linear function between these samples and the old samples from the radio map. In the online phase, new samples are first shifted to old samples using the estimated linear function, such that the original radio map can be re-used. The main assumption is that the

Jie Yin is with the Tasmanian Information Communication Technologies (ICT) Centre, Commonwealth Scientific and Industrial Research Organization (CSIRO), GPO Box 1538, Hobart, TAS 7001, Australia. Email: Jie.Yin@csiro.au

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

Lionel M. Ni is with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong. Email: ni@cse.ust.hk

adaptation can be performed independently of locations. However, in a real environment, RSS values can vary a lot from one location to another. The LANDMARC system [11] and the LEASE system [9] both utilize reference points to adaptively offset the variations of RSS samples caused by environmental changes. The advantage of these systems is that the location estimation can adapt to environmental dynamics by using real-time RSS samples received at reference points. However, experiments on these systems show that the accuracy of these systems can be guaranteed only when the reference receivers are densely distributed.

In this paper, we propose a novel method called LEMT for estimating locations even when RSS samples are dynamically changing over time. Our approach works in three steps: first, we place a number of RF receivers at fixed locations to detect real-time RSS samples; these receivers are called *reference points*. Second, we use the radio map collected at a certain time to learn the functional relationship in the RSS samples between the mobile client and the reference points. Third, we apply a nearest neighbor based method to find the most likely locations. This approach is referred to as the *adaptive temporal radio map* for location estimation. In our preliminary work [20], we have shown that it is feasible to use a model tree to adapt a radio map dynamically in WLANs. We extend this work by comparing our approach to existing adaptive approaches (the LANDMARC system [11] and the LEASE system [9]). In addition to the WLAN environment, we also evaluate our approach through extensive experiments in a RFID-based network environment.

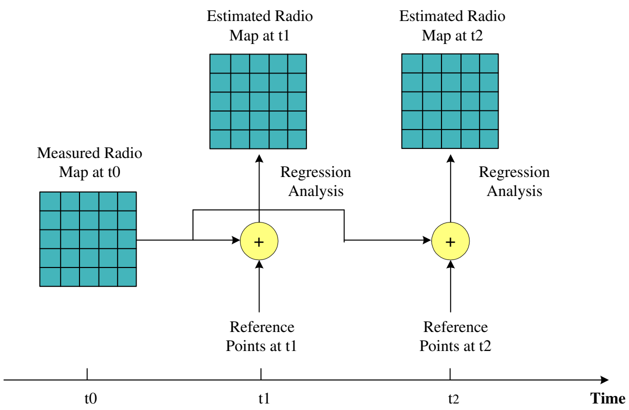


Fig. 1. Illustration of the adaptive temporal radio map based method

Figure 1 illustrates the idea behind the LEMT method. As in previous work, we start by collecting data to construct a static radio map in the time instant t_0. In any later time period t_i, where $i \geq 1$, instead of rebuilding the radio maps repeatedly, we place a few RF receivers which act as dynamic reference points throughout the geographic area. Based on real-time RSS samples received at reference points, we apply a regression analysis to obtain the estimated radio maps which comprise the corrections we need to make to the static radio map. In our approach, the static radio map is complied into a model-tree based model in which trees are built on the RSS values collected at the mobile client and those collected at reference points. In the online phase, the models are used to predict the most likely location of the mobile client. We show that this method is more robust as time evolves and the environment changes. We demonstrate the capability of this method in two real wireless network domains.

In this work, our objective is to extend current indoor location-estimation techniques to cope with the variations of the radio maps at different time periods. This extension would allow the radio map built at one time instant to be adaptable and usable for other time instants. We mainly focus on dealing with environmental changes caused by the short term fading, like unpredictable people moving and door opening or closing, in the building. We can also cope with small environmental changes caused by the long term fading, like the changes of light, temperature and humidity in the environment. However, if significant infrastructure changes occur, such as the change of the building layout and the moving of signal transmitters, the radio map needs to be rebuilt by calibrating new RSS samples. Our basic intuition is that, for a mobile client at a specific location, its neighbors can reflect similar dynamic changes in its surrounding environment. Therefore, even though the values of RSS samples may change greatly over time even at the same location, the relation of how signal strength depends on its neighboring reference points remain relatively constant. In other words, the local neighborhood relationship stays the same while each neighbor may change with time. This constraint is typically used in machine learning, when dimensionality reduction is applied to complex data [16]. We can thus adapt the radio map built at a certain time instant t_0 using real-time RSS samples received at references points at other time instants t_i. This assumption will be thoroughly verified using extensive experiments presented in Section IV.

The novelty of our work can be summarized as follows:

- Compared with previous static fingerprint-based techniques, our proposed LEMT method can better adapt to the variations of RSS values caused by the environmental dynamics.
- Our proposed LEMT method can achieve higher localization accuracy than existing adaptive techniques, even with a low density of reference points.

The rest of the paper is organized as follows. Section II reviews related work on location estimation using RF signal strength. Section III presents our proposed algorithm for location estimation in detail. Section IV presents extensive experimental evaluation of our proposed algorithm. Section V concludes the paper and discusses directions for future work.

II. Location Estimation Based on RF Signal Strength

In this section, we review two major approaches to location estimation using RF signal strength. Section II-A reviews fingerprint-based techniques for location estimation. Section II-B presents the noisy characteristics of RF signal strength. Section II-C discusses adaptive techniques to tackle the variations of signal strength due to environmental changes.

A. Static Fingerprint-Based Techniques

Significant research has been undertaken on location estimation using static fingerprint-based approaches. The basic idea is to build a radio map by collecting RSS samples in predefined locations in the offline phase and apply the radio map to estimate the locations in the online phase. Depending on how the radio map is built, we classify fingerprint-based approaches into *deterministic techniques* and *probabilistic techniques*. Deterministic techniques [1] [2] apply deterministic inference to estimate a client's locations. For example, the RADAR system by Microsoft Research [1] uses the nearest neighbor method to infer a user's locations. In the offline phase, RADAR builds a radio map by storing the average RSS value for each AP at each location. In the online phase, new RSS samples are compared against the radio map and the coordinates of the best matches are averaged to give the location estimate. The accuracy of RADAR is about three meters with 50% probability. Since RADAR only represents RSS samples using a simple mean instead of the whole signal distribution, its localization performance is limited.

Probabilistic techniques [4] [10] [15] [21] [22] [23] form the second category of fingerprint-based approaches. They tackle the uncertainty problem in indoor wireless networks by constructing the RSS distributions over locations in the radio map and use probabilistic inference methods for localization. For example, the robotics-based location sensing system [10] first computes the conditional probabilities over locations based on RSS samples. Then a post-processing step, which utilizes the spatial constraints of a user's movement trajectories, is used to refine the location estimation and to reject the estimates showing significant changes in the location space. Depending on whether the postprocessing step is used or not, the accuracy of this method is 83% or 77%, respectively, within 1.5 meters. Youssef et al. [23] apply a joint clustering technique to group locations so as to reduce the computational cost of the system. The method first determines a most likely cluster within which to search for the most probable location, and then applies Bayesian inference to estimate the most probable location within the cluster. The core technique of these approaches is the use of the Maximum Likelihood (ML) method which computes a probability distribution over locations conditioning on RSS samples and estimates the location to be the one with the maximum likelihood in the distribution. The advantage of the ML method is that it captures the noisy characteristics in signal propagation using conditional probabilities. Therefore, it can preserve complete information contained in the RSS samples for further localization.

Most of the above fingerprint-based approaches are based on a common assumption that the radio map built in the offline phase does not change much later in the online phase. A major limitation with this assumption stems from the dynamic characteristics of signal propagation and the environment, where the RSS values measured in the online phase can significantly deviate from those stored in the radio map, thereby limiting the localization accuracy in practical location-estimation systems.

B. Noisy Characteristics of RF signal strength

As we have mentioned in Section I, our work is motivated to cope with the variations of radio maps at different time periods. In this section, we demonstrate the need for the radio map adaptation by showing the uncertain nature of RF signal strength. We illustrate using two particular experimental test-beds: an indoor WLAN environment and a RFID-based network environment. Below we analyze the noisy characteristics of RF signal strength in the two networks.

The IEEE 802.11b WLAN uses radio frequencies in the 2.4 GHz band, which is attractive because it is license-free in most places around the world. However, it does suffer from inherent disadvantages. In the 2.4 GHz band, microwave ovens, Bluetooth devices, 2.4 GHz cordless phones and other devices can be sources of interference. Subject to reflection, refraction, diffraction and absorption by structures and humans, signal propagation suffers from severe multi-path fading effects [8]. A transmitted signal can reach the receiver through different paths, each having its own amplitude and phase. These different components are then combined to reproduce a distorted version of the original signal. These phenomena are particularly severe when operating indoors because there is rarely a line of sight between the transmitter and the receiver. In addition, RF signal strength also varies at different time periods due to time-correlated phenomena [7]. These phenomena include changes in environmental conditions caused by people moving, or doors opening and closing in the building, and transient interference caused by other electronic devices. These changes can cause signal strength to vary from time to time over both small and large timescales, which in turn makes the RSS radio maps collected at one time period become invalid at later time periods.

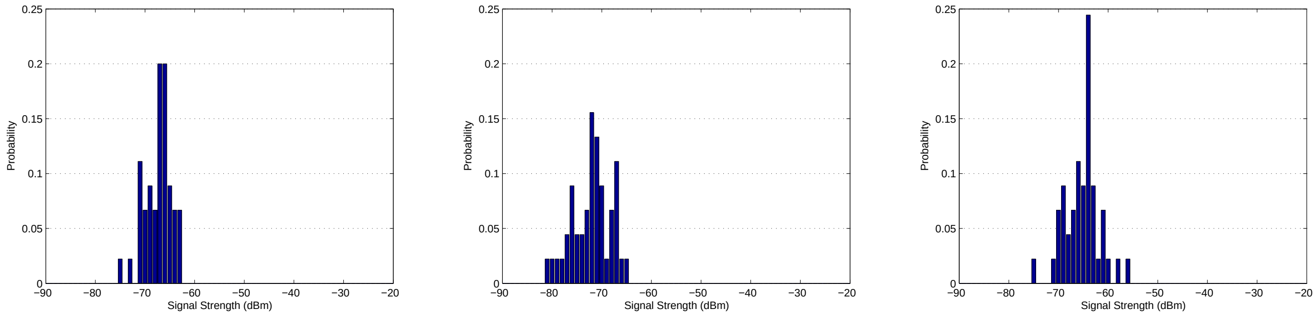
In Figures 2, we give a typical example to illustrate the variations of RF signal strength over different time periods. The figure shows three signal-strength histograms at different time periods at a particular location 20 meters away from a fixed AP. To build each histogram, at each location we took 450 RSS samples within a time period of 45 seconds. From these histograms, we can clearly observe that, these distributions, asymmetric and having multiple modes, are essentially non-Gaussian. More importantly, the signal-strength histograms vary noticeably over different time periods. These variations suggest that, depending on the histograms trained in the offline phase, location estimation might be inaccurate if RSS samples measured in the online phase deviate significantly from those collected in the offline phase.

Figure 3 shows two signal-strength histograms received by a RFID reader in a RFID-based network. To build each histogram, we collected 150 RSS samples at a location three meters away from the RFID reader. For data calibration, we used the RF Code MANTIS^{TM} active readers and tags [14] in our experiment. The operating frequency is 303.8 MHZ and the transmission range is up to 1500 feet. RFID readers detect and interpret the radio frequency beacon emitted by RFID tags to identify them and provide signal-strength information to determine their locations. We can see that, the signal-strength histogram in the daytime is quite different from that collected at night, although the uncertainty within the same time period is not as high as the RSS samples in a WLAN environment.

In summary, the RSS samples received in the WLAN and RFID-based network environments have similar uncertain characteristics in nature as time evolves. Therefore, it is a challenging task to accurately determine the locations of the tracked client in such dynamically changing environments.

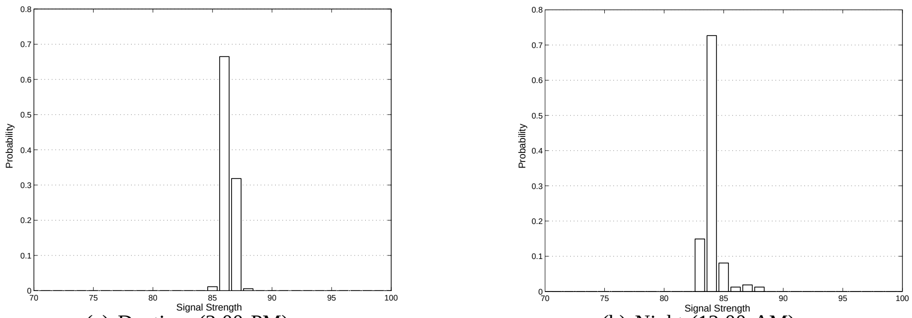
C. Adaptive Techniques in Previous Works

In recent years, several adaptive algorithms have been proposed to deal with the signal-strength variations caused by environmental changes. Haeberlen et al. [7] adapt the static radio map by calibrating new RSS samples at a few known locations and fitting a linear function between these values and the old values from the radio map. In the online phase, new RSS samples, independent of different locations, are first shifted to old samples using the estimated linear function, so that the original radio map can be reused. The main assumption of this method is that, the adaptation is uniformly performed across all the locations. However, this is not true in real wireless environments where the RSS values might vary a lot from one location to another.



(a) 10:00 AM (b) 2:00 PM (c) 10:00 PM

Fig. 2. The variations of signal-strength distributions over different time periods at a particular location 20 meters away from an AP



(a) Daytime (2:00 PM) (b) Night (12:00 AM)

Fig. 3. The variations of signal-strength distributions over different time periods at the same location 3 meters away from a fixed RFID reader

The LANDMARC system [11] and the LEASE system [9] both utilize the concept of referent points to alleviate the effects caused by the fluctuation in RF signal strength. LANDMARC [11] first computes the Euclidean distance in signal-strength vectors between a tracked client and reference points, and then uses k nearest reference points' coordinates to the location of the tracked client. The authors report that one reference tag is needed for each square meter to accurately locate the objects within the error distance between one and two meters. However, the accuracy of LANDMARC can only be guaranteed with a high density of reference points. The LEASE system [9] deploys a number of stationary emitters (SEs) and sniffers to assist location estimation in WLANs. In this system, SEs play the role of reference points as in our work. To obtain up-to-date values, LEASE applies Akima splines to interpolate RSS values at each grid using the known coordinates of the SEs and RSS values received at SEs. To estimate the location, a nearest-neighbor method was used. In an area of 68 meters $\times$ 44 meters, LEASE can achieve a median error of 4.5 meters and 2.1 meters, using 12 SEs and 104 SEs, respectively. However, since LEASE interpolates the RSS values for each grid based on the RSS values received at SEs and the locations of SEs, LEASE can work well only when the density of SE distribution is high.

III. The LEMT Algorithm: An Overview

In this section, we present our LEMT algorithm in detail. Since the data calibration process is labor-intensive and time-consuming, our main objective is to build an accurate location-estimation model that can work at different time periods even when limited training data are collected at a single time point. We accomplish this task by using real-time RSS samples received at reference points to adapt the static radio map over time. Once the model is constructed in an offline phase, we apply this model to the RSS values received in real time for online location estimation. Below, we first introduce the formal problem definition and notations that will be used in our algorithm description.

A. Problem Definition

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

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

where each tuple $(x_i, y_i), 1 \leq i \leq n$, represents the location of a tracked mobile client.

We define the signal-strength vector received by a tracked client as $\mathbf{s} = (s_1, \ldots, s_p)$, where $s_j, 1 \leq j \leq p$, denotes the RSS value received from the j^{th} signal transmitter (APs or RFID tags), and p is the number of signal transmitters in the environment. Suppose that there are m reference points placed in the environment, the signal-strength vector received at a reference point can be denoted as $\mathbf{r}_k = (r_{k1}, \ldots, r_{kp})$, where r_{kj}, $1 \leq k \leq m$, $1 \leq j \leq p$, represents the RSS value received at the k^{th} reference point from the j^{th} signal transmitter. The location estimation problem is, given a signal-strength vector $\mathbf{s}$ received by a tracked client and a set of signal-strength vectors $\mathbf{r}_k$, $1 \leq k \leq m$, received at reference points, we would like to estimate the client's location $\tilde{l}$ in the location space $\mathbb{L}$.

In our work, the performance of location estimation is measured using the notion of *localization accuracy*. Let d be a given error distance threshold measured in meters between two physical locations. If the distance between the estimated location $\tilde{l}$ and the actual location l^* is less than the error distance d, it is called a correct estimation. Given a test data set consisting of N signal-strength vectors received by the client, if the location-estimation algorithm makes C correct estimations, the algorithm is called to have an accuracy of C/N within an error distance d.

In the following, we detail the offline phase and the online phase respectively.

B. The Offline Training Phase

During the offline phase, which corresponds to the time period t_0, we apply a regression analysis to learn the predictive relationship between RSS values received at the reference points and at the mobile client which is tracked at each predefined location. Consider a location l_i, $1 \leq i \leq n$, for the j^{th} signal transmitter, $1 \leq j \leq p$, we learn a functional relationship f_{ij} which denotes the mapping from RSS values $r_{kj}(t_0)$ received at the k^{th} reference point, $1 \leq k \leq m$, to the RSS value received at the mobile client $s_j(t_0)$ at time t_0. In particular, we build a regression relationship using the following function $f_{ij}(t_0)$:

$$s_j(t_0) = f_{ij}(r_{1j}(t_0), r_{2j}(t_0), \ldots, r_{mj}(t_0)),$$
$$1 \leq i \leq n, 1 \leq j \leq p, \quad (1)$$

While this function $f_{ij}(t_0)$ is learned in time period t_0, the fundamental assumption in our work is that it captures the functional relationship between RSS values received at reference points and at the mobile device for each location, *regardless of the time period* t. In order to compute the expected RSS value received at the mobile device at time t, we simultaneously collect RSS values at reference points also at the time period t. The value $s_j(t)$ we obtain via Equation (1) is used to represent the estimated RSS value that may be received at the mobile device at each location at time t. In Section IV, we empirically show that the LEMT algorithm using this assumption gives the best result as compared to other competing systems.

C. The Online Localization Phase

During the online phase corresponding to time period t, based on the signal-strength vectors received at reference points, we compute a signal-strength vector $\tilde{\mathbf{s}}_i(t) = (\tilde{s}_{i1}(t), \ldots, \tilde{s}_{ip}(t))$ that may be received at each location l_i using the corresponding function f_{ij}. We refer to the signal-strength vector computed using the function f_{ij} as an *estimated signal-strength vector* $\tilde{\mathbf{s}}_i(t)$. Then, given an *actual signal-strength vector* $\mathbf{s}(t) = (s_1(t), \ldots, s_p(t))$ recorded by the mobile device at time t, we use the nearest neighbor method to compute the location of the mobile device. Specifically, for each location l_i, we compute the Euclidean distance D_i between its corresponding estimated signal-strength vector $\tilde{\mathbf{s}}_i(t)$ and the actual signal-strength vector $\mathbf{s}(t)$ as follows:

$$D_i(t) = \sqrt{\sum_{j=1}^{p} (\tilde{s}_{ij}(t) - s_j(t))^2}. \quad (2)$$

Finally, the estimated location $\tilde{l}$ is the one which can minimize the corresponding distance $D_i(t)$:

$$\tilde{l} = \arg\min_{l_i} D_i(t). \quad (3)$$

Since the neighboring reference points are subject to the same effect in the environment as the tracked mobile client, the newly observed RSS values at the reference points can be used to dynamically update the information for localization in real time. Therefore, this approach is more flexible and adaptive to the environmental dynamics. In order to achieve high accuracy, the critical issue is to model the functional relationship f_{ij} between the RSS values received at the reference points and at the mobile device during the offline phase, and use this relationship to compute the estimated signal-strength vectors that may be received at each location during the online phase.

D. Building Nonlinear Regression Relationship using Model Trees

In this section, we discuss how to model the functional relationship f_{ij} in Equation (1). Since the signal propagation in indoor environments is quite complex, we can never expect a globally linear relationship between the RSS values received at the reference points and at the mobile client. In particular, for a mobile client, its neighboring reference points can reflect the dynamical changes in its surrounding environment more accurately. Therefore, we propose a nonlinear approximation approach based on a model tree [12] [18] to model the functional relationship f_{ij}.

A model tree is a decision tree with linear regression functions at the leaf nodes. Thus it can represent any piecewise linear approximation to an unknown function. Figure 4 illustrates the basic idea behind the construction of a model tree. As we can see from the figure, the whole reference-point value space is partitioned into several regions, in each of which a different linear model is used for relating the RSS values received at reference points to the RSS value received at the mobile client.

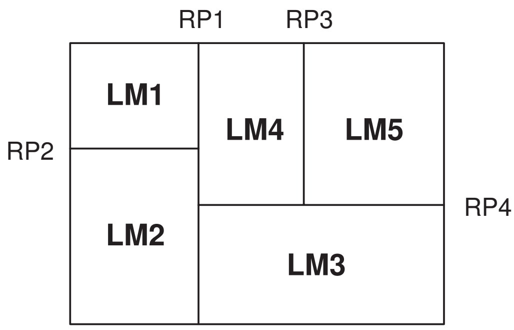


Fig. 4. Illustration of a model tree built for a signal transmitter at a location

Specifically, for each signal transmitter at each location, we build a model tree to learn the predictive relationship between the RSS values received at reference points and at the mobile device. As an example, Figure 5 shows such a tree structure built over four reference points ($RP_1 \sim RP_4$) to predict the RSS value received at the mobile device. Note that this tree structure is equivalent to the state-space structure shown in Figure 4. In the figure, each internal node corresponds to a binary test on the RSS value received at a specific reference point. Two subtrees are branched from an internal node, each corresponding to a binary range of values. For example, the root node corresponds to a binary test: $RP_1 < -73$ or $RP_1 \geq -73$. Starting from the root node, a test sample is asked through a sequence of questions

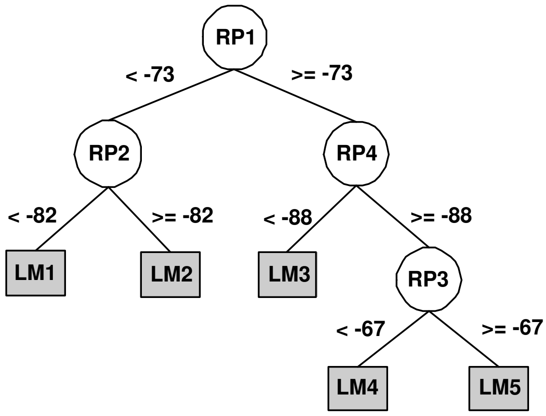


Fig. 5. An example of a model tree built for a signal transmitter at a location

until it reaches a leaf node. Each leaf node at the lowest level is attached with a linear regression function LM_i, from which the estimated RSS value that may be received at the mobile client can be calculated accordingly.

Now let us explain the process of building a model tree. In our work, we apply the M5' algorithm [17] to induce a model tree, which works in two stages: In the first stage, a decision-tree induction algorithm is used to build an initial tree by minimizing the intra-subset variation of the target value. In the second stage, the tree is pruned back by replacing subtrees with linear regression functions to minimize the estimated error. The two stages are detailed in the following discussions.

1) Building the Initial Tree: A model tree is initially built by the divide-and-conquer method, which splits the samples into subsets and applies the same process recursively to the subsets. The splitting criterion is used to determine which attribute is the best to split the samples that reach a particular node. It is based on treating the standard deviation of the class values as a measure of the error at that node, and calculating the expected reduction in error as a result of testing each attribute at that node. The expected error reduction, which is called the Standard Deviation Reduction (SDR), is calculated as follows:

$$SDR = sd(T) - \sum_i \frac{T_i}{T} * sd(T_i), \tag{4}$$

where T represents a set of samples that reach a particular node, and T_i represent the subsets that result from splitting the node according to the chosen attribute. sd denotes the standard deviation of a set of samples, which is computed as:

$$sd = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(y_i - \hat{\mu})^2}, \tag{5}$$

where y_i is the class value of each training sample, and $\hat{\mu}$ is the mean of class values for a set of M samples.

Based on the splitting criterion, the algorithm of building a model tree works as follows: Initially, all the training samples are placed in the root node. The algorithm then tries to break the samples into subsets using all possible splitting positions for each reference point, and chooses the one that maximizes the SDR as the splitting point. This splitting is then applied to each of the new branches. The splitting process continues until each node reaches a specified minimum node size and becomes a leaf node. If the standard deviation in a node reaches a user-specified minimum value, that node is also considered as a leaf node even if it has not reached the minimum node size.

In addition, the algorithm computes a multivariate linear model for each node of the tree. Each linear model takes the form of

$$LM = w_0 + w_1\alpha_1 + w_2\alpha_2 + \cdots + w_h\alpha_h, \tag{6}$$

where $\alpha_1, \alpha_2, \ldots, \alpha_h$ are the RSS values received at the reference points. The regression coefficients $w_0, w_1, \ldots, w_h$ are calculated using the least square estimation method [5]. However, the model is restricted to the reference points tested in the subtree below this node, because other reference points that affect the predicted value have been considered in the tests that lead to the node.

2) Pruning the Tree: After an initial tree is constructed, the algorithm prunes the tree based on cross-validation. The pruning procedure makes use of an estimate of the expected error at each node for unseen samples. First, the absolute difference between the predicted value and the actual class value is averaged over the training samples that reach that node. Since this average might underestimate the expected error for unseen samples, a multiplication factor $(u+v)/(u-v)$ is introduced to compute the estimated error, where u is the number of training samples that reach the node, and v is the number of parameters in the linear model at that node. The linear model at each node is simplified by dropping terms one by one, greedily, so long as the error estimate decreases. Finally, once an optimal linear model is in place for each internal node, the tree is pruned by turning some branch nodes into leaf nodes, and removing the leaf nodes under the original branch.

As an example of the online prediction process, consider estimating the RSS value received at the client from a signal transmitter using the model tree shown in Figure 5. Suppose that the RSS values received at reference points $(RP_1 \sim RP_4)$ are -78, -80, -90 and -70, respectively. Starting from the root node RP_1, the left branch would be followed because the condition $RP_1 < -73$ is satisfied. Subsequently, for the internal node RP_2, the right branch would be chosen because $RP_2 \geq -82$ is satisfied. Finally, the prediction process reaches the leave node LM_2. If $LM_2 = 0.5 * RP_1 + 0.5 * RP_2$, the estimated signal strength that would probably be received at the mobile client is -79.

E. Summary of the LEMT Algorithm

We now summarize the two phases of the LEMT algorithm, followed by a detailed discussion about its online computational complexity and robustness.

1) LEMT Algorithm Description: Our LEMT algorithm for location estimation is divided into two phases:

- *Offline Learning of Model Trees:* During the offline phase at time period t_0, at each location l_i, we use a series of q RSS samples received at the mobile device and reference points as the training data. Specifically, we use the following data:
  - D – a data set of RSS samples $\{\mathbf{s}_i, 1 \leq i \leq n\}$ collected at the mobile client in each of the n locations at time period t_0.
  - R – a set of RSS samples $\{\mathbf{r}_k, 1 \leq k \leq m\}$ collected at each of the m reference points at time period t_0.

  Then for each location l_i, we learn p different model trees, one corresponding to each signal transmitter.
- *Online Application of Learned Model Trees:* During the online phase, for each signal transmitter, given the RSS

samples received at reference points, we walk down the corresponding model tree until a leaf node is reached. Through the linear models attached to that leaf node (Equation (6)), we calculate an estimated signal-strength vector that may be received at the mobile client $\tilde{\mathbf{s}}_i = (\tilde{s}_{i1}, \tilde{s}_{i2}, \ldots, \tilde{s}_{ip})$ for each location l_i. Once $\tilde{\mathbf{s}}_i$ are obtained, we can use Equation (2) to compute their Euclidean distances to the actual signal-strength vector $\mathbf{s}$. Finally, the location l_i with the minimum distance D_i among all n locations is returned as the estimated location $\tilde{l}$ at time t.

2) Online Complexity Analysis: When we apply the learned model tree for localization during the online phase, the time complexity of the LEMT algorithm is $O(m'np)$, which is linear with the number of locations n, the number of signal transmitters p, and the average depths of learned model trees m'. Here we may have $m' \leq m$ because the LEMT algorithm always chooses an optimal subset of reference points to build the tree instead of using all the reference points. The space requirement is the number of model trees that must be stored, which equals the number of locations times the number of signal transmitters ($O(np)$).

3) Robustness Analysis: The LEMT algorithm is based on the absolute RSS values received at the reference points. If the paths between all the signal transmitters to all the reference points are blocked in the online phase, while not in the training phase, the RSS sample might be distorted, which causes location estimation to be inaccurate. However, in our work, we use m reference points where m is more than two. In such a case, there are two reasons why our method can still work (that is, robust), even though some of the reference points are blocked. First, the chance for all the paths from signal transmitters to all m reference points to be simultaneously blocked is fairly small. Second, when ”some” of paths (say, u) are blocked, it only affects u model trees to give inaccurate results, which may affect the distance function in Equation (2) used to calculate the nearest neighbor, but as long as $u < m$, the distance function can still reflect to some extent the true distance between two points in the signal space. Therefore, the LEMT algorithm is robust to small environmental changes.

IV. Experimental Evaluation

In order to evaluate the performance of our proposed algorithm, extensive experiments were carried out on two different test-beds: a WLAN-based environment and a RFID-based network environment. For comparison, three different algorithms were used as baselines. The first one is the Maximum Likelihood (ML) method, which is an essential fingerprint-based algorithm [10] [23]. This baseline is used to show the effect of dynamic environments on the localization accuracy. The other two approaches are used to test the sensitivity of adaptive algorithms against reference points. The first one is the interpolation-based algorithm used in the LEASE system [9], and the second one is the localization algorithm used in the LANDMARC system [11]. For LANDMARC, we set the number of nearest neighbors to be four because our experiments show that the highest accuracy is usually obtained at this point, as pointed out in [11]. In addition, since data calibration is labor-intensive and time-consuming, our experiments were designed to test the localization accuracy of different algorithms only based on limited training data collected at a single time instant. Therefore, in our experiments, we used the data collected at midnight for training, which span several hours in length, and tested different algorithms at different time periods including night and daytime.

A. Experiments on WLAN Data

We conducted the experiments in a section of the third floor of the Academic Building where the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology is located. The building is deployed with an IEEE 802.11b wireless network in the 2.4 GHz frequency bandwidth. The layout of the experimental test-bed is shown in Figure 6. This area measures 30m $\times$ 15m. We chose eight available PC machines along the horizontal hallway, each of which is equipped with a Linksys Wireless-B USB Network adapter, as the reference points. The placement of reference points is marked with solid circles in the figure. In this environment, nine APs can be detected, of which five APs distributed within this areas are marked with blank triangles in the figure. The other four APs are located either on the same floor outside this area or on the different floors. On average, the number of APs covering a location is six. In addition, an IBM 1.29GHz laptop with a Linksys Wireless-B USB Network adapter served as the tracked mobile client in our experiment. To make our RSS measurements, we developed an API program running under Windows XP to actively scan for APs, based on the NDIS User Mode I/O (NDISUIO) driver [13] provided by Microsoft.

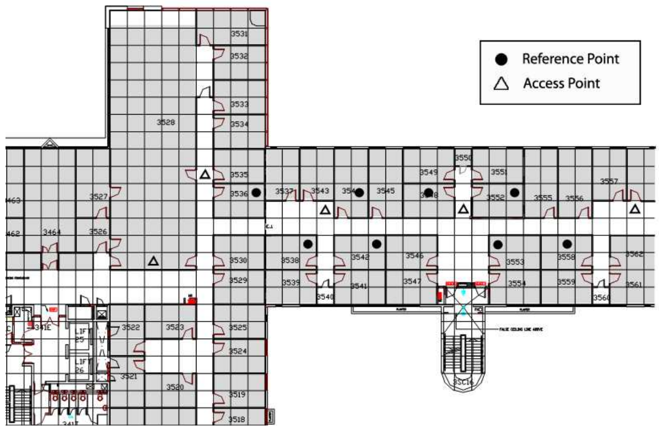
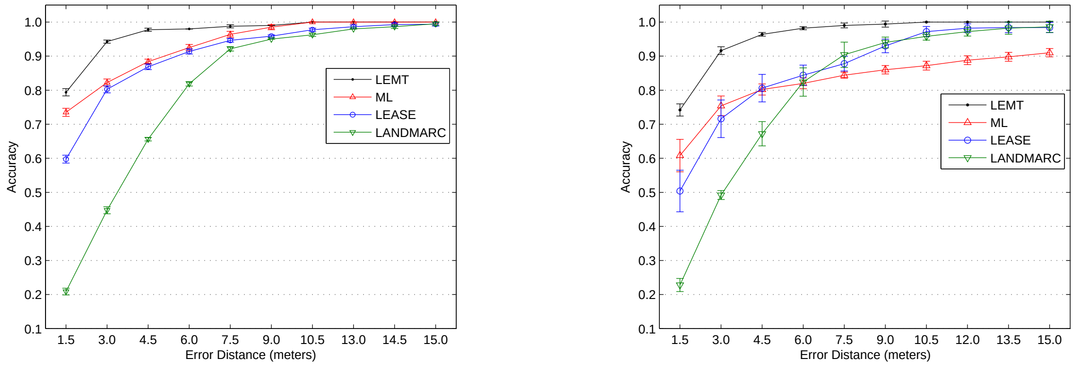


Fig. 6. The layout of the experimental test-bed

With the placement of reference points shown in the figure, we repeatedly collected RSS samples at the reference points over different time periods across three days. While the data were continuously collected at the reference points, two persons simultaneously used an IBM laptop to collect RSS samples at various positions in the horizontal hallway, along which reference points are placed. Each grid has a size of 1.5×1.5 meters, and we have a total of 55 grids. At each grid, RSS samples were collected at various positions and with different orientations. In the collection process, each scan of the APs produces a signal vector. We had 10 active scans every second and took the mean as one sample because we may miss some APs in a single scan. At each grid, 90 samples were collected separately for training and testing at different time periods.

To test the validity of our LEMT algorithm, we partitioned the data set into two separate parts: night and daytime.

- Let D_{night} be the data set collected at the time period t_n, where t_n is between 8:00 PM and 12:00 AM at *night*.



(a) Night

(b) Daytime

Fig. 7. Localization accuracy vs. different error distances

$D_{night} = \{Y_i, 1 \leq i \leq n = 55\}$ is a collection of 90 RSS samples $Y_i = \{S_1, S_2, \ldots, S_{90}\}$ calibrated at each of 55 grids. D_{night} is used later for building the training data.

- Let D_{day} be the data set collected at the time period t_d, where t_d is between 8:00 AM and 4:00 PM during the *daytime* over three days. $D_{day} = \{Y_i, 1 \leq i \leq n\}$ is a collection of 450 samples $Y_i = \{S_1, S_2, \ldots, S_{450}\}$ collected during the time period t_d at each of the 55 grids.

We took special care to account for small-scale variations of the RSS samples defined in [21], where "small-scale variations" refer to significant signal-strength changes at small distances. At each grid, we allowed the person to vary his positions and orientations while collecting the 90 samples. However, since the orientations of the person were not typically changed within one second, we did not average signal-strength values of different directions. In addition, we collected the test data in a time span of three days to capture day-to-day variations of the RSS samples.

1) Impact of environmental factors: Experiments were first performed to compare the four algorithms (LEMT, ML, LEASE and LANDMARC) with respect to their ability to adapt to the environmental factors. In this experiment, we used the RSS samples D_{night} collected at night to train the radio map for ML and LEMT. We bootstraped D_{night} to build different training sets as follows. Let Tr_n be a set of samples $\{Y_i', 1 \leq i \leq n\}$, where each Y_i' is a subset of 45 RSS samples that are randomly selected from 90 samples Y_i at each grid. Tr_n is used as one set of training data in our experiments, and repeating this process provides us with different training data sets. For each set of Tr_n, we derive a non-overlapping subset of samples $Ts_n = \{Y_i - Y_i', 1 \leq i \leq n\}$ as the testing samples during the time period t_n.

For ML, Tr_n were used as the training data. For LEMT, Tr_n and the data calibrated at the reference points were used for training. The test data sets Ts_d for daytime were constructed similarly from the data set D_{day}, by randomly selecting 45 samples at each of the 55 grids from the data set D_{day}. For LANDMARC and LEASE, the data Ts_n or Ts_d were directly used for localization because training is not required for the two systems. The four algorithms were tested on the disjoint data sets Ts_n and Ts_d, for night and daytime, respectively.

Figure 7(a) shows the localization accuracy tested at night (8:00 PM to 12:00 AM) with respect to different error distances. Here the error distance is defined as distance between the predicted grid and the actual grid during the localization phase. For each value of the error distance, we tested the performance of the four algorithms over 10 trials. For training in LEMT and ML, we used the data D_{night} collected at night, where we randomly selected Tr_n for each of the 10 trials. For testing, we took the corresponding disjoint data Ts_n, also over 10 trials. We can see from the figure that LANDMARC performs poorly because it cannot work well with such a sparse density of reference points. As a whole, LEMT outperforms the other three algorithms. For example, LEMT can achieve the accuracy of 95% within three meters at night. We can also observe that, the variations in accuracy for each algorithm are very small. This is because the environmental conditions in the department at night are relatively static, when the building is quiet. Therefore, the variations of RSS samples collected at night are relatively small. Also, for ML, since the static radio map built offline can model the RSS samples collected in the online phase, it can be observed to outperform LANDMARC and LEASE by making use of the training process.

Figure 7(b) shows the localization accuracy tested during the daytime at different time periods with respect to different error distances. Similar to the night time, we performed 10 trials for each value of error distances. In each trial, for ML and LEMT, we used Tr_n collected at night for training, and randomly selected testing data Ts_d during the daytime period for testing. The same Ts_d were also used in testing LEASE and LANDMARC. We can see from the figure that LEMT can achieve higher accuracy than ML. Also, LEMT has much smaller variance in accuracy than ML over different daytime periods. This is because the environment during the daytime is more complex than at night due to people moving, doors opening or closing, changing temperatures, as well as other environmental conditions. These conditions cause the RSS samples measured during the daytime to be significantly different from those stored in the radio map, which was collected at night. Therefore, the performance of ML may degrade dramatically depending on the environmental dynamics. Accordingly, ML can also be observed to perform worse than LEMT and LANDMARC at certain error distances. We can also observe that LEMT outperforms LEASE and LANDMARC by a large margin. Subject to dynamic environmental conditions, the real-time RSS samples measured at reference points and the mobile client may vary a lot over different daytime periods. Therefore, for LEASE and LANDMARC, the accuracy varies much over different daytime periods. In contrast, by using training data, LEMT can achieve higher accuracy with smaller variance.

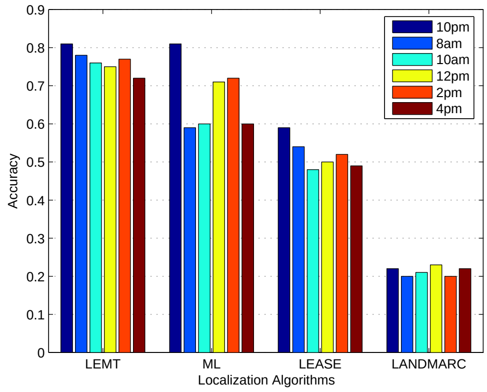


Fig. 8. Localization accuracy within 1.5 meters vs. different time periods

Figure 8 compares the localization accuracy of the four algorithms at six different time periods including 10:00 PM, 8:00 AM, 10:00 AM, 12:00 PM, 2:00 PM and 4:00 PM. Similarly, we performed 10 trials for each time period and plotted the mean value. For each trial at a certain time period, we used Tr_n collected at night for training, and used Ts_n or Ts_d at the corresponding time period for testing. We can see from the figure that LEMT and ML can achieve comparable localization accuracy at 10pm, a quiet time in the department. This is because the environmental conditions at night are relatively static. For ML, the radio map built in the training phase can accurately model the RSS samples observed in the localization phase in these quiet time periods. Therefore, there is not much difference in accuracy between ML and LEMT. However, the situation is quite different in the daytime periods, when LEMT can be seen to outperform ML to a large extent. Also, LEASE and LANDMARC perform poorly in this environment with a low density of reference points.

2) Impact of reference points: We also carried out experiments to investigate the effect of reference points on the localization accuracy. Intuitively, the placement and number of reference points are related to the technique used to build the model. For LEMT, the model is built by first dividing the whole reference-point value space into sub-regions and then fitting a different linear function to each sub region. In each sub region, at least two reference points are needed for reasonable smoothing in order to learn a linear function. Thus, we divided the horizontal hallway into four sub squares with approximately equal area, in each of which at least two reference points are placed on two sides respectively along the hallway.

Figure 9 shows the localization accuracy within 1.5 meters by varying the number of reference points. In this experiment, for both LEMT and ML, we still randomly chose 45 samples from the data D_{night} collected at night for each grid as the training data. The testing data were 45 samples randomly chosen from the data D_{day} collected at different daytime periods. For a given number of reference points, we chose 20 random subsets of reference points and compared the four algorithms. In the figure, the accuracy of ML is a horizontal line because it does not utilize the information about reference points. We can see that, as the number of reference points increases, the accuracy values of LEASE and LANDMARC increase as well.

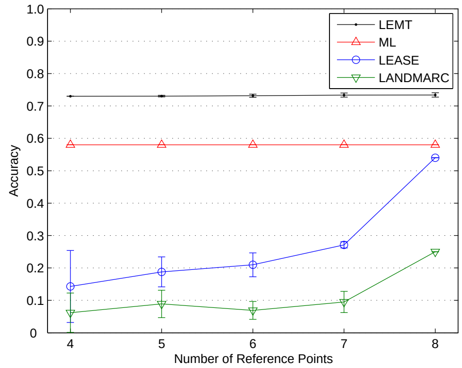


Fig. 9. Localization accuracy within 1.5 meters vs. different numbers of reference points

In this experiment, although LEASE can be seen to outperform LANDMARC, their performance depends much on the number of reference points. When the number of reference points is eight, the performance of ML and LEASE is comparable to each other. Another interesting observation is that, the accuracy of LEMT is insensitive to the number of reference points. This is because LEMT always chooses an optimal subset of reference points to construct the model tree according to their capability of predicting the RSS value received at the mobile client, even when more reference points are provided. From the perspective of system design, it is difficult to specify the appropriate number of reference points before the system starts to work. Therefore, LEMT is more feasible than LEASE and LANDMARC in time-dependent location-based applications.

3) Impact of access points: Experiments were also conducted to study the effect of the number of APs on the localization accuracy. In this experiment, 45 samples at each location collected at night were randomly chosen from D_{night} for training, and 45 samples in the daytime D_{day} were used for testing. For a given number of APs, we chose 20 random subsets of all the nine APs and ran the four algorithms 20 times. Figure 10 shows the accuracy within 1.5 meters with respect to different numbers of APs. We can see that, as the number of APs initially increases from one, the accuracy of the four algorithms increases and the variances in accuracy decrease at the same time. This is because when more APs are used, we have more information for localization and thus the systems become more robust. One interesting observation is that, when the number of APs increases to six, the accuracy of ML begins to decrease with more added APs. This occurs because, as the number of APs increases, more information is added for localization while more noise is incurred on the other hand. Therefore, ML can achieve the best performance using a subset of APs. This result is consistent with the work of [3], in which an optimal subset of APs is claimed to be able to produce the highest localization accuracy. For LEMT, when the number of APs increases to six, the accuracy remains almost the same. Therefore, we only need six APs to accurately locate a mobile client in our WLAN-based environment.

From the experiments on WLAN data, we can conclude that LEMT can adapt best to the dynamics of environmental conditions by leveraging the offline training process and reference points. If the environmental conditions are relatively stable, ML can outperform LANDMARC and LEASE by taking advantage

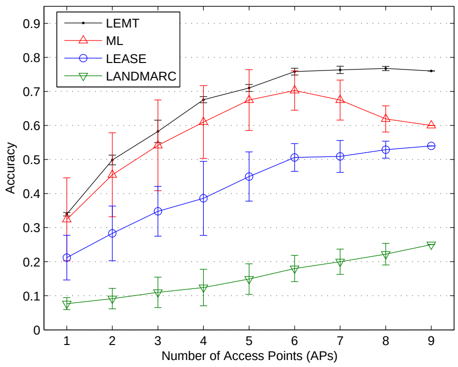


Fig. 10. Localization accuracy within 1.5 meters vs. different numbers of APs

of the offline training process. In contrast, if the environmental conditions change a lot over time, LANDMARC and LEASE can perform better than ML by using densely deployed reference points. However, since the reference points are sparsely deployed in the WLAN-based environment, ML usually outperform LEASE and LANDMARC in most cases.

B. Experiments on RFID Data

To show the generality of our LEMT algorithm, we also conducted a series of experiments on real data sets collected from a RFID-based network environment. For data calibration, we used the RF Code MANTIS^{TM} active readers and tags [14] in our experiment. The operating frequency is 303.8 MHZ and the transmission range is up to 1500 feet. Each RFID reader can detect up to 500 tags in 12.5 seconds. Each tag is pre-programmed with a unique 8-character ID for identification by readers. RFID readers detect and interpret the radio frequency beacon emitted by RFID tags in order to identify them and provide signal-strength information to determine their locations.

1) RFID Experimental Setup: In our standard setup shown in Figure 11(a), we place four RFID readers (p=4) and 16 tags (m=16) as reference points in our pervasive computing lab. The reference tags are placed every one meter apart from each other (each grid is 1m $\times$ 1m), which are marked with blank squares in the figure. Another tag that is placed at different positions within each grid serves as the tracked object in our experiment.

With the placement of the reference tags and the tracked tags shown in the figure, we collected two groups of RSS samples from four RFID readers continuously. A first data set D_1 was collected at multiple nights from 12:00 AM to 6:00 AM when there is little noise. The other data set D_2 was collected during multiple days from 2:00 PM to 6:00 PM, during which various activities were carried out in our lab that would result in different levels of noise. Specifically, each RSS sample is a 4-dimensional vector. To locate a tracked tag, we took 200 RSS samples for both D_1 and D_2, at a sampling rate of one sample every two seconds, at the tracked tag and the reference tags. For LEMT and ML, to locate a RFID tag, we randomly chose 100 samples at each grid from D_1 for training and 100 samples at each grid from D_2 for testing, or vice versa. For LEASE and LANDMARC, the process of training is not required.

a) Impact of environmental factors: We first performed experiments to compare the localization accuracy of the four algorithms with respect to their adaptive abilities to the environmental conditions. To avoid statistical variability, the reported results are based on 10 trials. Figure 12(a) shows the accuracy tested at night with respect to different error distances. For ML and LEMT, 100 samples randomly chosen at each grid from D_2 were used for training. The four algorithms were tested on an independent data set, which consists of 100 samples randomly selected from D_1. In the figure, LEMT can be seen to outperform the other three algorithms, and the performance of LEASE and LANDMARC is comparable. Figure 12(b) shows the accuracy tested in the daytime with respect to different error distances. For ML and LEMT, 100 samples randomly selected at each grid from D_1 were used for training. We also tested the four algorithms on an independent data set, which consists of 100 samples randomly chosen for each grid from D_2. Note that we adopt the same process of training and testing in the following experiments. We can see that, the performance of LEMT is close to that of LEASE and LANDMARC, whereas three of them outperform ML. From this part of experiments, we can conclude that, LEMT outperforms ML, LEASE and LANDMARC, while the performance of LEASE and LANDMARC is still good. For example, both of them can achieve about 90% accuracy within 2 meters, with a 1m $\times$ 1m density of reference tags.

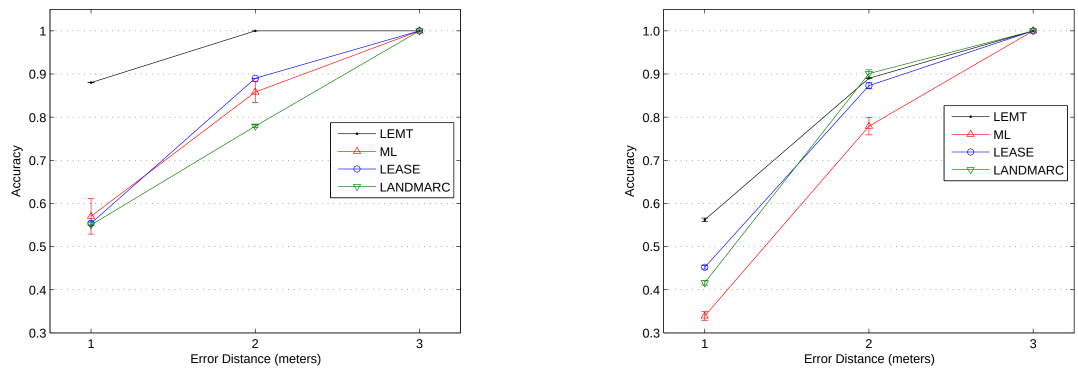
b) Impact of RFID readers: We performed another set of experiments to investigate the effect of the number of RFID readers on the localization accuracy. Figure 13 shows the accuracy within 2 meters using the four algorithms with respect to different numbers of RF readers. For each grid, we still randomly chose 100 samples from one group as the training data, and 100 samples from the other group as the test data. For a given number of RFID readers, we chose 10 random subsets of RFID readers and ran the four algorithms. Figure 13(a) and Figure 13(b) show the localization accuracy tested at night and in the daytime respectively. We can see that, in general, the accuracy of the four algorithms increases as the number of RF readers increases. Moreover, LEASE and LANDMARC can usually outperform ML in most cases, with such a high density of reference points.

c) Impact of reference tags: Experiments were also carried out to study the effect of the number of reference tags on the localization accuracy. Figure 14 compares the accuracy within 2 meters using the four algorithms with respect to different numbers of reference tags. In this experiment, the number of readers is fixed at 4. For each grid, 100 samples randomly chosen from one group were used for training and 100 samples from the other group for testing. For a given number of reference tags, we randomly chose 20 subsets of reference tags and ran the four algorithms. We can see that, the accuracy of ML is a horizontal line because it does not use the reference points. As the number of reference tags increases, the accuracy of LEASE and LANDMARC increases and the highest accuracy can be achieved when the number of reference tags is 16. In Figure 14(a) and 14(b), we can observe that, the performance of LEASE and LANDMARC is close to or better than that of ML, when the number of reference points reach 15 and 16, respectively. In contrast, the performance of LEMT does not depend on the number of reference tags to a large extent because it can intelligently choose an optimal subset of reference tags.

2) RFID Experimental Setup with a lower density of reference tags: In order to study how the density of reference tags affects the localization accuracy, we conducted our experiments on

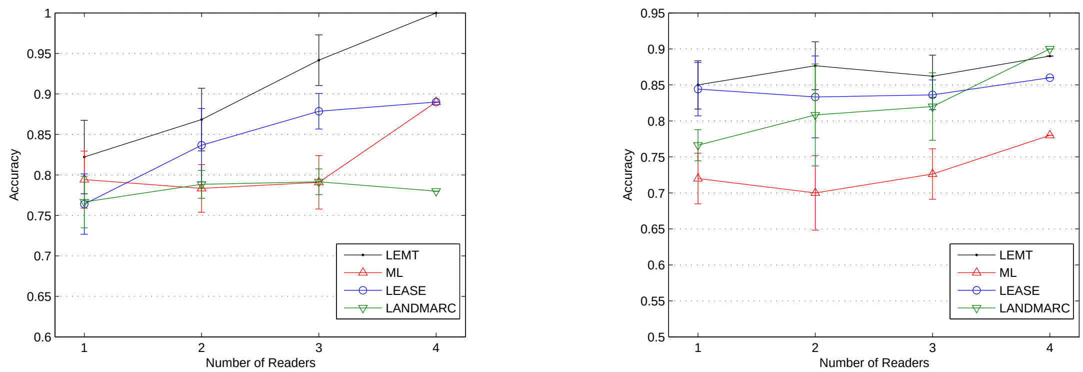
(a) (b)

Fig. 11. Experimental setup: (a) Standard placement of RF readers and reference tags; (b) Placement of RF readers and reference tags with a lower density



(a) Night (b) Daytime

Fig. 12. Localization accuracy vs. different error distances



(a) Night (b) Daytime

Fig. 13. Localization accuracy within 2 meters vs. different numbers of RFID readers

another setup with a lower density of reference tags, as shown in Figure 11(b). In this setup, we place four RFID readers (p=4) and 15 reference tags (m=15) in our lab. The reference tags are placed every 2 meters apart from each other and thus each grid is 2m $\times$ 2m. The tracked tags are placed at different positions marked with solid dots in the figure. Similar to the previous standard setup, we collected two groups of data D_1 at night and D_2 in the daytime respectively. Also, in the following experiments, we use the same

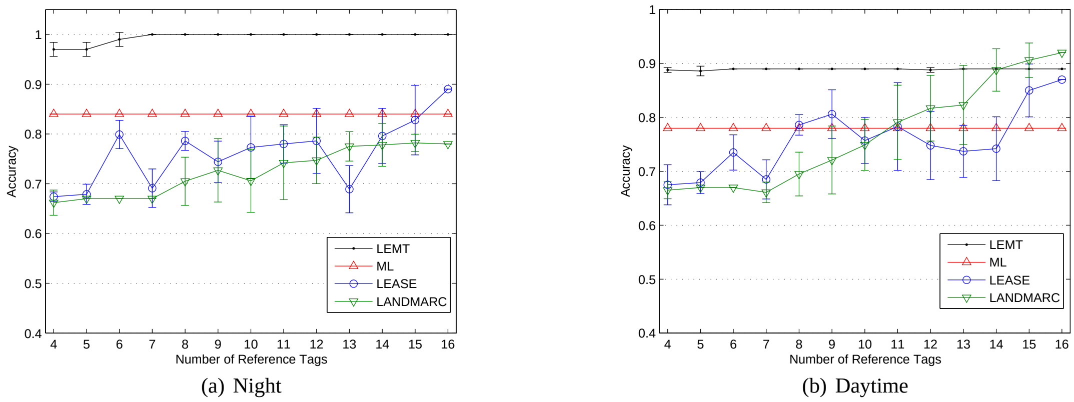


Fig. 14. Localization accuracy within 2 meters vs. different numbers of reference tags

experimental procedure as in the previous standard setup.

a) Impact of environmental factors: Figure 15 shows the accuracy of the four algorithms tested in two different environments. We can see that, LEMT consistently yields higher accuracy than the other three algorithms. Let us further compare Figure 15 with Figure 12 to analyze the effect of the density of reference tags on the localization accuracy. It is clearly noticed that the accuracy of LEASE and LANDMARC decreases significantly with a lower density of reference tags. For example, the accuracy of LANDMARC at night decreases from 78% to 48% and the accuracy in the daytime decreases from 91% to 48% within 2 meters. In contrast, by making use of training process, LEMT can achieve higher accuracy than LEASE and LANDMARC with a lower density of reference tags. We can also see from Figure 15, ML can outperform LEASE and LANDMARC in most cases with a sparse deployment of reference points.

b) Impact of RF readers: Experiments were also carried out to investigate the effect of the number of RF readers on the accuracy in this experimental setup. Figure 16 shows the accuracy within 2 meters with respect to different numbers of RF readers. Again, the accuracy of the four algorithms increases as the number of RF readers increases.

c) Impact of reference tags: Again, experiments were performed to investigate the effect of the number of reference tags on the localization accuracy. Figure 17 shows the accuracy within 2 meters with respect to different numbers of reference tags. Similarly, the number of readers is set to be 4 in this experiment. We can conclude from the figure that LEMT is less sensitive to the number of reference tags than LEASE and LANDMARC.

From the experiments on RFID data, we can conclude that LEMT can also perform best to offset dynamic environmental changes in two different experimental setups shown in Figure 11. When the envionmental conditions change a lot, LEASE and LANDMARC can outperform ML with a dense deployment of reference points (Figure 11(a)). However, if reference points are sparsely deployed (Figure 11(b)), the performance of LEASE and LANDMARC may remarkably degrade and they become less accurate than ML.

C. Experimental Summary

Based on extensive experiments presented above, we now summarize the advantages of our LEMT algorithm as follows:

1) By using reference points, LEMT is more robust than ML when the environmental conditions change over time.
2) LEMT is more accurate than LEASE and LANDMARC by making advantage of the offline training process, in particular, in a lower density of reference points.
3) LEMT is less sensitive to the number of reference points than LEASE and LANDMARC.

Therefore, LEMT can achieve better localization performance by leveraging the offline training process and real-time RSS samples received at reference points.

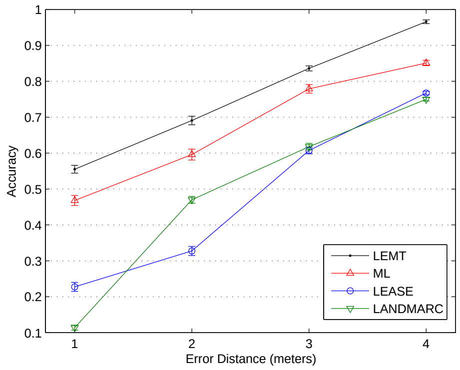
Furthermore, we summarize the comparison among the other three algorithms (ML, LANDMARC and LEASE) as follows:

1) If the environmental conditions are relatively static, ML can outperform LANDMARC and LEASE by taking advantage of the training process.
2) If the environmental conditions change dynamically over time, LANDMARC and LEASE can outperform ML with a dense deployment of reference points, as shown in our experiments on RFID data using the experimental setup in Figure 11(a). However, if reference points are sparsely deployed, as shown in our experiments on both WLAN data and RFID data using experimental setup in Figure 11(b), ML can still achieve higher accuracy than LANDMARC and LEASE.

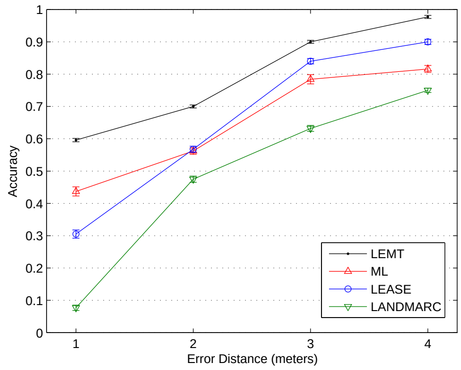
V. Conclusions and Future Work

In this paper, we have proposed a novel RF-based location-estimation algorithm called LEMT, which can well adapt to dynamic environmental changes. Our extensive experiments show that the LEMT algorithm can achieve a large advantage over the ML method in terms of localization accuracy using adaptive temporal maps via reference points. Compared with existing adaptive techniques, LEMT is much more robust to the reduction in the number of reference points. For LEMT, the number of reference points and signal transmitters is known, but their physical locations are not required as an input.

Our work can be extended in several directions. First, we will consider reducing the online computational complexity of the LEMT algorithm. LEMT has relatively high computational overhead, mainly due to the model tree algorithm used to estimate the signal strength that may be received at all the locations, and the nearest neighbor method used to search the best location

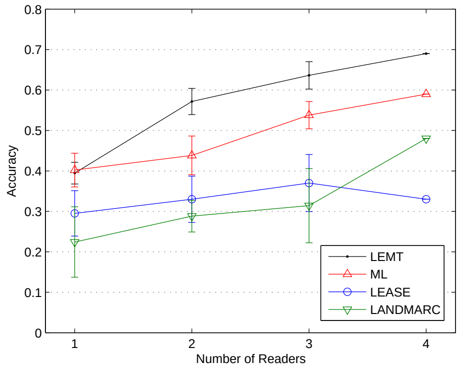


(a) Night

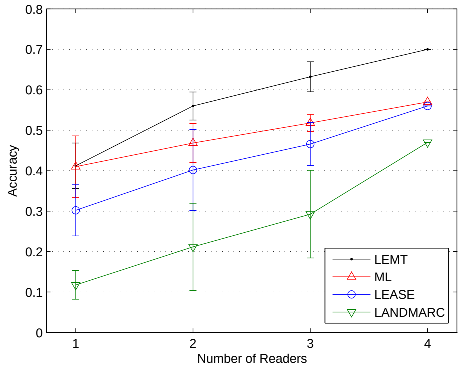


(b) Daytime

Fig. 15. Localization accuracy vs. different error distances

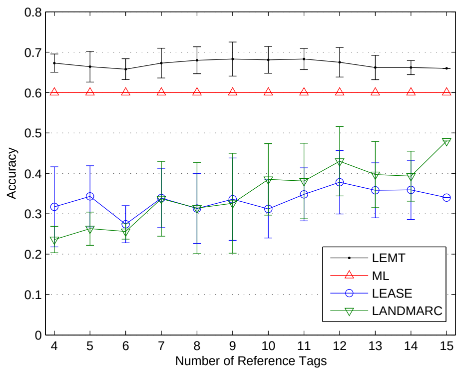


(a) Night

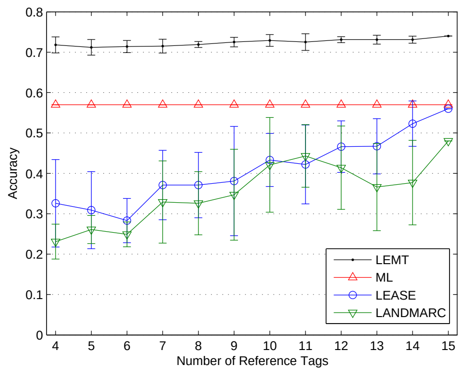


(b) Daytime

Fig. 16. Localization accuracy within 2 meters vs. different numbers of RFID readers



(a) Night



(b) Daytime

Fig. 17. Localization accuracy within 2 meters vs. different numbers of reference tags

in the location space. In addition, the computational complexity increases as the number of signal transmitters increases. However, the model tree algorithm itself does not incur much computational overhead in the online phase, because it only requires comparison operations when walking along the tree to estimate the signal strength that may be received by the mobile device. Therefore, the computational complexity of the LEMT algorithm can be further reduced by using clustering techniques [23] or by intelligently selecting signal transmitters [3]. Second, we will also consider applying additional nonlinear approaches to build the radio map at each grid point using the signal-strength values received at the reference points. Third, we wish to incorporate the users' movement trajectories to further improve the localization accuracy of the LEMT algorithm.

ACKNOWLEDGEMENT

Jie Yin is supported by Tasmanian ICT Centre, which is jointly funded by the Australian Government through CSIRO and the Intelligent Island Program administered by the Tasmanian Department of Economic Development. The authors also thank the Hong Kong RGC Grant HKUST6187/04E and HKUST6183/05E, Hong Kong CAG Grant HKBU1/05C, and the National Basic Research Program of China (973 Program) under grant No. 2006CB303000. We also thank the anonymous referees for their valuable comments.

REFERENCES

[1] P. Bahl and V. N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *Proceedings of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, pages 775–784, Tel-Aviv, Israel, March 2000.

[2] E. S. Bhasker, S. W. Brown, and W. G. Griswold. Employing user feedback for fast, accurate, low-maintenance geolocationing. In *Proceedings of the IEEE International Conference in Pervasive Computing and Communications (PerCom)*, Orlando, FA, USA, March 2004.

[3] Y. Chen, Q. Yang, J. Yin, and X. Chai. Power-efficient access-point selection for indoor location estimation. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 18(7):877–888, 2006.

[4] D. Fox, J. Hightower, L. Liao, and D. Schulz. Bayesian filtering for location estimation. *IEEE Pervasive Computing*, 2(3):24–33, 2002.

[5] R. J. Freund and W. J. Wilson. *Regression Analysis: Statistical Modeling of a Response Variable*. Academic Press, 1998.

[6] C. Gentile and L. K. Berndt. Robust location using system dynamics and motion constraints. In *Proceedings of IEEE International Conference on Communications (ICC)*, pages 1360–1364, Paris, France, June 2004.

[7] A. Haeberlen, E. Flannery, A. Ladd, A. Rudys, and D. Wallach L. Kavraki. Practical robust localization over large-scale 802.11 wireless networks. In *Proceedings of the Tenth ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 70–84, Philadelphia, PA, USA, September–October 2004.

[8] H. Hashemi. The indoor radio propagation channel. *Proceedings of the IEEE*, 81(7):943–968, July 1993.

[9] P. Krishnan, A.S. Krishnakumar, W. H. Ju, C. Mallows, and S. Ganu. A system for LEASE: Location estimation assisted by stationary emitters for indoor RF wireless networks. In *Proceedings of the Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, pages 1001–1011, Hong Kong, China, March 2004.

[10] A. Ladd, K. Bekris, G. Marceau, A. Rudys, L. Kavraki, and D. Wallach. Robotics-based location sensing using wireless Ethernet. In *Proceedings of the Eighth Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 227–238, Atlanta, GA, USA, September 2002.

[11] L.M. Ni, Y. Liu, Y.C. Lau, and A.P. Patil. LANDMARC: Indoor location sensing using active RFID. In *Proceedings of IEEE International Conference in Pervasive Computing and Communications (PerCom)*, pages 407–415, Dallas, TX, USA, March 2003.

[12] J. R. Quinlan. Learning with continuous classes. In *Proceedings of Australian Joint Conference on Artificial Intelligence*, World Scientific, Singapore, 1992.

[13] NDIS Developer's Reference. Available at: `http://www.ndis.com/`.

[14] Inc. RF code. Available at: `http://www.rfcode.com`.

[15] 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):155–164, July 2002.

[16] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

[17] Y. Wang and I. Witten. Inducing model trees for continuous classes. In *Proceedings of the Ninth European Conference on Machine Learning*, pages 128–137, Prague, Czech, April 1997.

[18] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, 2000.

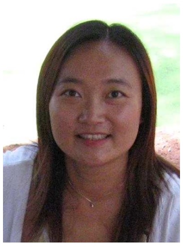
[19] J. Yin, X. Chai, and Q. Yang. High-level goal recognition in a wireless LAN. In *Proceedings of the Nineteenth National Conference in Artificial Intelligence (AAAI)*, pages 578–584, San Jose, CA, USA, July 2004.

[20] J. Yin, Q. Yang, and L.M.Ni. Adaptive temporal radio maps for location estimation. In *Proceedings of IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 85–94, Kauai Island, HW, USA, March 2005.

[21] M. Youssef and A. Agrawala. Small-scale compensation for wlan location determination systems. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, volume 3, pages 1974 – 1978, New Orleans, LA, USA, March 2003.

[22] M. Youssef and A. Agrawala. Handling samples correlation in the horus system. In *Proceedings of the 23rd IEEE Conference on computer communications and networking (INFOCOM)*, pages 1023–1031, Hong Kong, China, March 2004.

[23] M. Youssef, A. Agrawala, and U. Shankar. WLAN location determination via clustering and probability distributions. In *Proceedings of IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 143–150, Dallas, TX, USA, March 2003.



Jie Yin received the BE degree from the Computer Science Department, Xi'an Jiaotong University, China, in 2001 and the PhD degree from the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology, Hong Kong, in 2006. She is currently a researcher at the CSIRO ICT Centre, Australia. She is working on outlier detection and event detection in wireless sensor networks. Her research interests include data mining, machine learning, sensor-based activity recognition, data processing in sensor networks, and decision support systems. More information about her can be found at http://www.ict.csiro.au/staff/Jie.Yin/

Qiang Yang received the PhD degree from the University of Maryland, College Park. He is a professor of the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology. He is also an adjunct professor at Peking University, Beijing, and at Zhongshan University in Guangzhou, China. His research interests include AI planning and sensor-based activity recognition, machine learning and case-based reasoning, and data mining. He is a senior member of the IEEE, the AAAI, and the ACM, and an associate editor for the IEEE Transactions on Knowledge and Data Engineering and IEEE Intelligent Systems, as well as the International Journal of Knowledge and Information Systems. More information about him can be found at http://www.cse.ust.hk/˜qyang.

Lionel M. Ni is Chair Professor and Head of the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology (HKUST). He is Chief Scientist of the National Basic Research Program of China (973 Program) on wireless sensor networks and Director of HKUST's Digital Life Research Center. He earned his Ph.D. degree in Electrical and Computer Engineering from Purdue University in 1980. His research interests include high performance computer systems, high-speed networks, pervasive computing, wireless sensor networks, and distributed systems. A fellow of IEEE, Prof. Ni has chaired many professional conferences and has received a number of awards for authoring outstanding papers. More information about him can be found at http://www.cse.ust.hk/˜ni.