GEM : A Learning Approach for Indoor Localization

[Paper Id : 1569469851. Number of Pages : 12]

Abhishek Goswami, Luis E. Ortiz, Samir R. Das
Computer Science Department
Stony Brook University
agoswami,leortiz,samir@cs.stonybrook.edu

ABSTRACT

We consider the problem of localizing a wireless client in an indoor environment based on the signal strength of its transmitted packets as received on stationary sniffers or access points. Current state-of-the art indoor localization techniques have the drawback that they rely extensively on a 'training' phase. This 'training' is a labor intensive process and must be done for each target-area under consideration for various device types. This clearly does not scale for large target areas. The introduction of unmodeled hardware with heterogeneous power-levels etc further reduces the accuracy of these techniques.

We propose a solution in which the received signal strength is modeled as a Gaussian Mixture Model (GMM). Expectation maximization (EM) is used to find the parameters of our GMM. This approach is able to provide a location fix for a transmitting device based on the maximum likelihood estimate. This way, not only the costly training phase is avoided but the location estimates are much more robust in the face of various form of heterogeneity and time varying phenomena. We present our results on two different indoor testbeds with multiple WiFi devices (iphones, android, laptops, netbooks). We demonstrate that the accuracy is at par with state-of-the-art techniques but without requiring any training.

1. INTRODUCTION

Over the past decade, the increasing use of wireless networking has fueled the use of wireless links to localize wireless clients in indoor spaces. This issue is increasingly finding attention both from research and business communities as a perfect, general-purpose solution such as outdoors GPS has been elusive. Close scrutiny of available techniques reveal that the more successful techniques require a substantial 'pre-deployment' effort by way of creating RF maps, for example. Technically, this is equivalent to 'training' in a learning technique. Finer grain map creation makes localization more accurate, but requires more effort. This additional burden has made these localization techniques less appealing in practice.

The received signal strength (RSS) based techniques are the most popular as commodity wireless devices are all capable of measuring RSS. Two general directions have emerged – (i) client-based approach [10, 9, 21, 5, 12, 22] and infrastructure-based approach [6, 13, 19, 11]. In the client-based approach, the client device measures the RSS as seen by it from various APs (access points). This information is used to locate the client. In the infrastructure-based approach, the network administrator can use simple sniffing devices (or APs doubling as sniffers) to monitor clients and record RSS from the client transmissions. This sniffed RSS is used to localize the client. The infrastructure-based approach is more attractive for large scale deployment, because any arbitrary client without any specific installed application can still localize itself with the assistance of the infrastructure. It is also easier to deploy, manage and maintain.

In the discussion that follows, we specifically focus on WiFi-based localization using an infrastructure-based approach. WiFi is chosen because of the popularity of WiFi devices and WiFi-based WLAN systems. But the techniques used are not specific to any link layer technology.

1.1 Limitations of Pre-Deployment Effort

In the existing indoor WiFi localization solutions the first phase is a pre-deployment 'offline phase' or training phase aimed at building detailed RF maps or RF propagation models based on a survey of the target area. The second phase is the 'online phase,' where a localization algorithm is used to provide a location estimate for an observed set of RSS measurements from the transmissions from a mobile client to be localized. There are three major drawbacks for this general approach.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
ACM CoNEXT 2010, November 30 - December 3, 2010, Philadelphia, USA.
Copyright 2009 ACM X-X-X-X/XX/XX ...$5.00.

1. The device used during the 'offline phase' may differ from the target device in the 'online phase.' Unmodeled hardware devices operating at different transmit power levels can introduce significant variations in the signal patterns between the training device and the target device. This adversely affects the accuracy of location estimation [20]. Experiments described later in this paper indicate how hardware variance between four common commodity WiFi devices can significantly degrade the positional accuracy of two commonly used localization algorithms.

2. The 'offline phase' itself involves labor-intensive sampling of signal strength values at discretized locations in the target space. Again, experiments show that location accuracy depends significantly on the granularity of the training locations. If the training locations are sparse, the location estimates become substantially poorer.

3. Static models built during the 'offline phase' cannot counter time varying phenomena like movement of people, changing occupancy and surroundings etc. Most 'killer' applications of indoor localization would be in large shopping malls, airports, convention centers etc., where such changes would be routine. On the other hand, due to the reason 2 above, such models are difficult update regularly.

1.2 Approach

We propose GEM, a novel localization algorithm that uses the *Gaussian Mixture Model* (GMM) and solves the model to determines maximum likelihoods using *Expectation Maximization* (EM). GEM leverages the infrastructure based model while eliminating any pre-deployment effort. Packet transmissions made by a client are received on stationary sniffers (or access points doubling as sniffers) that extract the RSS and MAC id of the target client and report this information to a central localization server. Using this information, GEM builds a model for the target device and provides a location estimate. The estimate can be made available to the client via a simple web-based application, e.g.

GEM provides several key benefits by eliminating the 'offline phase.' First, by building a model for each target device effectively addresses the hardware variance problem. Thus GEM can be used across heterogeneous devices, each operating at different power levels. Second, zero pre-deployment effort makes GEM particularly attractive for large indoor spaces. Third, GEM is a purely online algorithm : the model parameters get updated and modified based on real-time RSS observations. As such, GEM is able to adapt to dynamic changes in the target space.

Our results of deploying GEM in two different office buildings are promising. We specifically note that when measurements made using one device are used to localize a different device, GEM is seen to perform better that RF signal map based techniques like RADAR[x] and Probabilistic[y]

2. RELATED WORK

In this section we provide a brief overview of some fundamental techniques in the field of indoor localization. Over the past two decades, this field has seen tremendous push, both from the research community and from industrial circles. The advent of pervasive and mobile computing has fueled tremendous interest in this field in recent years.

Calibration-free techniques: An indoor path-loss propagation model essentially forms the bedrock for these techniques. In RADAR [1] Bahl et al give a indoor radio propagation model to calculate RSS at various locations in the building based on the distance, number of walls etc. The NNSS metric is then used to estimate the location of the mobile user by matching the observed RSS to the theoretically computed signal strength values at these locations. Both [6] and [13] give sniffer based techniques for localization based on propagation models. In [6] Moraes et al use a naive propagation model to generate a 'radio propagation map' at each sniffer. They use RSS measurements between the sniffers and a reference Access Point(APRef) to reconstruct the RPM, either periodically or when there are synificant variations in the RSS. A probabilistic model is then used to give a location estimate. In [13] Lim et al consider online measurements of RSS between 802.11 APs and between a client and its neighboring APs, to create a mapping between the RSS measure and the actual geographic distance. TIX [9] by Gwon et al uses a similar setting whereby inter-AP and client-AP RSS measurements are used to perform linear interpolation for estimating the RSS at distinct locations in the target space. In [14] Madigan et al propose a client-based scheme that uses a bayesian hierarchical graphical model. By making the assumption that different access points behave similarly, they develop a model which avoids the need to know the location of training points. *While most of these schemes are designed to be responsive to real time changes in the environmental dynamics of the target space, none of them model variations in client hardware and transmission power , factors which can significantly degrade the accuracy estimates of RSS based WiFi localization schemes.*

Techniques that build RF signal maps: Several client-based schemes and infrastructure-based schemes rely on RF signal maps for localization. The basic ap-

proach is to have a pre-deployment 'offline phase' or training phase aimed at building detailed RF maps or RF propagation models based on a survey of the target area. The client device is then localized by matching the observed RSS against the signal map. RADAR-empirical [1] was one of the first RF-based schemes to use this model. In recent years, a number of probabilistic techniques [21, 12, 10] have been proposed to enhance the robustness of localization. In such techniques, the offline phase corresponds to the construction of conditional probability distributions which map signal intensities to locations on a map. Thus, we first build up a *signal map* database for the area being covered. During the location determination phase, given a real-time RSS-signature, a probabilistic inference algorithm is used to select the most likely location from all possible locations in the target space. *As mentioned in section 1.1, these techniques require considerable 'pre-deployment' training effort, are difficult to maintain and update with changing dynamics in the target space and are inherently susceptible to the hardware variance problem [20]*

Prior work on hardware variance: In [20] Tsui et al. observe that hardware variance can significantly degrade the positional accuracy of RSS-based Wi-Fi localization systems. Infact they note that the hardware variance problem is not limited to differences in the WiFi chipsets used by training and tracking devices but also occurs when the same Wi-Fi chipsets are connected to different antenna types and/or packaged in different encapsulation materials. The authors stick to the *online*-training and *offline* location-determination model but add an intermediate online-adjustment phase . In this intermediate phase they use unsupervised learning methods to construct a signal transformation function between the training device and a new tracked device. Prior work on hardware variance [10] observe a linear relationship between the RSS mappings of several commodity Wi-Fi cards and suggest a manual calibration effort to identify this relationship between different cards. *The ever-increasing number of wifi chipsets, antennas and encapsulating materials make this manual adjustment effort impractical in real-world deployments.*

In [19] Tao et al. have an interesting take on unmodelled hardware and transmission power variations being effected by a transmitting client. They also have an infrastructure based model though they stick to building an RF-map first. They observe that RSS is linearly proportional to transmission power. Thus the difference in received signal strengths between a pair of sniffer devices would not vary dramatically as the transmission power of a client device changes. Based on the difference in signal strength between every pair of sniffers, they suggest a weighted heuristic to estimate a location-fix for a given target RSS fingerprint. With such a 'difference' based approach, we can no longer assume that the sniffers are independent. Thus, we are restricted to the use of a heuristic in this model. However, the observation that RSS is linearly proportional to transmission power is very interesting. *Infact, we use this observation in building our model.*

GEM compared to prior work: The major contribution of this work is to develop an algorithm that does not rely on training data. Instead, the algorithm can learn the parameters of the model from real-time transmissions being made by a Tx-client. Thus it can adapt to variations in transmit power across heterogeneous devices which makes it particularly suitable for server-side localization techniques across large target areas. Moreover, this model can also factor in real-time changes in the environmental dynamics of the target space.

3. PROBLEM FORMULATION

Assume that the target space is discretized in J locations. This can be of any level of granularity depending on the desired accuracy. Finer granularity does increase computational load, but does not seem to be a bottleneck. There are a set of sniffers or APs doubling as sniffers (a larger number is expected to improve accuracy) that report a vector of RSSs from a target device to be localized to a server that performs the necessary computation. The target device can be static or mobile. In fact, mobility tends to improve performance (more on this later). The location of the sniffers themselves are assumed to be known with respect to which the J locations are specified. No prior wireless measurements are needed, providing our approach with a significant leverage.

3.1 Using Gaussian Mixture Model

We use the well-known idea of *mixture models* in statistics. The idea is to first make a very general assumption that the target could be in any of the J possible locations with varying probabilities. Each of these possibilities can potentially generate a distribution of RSSs at the set of sniffers. Now, given the vector of RSSs sniffed at the set of sniffers, the problem is to estimate the most likely target location out of the J possibilities that could have generated that vector of RSSs. Since the same device at the same location but with a different transmit power can generate different distributions of RSSs, an additional subtlety we handle is that the most likely power level (actually an abstract sense of it) is also determined as a part of the process. **This subtle addition makes the method adaptive for different devices having their own default power levels for wireless transmission.**

Before a more formal presentation a key assumption

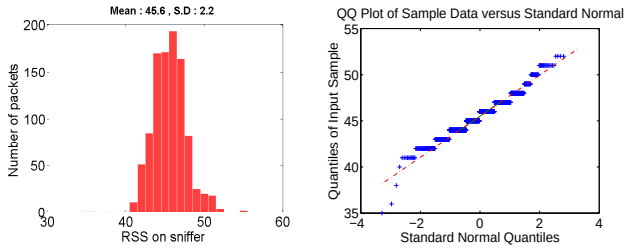


Figure 1: The distribution of RSS observed on a sniffer

we must make upfront is that the distribution of RSS at a sniffer (more specifically an indicator representing RSS, commonly known as RSSI) is Gaussian given the target device is stationary at a location **and transmitting at a fixed power level** The Gaussian assumption is not uncommon in wireless link modeling [10, 6, 19]. In fact, the log-normal shadowing model [] is widely used albeit in a slightly different context. To lend confidence to this assumption on our specific hardware setup, we have performed a set of measurements using the same sniffer and target device hardware used in later experiments. Figure 1 shows the RSSI distribution from this set of measurements and the quality of Gaussian fit.

The Gaussian assumption makes our approach amenable to well-known machine learning tools. Now, the distribution of the RSSs on the sniffers can be represented by the *Gaussian Mixture Model* or GMM [17, 3] – a simple linear superposition of Gaussian components. Nothing is known a priori about the nature of these Gaussian and in what proportion they are mixed. They are modeled in terms of discrete latent variables. We describe the modeling approach below.

3.2 Latent Variables for Target Locations and Power Levels

Assume that a J-dimensional binary random variable $\mathbf{x}$ representing possible target locations. $\mathbf{x}$ has a 1-of-J representation in which a particular element x_j is equal to one and all other elements are equal to 0. The values of x_j therefore satisfy $x_j \in \{0,1\}$ and $\sum_j x_j = 1$. Thus, there are J possible states for the vector $\mathbf{x}$.

The probability distribution over $\mathbf{x}$ can be specified as a multinomial

$$p(x_j = 1) = \upsilon_j, \tag{1}$$

where the parameters $\{\upsilon_j\}$ must satisfy

$$0 \leq \upsilon_j \leq 1 \; and \; \sum_{j=0}^{J} \upsilon_j = 1. \tag{2}$$

Similarly, assume that a K-dimensional binary random variable $\mathbf{z}$ representing Power Levels. $\mathbf{z}$ has a 1-of-K representation in which a particular element z_k is equal to 1 and all other elements are equal to 0. The values of z_k therefore satisfy $z_k \in \{0,1\}$ and $\sum_k z_k = 1$. Vector $\mathbf{z}$ has K possible states.

The distribution over $\mathbf{z}$ is specified as a multinomial

$$p(z_k = 1) = \tau_k, \tag{3}$$

where the parameters $\{\tau_k\}$ must satisfy

$$0 \leq \tau_k \leq 1 \; and \; \sum_{k=0}^{K} \tau_k = 1. \tag{4}$$

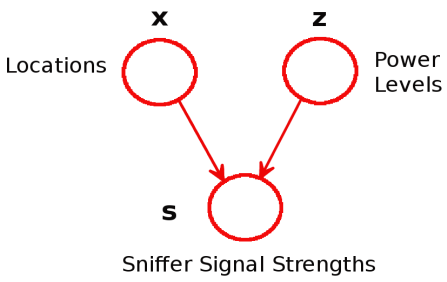


Figure 2: The GMM for our problem

3.3 RSSI Distribution

Let $\mathbf{s}$ be the N-dimensional vector representing the RSSI observed by the N sniffers in the target area. Using the chain rule of probability, we can now define the joint distribution $p(\mathbf{s}, \mathbf{x}, \mathbf{z})$ in terms of the distribution $p(\mathbf{x}, \mathbf{z})$ and the conditional distribution $p(\mathbf{s}|\mathbf{x}, \mathbf{z})$, **corresponding to the graphical model in Figure 2**:

$$p(\mathbf{s}, \mathbf{x}, \mathbf{z}) = p(\mathbf{x}, \mathbf{z}) p(\mathbf{s}|\mathbf{x}, \mathbf{z}). \tag{5}$$

Since $\mathbf{x}$ and $\mathbf{z}$ are independent random variables,

$$\begin{aligned} p(\mathbf{s}, \mathbf{x}, \mathbf{z}) &= p(\mathbf{x}, \mathbf{z}) p(\mathbf{s}|\mathbf{x}, \mathbf{z}) \\ &= p(\mathbf{x}) p(\mathbf{z}) p(\mathbf{s}|\mathbf{x}, \mathbf{z}). \end{aligned} \tag{6}$$

Equation 6 gives us the joint distribution of $p(\mathbf{s}, \mathbf{x}, \mathbf{z})$. The marginal distribution of $\mathbf{s}$ is then obtained by summing the joint distribution over all possible states of $\mathbf{x}$ and $\mathbf{z}$:

$$p(\mathbf{s}) = \sum_{\mathbf{x}} \sum_{\mathbf{z}} p(\mathbf{x}) p(\mathbf{z}) p(\mathbf{s}|\mathbf{x}, \mathbf{z}). \tag{7}$$

Now assume that the RSSIs observed at different sniffers are independent. This is justified as the sniffers are typically at disparate locations and thus the wireless propagation path loss can be assumed independent. Thus, the term $p(\mathbf{s}|\mathbf{x}, \mathbf{z})$ in equation 7 can be simplified as

$$p(\mathbf{s}|\mathbf{x}, \mathbf{z}) = \prod_{i=1}^{N} p(s_i|\mathbf{x}, \mathbf{z}). \tag{8}$$

Based on the Gaussian assumption made before, the RSSI can be modeled as Gaussian random variables determined by the (location, power-level) pair. Thus,

$$s_i|(x_j, z_k) \sim Gaussian(\mu_{i\ (j,k)}\ , \sigma_{i\ (j,k)}). \quad (9)$$

This lends simplicity to our model since the term $p(\mathbf{s}|\mathbf{x}, \mathbf{z})$ in equation 8 can be further simplified as

$$p(\mathbf{s}|\mathbf{x}, \mathbf{z}) = \sum_{j=1}^{J}\sum_{k=1}^{K}(\prod_{i=1}^{N}\mathcal{N}[s_i|\mu_{i\ (j,k)}\ , \sigma_{i\ (j,k)}]). \quad (10)$$

3.4 Model Parameters

Putting equations 7 and 10 together we get the distribution of $\mathbf{s}$ as

$$p(\mathbf{s}) = \sum_{j=1}^{J}\sum_{k=1}^{K}(\upsilon_j\tau_k\prod_{i=1}^{N}\mathcal{N}[s_i|\mu_{i\ (j,k)}\ , \sigma_{i\ (j,k)}]). \quad (11)$$

Thus we have modeled the marginal distribution of $\mathbf{s}$ as a Gaussian mixture with target locations and power levels as the latent variables. The parameters of the model are

$$\theta = \left(\upsilon_j, \tau_k, (\mu_{i\ (j,k)}, \sigma_{i\ (j,k)})\right), \quad (12)$$

where $j \in \{1, \ldots, J\}$, $k \in \{1, \ldots, K\}$ and $i \in \{1, \ldots, N\}$. We now use the widely used *Expectation Maximization* (EM) algorithm [7, 4, 2, 8, 3] to estimate the parameters of the model.

4. EM ALGORITHM

An elegant and powerful method for finding maximum likelihood solutions for models with latent variables is the *Expectation Maximization* algorithm. The EM algorithm is an iterative process through two steps: an expectation step (E-step) and a maximization step (M-step). During the iterations, a sequence of model parameters $\theta^{\mathbf{0}}$, $\theta^{\mathbf{1}}$, ..., θ^* is generated where $\theta^{\mathbf{0}}$ is the initial parameter and θ^* is the converged parameter when the algorithm terminates.

4.1 E-step

Suppose we have a data set of RSSI observations at the sniffers from the target device: $\overline{\mathbf{S}} = \{\ \mathbf{s}^0, \mathbf{s}^1, \ldots, \mathbf{s}^M\}$. The E-step corresponds to finding the expected value of the latent or hidden component ($\mathbf{x}$ and $\mathbf{z}$) values given the observed data $\overline{\mathbf{S}}$ and the current parameter estimates. Using this observation set and the current parameter estimates, we find out the posterior probabilities **(or responsibilities)** as follows. For each observation $\mathbf{s}^l$,

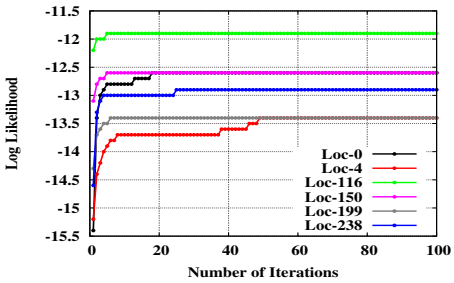


Figure 3: Convergence of log likelihood for 6 different instances of using GEM.

$$\pi^l_{(x_j,z_k)} = p(x_j = 1, z_k = 1|\mathbf{s}^l) \quad (13)$$

$$= \frac{p(x_j = 1)p(z_k = 1)p(\mathbf{s}^l|x_j = 1, z_k = 1)}{\sum_{p=1}^{J}\ \sum_{q=1}^{K} p(x_p = 1)p(z_q = 1)p(\mathbf{s}^l|x_p = 1, z_q = 1)}$$

$$= \frac{\upsilon_j\ \tau_k N(\mathbf{s}^l|\mu_{j,k}, \sigma_{j,k})}{\sum_{p=1}^{J}\sum_{q=1}^{K}\left[\upsilon_p\tau_q N(\mathbf{s}^l|\mu_{p,q}, \sigma_{p,q})\right]}. \quad (14)$$

The posterior probability value $\pi^l_{(x_j,z_k)}$ can be viewed as the *responsibility* that component (x_j, z_k) takes for explaining observation $\mathbf{s}^l$. We determine this measure of responsibility for each observation in the data set $\overline{\mathbf{S}}$.

4.2 M-step

The M-step of the algorithm corresponds to 'maximizing the likelihood' of the observed data. This leads us to re-estimating the parameters for the next iteration based on the posterior probabilities calculated in the expectation step of the algorithm:

$$\upsilon_j = \frac{\sum_{l=1}^{M}\ \sum_k \pi^l_{(x_j,z_k)}}{M}, \quad (15)$$

$$\tau_k = \frac{\sum_{l=1}^{M}\ \sum_j \pi^l_{(x_j,z_k)}}{M}, \quad (16)$$

$$\mu_{i\ (j,k)} = \frac{\sum_{l=1}^{M}\pi^l_{(x_j,z_k)}s^l_i}{N_{j,k}}, \quad (17)$$

where

$$N_{j,k} = \sum_{l=1}^{M}\pi^l_{(x_j,z_k)}. \quad (18)$$

The variance parameter can also be updated accordingly.

4.3 Convergence of Log Likelihood

Each update of the parameters resulting from an E-step followed by an M-step is guaranteed to increase the

log likelihood function:

$$\ln p(\overline{\mathbf{S}}|\theta) = \sum_{l=1}^{M} \ln \left\{ \sum_{j=1}^{J} \sum_{k=1}^{K} \upsilon_j \tau_k \mathcal{N}(\mathbf{s}^l | \mu_{j,k}, \sigma_{j,k}) \right\}. \tag{19}$$

The algorithm is deemed to have converged when the change in the log likelihood function falls below a threshold (10^{-6} in the experiments described later). Figure 3 shows how the log-likelihood converges for six different instances of running GEM. Each instance here was to localize an android phone on the CEWIT testbed.

4.4 Handling Identifiability

There is an identifiability problem in this general approach that is well understood [3]. This arises because there are $P!$ equivalent solutions in a P component mixture model. In our case, each component is a (location, power-level) pair. We handle the problem of identifiability by using the knowledge of sniffer locations and initializing the EM algorithm using the basic log-distance radio propagation model [16, 15] below:

$$P_r(d) = G\frac{P_t}{d^{\alpha}}, \tag{20}$$

where $P_r(d)$ is the received power at distance d and P_t is the transmit power. α is the path loss exponent which is simply a model parameter. In free space $\alpha = 2$, but it typically increases somewhat in complex environments. G is a frequency and antenna dependent constant. Often the above equation is expressed somewhat differently as:

$$P_r(d) = P_r(d_0) - 10\alpha \log\left(\frac{d}{d_0}\right), \tag{21}$$

where P_r is now expressed decibel (dB) units. This emphasizes that when powers are expressed in dB units transmit power changes expressed in dB causes the same dB change at all receivers regardless of location. In our experiments we will use RSSI in dB units. We independently verified (not reported here for brevity) that the RSS measurement on our sniffer hardware is accurate at least to the extent that a dB shift in the transmit power does get recorded as a similar shift at the sniffer regardless of location.

4.4.1 *Initializing the components of our Model*

In Equation 21, $P_r(d_0)$ is the signal power at some reference distance d_0 from the sniffer. This reference signal strength $P_r(d_0)$ can be derived empirically or obtained from wireless network hardware specifications [1]. In our deployment all our sniffers have the same hardware (described in detail in Section 6) . The value of $P_r(d_0)$ was empirically found to be approximately 60 dB when $d_0 = 1$ meter . No assumption is made about the target space, so $\alpha = 2$. The path loss equation of equation 21 can now be succintly expressed as:

$$P_r(d) = 60 - 10\alpha \log(d), \tag{22}$$

For a location L at a distance d from a sniffer, equation 22 gives us the theoretical RSSI value P at the sniffer. However, different devices may work at different power-levels for doing wireless transmissions. Thus we generate k values ranging from $[P - \frac{k}{2}, P + \frac{k}{2}]$ to initialize the means for the k component from location L. We do this procedure for every possible target location in the map. The standard deviation ($\sigma_{j,k}$) was chosen as 5 (and kept fixed to reduce computation time). This choice was mostly arbitrary though some previous work [19] also use fixed values of standard deviation ($\sigma = 12$) in their work.

4.5 Final Location Estimate

Given a real-time received RSS vector $\mathbf{s}^{(obs)}$, we can now find the location with the highest probability. We do this by first finding the probability for each (location, power-level) pair and then marginalizing over the power-levels. This gives us a probability distribution over the possible locations inside the target space. The location with the highest probability is returned as the answer. Thus the estimated location index is given by j^* where

$$j^* = max_j \sum_k P(x_j = 1, z_k = 1 | \mathbf{s}^{(obs)}) \tag{23}$$

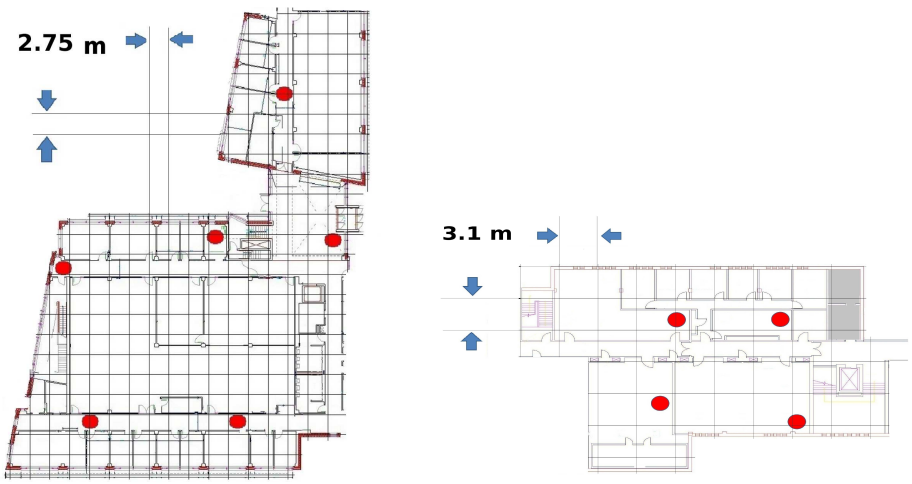
5. EXPERIMENT METHODOLOGY

We start with a description of our system setup, including an overview of the components of our sniffer devices. We then present details about the two testbeds where we conducted our experiments. Finally, we round up this section by discussing the data collection process.

5.1 System Setup

As mentioned briefly in Section 1, GEM uses an infrastructure based architecture. The system has two main components: stationary sniffer devices in the target space and a centralized server running the GEM algorithm. Sniffers provide overlapping coverage of the target area (similar to WLAN APs). The server notifies the sniffers about the MAC id of the target device, the channel number and the listening period. The sniffers then record the RSSI of all packets received that match the server's query. The recorded information is sent to the server which then makes a location estimation using the GEM algorithm.

In the current prototype, the server communicates with the sniffer devices using in-building power-line network. In the ultimate embodiment, the sniffer functionality could be integrated directly into the WLAN APs. If necessary and appropriate, a localization application



(a) CEWIT **(b)** CSD

Figure 4: Two testbeds for validation experiments. The red circles represent sniffer locations.

can also run on the client that downloads the building map as soon as gets connected to the WLAN, sends a localization request to the WLAN and shows the location on the map.

5.1.1 *Sniffer Information*

We use Soekris net4801 [] SBCs as sniffer devices with atheros-based CM9 cards for wireless captures. The sniffers run Pyramid Linux (version 2.6.16-metrix). The default MadWiFi driver is used comes with this distribution (0.9.4.5:svn 1485).

To capture packets the standard Tcpdump tool (version 4.0.0/libpcap version 0.9.8) is used. To obtain signal strength information, the MadWiFi driver allows a monitor mode interface to be created and configured with the radio tap header support. The radio tap header reports the SNR (in dB) as the RSSI. This is what we use directly. Since the noise floor reported by the cards is constant (-95dBm), the RSSI value is also the same as the RSS (in dBm) with a constant difference.

5.2 Testbed Details

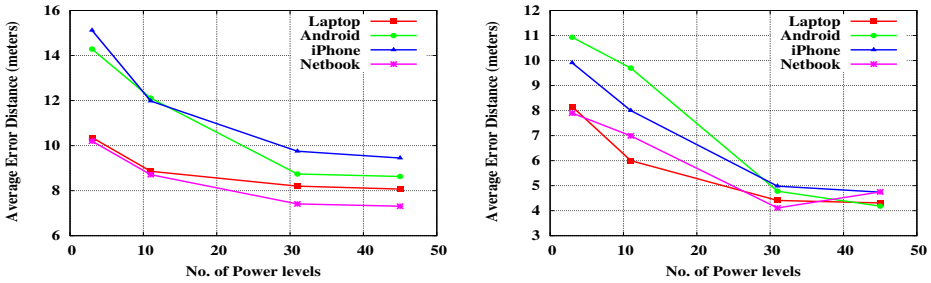
Two different indoor testbeds are used for validation. The first building, henceforth called CEWIT, is a research and development center in the university with a dimension of 65 meter × 50 meter. The L-shaped floor comprises of several obstructions in the form of walls of various types, glass and metal doors, office furnitures, server-rack cabinets etc. The second, building henceforth called CSD, is part of the building housing the computer science department. See Figure 4. This rectangular-shaped floor has a dimension of 20 meter × 30 meter and also have walls and various partitions and office furnitures. Both these testbeds had a continuous flux of people moving around in the building at the time the experiments were conducted. The CEWIT and CSD testbeds use 6 and 4 sniffers respectively. See Figure 4 for the sniffer locations.

5.3 Data Collection Methodology

The CEWIT testbed is discretized into 45 distinct locations roughly every 5.5 meters. The CSD testbed is divided into 27 distinct locations roughly every 3.3 meters. See Figure 4. Multiple device types are used. For each device, we transmit 200 ping packets from every distinct location of the corresponding testbed. This is typically accomplished by having the user hold the mobile device and walk across the floor of the building briefly stopping at each marked location to transmit 200 ping packets. The ground truth is noted at each location before moving on to the new location. Note that the ground truth information is used only for evaluation of the localization error and is not supplied to GEM for training. Each ping packet is separated uniformly apart at a rate of 1 per second. On the server, the sequence number in the ping packet is used to form the vector of RSSI values recorded by individual sniffers for each transmission. Thus, from each distinct location on the map and for each device type, we have a set of 200 RSSI tuples. This comprises our entire data set that we use in this paper. Experiments on RADAR and Probabilistic (described later in this paper) use a subset of this dataset for building the RF signal map and the remainder data for calculating localization error.

5.3.1 *Test Devices*

Four different wireless devices are used - a laptop, an android phone, an iphone, a netbook. The laptop is a Dell Inspiron 1545 running Ubuntu v9.04. The android phone is a Google Nexus One. An iphone 3GS (iOS version 4.2.1) is also used. The netbook used is a



(a) CEWIT **(b)** CSD

Figure 5: Avg Error distance as a function of the number of power levels

Dell Latitude 2110 running Ubuntu v9.10. Each device is using its default driver and default power levels for WiFi transmissions. The data is collected over a span of several days. The devices are not oriented in any specific direction while making the ping transmissions. The orientation is simply left to the user's choice or convenience.

6. EVALUATION

In this section, we present a comprehensive overview of our experimental results. We evaluate the performance of GEM on our two experimental testbeds. We attempt to answer the following questions :

- What is the number of power-levels that we should use in GEM i.e what is the value of K (mentioned in Section 3.2 above) that we should use when we run GEM on the back end localization server.
- How does the localization accuracy vary as the size of the learning data-set increases.
- GEM accuracy for heterogenous devices with unmodelled hardware and power-level characteristics
- How does GEM perform with respect to a model-based scheme that uses the indoor radio path loss propagation model. This presents a true head-to-head comparison because both the techniques do not need pre-deployment effort and can work on the same granularity of discretization of the target space.
- How does GEM perform with respect to schemes that build RF signal maps like RADAR and Probabilistic. This experiment shows how the WiFi hardware variance problem can impact the accuracy of RF signal map schemes and also show the impact of training granularity for signal map based schemes.
- We also study how the mobility of a client can actually improve GEM's localization accuracy.

6.1 Number of powers levels to use in GEM

As mentioned in Section 5.3 above, the CEWIT testbed has 45 distinct locations and CSD has 27 distinct locations, and for each distinct location on the map and for each device type we have a set of 200 RSS tuples. We divide these 200 tuples into two sets of 100 tuples each: one for learning the GEM parameters and the other for testing the GEM localization results. Each device type is considered separately. Figure 5 shows the results of the average error distance (in meters) for the four devices across varying number of power levels used in GEM. We see that the average error distance hits a plateau after K = 31. This is an interesting result because it helps us bound the number of power levels to use. We use a value of K = 45 in the subsequent experiments.

6.2 Localization accuracy as a function of the learning set size in GEM

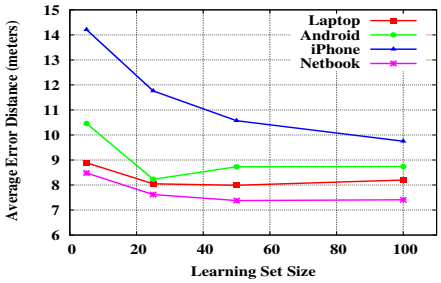


Figure 6: Average Error distance on the CEWIT Dataset as a function of the learning set size

Having fixed the number of power levels to use, we now study how the size of the learning data-set changes the average error distance. Recollect here that as part of our data collection methodology, we have 200 RSS tuples for every location on the map for each of the four device types. This time we again divide the 200 tuples into two sets : one set for learning and the other for testing. The test set size is kept fixed at 100 RSS

tuples. From the remaining tuples, the learning set size is varied from 2 tuples going up to 100 tuples. Each device type is considered separately. Figure 6 shows the results of the average error distance (in meters) in the CEWIT testbed as the size of the learning set varies. We observe that for all the four devices, the average error does not vary much as the as we move from 50 training samples to 100 training samples. The CSD testbed results (not included here) converged after 25 training samples itself. The experiments which follow have been done keeping the GEM learning set size at 100 and using the remaining 100 samples for testing the localization accuracy.

6.3 GEM accuracy for heterogeneous devices

Figure 7 shows how GEM performed across the four test devices 5.3.1 on both the testbeds. We see that for both the testbeds, the accuracy estimates are pretty similar for all the devices. Thus we see that GEM can adapt itself for heterogeneous devices working at different power levels. Section 6.5 below shows how RF-signal map based techniques show substantial degradation in accuracy because of hardware variance.

6.4 Baseline Comparison with a model-based scheme

Here we analyze the performance of GEM with respect to a model-based scheme that uses the indoor radio path loss propagation model (Section 4.4). This presents a true head-to-head comparison, because both the techniques do not need pre-deployment effort and can give a location estimate at the same granularity of discretization of the target space. Both our testbeds, CEWIT and CSD, have been discretized as shown in Figure 4. There are 267 grid vertices inside the CEWIT testbed roughly every 2.75 meters. The CSD testbed has 36 grid vertices roughly every 3.1 meters. GEM can localize a given target RSS vector to any of these grid vertices. As mentioned in Section 5.3, the data for our experimental evaluation is coming from 45 distinct locations on the CEWIT testbed and 27 distinct locations in the CSD testbed. There are 200 RSS tuples for every location on the map for each of the four device types. As mentioned above in Section aa, GEM is using a learning set size of 100 RSS samples with 45 power levels to build the model. Thus the test-set for both the algorithms is remaining 100 RSS tuples from each location. Each device type is evaluated separately.

The log-distance path loss (LDPL) mentioned in Section 4.4 is used to estimate the RSS that should be observed at a sniffer for each grid vertex inside the target-space. These RSS values are used to initialize GEM, as mentioned in section 4.4. The Model-based algorithm also uses these same RSS values with a suitable metric to give a final location estimate. Similar to [1], the model-based algorithm that we use here uses nearest neighbor in signal space (NNSS) as the metric of choice.

Figure 8 shows the median error for both techniques. We see that GEM performs better than the model-based scheme across all device types in both the testbeds.

6.5 Comparisons with schemes that use RF signal maps

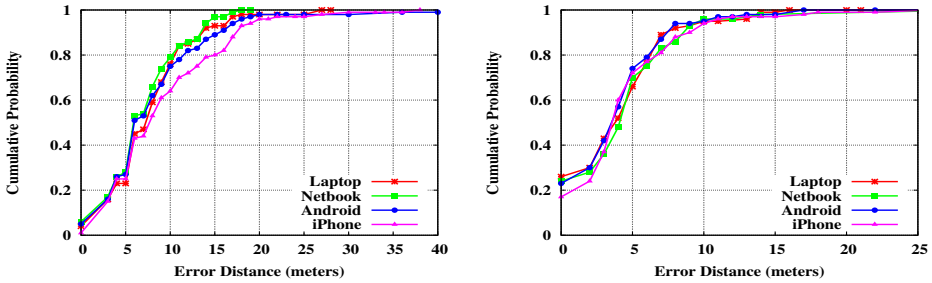
We now compare the performance of GEM again two schemes that spend considerable pre-deployment effort in first building an RF signal map from RSS signatures collected from various locations inside the target space. Both schemes differ in the way they handle an incoming signature to provide a location estimate.

- Deterministic schemes like RADAR [1] use the nearest neighbor in signal space (or an average of k-nearest neighbors) as the metric to give a location estimate.
- Probabilistic schemes [10, 21, 18] on the other hand maintain a probability distribution of the RSS value from various locations. For the incoming signature, a probability distribution is built over the location space and a maximum likelihood estimate is used to determine position. As in [10], we model signal intensity as a normal distribution determined by the location and sniffer pair.

6.5.1 Discussion

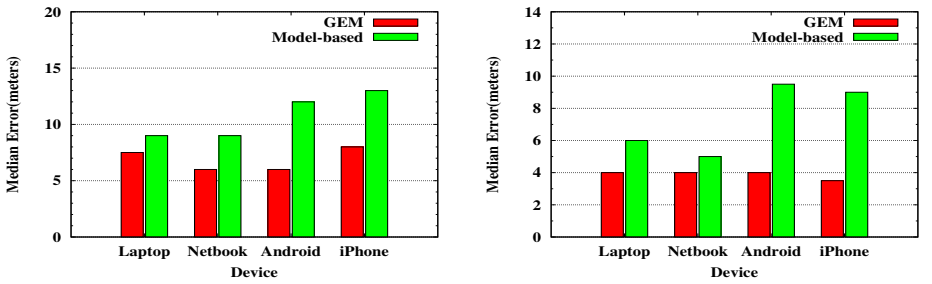
1. For all three techniques that we compare here viz GEM, RADAR and Probabilistic, we consider the best location as our location estimate. We do not consider the weighted average of the top few locations for any of the three techniques compared here.
2. To show the WiFi hardware variance problem mentioned in Section 1, we use different devices for location estimation. For the CEWIT testbed, a DELL Laptop was used to train the radio map during the *'offline' phase*. In the CSD testbed, an android phone was used to build the radio map. Based on the radio maps, four different device types are used to give location estimates.
3. As mentioned in Section 5.3, the data for our experimental evaluation is coming from 45 distinct locations on the CEWIT testbed and 27 distinct locations in the CSD testbed.

   For RADAR and Probabilistic, we try to understand the effect of the granularity of training locations on the final location estimate. We consider two scenarios : one optimistic and the other more



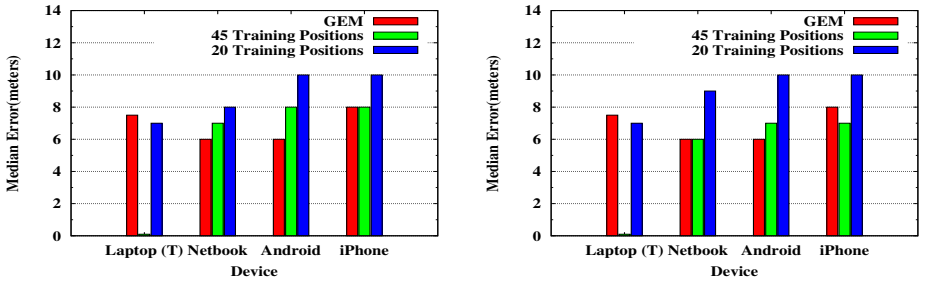
(a) CEWIT (b) CSD

Figure 7: GEM location accuracy for multiple devices



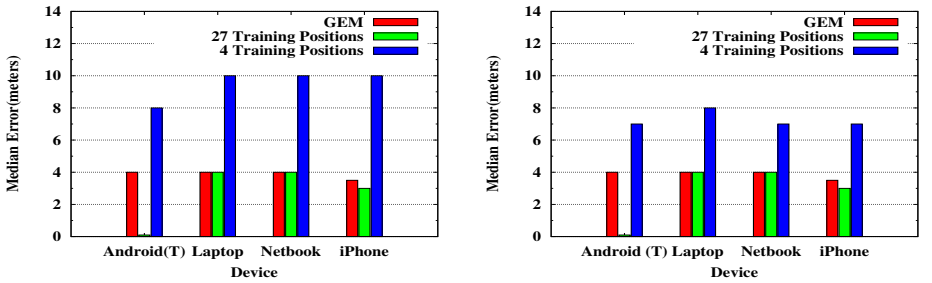
(a) CEWIT (b) CSD

Figure 8: Baseline Comparisons



(a) Probabilistic (b) RADAR

Figure 9: Comparisons on the CEWIT testbed



(a) Probabilistic (b) RADAR

Figure 10: Comparisons on the CSD testbed

realistic. In the first scenario, we consider the optimistic secnario where the training and test data sets (mentioned below) are collected at the same physical locations, e.g on the CEWIT testbed, the signal map is built from the same 45 distinct locations where the position estimation is being done. The second scenario, is the more realistic scenario where the training is done from only a subset of the test locations. Thus the testing locations are no longer strictly co-located with the training locations. In the CEWIT testbed, the training is done from 20 locations (out of the 45 possible) roughly every 10.3 meters apart. In the CSD testbed, the subset comprises of 4 locations (out of the 27 possible) roughly every 12.7 meters apart. We note here that GEM on the other hand requires no training effort. For GEM we use the same discretization of the target space that we use in section 6.4 i.e the GEM location estimate can be on any of the grid points inside the building.

4. There are 200 RSS tuples for every location on the map for each of the four device types. We divide these 200 tuples into two disjoint sets of 100 tuples each. The first set is used by RADAR and Probabilistic to build their RF signal maps while GEM uses it as the learning data to build the localization model. The second set of 100 tuples from each location is used as the test data for all three algorithms. Every device type is evaluated separately.

6.5.2 Observations

Figure 9 and Figure 10 shows the median error comparison between the three techniques . We make some interesting observations here:

- Hardware variance is a major issue for both RADAR and Probabilistic. When the same device is used for training and testing, the median error is zero. For other devices it jumps up dramatically. This is a critical problem for such techniques because a real-world deployment would be typically unaware of the device type being localized. In fact, we see GEM performing as good as RADAR and Probabilistic for other device types. This is particularly promising because unlike RADAR and Probabilistic, GEM did not have the overhead of a pre-deployment effort.

- When the granularity of sampling is reduced, RADAR and Probabilistic start showing substantially poorer accuracy estimates. Thus location estimates for such techniques are tightly bound to the granularity of the training effort. This makes them unattractive for performing localization in large target spaces. A heavy pre-deployment effort also make these techniques difficult to maintain and update in a dynamic environment.

6.6 Impact of Mobility on GEM's localization accuracy

In this experiment we try to understand how the mobility of a client can effect the location estimates made by GEM. To create the test data set, a user initially walks across all distinct location on the map, making 100 ping transmissions from each location. This forms our test data set which is basically a union of 100 RSS tuples from each distinct location on the map. Now we try to observe the effect of client mobility in localizing this test data set. To do this, the user follows a random walk scheme. In each random walk, the user visits every distinct location in the building floor once, and makes a single ping transmission from that location. This serves as the learning data-set for GEM, in order to build a model for the device and give location estimates for the test set.

We evaluate this scheme on both testbeds across four different devices. Figure 11 shows how the median error of the test data set varies as the mobility (i.e the number of random walks) increases. We observe that mobility actually helps GEM localization accuracy.

7. CONCLUSIONS

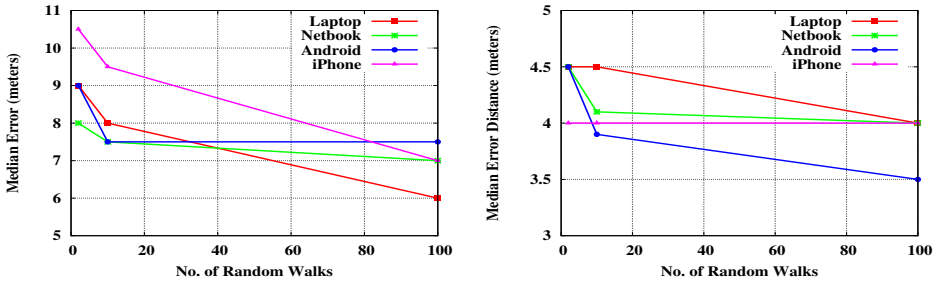
In this work, we have developed a server-side technique to localize a wireless client in an indoor environment based on the signal strength parameter of its transmitted packets. We developed a learning-based algorithm that can learn the parameters of the model dynamically from packets captured by the stationary sniffers / APs inside the building. By using dynamic packet captures for parameter estimation, we can provide location estimates which are much more robust in the face of time varying phenomena like movement of people inside the building, opening closing of doors etc. Moreover, this technique can be used on a host of heterogeneous devices operating at different power levels.

We do not have an explicit *training* phase in our technique. Infact, we showed that we can achieve accuracy that is at par with state-of-the-art techniques that use training to build RF-signal maps first. Thus, our technique not only eliminates the intensive time-consuming (often manual) training phase but also makes our technique scalable for large target spaces.

8. REFERENCES

[1] P. Bahl and V. N. Padmanabhan. Radar: an in-building rf-based user location and tracking system. In *in Proceedings of IEEE INFOCOM*, pages 775–784, 2000.

[2] J. Bilmes. A gentle tutorial on the em algorithm and its application to parameter estimation for



(a) CEWIT **(b)** CSD

Figure 11: Mobility

gaussian mixture and hidden markov models. Technical report, 1997.

[3] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[4] S. Borman. The expectation maximization algorithm: A short tutorial. Technical report.

[5] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan. Indoor localization without the pain. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, MobiCom '10, pages 173–184, New York, NY, USA, 2010. ACM.

[6] L. F. M. de Moraes and B. A. A. Nunes. Calibration-free wlan location system based on dynamic mapping of signal strength. In *Proceedings of the 4th ACM international workshop on Mobility management and wireless access*, MobiWac '06, pages 92–99, New York, NY, USA, 2006. ACM.

[7] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, (39), 1977.

[8] I. D. Dinov. Expectation maximization and mixture modeling tutorial. *UC Los Angeles: Statistics Online Computational Resource. Retrieved from: http://escholarship.org/uc/item/1rb70972*, 2008.

[9] Y. Gwon and R. Jain. Error characteristics and calibration-free techniques for wireless lan-based location estimation. In *Proceedings of the second international workshop on Mobility management & wireless access protocols*, MobiWac '04, pages 2–9, New York, NY, USA, 2004. ACM.

[10] A. Haeberlen, E. Flannery, A. M. Ladd, A. Rudys, D. S. Wallach, and L. E. Kavraki. Practical robust localization over large-scale 802.11 wireless networks. In *Proceedings of the 10th annual international conference on Mobile computing and networking*, MobiCom '04, pages 70–84, New York, NY, USA, 2004. ACM.

[11] P. Krishnan, A. 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 *in Proceedings of IEEE INFOCOM*, pages 1001–1011, 2004.

[12] A. M. Ladd, K. E. Bekris, A. Rudys, G. Marceau, L. E. Kavraki, and D. S. Wallach. Robotics-based location sensing using wireless ethernet. In *Proceedings of the 8th annual international conference on Mobile computing and networking*, MobiCom '02, pages 227–238, New York, NY, USA, 2002. ACM.

[13] H. Lim, L.-C. Kung, J. C. Hou, and H. Luo. Zero-configuration indoor localization over ieee 802.11 wireless infrastructure. *Wirel. Netw.*, 16:405–420, February 2010.

[14] D. Madigan, Elnahrawy, R. P. Martin, W. hua Ju, P. Krishnan, and A.Krishnakumar. Bayesian indoor positioning systems. In *in Proceedings of IEEE INFOCOM*, pages 1217–1227, 2005.

[15] D. Molkdar. Review on radio propagation into and within buildings. In *Microwaves, Antennas and Propagation, IEE Proceedings H*, pages 61–73, 1991.

[16] T. Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 2001.

[17] D. Reynolds. Gaussian mixture models. Technical report.

[18] T. Roos, P. Myllymki, H. Tirri, P. Misikangas, and J. Sievnen. A probabilistic approach to wlan user location estimation. In *International Journal of Wireless Information Networks*, pages 155–164, 2002.

[19] P. Tao, A. Rudys, A. M. Ladd, and D. S. Wallach. Wireless lan location-sensing for security applications. In *Proceedings of the 2nd ACM workshop on Wireless security*, WiSe '03, pages

11–20, New York, NY, USA, 2003. ACM.

[20] A. W. Tsui, Y.-H. Chuang, and H.-H. Chu. Unsupervised learning for solving rss hardware variance problem in wifi localization. *Mob. Netw. Appl.*, 14:677–691, October 2009.

[21] M. Youssef and A. Agrawala. The horus location determination system. *Wirel. Netw.*, 14:357–374, June 2008.

[22] M. A. Youssef, A. Agrawala, and A. U. Shankar. Wlan location determination via clustering and probability distributions. In *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications*, PERCOM '03, pages 143–, Washington, DC, USA, 2003. IEEE Computer Society.