WiGEM : A Learning-Based 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. Several state-of-the-art indoor localization techniques have the drawback that they rely extensively on a 'training' phase that does not scale well. The 'training' is a labor intensive process and must be done for each target area under consideration. The introduction of unmodeled hardware with heterogeneous power-levels further reduces the accuracy of these techniques.

We propose a learning-based approach, WiGEM, where the received signal strength is modeled as a Gaussian Mixture Model (GMM). Expectation Maximization (EM) is used to learn the maximum likelihood estimates of the model parameters. This approach enables us to localize a transmitting device based on the maximum a posteriori estimate. The key insight is to use the physics of wireless propagation, and exploit the signal strength constraints that exist for different power levels. A learning approach not only avoids the labor-intensive training phase, but also makes the location estimates considerably robust in the face of various form of heterogeneity and time varying phenomena. We present evaluations on two different indoor testbeds with multiple WiFi devices (iPhones, Android-based phones, Laptops and Netbooks). We demonstrate that WiGEM's accuracy is at par with or better than 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 because a perfect, general-purpose solution such as outdoor 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.' Fine grain RF map creation makes localization more accurate, but requires proportionately more effort. On the other hand, any RF map is inherently device specific. *This pre-deployment burden that lacks generality has made these localization techniques less appealing in practice.* This paper develops a new machine learning-based localization algorithm, WiGEM, that removes these limitations.

Over time, two general localization approaches have emerged in literature – (i) client-based approach [12, 11, 23, 7, 14, 24] and (ii) infrastructure-based approach [8, 15, 21, 13]. In the client-based approach, the client device measures the RSS (received signal strength) as seen by it from various APs (access points). This information is used to localize 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 technique we develop is not specific to any link layer technology. At this point we also make a distinction between 'learning' and 'training'. By 'learning' we actually mean unsupervised learning, whereby we automatically try to estimate our model parameters from unlabeled data. On the other hand, 'training' is ak-

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.

ing to supervised learning, and in our scenario leads to substantial limitations as discussed next.

1.1 Limitations of Training

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 mobile device being localized. There are three major drawbacks of this general approach.

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 [22]. Experiments described later in this paper indicate how hardware variations between four common commodity WiFi devices can significantly degrade the accuracy of two commonly used localization algorithms.

2. The 'offline phase' itself involves labor-intensive sampling of signal strength values at discrete 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 to update regularly.

1.2 Approach

We propose WiGEM, a novel **Wi**reless localization algorithm that uses the **G**aussian Mixture Model (GMM) and employs **E**xpectation **M**aximization (EM) to estimate the model parameters. WiGEM leverages the infrastructure based approach while eliminating any 'pre-deployment' effort. Packet transmissions made by a client are received on stationary sniffers (or APs 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, WiGEM 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, for example, depending on the intended application. But this is not a part of the current work.

At the heart of our technique is a radio propagation model [18, 17] and the typical constraints that exist between the signal strength intensities for different power levels at the same location. This information is used to initialize our model. (Section 4.4)

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

The remainder of the paper is organized as follows. In Section 2, we survey related work in Indoor Localization. In Section 3 we introduce GMM, the modeling approach we use to localize a target device, and discuss the parameters of the model. In Section 4 we discuss the EM algorithm, which is used to estimate the parameters of our model. Section 5 provides details on the experiment methodology and Section 6 presents the experimental results obtained from two different testbeds. In Section 7 we discuss WiGEM and identify items for future work. Finally, we present our conclusions in Section 8.

2. RELATED WORK

In this section we provide a brief overview of some of the fundamental techniques in the field of indoor wireless localization. In passing, we also point out their limitations with respect to the proposed WiGEM technique.

Calibration-free techniques: An indoor path-loss propagation model essentially forms the bedrock for these techniques. In RADAR, Bahl *et al* [3] provide a indoor radio propagation model to calculate RSS at various locations in the building based on distance, number of walls etc. The nearest neighbor in signal space (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. In [8, 15] the authors describe sniffer based techniques for localization based on propagation models. Moraes *et al* [8] use a naive propagation model to generate a 'radio propagation map' (RPM) at each sniffer. They use RSS measurements between the snif-

fers and a 'reference AP' (APRef) to reconstruct the RPM, either periodically or when there are significant variations in the RSS. A probabilistic model is then used to compute a location estimate. Lim *et al* [15] 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 [11] 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. Madigan *et al* [16] 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 the 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 approach 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 [3] was one of the first RF-based schemes to use this model. In recent years, a number of probabilistic techniques [23, 14, 12] have been proposed to enhance the robustness of localization. For the probabilistic techniques, the 'offline phase' corresponds to the construction of conditional probability distributions that map signal intensities to locations on a map. 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 [22].*

Prior work on hardware variance: Tsui *et al* [22] observe that hardware variance can significantly degrade the positional accuracy of RSS-based Wi-Fi localization systems. In fact, 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 introduce an intermediate 'online adjustment' phase where they use unsupervised learning to construct a signal transformation function between the training device and a new target device. Prior work on hardware variance [12] 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.*

Tao *et al* [21] have an interesting take on unmodelled hardware and transmission power variations. They observe that RSS is linearly proportional to transmission power. Based on the difference in signal strength between every pair of sniffers, they suggest a weighted heuristic to give a location estimate for a target RSS fingerprint.

WiGEM compared to prior work: The major contribution of this work is to develop an algorithm, WiGEM, that eliminates the expensive 'training' phase. While a similar attempt has also been made in a recent work [7], this technique depends on the availability of GPS feed in some indoor locations, more the better. WiGEM does not depend on availability of GPS. WiGEM can adapt to variations in transmit power across heterogeneous devices, which makes it particularly suitable for infrastructure-based localization schemes. The algorithm also 'learns on the go' and thus can factor in real-time changes in the environmental dynamics of the target space.

3. PROBLEM FORMULATION

Assume that the target space is discretized into 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 as-

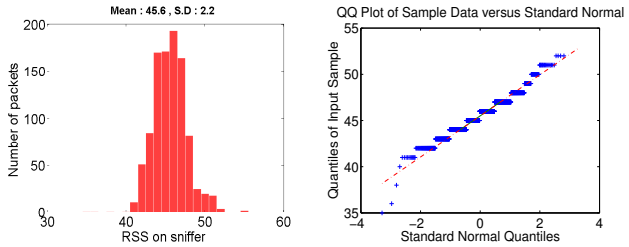


Figure 1: The distribution of RSS observed on a sniffer.

sumption 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 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 [12, 8, 21]. In fact, the log-normal shadowing model [18] 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 measured RSSI distribution observed on a sniffer in our testbed for a stationary client transmitting at a fixed power level. The quality of the Gaussian fit for this distribution is also shown. The fit is very good.

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 [19, 5] – 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

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

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

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

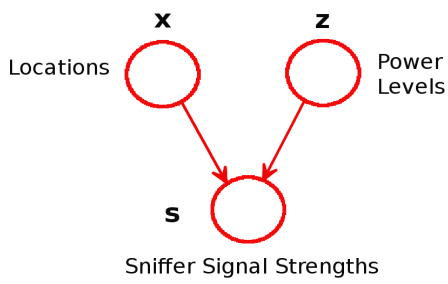


Figure 2: The GMM for our problem.

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, so that

$$s_i|x_j = 1, z_k = 1 \sim \text{Gaussian}(\mu_{i,(j,k)}, \sigma_{i,(j,k)}). \tag{9}$$

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

$$p(\mathbf{s}|\mathbf{x}, \mathbf{z}) = \prod_{j=1}^{J} \prod_{k=1}^{K} \prod_{i=1}^{N} \mathcal{N}(s_i|\mu_{i,(j,k)}), \sigma^2_{i,(j,k)})^{x_j z_k} . \tag{10}$$

Note that for any given $\mathbf{x}$ and $\mathbf{z}$ only one term in the product is actually active for all i, because the exponent $x_j z_k$ acts as a selector: $x_j z_k = 1$ for exactly one index pair (j, k), and equals 0 for all others.

From now on, for notational convenience we define

$$\mathcal{N}(\mathbf{s}|\boldsymbol{\mu}_{(j,k)}), \boldsymbol{\sigma}^2_{(j,k)}) \equiv \prod_{i=1}^{N} \mathcal{N}(s_i|\mu_{i,(j,k)}), \sigma^2_{i,(j,k)}) .$$

3.4 Model Parameters

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

$$p(\mathbf{s}) = \sum_{j=1}^{J} \sum_{k=1}^{K} \upsilon_j \tau_k \mathcal{N}(\mathbf{s}|\boldsymbol{\mu}_{(j,k)}), \boldsymbol{\sigma}^2_{(j,k)}) . \tag{11}$$

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

$$\boldsymbol{\theta} = \left(\boldsymbol{\upsilon}, \boldsymbol{\tau}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2\right) . \tag{12}$$

Henceforth, we refer to this model as **WiGMM**. We now use the widely popular *Expectation-Maximization* (EM) algorithm [9, 6, 4, 10, 5] to estimate the parameters of the model.

4. EM ALGORITHM

An elegant and powerful method for finding maximum likelihood parameter estimates for probabilistic models with latent variables is the *Expectation Maximization* algorithm. The EM algorithm is an iterative process consisting of 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. Under typical conditions, which hold in our model, the sequence of parameters guarantees monotonic improvement of the likelihood function and almost always converges to a (local) maximum-likelihood estimate.

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 $\mathbf{s}^l$,

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

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

$$= \frac{\upsilon_j \ \tau_k \mathcal{N}(\mathbf{s}|\boldsymbol{\mu}_{(j,k)}), \boldsymbol{\sigma}^2_{(j,k)})}{\sum_{p=1}^{J} \sum_{q=1}^{K} \upsilon_p \tau_q \mathcal{N}(\mathbf{s}|\boldsymbol{\mu}_{(p,q)}), \boldsymbol{\sigma}^2_{(p,q)})} . \tag{14}$$

The posterior probability value $\pi^l_{(j,k)}$ can be viewed as the *responsibility* that component (j, k) takes for explaining observation $\mathbf{s}^l$. We compute 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 *expected* log-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

$$\upsilon_j = \frac{\sum_{l=1}^{M} \sum_k \pi^l_{(j,k)}}{M}, \tag{15}$$

$$\tau_k = \frac{\sum_{l=1}^{M} \sum_j \pi^l_{(j,k)}}{M}, \tag{16}$$

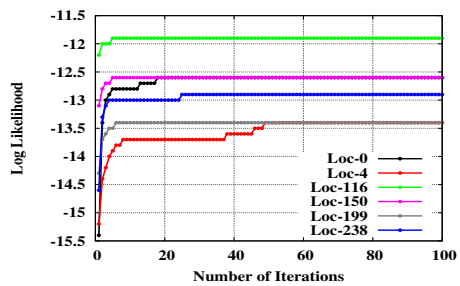


Figure 3: Convergence of log likelihood for 6 different instances of using WiGEM.

$$\mu_{i\ (j,k)} = \frac{\sum_{l=1}^{M} \pi^{l}_{(j,k)} s^{l}_{i}}{N_{j,k}}, \tag{17}$$

where

$$N_{j,k} = \sum_{l=1}^{M} \pi^{l}_{(j,k)}. \tag{18}$$

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

$$\ln p(\overline{\mathbf{S}}|\theta) = \sum_{l=1}^{M} \ln \left\{ \sum_{j=1}^{J} \sum_{k=1}^{K} \upsilon_j \tau_k \mathcal{N}(\mathbf{s}|\boldsymbol{\mu}_{(j,k)}), \boldsymbol{\sigma}^2_{(j,k)}) \right\}. \tag{19}$$

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 WiGEM. Each instance here was to localize an android phone on the CEWIT testbed (testbed description in the next section).

4.4 Handling Identifiability in WiGEM

There is an identifiability problem in this general approach that is well understood [5]. This arises because there are $U!$ equivalent solutions in a U 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 to initialize the EM algorithm using the basic log-distance radio propagation model [18, 17] 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 cause 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 Model Parameters

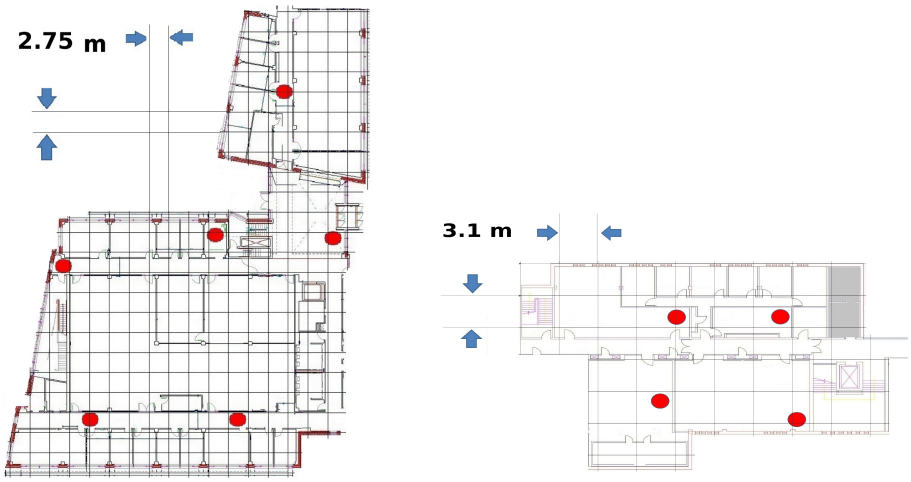
- $\boldsymbol{\upsilon}$ and $\boldsymbol{\tau}$ are initialized as being from a uniform distribution over locations and power levels respectively.
- For a location l_j, Equation 21 gives us the theoretical RSSI value w_{ij} (say) at a sniffer s_i. The reference power $P_r(d_0)$ is assumed to be 60 (in dB units) at $d_0 = 1$ meter. Note that the reference power assumption is somewhat arbitrary. The parameter α is assumed to be 2.

  We consider K values
  $$\left\{(w_{ij} - \frac{K}{2}), (w_{ij} - \frac{K}{2} + 1), \ldots, (w_{ij} + \frac{K}{2})\right\}.$$
  The values are used to initialize the means, $\mu_{i,(j,k)}$ of the K components corresponding to location l_j and sniffer s_i. We do this for every target location in the map and for each sniffer in the building. This effectively initializes parameter $\boldsymbol{\mu}$ in WiGMM.

  Note: negative values are not allowed and are set to 0 during initialization.
- The standard deviation parameter, $\boldsymbol{\sigma}$, is initialized to 5 for each component in WiGMM (and kept fixed to reduce computation time). This choice is mostly arbitrary. Some previous work [21] 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



(a) CEWIT (b) CSD

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

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

$$j^* = \arg\max_j \sum_k p(x_j = 1, z_k = 1 | \mathbf{s}^{(obs)}) \,. \tag{22}$$

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, WiGEM uses an infrastructure based architecture. The system has two main components: stationary sniffer devices in the target space and a centralized server running the WiGEM 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 WiGEM 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 can also run on the client that downloads the building map as soon as it gets connected to the WLAN, sends a localization request to the WLAN and shows the location on the map.

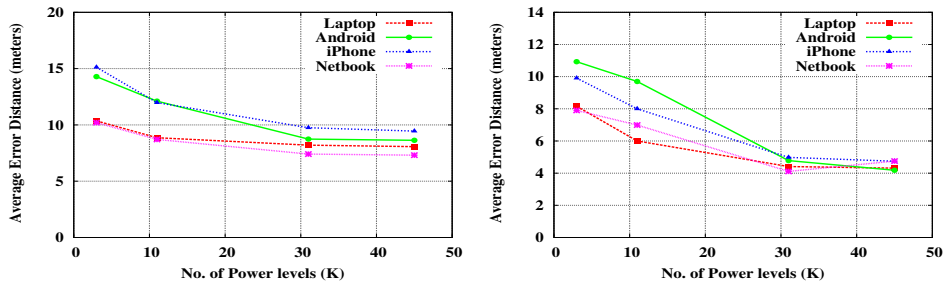
5.1.1 Sniffer Information

We use Soekris net4801 [1] 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 which 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 at Stony Brook University with a dimension of 65 meter $\times$ 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 $\times$ 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.



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

Figure 5: Average error distance results for WiGEM as a function of the number of power levels.

See Figure 4 for the sniffer locations.

5.3 Data Collection Methodology

The CEWIT testbed has 45 distinct test locations roughly every 5.5 meters. The CSD testbed has 27 distinct test locations roughly every 3.3 meters. The testing locations are uniformly distributed in each testbed. Note that test locations are a subset of the total number of vertices formed by a superimposed (virtual) grid. The grid vertices are 2.75 meters apart for the CEWIT testbed and 3.1 meters apart for the CSD testbed. 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 WiGEM 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 and 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 Dell Latitude 2110 running Ubuntu v9.10. Each device is using its default driver and default power levels for WiFi transmissions. Nothing specific for the experiments were done on the devices and they could do their default rate and power control, if any. 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 the evaluation of WiGEM on the two different testbeds. As a part of the evaluation, we determine the number of power levels (K) that should be used for the modeling. Our test cases include heterogeneous devices as described before with unmodeled hardware and power level characteristics, this providing a very realistic benchmarking. We also compare WiGEM with respect to a simple propagation model based scheme (that also requires no pre-deployment effort) as well as well-known, high-performing schemes such as RADAR and Probabilistic [12, 23, 20] (they require significant pre-deployment effort, but provide the best accuracy). In addition we evaluate how the size of the learning data set or mobility impacts the accuracy of WiGEM.

All reported experiments with WiGEM uses half of the measured data set at each location for learning and the other half for testing and validation. The learning part reflects the typical learning process for WiGEM. The general idea is that a typical client device will naturally transmit multiple (likely many) packets for its own network use. It can always be forced to transmit some number of packets for use in localization if it does not naturally transmit anything. The sniffed RSSI vectors for these packets will form the learning set to be used in WiGEM localization. The client does not need to be stationary and is free to move about while the learning set is gathered. In practice, we could keep doing the EM steps and generating location estimates as the device moves through the environment. Two important questions now are the determination of a suitable number of power levels (K) to use for the learning and the

size of learning set. These are addressed next. Unless otherwise noted, we conduct our experiments across all test locations (Section 5.3). Likewise the error distances reflect the aggregate metric across all test locations.

6.1 Number of Powers Levels and Learning Set Size

Figure 5 shows the results of the average error distance (in meters) for the four devices across varying number of power levels used in WiGEM. 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.

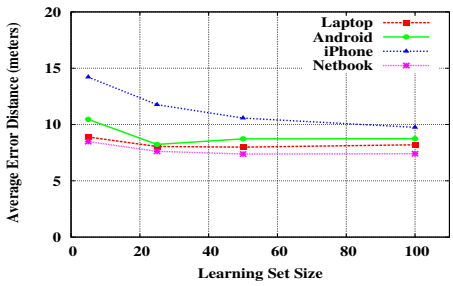


Figure 6: Average error distance results for WiGEM as a function of the learning set size (CEWIT testbed).

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. Recall 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. 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 we move from 50 training samples to 100 training samples. The CSD testbed results (not included here) converged after 25 samples itself. The experiments that follow have been done keeping the WiGEM learning set size at 100 and using the remaining 100 samples for testing the localization accuracy.

6.2 WiGEM Accuracy With Heterogeneous Devices

Figure 7 plots CDFs of error distances to show how WiGEM performed across the four test devices (Section 5.3.1) on both the testbeds. For both the testbeds, the accuracy estimates are pretty similar for all the devices. Thus, we see that WiGEM can adapt itself for heterogeneous devices that possibly work at different transmit power levels. In Section 6.4 below, we show how RF-signal map based techniques show substantial degradation in accuracy owing to such hardware variations.

6.3 Baseline Comparison with a Model-based Scheme

Now we analyze the performance of WiGEM with respect to a model-based scheme that uses the indoor radio propagation model (Section 4.4). Neither of the techniques need pre-deployment effort. We want to basically establish that the learning technique used in WiGEM is buying us something relative to a generic radio propagation model.

The log-distance path loss 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 WiGEM, 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 [3], the model-based algorithm that we use here uses the 'nearest neighbor in signal space' as the metric of choice. Figure 8 shows the median error for both techniques. Note that WiGEM performs better than the model-based scheme across all device types in both testbeds. The performance improvement is quite substantial for the phone-based devices – with the median error distance reducing to roughly half.

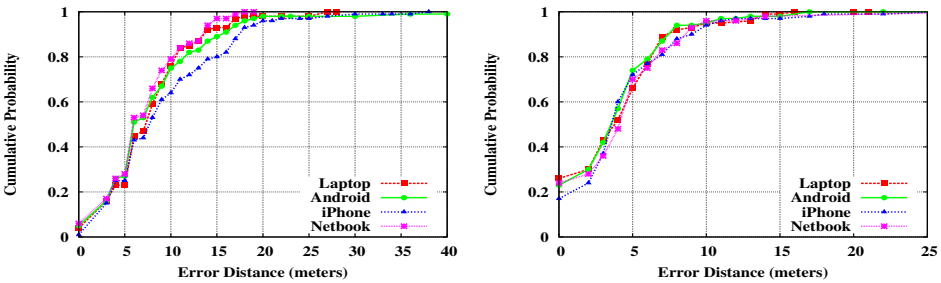
6.4 Comparison with Schemes Using RF Signal Maps

We now compare WiGEM against two popular RF signal map-based schemes that also performed well in literature. We have already reviewed such schemes in Section 2. We have picked two representative schemes – (i) a deterministic scheme, RADAR [3], that uses the nearest neighbor in signal space (or, an average of k nearest neighbors) as the metric; and (ii) a probabilistic scheme [12, 23, 20] that maintains a probability distribution of the RSS values from various locations. For the incoming RSS signature, a probability distribution is built over the location space and a location estimate is made. Here, we mainly follow [12], and model RSS as a normal distribution determined by the location and sniffer pair.

6.4.1 Evaluation Setup

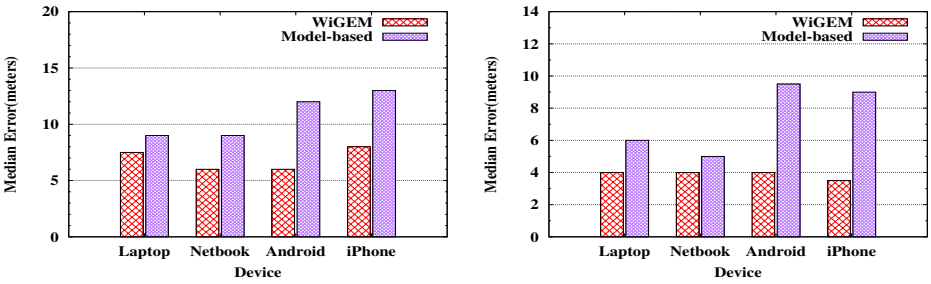
For all three techniques we present here, viz., WiGEM, RADAR and Probabilistic, we consider the best location as the location estimate (and not any weighted average of the top few locations).

As discussed in Section 1, the RF map-based techniques need significant 'off-line' training and thus are



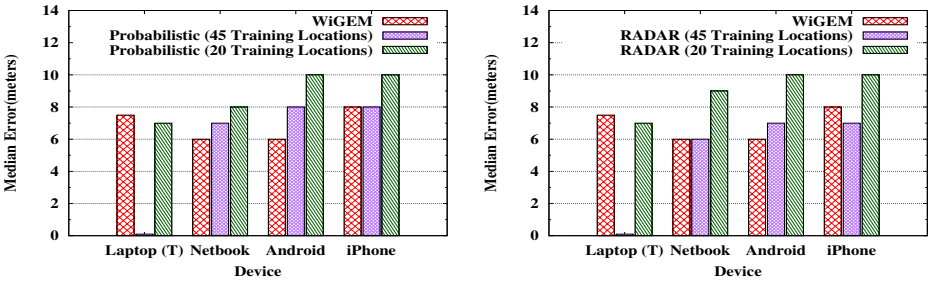
(a) CEWIT testbed (b) CSD testbed

Figure 7: CDF of WiGEM's location accuracy for multiple devices.



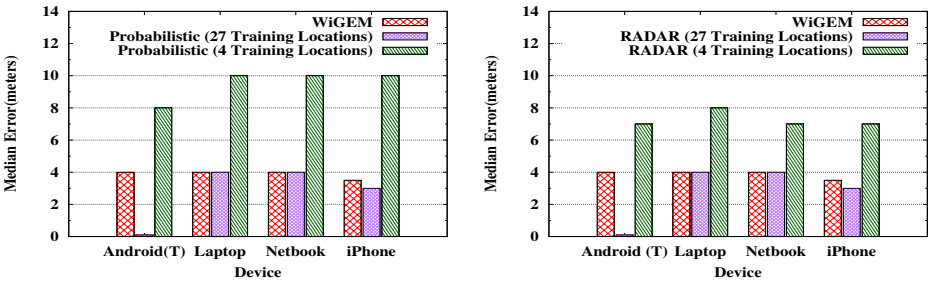
(a) CEWIT testbed (b) CSD testbed

Figure 8: Baseline Comparisons.



(a) WiGEM vs. Probabilistic (b) WiGEM vs. RADAR

Figure 9: Comparisons on the CEWIT testbed.



(a) WiGEM vs. Probabilistic (b) WiGEM vs. RADAR

Figure 10: Comparisons on the CSD testbed.

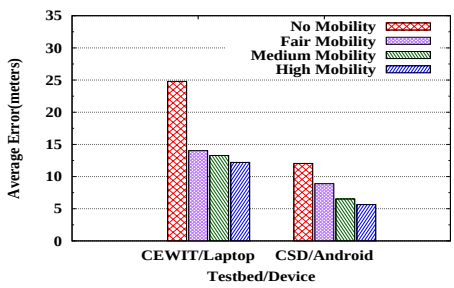


Figure 11: Impact of Mobility.

vulnerable to accuracy issues in realistic scenarios as training and test devices differ. For the CEWIT and CSD testbeds, the Dell laptop and the android phone respectively were used for this off-line training. All four devices were used for testing in all cases.

Similarly as mentioned in Section 1, the accuracy of RF map-based techniques depends on the training granularity. To evaluate its impact, we consider two scenarios for RADAR and Probabilistic techniques – one 'optimistic' and the other more 'realistic.' In the optimistic scenario, the training and test data sets are collected at the same physical locations (i.e., 45 and 27 locations in CEWIT and CSD, respectively). In the realistic scenario, the training is done only at a subset of the test locations, specifically, 20 and 4 locations in CEWIT and CSD, respectively. The testing locations are still uniformly distributed on the grid (Section 5.3)

6.4.2 *Observations*

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

- Hardware variations is a major issue for both RF map based techniques. When the same device is used for training and testing and the same locations are used for training and testing, the median error is zero. However, when the devices differ the error jumps up dramatically. This is a critical problem for such techniques, because device hardware will vary widely in a real-world deployment.
- On the other hand, WiGEM cannot match RADAR and Probabilistic for their most favorable case (same device, same test locations as training). But it performs at par with RADAR and Probabilistic when devices vary. This is particularly promising because unlike RADAR and Probabilistic, WiGEM does not have the overhead of a pre-deployment training.
- When the granularity of training is coarse, RADAR and Probabilistic show substantially poorer accuracy estimates. Thus, location estimates for such techniques are tightly bound to the granularity of the training effort. WiGEM is almost always better that RADAR and Probabilistic when they use coarser grain training. Sometimes the reduction in error is substantial (more than half).

6.5 Impact of Mobility

Finally, we show that the mobility of a client does not adversely affect WiGEM's operation or accuracy. In fact, mobility can be helpful. The mobility issue is important as for many practical use of WiGEM (e.g., indoor navigation) the client device can be continuously moving, providing RSS samples from different locations that may form the learning data set.

To model the impact of mobility, we designed the following simple experiment : Find the accuracy estimate for localizing a client device transmitting from a new location 'x' (meaning that the learning data-set has no samples from location 'x'). We consider 4 mobility scenarios : *No Mobility* , *Fair Mobility, Medium Mobility, High Mobility.*

No Mobility is the case of a static user transmitting from a fixed location, other than location 'x'. Using RSSI samples from this location, we try to provide location estimates for 200 RSS tuples from location 'x'. The experiment is repeated with the static user exclusively transmitting from every location in the testbed other than location 'x'.

Fair Mobility is the case of the mobile user who has covered 33% of the locations in the testbed, except location 'x'. The samples in the learning set are distributed uniformly across the locations he has covered. We then try to localize the same 200 RSS tuples from location 'x' as before. For *Medium Mobility*, the mobile user covers 67% of the locations in the testbed. In the Fair and Medium mobility scenarios, we repeat the experiment 10 times with randomly chosen locations which the mobile user visits. *High Mobility* is the scenario where the mobile user has covered all locations in the testbed except location 'x'.

It is important to note that across all experiments, the learning data-set had a fixed size, and the test data (200 RSS tuples from location 'x') was kept consistent. Moreover in each of the two testbeds, we repeated the experiment for ten different locations of 'x', chosen at random from the set of locations available in the dataset for that testbed. (Section 5.3). We use the Dell laptop and the Android phone for the CEWIT and CSD testbeds respectively. Figure 11 shows the average error across all experiments for each of the four respective mobility scenarios. The average error is high for the static user but decreases with increased mobility. Mobility during learning clearly helps localization.

6.6 WiGEM Running Time

Num. of learning samples	10	20	50	100
Running time (seconds)	30	61	169	348

Table 1: WiGEM Running times

WiGEM being an infrastructure based approach, running the backend computations can easily be offloaded to server machines in the cloud. Note that the running time for building the WiGMM for a device would depend on a number of factors like the number of possible target locations in the testbed, the number of samples in the learning data, the number of power-levels used, the log likelihood convergence threshold etc. Table 1 shows the typical running time required to build the WiGMM model for a single device one a single machine, as a function of the learning data-set size. This is on the CEWIT testbed and all other parameters are kept same as discussed in the paper. The machine used is a Dell PowerEdge SC1420 with two two Intel Xeon processors.

7. DISCUSSION

In its current embodiment, WiGEM uses a radio propagation model (Section 4.4) for initializing the WiGMM model. For handling identifiability in our model, we exploit the typical constraints between the means of the Gaussians at the same location for different power levels. Future work can explore whether enforcing similar constraints during run time increases the localization accuracy. Our framework could also be used to do a more efficient training process, whereby the radio propagation model is substituted by a few carefully done measurements. In this case, including the power of the source into the model may increase the robustness of the method and make it work for various devices. As future work, we plan to do adaptive localization by doing learning and using the adjacency of the locations as information to track how motion could evolve. This seems to have been done with EM before [2] and might be nicely combined with our technique. Additional factors like the number and location of sniffers, the size of the grid etc., and their effect on localization accuracy can also be explored.

8. CONCLUSIONS

In this work, we have developed WiGEM, an infrastructure based technique to localize a wireless client in an indoor environment based on the RSS of its transmitted packets as received by a sniffer (or an AP doubling as sniffer). WiGEM is based on a learning-based algorithm that can learn the parameters of a Gaussian Mixture Model dynamically from packets captured by the sniffers. By using dynamic packet captures for parameter estimation, WiGEM can provide location estimates that are much more robust in the face of device and power level variabilities, mobility, and changes and reconfiguration of indoor spaces that many training-based systems are susceptible to. The biggest advantage of WiGEM is that there is no explicit training phase. This saves a significant pre-deployment effort that is also difficult to maintain and update. Performance evaluations with a range of different WiFi devices in two different indoor testbeds demonstrate that WiGEM performs better than model-based techniques and at par or better than state-of-the-art RF map-based techniques. Of particular importance is WiGEM's superior performance when heterogeneous devices are used and when the RF map-based techniques have coarser training locations.

9. REFERENCES

[1] http://soekris.com/products/net4801.html.

[2] P. Addesso, L. Bruno, and R. Restaino. Adaptive localization techniques in WiFi environments. In *Proceedings of the 5th IEEE international conference on Wireless pervasive computing*, ISWPC'10, pages 289–294, Piscataway, NJ, USA, 2010. IEEE Press.

[3] P. Bahl and V. N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *Proceedings of IEEE INFOCOM*, pages 775–784, 2000.

[4] J. Bilmes. A gentle tutorial on the EM algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Technical report, 1997.

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

[6] S. Borman. The Expectation Maximization algorithm: A short tutorial. Technical report.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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 *Proceedings of IEEE INFOCOM*, pages 1001–1011, 2004.

[14] 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.

[15] 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.

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

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

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

[19] D. Reynolds. Gaussian Mixture Models. Technical report.

[20] 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.

[21] 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.

[22] 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.

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

[24] 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.