# Profiling-Based Indoor Localization Schemes

Israat Tanzeena Haque, *Member, IEEE*, and Chadi Assi, *Senior Member, IEEE*

***Abstract*—In this paper, we consider the indoor localization problem, i.e., identifying the Cartesian coordinates of an object or a person under the roof. To solve this problem we consider an RF-based localization method called *profiling*, a two-step process, where a radio map of the monitored area is first constructed by collecting signal strength from known locations. An unknown location is then predicted using this radio map as a reference. In this paper, we first propose a K nearest neighbor (KNN) profiling-based localization method dubbed *LEMON* (location estimation by mining oversampled neighborhoods). It is based on a low-cost, low-power wireless devices and ensures good accuracy compared to the state-of-the-art. We then propose a variant of LEMON called *combinatorial localization* which exhaustively searches for the best possible set of nearest neighbors. We further define a Bayesian network model for the same localization problem. The performance of these methods is evaluated through extensive experiments in various indoor areas. We found an interesting outcome that the simple KNN-based approach can offer better localization accuracy compared to other complex localization methods. Thus we further enhance the performance of the KNN-based approach using multiple RF channels.**

***Index Terms*—Indoor localization, localization error, profiling-based localization, received signal strength (RSS), RF-based localization.**

## I. Introduction

KNOWING the location of an object or a person is an interesting and challenging problem in many areas of practical applications, including security, health care, behavioral pattern recognition, geographic routing, robotics, etc. While the problem is well addressed by GPS [1] in outdoors, locating a device in indoor environment still remains a challenging problem due to the poor characteristics of GPS signals under the roof. Formally, an indoor *localization* problem can be defined as a procedure of determining the Cartesian coordinates of a wireless device within some monitored area. The problem can be tackled via several different technological approaches (e.g., infrared or acoustic sensing [2], [3], as well as various pressure/vibration sensors [4]), but the generic RF-based approach has the promise of simplicity, ubiquity, low cost, and un-obtrusiveness. Hence, in this study we select RF-based techniques for indoor localization. This class of techiques may either rely on some existing wireless infrastructure [most prominently, WiFi access points (APs)] or use special wireless nodes dedicated to the specific task of localization. However, using APs as infrastructure nodes may be challenged by the constraints on their number and locations. Thus, an AP deployment that can support localization should be designed with the purpose of aiding the localization in mind. Alternatively, one may consider dedicated low-cost low-power infrastructure nodes to design a localization system to avoid any constraints on their number and locations. We consider such devices in our localization scheme.

Unlike other techniques which rely on using the time of arrival (TOA) [5], or the angle of arrival (AOA) [6], [7], in this work we consider the received signal strength (RSS) [8]–[11] between the transmitter and the receiver as the localization attribute. This is because the RSS-based approach poses minimum requirements on the RF technology of the requisite modules, which translates into low cost and off-the-shelf availability. Within its realm, the RSS-based localization can be carried out using methods that are *range-based* [8], [12], *range-free* [6], [10], [13], or based on *fingerprinting* or *profiling* [9], [14]–[18].

In profiling-based schemes, the RSS metric depends on the distance between the transmitter and the receiver as well as the indoor environment. Accordingly, we may expect RSS readings from similar environment to behave similarly. This hope lies behind the construction of the *radio map* of the monitored area by gathering the RSS readings from known locations. This process of radio map generation is called *RSS profiling*. Now, to estimate the location of a query $q$ (we call this query sending device *tag*) based on a given set of RSS readings $\Phi$, this map is explored to search for a set of *$K$nearest neighbors (KNN)* of $\Phi$ in terms of minimizing the RSS distance. In the radio map, the locations of these chosen neighbors are also stored which can be used to predict the location of $q$. The procedure of localization then becomes a two-stage process: *profiling* and *estimation*. It has been demonstrated that this approach offers consistently a better location estimation accuracy than other categories of schemes [9], [11], [19]. Hence, we have considered RSS profiling as our localization approach.

In this paper, we first present our proposed KNN-based localization method *LEMON* (location estimation by mining oversampled neighborhoods) [19]. LEMON is based on low-cost low-power wireless sensors, which can be placed at any required indoor areas without any restriction on their numbers. This flexibility (unlike existing KNN-based methods RADAR [9] and LANDMARC [14]) of using a large number of low-cost low-power infrastructure nodes (we call them *pegs*) helps LEMON to generate the radio map precisely. Consequently, LEMON offers better accuracy compared to the state-of-the-art (which is verified through extensive experiments and the result is presented later in this paper). We next propose a

Manuscript received November 21, 2011; revised June 6, 2012, September 5, 2012, and September 30, 2012; accepted August 9, 2013. Date of publication September 25, 2013; date of current version March 2, 2015.

I. T. Haque is with the Computing Science, University of Alberta, Edmonton, AB T6G 2E8, Canada (e-mail: israat@cs.ualberta.ca).

C. Assi is with the Concordia Institute for Information Systems Engineering, Concordia University, Montréal, QC H3G 2W1, Canada (e-mail: assi@ciise.concordia.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSYST.2013.2281257

1932-8184 © 2013 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

combinatorial variety of the LEMON, where $\binom{S}{K}$ sets of $K$ nearest neighbors are generated out of $S$ profiled samples. Each one of these sets provides an intermediate estimation. The final estimation is computed as the average of these estimates across all sets. We also propose a model for Bayesian network and maximum likelihood estimation (MLE)-based localization.

The profiling data can be used to learn the required parameters of the Gaussian process (GP)- [20] and the support vector machine (SVM)- [21] based localization. Similarly the required parameters of the range-based *lateration* method can be obtained from the profiled data. Once the parameters are known, an unknown location can be predicted relying on the learned parameters. Thus we compare the performance of our proposed localization methods with these existing approaches. In particular, extensive experiments are performed at various indoor areas of the University of Alberta campus to evaluate the performance of the above localization methods. The results reveal that the simple KNN-based approach offers an impressive performance compared to complex methods.

All the above experiments are based on a single RF channel of communication. However, it is also possible to consider multiple-RF-channel-based approach to gather both the profiling and the localization data. We introduce such variant of the LEMON. This technique improves the localization performance as we get multiple estimates of the same location through multiple channels and use them along with the corresponding channel reliability as a weighting factor to get the final estimation.

The rest of the paper is organized as follows. Section II presents our proposed localization methods as well as the state-of-the-art. The hardware and the logistics of our experiments are presented in Section III. Discussion on the experimental results appears in Section IV. The next section presents the localization performance with multiple RF channels following the concluding remarks.

## II. Transforming RSS Samples Into Locations

In our localization system, a single RSS profiled sample (reference point) stored in a database can be viewed as a triplet $\langle C, \Omega, \tau \rangle$, where $C$ stands for the known coordinates of the profiled sample, $\Omega$ is the *association list*, and $\tau$, is the sample's *class*, where a class identifies the RF parameters of the transmitter (typically transmission power, bit rate, and channel number). The association list $\Omega$ is a set of pairs $\langle p, r \rangle$, where $p$ identifies a peg, and $r$ is the signal strength value observed at that peg.

In the following subsections, we first present our proposed localization methods and then briefly outline existing methods including lateration, GP, and SVM. Before we start to describe the methods, we would like to define the notation used throughout the description of these methods. Let $S \in \mathbb{R}^{n \times m}$ be the collection of the profiled samples, which is a $n \times m$ matrix, i.e., there are $n$ profiled samples each composed of a $m$-dimensional RSS vector produced across $m$ pegs in the monitored area. Also, $s_1 \in \mathbb{R}^{1 \times m}$ and $s_2 \in \mathbb{R}^{n \times 1}$ denote $m$-dimensional row vector and $n$-dimensional column vector, respectively. The former one represents a profiled sample across all the pegs while the latter one stands for all the profiled samples across a particular peg.

### A. *Our Methods of Localization*

*1) $K$ Nearest Neighbor Search:* In general, the KNN-based localization method searches the profiled data to choose the $K$ closest profiled samples in terms of minimizing the RSS discrepancy between the query RSS sample ($\Phi$) and the profiled ones. The average (weighted) of the coordinates (locations) of these $K$ samples generates the estimated location. Examples of such localizations are RADAR [9] and LANDMARC [14]. The limitations of these methods are related to the number of infrastructure nodes that need to be deployed and hence the higher cost associated with them. We overcome this limitation by introducing the KNN-based LEMON that relies on the low-cost low-power wireless sensors. Thus, we can use a large number of them to generate a rich set of profiled data of the monitored area to learn the environment precisely. The localization method of the LEMON is defined in Algorithm 1.

**Algorithm 1** LEMON $(\Phi, K)$

1: Find peg $p_h$ in $\Phi$ that received highest signal strength $r_h$ from the tag.
2: Find all the profiled RSS samples that perceived signal strength from $p_h$.
3: Sort these samples in terms of the RSS discrepancy between these samples and $\Phi$.
4: Choose top $K$ profiled samples (nearest neighbors).
5: $x_e \leftarrow \sum_{i=1}^{K} w_i x_i$
6: $y_e \leftarrow \sum_{i=1}^{K} w_i y_i$

The localization steps of the LEMON can be described as follows:

Having received a location estimation request, the server will search through the database of the profiled samples in order to choose a small number of best matching ones. The input to the estimation process is a list of readings $\langle p, r \rangle$, denoted by $\Phi$, representing the set of pegs that have received the current packet sent by the tag (at the unknown location) along with their RSS readings. One simple heuristic used to narrow down the subsequent search consists of selecting from $\Phi$ the pair $\langle p_h, r_h \rangle$, such that $r_h$ is the highest among all pairs in $\Phi$. Then, the procedure will only consider those profiled samples from the database whose association lists include $p_h$ as one of the pegs.

After selecting a subset of relevant profiled samples from the database, the server will sort them according to their discrepancy from the query sample $\Phi$. Then, it will select $K$ profiled samples in the signal space that are "closest" to the query sample. Suppose that $\Omega = \{\omega_1, \ldots, \omega_m\}$ and $\Psi = \{\psi_1, \ldots, \psi_m\}$ are two association lists. The discrepancy between these lists can be defined as

$$D(\Omega, \Psi) = \sqrt{\sum_{j=1}^{m} \left(R_\Omega(j) - R_\Psi(j)\right)^2} \tag{1}$$

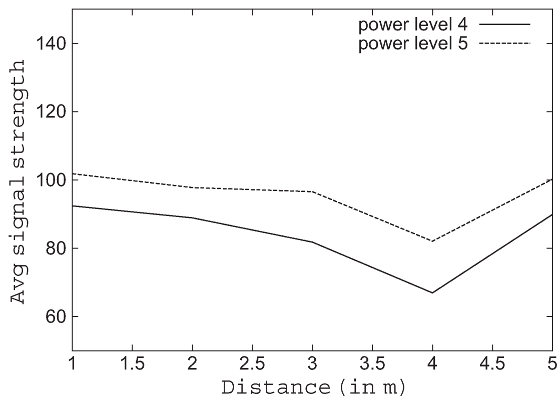



Fig. 1. Sample distribution of the RSS with the distance.

where $m$ is the total number of pegs in the network and $R_\Omega(j)$ is defined as $r_j$, if the pair $\langle p_j, r_j \rangle$ occurs in $\Omega$, and 0 otherwise. The same is true for $R_\Psi(j)$.

In our current experimental setup, both the profiling and testing consider the same transmit power. Thus (1) measures the discrepancy correctly. If the transmitter of the tag is tuned to a power level different from the profiled one, (1) will provide erroneous measurement. One way to amortize this problem is to consider the RSS normalization or calibration [22], [23]. As the RSS linearly varies with the distance (see Fig. 1), a linear normalization model can be considered to learn the necessary normalization parameters. Once such parameters are known, the RSS from a tag can be normalized to match the generated radio map. This way different power levels between the radio map and the tag can be handled and (1) can be applied to the RSS discrepancy measurement.

In the last step, the coordinates of the $K$ selected profiled samples are averaged to produce the estimated location as

$$(x_e, y_e) = \frac{\sum_{i=1}^{K}(x_i, y_i) \times (d_{\max} - d_i)}{K \times d_{\max} - D} \tag{2}$$

where $(x_i, y_i)$ are the coordinates associated with the profiled sample $i$. $d_{\max}$ is the maximum discrepancy from $\Phi$ among the best $K$ selected profiled samples and $D = \sum_i^K d_i$ is the sum of all those discrepancies. Note that we assign a zero weight to the farthest neighbor and consider the remaining $(K-1)$ nearest neighbors for the estimation.

LEMON performs the localization by searching through $n$ profiled records, where the RSS discrepancy between the $m$-dimensional query and the profiled data is measured to choose the $K$ nearest neighbors. The computational complexity for these two steps is $O(mn)$. Once we have the coordinates of the nearest neighbors, we can use any weighted averaging technique to get the estimated location of the tag. The complexity of this last step is thus constant.

*2) Combinatorial Localization:* In our proposed combinatorial localization, we first consider all the profiled samples, say $S$, from the database and sort them in terms of the RSS discrepancy between the tag and these samples. Now, instead of choosing the closest $K$ of them as the nearest neighbors, we generate $\binom{S}{K}$ combinations, say $I$, to get that many intermediate $K$ nearest neighbors. Therefore, we can compute $I$ intermediate locations as the weighted average of the $K$ members. The final estimated location $(x_e, y_e)$ is simply the weighted average of the $I$ intermediate estimations. We found that instead of considering all $S$ samples, a subset of them is actually more useful. In particular, we may choose the closest $s \subset S$ profiled samples and apply the above trick to estimate the location. The algorithm for the combinatorial localization is defined in Algorithm 2.

**Algorithm 2** Combinatorial Localization $(\Phi, s, K)$

Find the $s$ nearest neighbors of the query tag $\Phi$.
Generate $I = \binom{s}{K}$ set of $K$ nearest neighbors.
**for** $i = 1$ to I **do**
  $x_i \leftarrow \sum_{j=1}^{K} w_j x_j$
  $y_i \leftarrow \sum_{j=1}^{K} w_j y_j$
**end for**
$x_e \leftarrow \sum_{l=1}^{I} w_l x_l$
$y_e \leftarrow \sum_{l=1}^{I} w_l y_l$

In the combinatorial localization, the complexity for the first step is $O(mn)$. Let $\bar{s} = \min(K, s-K)$, then the complexity for $\binom{s}{K}$ is $O(\bar{s})$.

Both the combinatorial and the KNN-based localizations need $K$ as the input to compute the location. In the former approach if we consider the profiled data and estimate the location of each profiled point, then for each such estimation we will have a set of $K$ neighbors that will generate the most accurate location. By exploring all these $K$ values over all the profiled points, the most frequent one can be chosen as the input for the single-$K$-input-based localization like LEMON. Note that the same trick can be applied to LEMON over the profiled data to figure out which value of $K$ gives the best estimation of the profiled points. However, in case of the combinatorial localization, we have the opportunity to explore a large number of $K$ compared to the single-$K$-input-based LEMON, which may help to choose more accurate value of $K$ to use during the online location estimation phase. In a nutshell, the input parameter $K$ for the LEMON might be learned and tuned through the combinatorial approach.

*3) Bayesian Network Approach:* In our localization problem, the system variables can be defined as the set $\mathcal{X} = \{X, Y, r_1, r_2, \ldots, r_m\}$. $X$ and $Y$ are random variables associated with the location and $r_i$ are variables corresponding to the RSS readings at the m pegs. We may say that the RSS reading at a peg is a function of its distance to the transmitter, which could be captured with the distribution $P(r_i|X, Y)$. Furthermore, we can say that once we know the position of the transmitter the probability of observing a specific RSS at peg $i$ is independent of the readings at the other pegs. Thus we define the joint distribution as follows:

$$P(\mathcal{X}) = P(X, Y) \prod_{i=1}^{m} P(r_i|X, Y). \tag{3}$$

If we make the further assumption that the $X$ and $Y$ coordinates are independently distributed (if they are correlated we can still use the chain rule to decompose the distribution

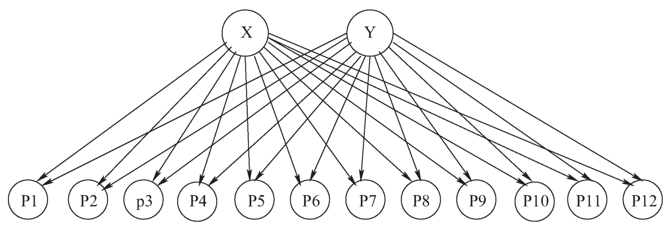



Fig. 2. DAG representation of the Bayesian network model.

to $P(X,Y) = P(X|Y)P(Y)$), we can also break down their marginal distribution $P(X,Y) = P(X)P(Y)$. Thus we may define a DAG (directed acyclic graph) shown in Fig. 2 as the RSS-based localization model.

Given the grid nature of the locations we treat $X$ and $Y$ as discrete random variables in our model, but we model RSS data as Gaussian random variables. In particular, at any particular node in the DAG we store a table with the probability of the variable taking each of the specific values. To store the Gaussian models we fit a distribution of the form $P(r_i|X,Y) \sim N(\mu_{x,y}, \sigma^2)$, which means that for each assignment to its discrete parents we get a different mean. Though the number of readings per sample (five readings were taken for each profiled and testing samples) might be sufficient to estimate the mean of a distribution, it might not be reasonable to estimate its variance, especially when all five data points are pretty much centered on the mean (i.e., almost constant). Thus it would be reasonable to assume that all pegs have the same variance and treat this as a tuning parameter that we can optimize on. In particular, we can learn a global variance based on the profiled data that is applicable to all distributions.

Once we have our model and distributions, we are ready to predict the location of the tag. The association list of that tag could be used to compute $P(\mathcal{X})$ for every profiled points. The weighted average of all the samples thus could be defined as

$$(x_e, y_e) = E\left(P(\mathcal{X})|r_i\right) \quad \text{where} \quad i = 1 \text{ to } m. \tag{4}$$

In the Bayesian networks, we first need to learn the model, which is proportional to the size of the profiled data, i.e., $O(mn)$. Once we learn the model, the prediction is then performed by using the probability score of the profiled data for the query. The complexity for this step is then $O(mn)$. Similarly, in the MLE we also need to learn the model and compute the probability score of the profiled data for a given query. Then, we report the most probable profiled location as the estimation.

*4) Maximum Likelihood Estimation:* With the MLE [24], [25] approach, we estimate the tag's location as the single profiled sample $(x_j, y_j)$ that maximizes $P(\mathcal{X})$. If we assume that all $(x_j, y_j)$ are equally likely and the pegs are independent, we shall obtain (3). However, instead of taking the weighted average we report here the most likely profiled point as the estimated location. Note that when we have no knowledge about prior probability, we may use the MLE.

### *B. Other Methods of Localization*

In this section, we briefly outline the lateration [26], [27], the GP [20], and the SVM [21] that can be used for the profiling-based localization.

The idea of lateration [26], [27] is to estimate the physical distance between the pegs and the tag by using a path-loss model on the RSS. Once such distances are known, a system of nonlinear equations is solved to estimate the location of the tag. In lateration the average loss is defined as

$$\bar{L}[\text{dB}] = C - 10\alpha \log(d) \tag{5}$$

where $d$ is the distance between the transmitter (tag) and receiver (pegs) expressed in units of distance $d_0$ (e.g., meters). The term $C$ accounts for the frequency dependent component and antenna gains and $\alpha$ is known as the *path loss exponent*.

Thus, the path-loss could be described as a straight line equation with respect to $\log(d)$. Simple least-square fitting can be applied to determine the path-loss exponent $\alpha$ (as well as $C$). In our localization experiment, we use the profiled points to learn $C$ and $\alpha$ based on (5). However, we define two kinds of parameters namely, *local* and *global*. The former approach includes a pair of parameters for every peg in the system, i.e., 12 pairs of parameters are learned for 12 pegs. The latter option deals with only one pair of parameters for all the pegs by considering $S \in \mathbb{R}^{n\times m}$ and uses them to compute the distance.

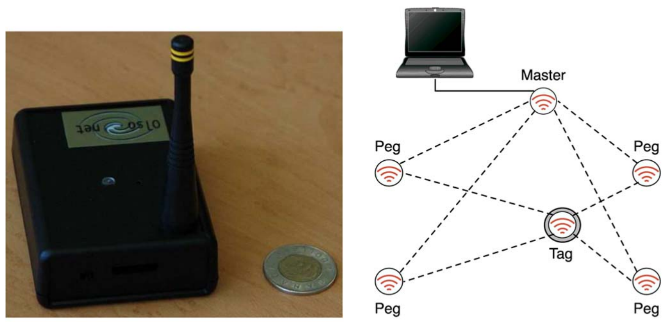
Note that our localization problem is considered as a regression problem but not a classification problem as we need to predict the coordinates (locations of the tag) rather than identifying the region of the query tag. For this purpose we may consider nonlinear-regression-based Gaussian process (GP) and support vector machine (SVM) [28], [29], where the nonlinear "kernel trick" is used to project the input space to higher-dimensional feature space. For the estimations, we consider the relationship between the RSS vector and the locations in the feature space, where these two may be linearly related. Therefore, we may consider the following linear model:

$$f(\mathbf{x}) = \mathbf{x}^T\mathbf{w}, \quad y = f(\mathbf{x}) + \varepsilon \tag{6}$$

where $\mathbf{x} \in \mathbb{R}^m$, $\mathbf{w} \in \mathbb{R}^m$, $f \in \mathbb{R}$, and $y \in \mathbb{R}$ are the input vector, the weight vector, the function value, and the observed target value, respectively.

In case of the SVM, the kernel function that gave us the best performance is the radial basis function (RBF). This is chosen based on the model defined for the training profiled data. The model also determines the weight $\mathbf{w}$ and the bias $\varepsilon$. An optimization problem (see [29] for the details) is solved to learn these parameters. The location of a query tag is performed by first projecting the input vector to the feature space using the chosen RBF kernel. In that space the weighted average of the support vectors is reported as the estimation. We considered Weka [30] to implement the SVM-regression-based localization, where $X$ and $Y$ coordinates are trained and tested individually as per the above description.

We also consider (6) for the regression while using GP, where the observed values $y$ are assumed to differ from function values by an additive noise $\varepsilon \in \mathbb{R}$, which is assumed to be Gaussian distributed with zero mean and variance $\sigma^2$, i.e., $\varepsilon \sim \mathcal{N}(0, \sigma^2)$. The input vector is projected on the feature space using the RBF kernel, which is chosen based on the model defined over the profiled data. The model also provides us the posterior distribution of the weight. During the prediction, the




**The EMSPCC11 mote** **Data collection**

Fig. 3. EMSPCC11 mote and the outline of the experimental setup.

RSS vector of the query tag is first projected on the feature space using the RBF kernel. The posterior distribution of the weight for the tag is then determined on this space to get the mean as the estimation. We used Weka for the GP regression and also trained and tested $X$ and $Y$ coordinates separately.

We compare our proposed localization method with the above ones in terms of simplicity and accuracy. The analysis of our experimental results reveals that not only these methods may introduce complexity but also may not provide adequate accuracy required in indoor environment.

## III. Experiments

A prototype system was implemented for collecting the RSS measurements at various indoor areas on the campus of the University of Alberta.

### *A. Hardware*

In our experiments, the wireless device used for both pegs and tags alike is the EMSPCC11[1] from Olsonet Communications shown in Fig. 3. EMSPCC11 is a low-cost low-power mote for wireless sensor networking programmable in PicOS [31]. The mote employs the MSP430F1611 microcontroller and the CC1100 RF module, both from Texas Instruments. The RF module operates within the 916 MHz band. The transmission power is settable from $-30$ dBm to 10 dBm (in 8 discrete steps), the bit rate options are 5 kb/s, 10 kb/s, 38 kb/s, and 200 kb/s, and there are 256 different channels (numbered 0 to 255) with 200 kHz spacing. The experiments reported in the rest of this paper were carried out at the second lowest power setting with 5 kb/s transmission rate using channel 0.

### *B. Logistics*

The program runs in the nodes (the *praxis* according to PicOS terminology [31]) allowed us to obtain the RSS readings between all pairs of directly reachable nodes and for any selected setting of the transmitter (output power, bit rate, channel number). Thus, the deployment of nodes was usually followed by data collection: the nodes would exchange a massive number of packets, conveying to the central node (dubbed the *master*) their parameters (sender/receiver ID, serial number, transmitter parameters, RSS). The master node was connected to a laptop, where all the data collected by the network were deposited.

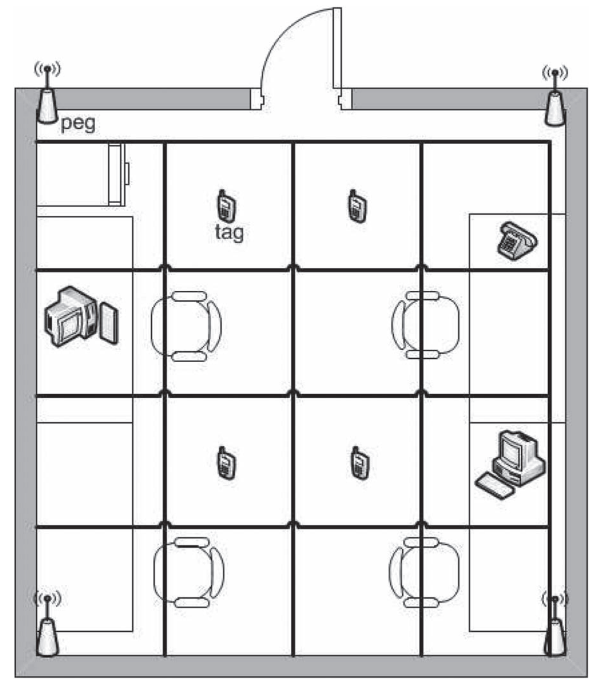



Fig. 4. Sample distribution of furniture.

We generated a grid in the monitored area with each grid cell size 1 m $\times$ 1 m. The RSS readings for the profiled points were gathered from every grid point, whereas the test data was collected from the center of each grid cell (see Fig. 4). Note that the aforementioned cell size was chosen based on an initial experiment. We collected two sets of RSS data a) the profiled samples from known locations and b) the test data to be localized. We initially tested the impact of time on the localization performance. In particular, we gathered the profiled data over three days, then started gathering the test data over more than a day. We noticed that the gathered RSS has almost same behavior as long as the indoor environment (setup) remains same. However, if we rearrange the setup of furniture, a change in the RSS was observed. In order to observe the impact of the body blockage, we did a series of experiments where tag was carried by a person. The outcome of the localization in such environment is presented in Section IV.

Our experiments were carried out in a three-room scenario: three adjacent graduate student offices (size of each room was 3 m $\times$ 5 m), with each office outfitted with four wooden tables, four chairs, several wooden shelves mounted on the wall, two PC computers, and a steel file cabinet (see Fig. 4). We placed four pegs at the corners of each room and gathered 52 profiled points as part of the radio map construction to test 33 tag locations.

We considered a monitored area of reasonable size, e.g., 3 $\times$ (3 m $\times$ 5 m) and 13 m $\times$ 3 m, which is bigger than the monitored area considered in [14]. Our methods can also be used to monitor areas of larger sizes, but this was not considered in this work.

We conducted initial experiments in an office room (first floor), seminar room (third floor), class room (fifth floor),

[1]See http://www.olsonet.com/Documents/emspcc11.pdffordetails.

and corridors in the presence of various indoor furniture and people. The empirical data was used to learn and tune different parameters. Once such initial investigation was completed, we considered multi-room scenario, where three consecutive office-rooms were used, one located on the first floor and the other one on the fourth floor. In the latter case, a heating furnace was present between two rooms. This scenario was considered to include diverse propagation conditions. Both of these multi-room empirical results are presented in this paper.

### C. Performance Evaluation

Once we finish the estimation, we measure the *error distance* (error in meter) between the actual and the estimated locations as the Euclidean distance between them. All our results report *average error distance*. Another evaluation technique is to measure the *figure of merit* of a localization based on the size of the monitored area $(Z)$ and the average error distance ($D_{er}$ in meter). It is defined as

$$\frac{D_{er}}{\sqrt{\frac{Z}{m}}}.$$

Thus the smaller this value the better the performance of the localization method.

## IV. Discussion of the Experimental Results

In this section we first define our proposed RSS scaling trick. Then we compare LEMON with existing profiling-based schemes that use KNN search.

The analysis of our experimental data revealed that most of the localization error occurred because of assigning too much relevance to low RSS values, i.e., corresponding to weak reception, which would exhibit large statistical fluctuations. With the linear factoring of the distance of the requisite RSS vectors into the coordinate averaging formulas (Section II-A1), such a fluctuation, additionally amplified by the presence of indoor obstacles, may cause the coordinates of a distant profile point to affect the estimation in a manner disproportionate to its *true* proximity to the tag.

Hence, we define the following scaling formula to control the impact of RSS values with a different magnitude:

$$R_s = \left(\frac{r - MIN}{MAX - MIN}\right)^{\gamma}$$

$MIN$ and $MAX$ represent the minimum and the maximum obtainable readings across all sets of measurements, and $\gamma$ is the amplification factor.

To evaluate the scaling performance, we first considered the impact of body blockage (tag was holding close to the body). In 90% of the cases we were able to estimate the tag location with an error less than 1 m. We then considered the experimental setup mentioned in Section III, where the entire experiment was carried out while people were roaming around. The best result for LEMON is obtained with $K = 6$ and a scaling factor $\gamma = 3.5$. We then compare this result with the state-of-the-art, which is presented in Fig. 5.

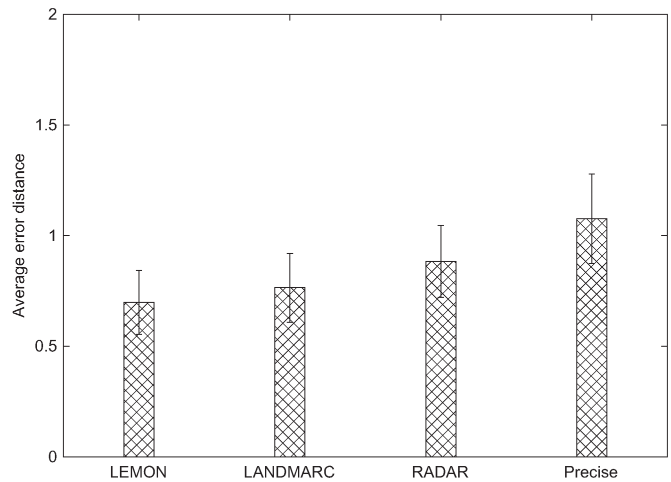



Fig. 5. Performance comparison using our experimental data.

Note that different localization schemes use different hardware and/or testbed size for the evaluation purposes. Thus it is not straightforward to compare them. The average error distance of the RSS-based profiling-based schemes in [9], [14], [15] is already larger than that of LEMON. However, for the sake of completeness and consistency, we use our gathered data to evaluate these existing schemes (see Fig. 5). Their difference lies in the averaging technique of the chosen $K$ coordinates. LEMON still performs better than the state-of-the-art. This indicates that the averaging technique that we consider in LEMON is quite useful.

We compared the performance of the LEMON with the state-of-the-art in terms of using the number of profiled points. We gathered 52 such points to estimate 33 locations, i.e., their ratio is around 1.5. With this setup, the average error of LEMON is 0.67 m. This error remains 0.9 m even when we use only half of these profiled points (placed at each corners of every 2 m $\times$ 2 m grid cells), i.e., we use only 27 profiled points to evaluate 33 tag locations. Both of these errors are better than the one reported in [9], [14], [15]. The ratio of the profiled points to the number of estimations is 4 in [9], where 280 profiled points are used to estimate 70 locations. This ratio is 2 in case of [14] and in [15] this ratio is not mentioned. Thus the LEMON ensures good accuracy compared to the state-of-the-art even using less number of profiled points, which in turn ensures simplicity of the profiling phase.

We further investigated the impact of the number of pegs on the localization though we do not impose any restriction on this number (we can use a large number of pegs to generate an accurate radio map of the monitored area, which is considered as a strength of the LEMON). The existing schemes use a limited number of the infrastructure nodes because of their high cost. This limitation has impact on the estimation accuracy. We avoid such constraint by using low-cost low-power sensors. For instance, we use 12 pegs in our experiments compared to 3 in [9] and 4 in [14]. However, even when we consider only four pegs located at the four corners of the monitored area, the average error distance (1.23 m) is still better than the one reported in the aforementioned schemes.

The LEMON exhibits a figure of merit of 0.36 using the scaled RSS. This value is slightly higher than the one presented in [32], [33]. Both the schemes dynamically update their estimations based on the change in the indoor environment, which seems to be an effective approach to ensure good figure of merit. In contrast, we use static KNN where a fixed $K$ number of neighbors is chosen from the generated radio map. from the generated radio map. This in turn gives us around 0.36 figure of merit compared to 0.2 in [32], [33]. However, the complexity of the schemes in [32], [33] higher than that of the LEMON, thus better suited for the resource constrained WSN. Thus we may improve the figure of merit of the KNN-based LEMON by dynamically updating the neighbor selection criterion.

Further, we observed that not only the obstacles but their types also have impact on the RSS; the RSS measured next to a steel filing cabinet is contaminated more compared to the one near a wooden chair. Thus, it is possible to generate a RSS distribution model in terms of the types of obstacles presents in the monitored area. This in turn will help to identify the region of interest during the estimation. Once such region is determined, a propagation model or the RSS distribution pattern for the detected obstacles can be used to dynamically choose nearby profiling points. In particular, instead of considering a fixed $K$ number of neighbors while choosing the profiling points, a dynamic approach will be considered where $K$ will be different for different measurements. In addition, it might also be possible to identify which profiling points are affected most by the multipath, and hence could be excluded from the radio map.

We finally compared the KNN-based LEMON with the other methods of localization, and considered both the unscaled and the scaled RSS. The best estimation of LEMON is obtained with $K = 8$ while using the unscaled RSS. The best performance of the lateration is obtained with 3 pegs and the local parameters. The environment with densely deployed furniture may need local parameters to better estimate the path loss exponent to take into account the small scale fading. It also imposes extra parameter estimation which is related to the number of pegs in the system. However, this extra estimation is worthwhile in terms of the measured accuracy. Thus we may suggest to consider local parameter estimation while using log-distance-based lateration to predict the tag's location, especially indoors where the presence of multipath is high. However, the performance of lateration is the worst compared to the other methods we considered. Lateration considers path-loss model that may be approximate. Thus the estimated path-loss exponent, even as a local parameter, may not provide a good distance measurement. Note that it gives the best results with a small (closest) number of pegs, where chance is high that those pegs have direct LOS (line-of-sight) with the tag. The KNN-based LEMON, on the other hand, does not try to separate LOS and NLOS (non-LOS) components of the signal strength. Indeed, the method considers signal strength as a quantity and compares the stored RSS with the query RSS with the expectation that the nearby RSS may experience the same environment thus yielding a similar measurement. The LEMON also does not assume that X and Y locations are independent. The performance comparison of different localization methods with the unscaled RSS is presented in Fig. 6.

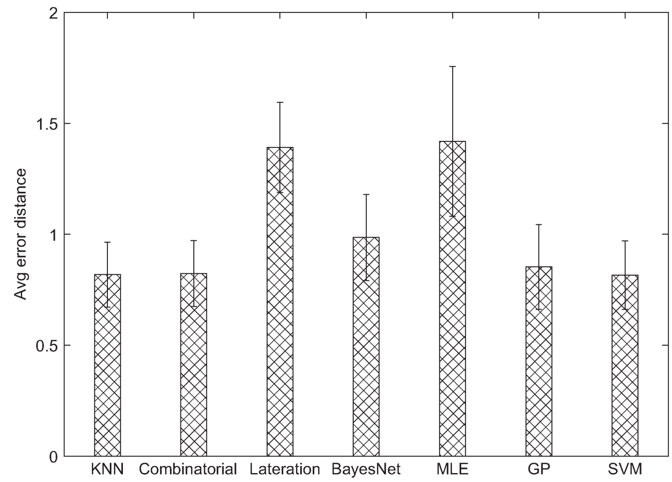



Fig. 6. Performance comparison of different localization methods without scaling.

In the combinatorial localization method, we try to generate, say $I$, intermediate estimations using a subset of the profiled samples and a suitable $K$. The idea is to use those $I$ estimations to produce the final prediction. The performance of this approach is almost the same as that of the LEMON. In the Bayesian networks, we learn the distribution of the profiled points and use that information to learn the probability for a query tag. The final estimation is the weighted average of the profiled points, where the weight is the corresponding probability. The performance of this method is worse compared to the LEMON. We use five samples per profiled point to generate the distribution, most of which are very consistent to each other. Thus this may not provide good distributions to be useful for the prediction. Note however that using more samples may help, but that increases the complexity of the data gathering phase. Without this extra burden we can choose the LEMON, which is simple and ensures good accuracy. The MLE performs worse compared to the Bayesian Networks as it reports the highest probable profiled point as the estimation. The last two methods the GP and the SVM both use the kernel trick to project the low-dimensional input space to the high-dimensional feature space. Then they perform linear regression in that feature space. The former learns the posterior distribution of the weight to generate posterior distribution for the query tag. The mean of this distribution is the estimation. The latter one estimates a weight vector that ensures minimum loss or error during the prediction. Their performance is similar to each other and also similar to the LEMON. Thus, a complicated regression after the kernel trick might not be useful for our localization purposes, where simple RSS scaling offers good estimation accuracy, as we will present next. Note that all the results shown in the remainder of the paper consider the scaled RSS.

The performance of different localization methods using the scaled RSS is given in Fig. 7. Note that the log-distance-based lateration is not evaluated with the scaled RSS as that may ruin the RSS versus the distance relationship. The performance of the remaining methods have similar trend as for the unscaled data. The LEMON performs best, followed by the Bayesian Networks and the GP. However, the SVM and the MLE are the

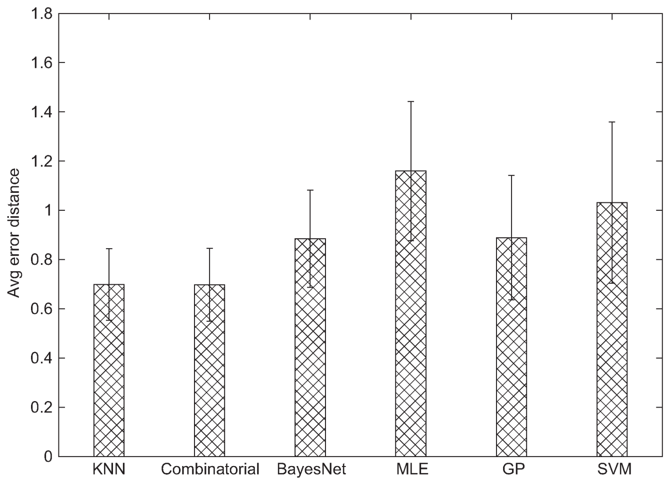



Fig. 7. Performance comparison of different localization methods with scaled RSS.

worst. In the SVM, scaling the RSS may affect its regression on the feature space.

The performance of all the algorithms does not vary significantly in case of the scaled RSS. It may happen that the dependency of these algorithms on the training data may not be useful; i.e., if the data has noise, learning may not provide quite useful information regardless of how sophisticated the localization algorithm is. All localization schemes can be affected by the noisy RSS captured through the low-cost low-power sensors. Nonetheless, simple KNN-based approach ensures better accuracy compared to others.

## V. Multi-Channel RF-Based Indoor Localization

Usually the range-based localization schemes use true physical distances among devices to learn the required parameters for the localization. The profiling-based schemes do not require such learning. However, to better utilize the profiled data, we learned further insight information like the cross correlation coefficient between the distance and the RSS to measure the reliability of the pegs. This information helped us to understand the behavior of different pegs and the RF channels. Channels may have different degree of noise sensitivity, and the RSS from the same location may behave differently across different channels. Some channels may be less affected by multipath fading, etc. compared to others. Thus, we may able to define the reliability of the channels in terms of the RSS fluctuations. In particular, the cross correlation of the pegs can be calculated to rank the channels in terms of the number of highly correlated pegs. The *cross correlation coefficientc* between the RSS and the Euclidean distance (between peg and tag) for each peg can be defined as

$$c = \frac{\sum_{i=1}^{n}(r_i - \bar{r})(d_i - \bar{d})}{\sqrt{\sum_{i=1}^{n}(r_i - \bar{r})^2 \sum_{i=1}^{n}(d_i - \bar{d})^2}} \quad (7)$$

where $n$, $r$, and $d$ are the number of profiled samples, the RSS value, and the known distance between the peg and the profiled point, respectively. The value of $c$ is between $-1$ and 1, and the smaller the value the weaker the correlation between the distance and the RSS.

TABLE I
Best Parameters for Channels

| Channel | $\gamma$ | $K$ |
|---|---|---|
| Channel 0 | 4.0 | 7 |
| Channel 84 | 3.5 | 8 |
| Channel 170 | 4.5 | 9 |
| Channel 255 | 3.0 | 5 |

*Definition 1:* A reliable peg is a peg with cross-correlation coefficient greater than a threshold, where threshold is an estimated parameter

*Definition 2:* Channel reliability is the ratio of the number of reliable pegs to the total number of pegs, $CH_{re} = (R_p/T_p)$.

$R_p$ and $T_p$ are the number of reliable pegs and total number of pegs, respectively.

The estimated locations across different channels can be averaged as

$$(x_e, y_e) = \sum_{i=1}^{ch} w_i(x_i, y_i), \quad \text{where} \quad w_i = \frac{CH_{re_i}}{\sum_{i=1}^{ch} CH_{re_i}} \quad (8)$$

$ch$ is the total number of channels and $(x_i, y_i)$ is the estimated location using channel $i$. Note that if $CH_{re}$ is 1 for each channel, i.e., all of them are perfectly reliable, the estimation becomes a simple averaging across the channels.

We performed an experiment using three consecutive rooms and set a 13 m $\times$ 3 m grid, where four different RF channels (0, 84, 170, and 255) (channel 0 is 800 MHz and consecutive channels were 200 kHz apart) were used to gather the signal strength. One of the walls was about 1 m thick (includes a heating furnace) whereas the other one was 0.15 m. As in other experiments, we gathered the profiled points from every grid point but the test points were located at the center of each grid-cell. There were 73 profiled points to test 36 test points. We placed 22 pegs to cover the entire area. The idea was to place more pegs near the walls shared by consecutive rooms. The best RSS scaling factor ($\gamma$) and the number of nearest neighbors ($K$) for each channel are presented in Table I.

The localization is first performed using individual channels and the performance is shown in Fig. 8. Then we take the average of these estimated locations across the channels as the new estimation. Indeed, applying weighted average, based on the above definitions, helps to reduce the error distance.

### A. RF-Based Room Localization

Buildings are usually composed of multiple rooms. Indeed, in many applications of indoor localization room identification may be sufficient instead of knowing the exact location of the tag. Given a set of samples from the reference points, the room localization problem is to identify the room from where a tag sends its query. We tested 33 tags queries using five nearest neighbors and identified the room where these tags belonged by using the label of these five nearest neighbors. In particular, we choose five nearest neighbors that minimize the RSS discrepancy. Then we check their room label and count for the majority, which is chosen as the label of the

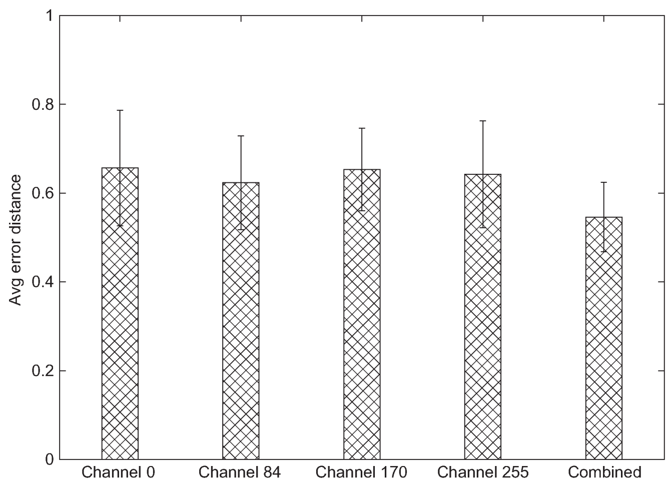



Fig. 8. Effect of using multiple channels on localization.

TABLE II
ROOM LOCALIZATION USING MULTIPLE RF CHANNELS

| Channel | Correct Detection | Wrong Detection |
|---|---|---|
| CHANNEL 0 | 35 | 1 |
| CHANNEL 84 | 35 | 1 |
| CHANNEL 170 | 34 | 2 |
| CHANNEL 255 | 34 | 2 |

query tag. Out of the 33 queries, only one is misclassified. As room localization performs adequately and has many potential applications, we perform further experiments in multiple rooms (mentioned above) and test the room localization for that setup. The outcome is shown in Table II.

## VI. CONCLUSION

We analyzed different localization methods ranging from simple nearest neighbor search to complex SVM. We also tested localization based on lateration, which estimates the path-loss exponent to relate the distance between the transmitter and the receiver and uses this distance to perform lateration. Experimental results show that the simple nearest-neighbor-search-based method like LEMON can be a good candidate for the profiling-based indoor localization in terms of offering adequate estimation accuracy. We have also performed localization while considering multiple RF channels. Finally, we have proposed a simple room localization technique and verified its performance under both single and multiple RF channels. Experimental results show that the multiple channels help to improve the localization accuracy. Also, the accuracy of the room localization is promising.

## REFERENCES


[1] *Global Positioning System Standard Positioning Service Specification*, U.S. Coast Guard Navigation Center, Alexandria, VA, USA, Jun. 1995.

[2] V. Fox, J. Hightower, L. Liao, D. Schulz, and G. Borriello, "Bayesian filtering for location estimation," *IEEE Pervasive Comput.*, vol. 2, no. 3, pp. 24–33, Jul.–Sep. 2003.

[3] S. Jung and W. Woo, "UbiTrack: Infrared-based user tracking system for indoor environments," in *Proc. ICAT*, Coex, Korea, Dec. 2004, pp. 181–184.

[4] S. Yeh, K. Chang, C. Wu, H. Chu, and J. Hsu, "GETA sandals: A footstep location tracking system," *Pers. Ubiquitous Comput.*, vol. 11, no. 6, pp. 451–463, Aug. 2007.

[5] A. Ward and A. Jones, "A new location technique for the active office," *IEEE Pers. Commun.*, vol. 4, no. 5, pp. 42–47, Oct. 1997.

[6] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low cost outdoor localization for very small devices," *IEEE Pers. Commun. Mag.*, vol. 7, no. 5, pp. 28–34, Oct. 2000.

[7] D. Niculescu and B. Nath, "VOR base stations for indoor 802.11 positioning," in *Proc. 10th Int. Conf. MobiCom Netw.*, New York, NY, USA, Oct. 2004, pp. 58–69.

[8] S. Ali and P. Nobles, "A novel indoor location sensing mechanism for IEEE 802.11 b/g wireless LAN," in *Proc. 4th WPNC*, Hannover, Germany, Mar. 2007, pp. 9–15.

[9] P. Bahl and V. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proc. IEEE INFOCOM*, Tel Aviv, Israel, Mar. 2000, pp. 775–784.

[10] D. Niculescu and B. Nath, "Ad hoc Positioning System (APS)," in *Proc. IEEE Int. Conf. GLOBECOM*, San Antonio, TX, USA, Nov. 2001, pp. 2926–2931.

[11] K. Whitehouse, C. Karlof, and D. Culler, "A practical evaluation of radio signal strength for ranging-based localization," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 11, no. 1, pp. 41–52, Jan. 2007.

[12] J. Kang, D. Kim, and Y. Kim, "RSS self-calibration protocol for WSN localization," in *Proc. 2nd Int. Symp. Wireless Pervasive Comput.*, San Juan, Puerto Rico, Feb. 2007, pp. 181–184.

[13] A. Küpper, *Location-Based Services: Fundamentals and Operation*. Hoboken, NJ, USA: Wiley, 2005.

[14] L. Ni, Y. Liu, Y. Lau, and A. Patil, "LANDMARC: Indoor location sensing using active RFID," *Wireless Netw.*, vol. 10, no. 6, pp. 701–710, Nov. 2004.

[15] E. Martin, O. Vinyals, G. Friedland, and R. Bajcsy, "Precise indoor localization using smart phones," in *Proc. Int. Conf. Multimedia*, Firenze, Italy, Oct. 2010, pp. 787–790.

[16] T. Oktem and D. Slock, "Pairwise error probability analysis for power delay profile fingerprinting based localization," in *Proc. IEEE 73rd VTC Spring Conf.*, Yokohama, Japan, May 2011, pp. 1–5.

[17] M. Hossain, H. N. Van, Y. Jin, and W. Soh, "Indoor localization using multiple wireless technologies," in *Proc. IEEE Int. Conf. Mobile Adhoc Sensor Syst.*, Pisa, Italy, Oct. 2007, pp. 1–8.

[18] N. Swangmuang and P. Krishnamurthy, "An effective location fingerprint model for wireless indoor localization," *Pervasive Mobile Comput.*, vol. 4, no. 6, pp. 836–850, Dec. 2008.

[19] I. Haque, I. Nikolaidis, and P. Gburzyñski, "A scheme for indoor localization through RF profiling," in *Proc. Int. Workshop SyCoLo*, Dresden, Germany, Jun. 2009, pp. 659–663.

[20] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.

[21] A. Ukil, *Support Vector Machine, Chapter 4, Intelligent Systems and Signal Processing in Power Engineering*. Berlin, Germany: Springer-Verlag, 2007.

[22] Y. Chen, D. Lymberopoulos, J. Liu, and B. Priyantha, "FM-based indoor localization," in *Proc. 10th Int. Conf. MobiSys, Appl. Services*, 2012, pp. 169–182.

[23] M. B. Kjrgaard, "Automatic mitigation of sensor variations for signal strength based location systems," in *Proc. 2nd Int. Workshop LoCA*, 2006, pp. 30–47.

[24] S. J. Russell and P. Norvig, *Artificial Intelligence: Modern Approach*, 3rd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2003.

[25] F. Seco, A. R. Jimenez, C. Prieto, J. Roa, and K. Koutsou, "A survey of mathematical methods for indoor localization," in *Proc. IEEE Int. Symp. Intell. Signal Process.*, Budapest, Hungary, Aug. 2009, pp. 9–14.

[26] K. Langendoen and N. Reijers, "Distributed localization in wireless sensor networks: A quantitative comparison," *Comput. Netw., Int. J. Comput. Telecommun. Netw.*, vol. 43, no. 4, pp. 499–518, Nov. 2003.

[27] T. S. Rappaport, *Wireless Communications: Principles and Practice,* 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2003.

[28] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York, NY, USA: Springer-Verlag, 2009.

[29] S. K. Shevade, S. S. Keerthi, C. Bhattacharyya, and K. R. K. Murthy, "Improvements to the SMO algorithm for SVM regression," *IEEE Trans. Neural Netw.*, vol. 11, no. 5, pp. 1188–1193, Sep. 2000.

[30] Machine Learning Group at the University of Waikato, Hamilton, New Zealand, "WEKA (Waikato Environment for Knowledge Analysis)," [Online]. Available: http://www.cs.waikato.ac.nz/ml/weka/

[31] P. Gburzynski and W. Olesinski, "On a practical approach to low-cost ad hoc wireless networking," *J. Telecommun. Inf. Technol.*, vol. 2008, no. 1, pp. 29–42, Jan. 2008.

[32] A. Bahillo, R. M. Lorenzo, P. Fernandez, F. A. Lago, E. Garcia, J. Blas, and E. J. Abril, "Robust indoor positioning provided by real-time rssi values in unmodified wlan networks," *IEEE J. Sel. Topics Signal Process.*, vol. 3, no. 5, pp. 821–831, Oct. 2009.

[33] A. Kushki, K. N. Plataniotis, and A. N. Venetsanopoulos, "Intelligent dynamic radio tracking in indoor wireless local area networks," *IEEE Trans. Mobile Comput.*, vol. 9, no. 3, pp. 405–419, Mar. 2010.

**Israat Tanzeena Haque** (M'13) received the B.S. degree in electronics and computer science in 1999, the M.S. degree in computer science from Concordia University, Montreal, QC, Canada, in 2004, and the Ph.D. degree from the Department of Computing Science, University of Alberta, Edmonton, AB, Canada, in 2011.

Currently, she is working in the same department as a Postdoctoral Fellow. Her research interest includes routing, localization and location-awareness, topology control, and resource management in adhoc and sensor networks and cloud computing. She worked as a Lecturer in the Department of Computer Science and Engineering, Asian University of Bangladesh, Dhaka, Bangladesh. She was the chair of the IEEE - Northern Canada Section: Computers and Communication chapter from 2010 to 2012. She is serving as a TPC member of the IEEE conferences ICC, ICCCN, ICNC, and WiMob. Also, she is actively reviewing articles from many other top conferences and journals.

Dr. Israat is an active IEEE and ACM member.

**Chadi Assi** (SM'12) received the B.Eng. degree from the Lebanese University, Beirut, Lebanon, in 1997 and the Ph.D. degree from the City University of New York (CUNY), Albany, NY, USA, in April 2003.

He is currently a full professor with the Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada. Before joining Concordia University in August 2003 as an assistant professor, he was a visiting researcher with Nokia Research Center, Boston, MA, USA, where he worked on quality of service in passive optical access networks. His research interests are in the areas of networks and network design and optimization. He received the prestigious Mina Rees Dissertation Award from CUNY in August 2002 for his research on wavelength-division multiplexing optical networks. He is on the Editorial Board of IEEE Communications Surveys and Tutorials, IEEE Transactions on Communications, and IEEE Transactions on Vehicular Technologies. His current research interests are in the areas of network design and optimization, network modeling and network reliability.