Wireless Indoor Positioning System with Enhanced Nearest Neighbors in Signal Space Algorithm

Quang Tran, Juki Wirawan Tantra, Chuan Heng Foh, Ah-Hwee Tan, Kin Choong Yow
School of Computer Engineering
Nanyang Technological University
Singapore

Dongyu Qiu
Concordia University
Canada

***Abstract*— With the rapid development and wide deployment of wireless Local Area Networks (WLANs), WLAN-based positioning system employing signal-strength-based technique has become an attractive solution for location estimation in indoor environment. In recent years, a number of such systems has been presented, and most of the systems use the common Nearest Neighbor in Signal Space (NNSS) algorithm. In this paper, we propose an enhancement to the NNSS algorithm. We analyze the enhancement to show its effectiveness. The performance of the enhanced NNSS algorithm is evaluated with different values of the parameters. Based on the performance evaluation and analysis, we recommend some guidelines on optimizing the parameters of our proposed enhanced algorithm.**

I. Introduction

The fantastic growth of mobile computing has fostered a strong interest in context-aware, especially location-aware services. Location-aware services have great potential in many areas such as navigation, entertainment, disaster response and security. These services make use of the knowledge about the position of the clients to give more intelligent and relevant responses or decisions. They can also be regarded as a filter for the huge amount of information available to the client. In the last couple of years, there is a number of research on radio received-signal-strength-based (RSS-based) indoor positioning systems proposed [1], [2], [3] to complement the Global Positioning System (GPS) in the indoor environment. The RSS-based technique may be implemented in an existing WLAN to achieve positioning. This WLAN-based positioning system employing RSS-based technique is an attractive solution for positioning since it is relatively simple and cheap to deploy compared to other techniques such as angle-of-arrival (AOA) or time-of-arrival (TOA).

Although the WLAN-based positioning system has been investigated in the past [4]-[9], most of the studies implement the common Nearest Neighbor in Signal Space (NNSS) algorithm to achieve location estimation. While NNSS algorithm yields promising results, it does not account for the variation of the received signal strength (RSS) from individual access points (AP) of a WLAN which may affect the accuracy of the estimation. Youssef *et al.* [8] and Xiang *et al.* [9] attempted to improve NNSS by taking the probabilistic approach which makes use of the signal probability distribution to calculate the probability that a mobile station (MS) is at a certain location. However, the main disadvantage of the probabilistic approach is the requirement of a large training set, which implies costly data collection. The main motivation of our work is to improve the accuracy of the NNSS algorithm while keeping the traditional data collection method.

In this paper, we introduce an enhancement to the NNSS algorithm. Our enhancement is based on the idea of finding the most probable location given the observation sequences of signal strength values ([8]) of the probabilistic approach. However, it is able to work with positioning systems that use the NNSS algorithm without additional data collection.

The remaining of the paper is organized as follows. Section II revisits the WLAN-based positioning system. Section III describes our enhancement for the NNSS algorithm. The effectiveness of the enhancement is analyzed in Section IV. In Section V, we evaluate the performance of the WLAN-based positioning system implementing the enhanced NNSS algorithm and provide several guidelines on optimizing the parameters of the algorithm. Some important conclusions are drawn in Section VI.

II. WLAN-Based Positioning Systems

In an infrastructure WLAN, APs are deployed in the network allowing MSs to access the WLAN. The WLAN-based positioning system uses the signal strength emitted from APs to estimate the location of a MS. As the signal attenuation in an indoor environment is very dependent to the physical characteristic of the surrounding, the WLAN-based positioning system often requires the survey of signal strength in the environment, known as the *off-line (data collection) phase* before the operation. During this off-line phase, the RSS statistics in the area are collected. The collected RSS and the related geographical information form a database to facilitate the *tracking phase* for the real-time positioning estimation.

During the tracking phase, a MS measures its RSS and estimates its current location based on its RSS and those collected RSS stored in the database. The WLAN-based positioning system usually employs the NNSS algorithm for the location estimation. In the following subsections, we briefly describe the off-line and tracking phase of an WLAN-based positioning system that uses NNSS algorithm.

A. *Off-line Phase*

During the off-line phase, a radio map is constructed by measuring the RSS at certain locations on the two dimensional floor plan. Commonly, a virtual rectangular grid is defined and measurements are taken at each of the grid intersections. Assuming that there are totally N APs visible throughout the

IN IEEE VTC 2006

floor plan, at intersection j^{th}, one can obtain the RSS vector for j as $\mathcal{M}_j =< \mu_{j,1}, \mu_{j,2}, \ldots, \mu_{j,N} >$, where $\mu_{j,i}$ is the RSS of AP_i.

Each record in the database includes a mapping of the grid intersection's coordinates (x_j, y_j) and the RSS vector $\mathcal{M}_j$. Each element in $\mathcal{M}_j$ is assumed to be the *mean of the RSS* from each of the N APs. This assumption can be achieved by collecting a large number of RSS samples for each orientation of the MS as suggested in [4] and [5].

B. Tracking Phase

During the tracking phase, the NNSS algorithm is used to determine the location of the MS. The RSS of all visible APs are measured at the MS to produce a vector, denoted as $\mathcal{R} =< r_1, r_2, \ldots, r_N >$, where r_i is the RSS of AP_i. By comparing $\mathcal{R}$ with $\mathcal{M}_j$ for each j, it is possible to identify a certain j such that the difference between $\mathcal{R}$ and $\mathcal{M}_j$ is the smallest compared to that of the other j values. If we only consider the simplest one nearest neighbor option of the NNSS algorithm, $\mathcal{M}_j$ is the nearest from $\mathcal{R}$, which also indicates that the two locations has the lowest *error distance* [1] , and hence one may conclude that MS is located at (x_j, y_j).

In NNSS algorithm, the difference between two vectors, say $\mathcal{R}$ and $\mathcal{M}_j$, is measured by the RSS distance between the two vectors. Let D_j be the RSS distance between $\mathcal{R}$ and $\mathcal{M}_j$, then D_j can be computed as

$$D_j = \sqrt{\sum_{i=1}^{N} (r_i - \mu_{j,i})^2} \tag{1}$$

The quantity D_j is a positive real value, where a lower value indicates a smaller difference between the two compared vectors. Using the resulting D_j, the NNSS algorithm ranks the list of $\mathcal{M}_j$ in ascending order. The location of the MS can then be estimated by triangulating based on the corresponding locations from the first few entries from the ranked list.

III. Enhancements to Neighbors Selection Procedure

We observe from (1) that the RSS distance, D_j, merely puts together RSS values from all APs without considering the variation of RSS from individual AP. It is indicated in [8], [9] that by considering the RSS distribution at each grid intersection, the accuracy of the location estimation will be improved. Instead of collecting the mean RSS from each AP at each grid intersection, the probabilistic approach records details of RSS distribution. Given $\mathcal{R}$, the joint probability distribution is computed for each j which then results in a probability. By ranking the probabilities for all j, nearest neighbors can be identified and hence the location of MS can be estimated.

Although the probabilistic approach reports an improved location estimation compared to that of the common NNSS algorithm, the probabilistic approach requires more costly data collection since details of RSS distribution must be recorded instead of the mean RSS value. The accuracy of this approach also depends on the recorded RSS distributions. Obtaining acceptable RSS distributions may result in a lengthy data collection process and therefore, hinder the deployment of such positioning systems. Motivated by the probabilistic approach, we propose an enhancement to the NNSS algorithm to improve its accuracy while keeping data collection phase unchanged.

[1]The actual Euclidean distance between the actual (physical) location of the MS and the estimated location [4].

TABLE I
Enhancement to NNSS Algorithm

Input: sorted neighbor list of size M, where $M \geq K$
begin
for m = 1 to K
    begin
        $count\ =\ 0$
        for i = 1 to N
        begin
            if ($|r_i - \mu_{m,i}| > \theta$)
                $count = count + 1$
        endfor
        if ($count \geq \tau$)
            remove neighbor m from neighbor list
    endfor
if (less than K entries in neighbor list)
    Return old list
else
    Return updated list
end

Our proposed enhanced NNSS algorithm takes two parameters θ and τ. The parameter θ (in dBm) acts as a threshold for the variation of the RSS around its mean. The parameter τ is an integer constant, which describes the maximum allowable number of APs from which the RSS varies beyond the threshold θ. Using these two thresholds, the enhanced algorithm further evaluates the accuracy of each record in the ranked list ascending order returned by the NNSS algorithm. Since the NNSS algorithm uses (1) which only evaluates the overall distance, it is possible that the NNSS draws a wrong conclusion due to high variation of the RSS for certain APs. Our enhancement compares the RSS of the MS and each of the RSS in the ranked list for each AP. If the difference between the two RSS values is greater than θ, the corresponding location of the record may not be the nearest neighbor. Our enhanced NNSS algorithm attempts to identify this inaccuracy of NNSS and to fix it by removing the neighbor wrongly identified as a nearest neighbor by NNSS from the ranked list. The details of the enhancement procedure is given in Table I. Note that the input of the algorithm is a neighbor list sorted according to (1) and the enhancement algorithm is performed on the first K entries of this list.

IV. Performance Analysis

Suppose that we have a WLAN-based positioning system deployed on a single floor plan inside a building. During the data collection phase, we define a virtual square grid over the floor plan. At each grid intersection, the RSS of visible APs are measured and their averages are recorded along with the physical position of the intersection. In the tracking phase, the RSS of each AP is measured at the MS and the real-time RSS vector $\mathcal{R} =< r_1, r_2, \ldots, r_N >$ is used to estimate the location of the MS. Each component in $\mathcal{R}$ can be considered as a random variable with the following assumptions:

- The random variables r_i (in dBm) are mutually independent, where $i = 1, 2, \ldots, N$.
- The random variables r_i (in dBm) are normally distributed.
- The standard deviation of all the random variables r_i is assumed to be identical throughout the area and denoted by σ (in dBm)

The assumption that r_i is normally distributed is supported by the *Log Distance Path Loss Model* [7] and the measurement results reported in [10]. The assumption that random variables r_i are independent is reasonable since there is no clear relationship between the signals transmitted by different APs. The last assumption is validated in [5] where the measurement of the RSS over a long period in an office room indicated that the RSS has a standard deviation σ of 2.13 dBm.

We shall now examine the performance of the enhancement algorithm when the NNSS algorithm return correct and incorrect physically nearest neighbors. For simplicity and mathematical tractability, we follow the assumption described in [5] that there are only two records $\mathcal{A}$ and $\mathcal{B}$ in the database and any estimation of the MS's location is limited to one of these two records.

We assume that the MS is physically located nearer to $\mathcal{A}$. In this case, the correct nearest neighbor of the MS is $\mathcal{A}$. However, due to the variation of RSS measured by the MS in the real-time, it is possible that the NNSS algorithm incorrectly identifies $\mathcal{B}$ as the nearest neighbor [5]. When this error occurs, we evaluate the probability that our enhancement identifies this incorrectness of the NNSS algorithm and corrects the error, which shows the effectiveness of our enhancement.

On the other hand, when the NNSS algorithm suggests a correct nearest neighbor, it is possible that our enhancement attempted to fix the already correct nearest neighbor which then eventually results in error. This incorrect estimation may occur when our enhancement is operated under incorrect setting of thresholds. The study of the probability of this incorrect estimation is presented in the next subsection.

A. Probability of Incorrect Estimation

Assuming that the NNSS algorithm returns $\mathcal{A}$ as the nearest neighbor. Define P_E to be the probability that our proposed enhanced NNSS algorithm removes $\mathcal{A}$ from the neighbors list resulting an error. Let $\overline{r_i}$ denote the mean value of random variable r_i. As in [5], the distribution of r_i can be described by a normal distribution $\mathcal{N}(\overline{r_i}, \sigma^2)$. Define P_{EI} to be the probability that r_i varies beyond the threshold θ, it is the probability that $|r_i - \mu_{\mathcal{A},i}|$ is larger than the value θ. This can be expressed as $1 - P(|r_i - \mu_{\mathcal{A},i}| < \theta)$, where $P(|r_i - \mu_{\mathcal{A},i}| < \theta)$ is the area under $\mathcal{N}(\overline{r_i}, \sigma^2)$ in the interval $(\mu_{A,i} + \theta, \mu_{A,i} - \theta)$. Hence P_{EI} can be expressed as

$$P_{EI} = P(|r_i - \mu_{\mathcal{A},i}| > \theta) = 1 - \int_{\mu_{A,i}-\theta}^{\mu_{A,i}+\theta} \mathcal{N}(\overline{r_i}, \sigma^2)(x)dx$$

Our algorithm stated that the neighbor record will be removed from the ranked neighbor list if more than $(\tau - 1)$ that $|r_i - \mu_{A,i}| > \theta$ is true for all i, where $i = 1, 2, ..., N$. Assuming that P_{EI} is independent of i, the probability that our enhanced NNSS algorithm results in incorrect estimation, denoted P_E, can then be determined by

$$\begin{aligned} P_E = P(&\text{More than } (\tau - 1) \text{ of } r_i \text{ where } i = 1, 2, ..., N \\ &\text{that satisfy } |r_i - \mu_{A,i}| > \theta) \\ = &\sum_{i=\tau}^{N} \binom{N}{i} (P_{EI})^i (1 - P_{EI})^{N-i}. \end{aligned} \quad (2)$$

B. Effectiveness of the Enhancement

We consider the case when the NNSS algorithm returns $\mathcal{B}$ as the nearest neighbor. We calculate the probability P_S that our enhanced NNSS algorithm corrects this inaccuracy by removing $\mathcal{B}$ from the ranked neighbors list.

Similar to the previous analysis, define P_{SI} to be the probability that r_i varies beyond the threshold θ for $\mathcal{B}$, we have

$$P_{SI} = P(|r_i - \mu_{\mathcal{B},i}| > \theta) = 1 - \int_{\mu_{B,i}-\theta}^{\mu_{B,i}+\theta} \mathcal{N}(\overline{r_i}, \sigma^2)(x)dx$$

Again, assuming that P_{SI} is independent of i, then the probability P_S can be evaluated as

$$\begin{aligned} P_S = P(&\text{More than } (\tau - 1) \text{ of } r_i \text{ where } i = 1, 2, ..., N \\ &\text{that satisfy } |r_i - \mu_{B,i}| > \theta) \\ = &\sum_{i=\tau}^{N} \binom{N}{i} (P_{SI})^i (1 - P_{SI})^{N-i} \end{aligned} \quad (3)$$

For the enhanced NNSS algorithm to be effective, we need a low probability of P_E while having a high probability of P_S. This can be accomplished by selecting a suitable combination of θ or τ. Some guidelines for choosing the value of θ and τ will be discussed in Section V.

V. Performance Evaluation

For performance evaluation purposes, we simulate an indoor positioning system as shown in Fig. 1. The grid spacing is set at 3 meters and we assume that three APs, located at (2,0), (0,3) and (3,4), are visible throughout the entire area. The Log-Distance Pathloss model is employed to calculate the pathloss at a certain distance d from the AP based on the free space pathloss at reference distance d_0 = 1 (meter) given as follows

$$Pathloss(d) = Pathloss(d_0) + 10\alpha log\left(\frac{d}{d_0}\right) + X_\sigma \quad (4)$$

where α is the pathloss exponent and X_σ represents a Gaussian random variable with zero mean and variance σ^2. The pathloss exponent $\alpha = 5$ and standard deviation $\sigma = 2.13$ dBm is used for the indoor environment as recommended in [7] and [5].

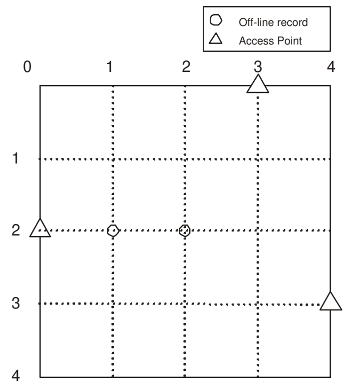

Fig. 1. Simulation scenario for performance evaluation of WLAN-based positioning system.

TABLE II
INCORRECT ESTIMATION OF NNSS ALGORITHM WITH 100000 SAMPLES

MS's Location	#Return Incorrect Neighbor	Percentage
(2,1.0)	7	0.007
(2,1.1)	190	0.190
(2,1.2)	2514	2.514
(2,1.3)	14291	14.291
(2,1.4)	41089	41.089

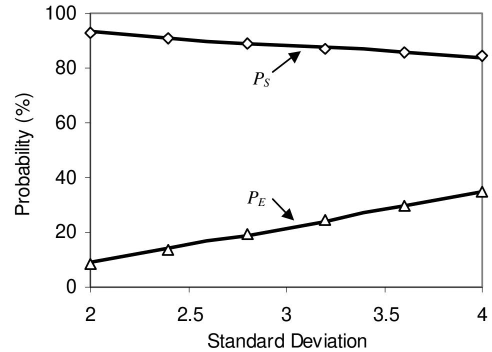

Fig. 2. P_E and P_S versus σ.

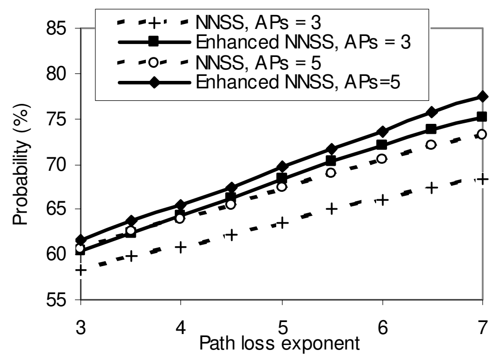

Fig. 3. Impact of the pathloss exponent on the effectiveness.

The RSS can be determined by subtracting the pathloss from the transmit power $Power_{transmit}$ of the AP, which is fixed at 15 dBm, as

$$RSS(d) = Power_{transmit} - Pathloss(d). \tag{5}$$

In our considered scenario, we assume that there are only two off-line records corresponding to intersections (2,1) and (2,2) and the MS moves on the segment along the two points. From Table II, we see that the NNSS algorithm has higher probability to return incorrect neighbor when the MS is nearer to the central of the line. Hence, in our simulation, we choose the MS's location at (2,1.4) so that the performance of the enhanced NNSS algorithm can be observed clearer.

A. *Effectiveness of the Enhanced NNSS Algorithm*

We first study the accuracy of our analysis. Fig. 2 plots the analytical and simulation results for the considered scenario. In our simulation experiment, 100000 random RSS samples are generated at the MS's location based on (4) and (5) for the study. The immediate result can be seen from the figure is that an excellent agreement is reached between the analytical results (shown in lines) and the simulation ones (shown in symbols). We further notice that in this scenario, we indeed achieve a low probability of P_E and a high probability of P_S, which indicates the effectiveness of the enhancement.

In Fig. 3, we study the effect of the pathloss exponent on the performance improvement of our enhanced NNSS algorithm with two different number of APs settings. As can be seen, the enhanced NNSS algorithm always provide higher probability of correct nearest neighbor estimation. Our improvement is even greater when the number of APs is small, this is because NNSS is less effective when the number of APs is small.

In a WLAN deployed in an office environment, the typical number of APs visible at a certain location is usually small due to the consideration of network cost. Hence, it is important that the location estimation algorithm can be maintained at a certain high accuracy in a WLAN environment with only a few overlapped APs installed. In Fig. 4, we present the results for the probability that a correct neighbor is returned by the NNSS and our enhanced NNSS algorithms with a range of the number of APs. The results show that the enhancement improves the probability that the nearest neighbor is correctly identified for all the studied numbers of APs. For a smaller number of APs, the improvement of our enhancement is high as the NNSS algorithm results in a relatively high number of inaccurate nearest neighbor detection. For a higher number of APs, even when the NNSS algorithm is relatively more effective, our enhancement still achieves improvement.

We next consider the impact of the standard deviation σ on the enhanced NNSS algorithm. The value σ varies between 2 and 4 (dBm), and the effectiveness of the NNSS and the enhanced NNSS algorithms is compared in Fig. 5. The standard deviation of the RSS of the MS is mainly influenced by the surrounding setup as well as the RSS sampling. Due to the real-time requirement for tracking, the MS usually takes only a few RSS samples, and hence the standard variation of the RSS will be high. As indicated in [5], a long period sampling

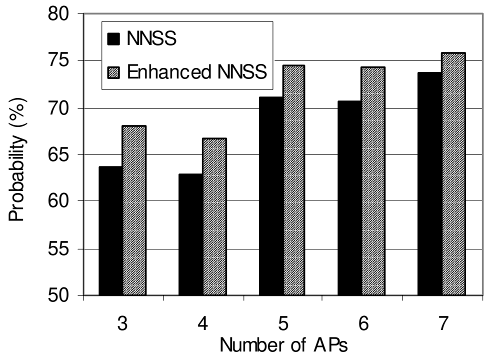

Fig. 4. Impact of the number of access points on the effectiveness.

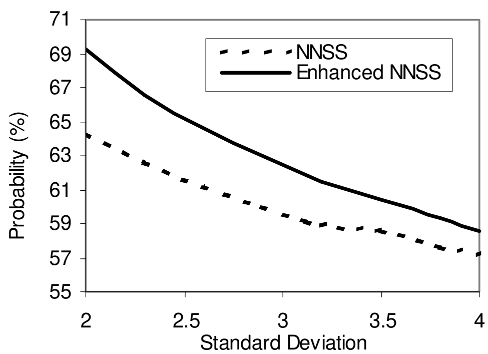

Fig. 5. Impact of the standard variation of RSS on the effectiveness.

of RSS, say over a few hours, gives $\sigma = 2.13$ dBm. For a short period sampling of RSS, a higher standard deviation will be expected. Here we study our enhanced NNSS algorithm for σ over the range between between 2 and 4 (dBm), and we see that our enhancement achieves higher accuracy.

B. Parameter Selection

In this subsection, we study the effect of the parameters of the enhanced NNSS algorithm on the improvement. In Fig. 6, the improvement of the enhanced NNSS algorithm over the NNSS algorithm with some combinations of θ and τ are depicted. We observe that an incorrect selection of τ and θ may reduce the benefit of our enhanced algorithm. From the results, we found that $\tau = 2$ is a better choice as it allows the enhanced algorithm to achieve a better performance for some θ in the range of between 3 dBm and 5 dBm. Table III summarized some typical selections for the two parameters recommended for operation.

TABLE III
RECOMMENDED VALUES FOR THE ENHANCED NNSS ALGORITHM PARAMETERS

Parameters	Suggested Range
τ	$\lceil \frac{N}{2} \rceil \leq \tau \leq N$
θ	$3 \leq \theta \leq 4$

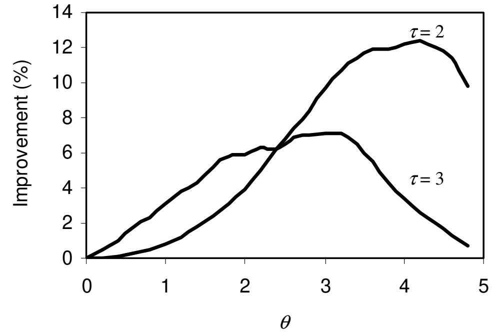

Fig. 6. Impact of θ and τ on Improvement.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed an enhancement to the NNSS algorithm to improve the accuracy of WLAN-based positioning systems. Our enhancement is based on the idea of the probabilistic approach but does not require additional data collection to work with systems employing the NNSS algorithm.

We analyzed the effectiveness of the enhanced NNSS algorithm with result showing that our enhanced NNSS algorithm indeed corrects errors generated by the NNSS algorithm. Further simulation experiments are also conducted to show the performance advantages of our proposed enhancement in the considered scenario. We also evaluated the performance for some parameter selections and recommended typical settings for our enhancement. In our ongoing work, we will try to minimize the effect of large standard deviations on the accuracy of the positioning system and validate our analysis with real measurements.

REFERENCES

[1] P. Prasithsangaree, P. Krishnamurthy, and P. K. Chrysanthis, "On Indoor Position Location With Wireless LANs," in *Proc. IEEE PIMRC*, Lisbon, Portugal, Sep. 2002.
[2] S. Saha, K. Chaudhuri, D. Sanghi, P. Bhagwat, "Location Determination of a Mobile Device Using IEEE 802.11b Access Point Signals," in *Proc. IEEE WCNC*, New Orleans, Louisiana, USA, Mar. 2003.
[3] Moustafa Youssef, and Ashok K. Agrawala, "The Horus WLAN Location Determination System," in Proc. MobiSys, Seattle, WA, USA, Jun. 2005.
[4] P. Bahl and V. N. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system," in *Proc. IEEE Infocom*, Tel-Aviv, Israel, Mar. 2000.
[5] K. Kaemarungsi and P. Krishnamurthy, "Modeling of indoor positioning systems based on location fingerprinting," in *Proc. IEEE Infocom*, Hong Kong, China, Mar. 2004.
[6] The Cricket Indoor Location System. [Online]. Available: http://nms.csail.mit.edu/projects/cricket/
[7] T. S. Rappaport, *Wireless Communications, Principles and Practice.* Prentice Hall, 2002.
[8] M. Youssef, A. Agrawala, A. U. Shankar, and S. H. Noh, "A probabilistic clustering-based indoor location determination system," University of Maryland, College Park, Tech. Rep. CS-TR 4350, 2002.
[9] Z. Xiang, S. Song, J. Chen, H. Wang, J. Huang, and X. Gao, "A wireless LAN based indoor positioning technology," *IBM Journal of Research and Development*, vol. 48, no. 5/6, 2004.
[10] J. Small, A. Smailagic, and D. P. Siewiorek, Determining User Location For Context Aware Computing Through the Use of a Wireless LAN Infrastructure. [Online] Available: http://www-2.cs.cmu.edu/∼aura/docdir/small00.pdf