APPLICATION OF LINEAR DISCRIMINANT FUNCTIONS TO LOCATION ESTIMATION

Galo Nuño-Barrau, José M. Páez-Borrallo

Centro de Domótica Integral.Universidad Politécnica de Madrid
galo@gaps.ssr.upm.es, jmpaez@etsit.upm.es

ABSTRACT

Location estimation is a recent interesting research area that exploits the possibilities of modern communication technology. In this paper we present a new location algorithm for wireless networks that is especially suitable for user-based systems, as it improves both the speed and the memory requirements. The algorithm is based on the application of Linear Discriminant Functions to location estimation and its performance has been compared with previous systems presented in the literature. Results show a very good performance, reducing the computing time and memory space and displaying an adequate behavior under conditions of few calibration points per location.

1. INTRODUCTION

Location estimation or location-aware computing is a recent interesting research area that exploits the possibilities of modern indoor communication technologies. Location estimation has a great potential in areas such as architecture, data-mining, security or tourism. The most obvious location-based service is the one answering questions like "Where is the Main Hall?" but much more complex services can be implemented, such as network security based on the physical location of the users, emergency services, or smart buildings that automatically turn off the lights when a worker goes home.

The possibility of implementing this kind of systems using an existing technology is very attractive. They are value-added services that should not imply any additional hardware once the technology has been deployed, so no initial investment is necessary. Additionally, a specifically designed location technology implementation would be more expensive, not only because of the hardware cost but also because of the design, deployment and integration with other systems.

Most of the traditional positioning methods cannot be used for this location estimation: GPS systems [1] have problems working indoors, infrared [2], ultrasonic [3] or computer vision -based systems [4] are not widespread and wide area cellular networks do not provide enough accuracy [5]. However, the increasing use of wireless networking, such as 802.11-based wireless systems, has gained attention recently as a potential technology for low cost indoor location services.

Indoor wireless LAN (WLAN) positioning systems should employ one of the different physical attributes of the medium for estimation. The typical features that might be used are: the received signal strength (RSS) of communication, the angle of arrival (AOA) of the signal and the time difference of arrival (TDOA). Among these, RSS is the only parameter that is measurable with reasonably priced current commercial hardware. Previous work [6, 7, 8, 9, 10, 11] has shown the feasibility of location estimation WLAN systems based on RSS measures.

Though in most of the applications commented above the network should know the user position, it does not mean that these applications should necessarily be network-based. As commented in [7], privacy is a great concern in a location system, and most users want to have the control to choose whether their location is transmitted to the network or not, so a terminal-oriented system is maybe more suitable. Scalability is another advantage of terminal-based systems; the more independent the location process is from the network, the more scalable the system is.

An additional reason for terminal-based systems is the variation of the transmitted power between different terminals. If the system is network-based, calibration is based on the power received at the Access Points (APs) from a calibrating terminal. However, when the system is on duty, mobile terminals (MTs) can be different to the ones that performed the calibration (for example different laptops, or laptops and cell phones) and therefore, the RSS from a MT can be different to the one received during the calibration phase from the same place with a different terminal, degrading the system performance. Furthermore, depending on the battery level, the transmitted power for a single device can vary, introducing a similar negative effect on the performance. A terminal-based system does not have this problem, as the APs do not change and are usually connected to the electrical network, so the RSS from the APs at a given position should be more stable in this case.

The design of a user-based RSS location system should take into account three additional considerations:

- Computation time. As the algorithm should run in the core of the MT, it should try not to reduce the processor performance. Therefore, algorithm speed is an important constraint to the feasibility of the system and it is also related to the battery life.
- Calibration. In order to work correctly, location systems should have been previously calibrated. As manual calibration reduces the flexibility of the system (because every time a change in the environment happens, it is necessary to recalibrate it), it is desirable to find an algorithm that can work properly with a reduced number of calibration samples, which makes the recalibration process easier and faster. It could even make possible to substitute manual calibration by a Ray-Tracing technique [12].
- Table size. When a mobile user connects to the network, it receives the *calibration information table* (CIT), i.e. the initial set of data that will allow the location estimation. These data have been gathered at the calibration phase and pre-processed at the network depending on the location algorithm. The calibration information should be transmitted through the wireless link and stored at the memory of the MT. Therefore, the greater the table is, the greater the transmission overhead and the memory occupation.

The aim of this paper is to present a new RSS location estimation algorithm that improves the location performance in a user-based system. This algorithm is based on the application of Linear Discriminant Functions (LDF) to the positioning problem and we have designed a software simulator where we have tested our algorithm against previous systems for different environments.

The structure of the paper is as follows: in Section 2 we present the general model of the system and we introduce the location algorithm that we are comparing with. In Section 3 we introduce our algorithm based on LDF. Finally, in Sections 4 and 5 we show some numerical results and comment the main conclusions.

2. PROPOSED APPROACH

2.1. Problem description

We consider a floor in a typical office building, as the one presented in Figure 1. The total surface can go from 500 to 2000 m^2. Employees can work either in open-air desks or in separate rooms. The average surface of a worker's *vital space* ranges from 6 to 10 m^2. Assuming a 20-40% of *common space*, i.e. space share by all employees (like corridors, stairs, elevators, bathrooms, etc), there would be a potential number of positions c from 30 to 300. Each position described by a 2-D position vector **l** (or 3-D if location estimation is possible in different floors).

We also consider that a WLAN network has been deployed, with d receivers. We will use the existing WLAN infrastructure for our location estimation services. User terminals can be laptop or desktop computers, PDA or even UMTS/ Wi-Fi dual cell phones. The location service is very simple; each user should be able to continuously know his/her position vector **l**, which shows the office/desk where he is located. To ensure privacy, location estimation should be performed by the terminals. Both the terminals and the network should have the location software previously installed. Every time a terminal connects to the network, it receives the CIT. Terminals store that information, and use it to locate themselves by analyzing the RSS from the Wi-Fi antennas.

The calibration information is obtained in the calibration phase when m calibration samples are taken by position. The calibration phase can be real, with measurements taken in different points, or simulated with a ray tracing model of the building floor. Calibration should be repeated whenever a major change happens in the floor distribution.

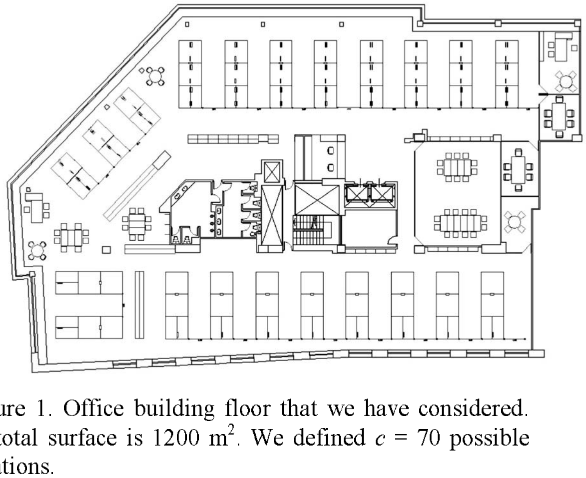
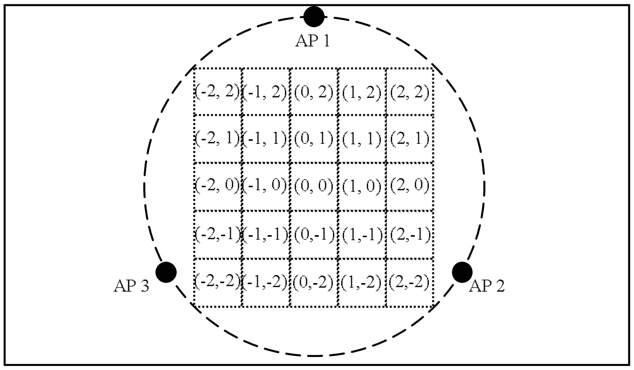


Figure 1. Office building floor that we have considered. Its total surface is 1200 m^2. We defined c = 70 possible locations.



(-2, 2)	(-1, 2)	(0, 2)	(1, 2)	(2, 2)
(-2, 1)	(-1, 1)	(0, 1)	(1, 1)	(2, 1)
(-2, 0)	(-1, 0)	(0, 0)	(1, 0)	(2, 0)
(-2,-1)	(-1,-1)	(0,-1)	(1,-1)	(2,-1)
(-2,-2)	(-1,-2)	(0,-2)	(1,-2)	(2,-2)

Figure 2. General building model for c = 25, d =3. Each square corresponds to a vital space of 9 m^2

2.2. System model

In order to compare our algorithm with the previous ones, we have designed a software model that simulates different environments. The model builds a square floor with c positions, surrounded by a circumference where the APs are equally distributed as it shown in Figure 2. Each position corresponds to a position vector, and it denotes a vital space of 9 m^2. We consider an error has occurred when we locate a user who is at position i as if he were at j. We cannot provide further accuracy inside a vital space.

We consider that the receiver averages the received samples to reduce the noise impact. The received signal power is simulated according to the models presented in [13, 14, 15]. The large-scale path loss of the locations is given by a log-normal model whereas the difference between power samples at a given location is considered to follow a Rayleigh distribution.

We can therefore simulate different buildings characterized by these parameters: number of rooms c, number of APs d, number of calibrations per room m, building complexity σ_N (log-normal standard deviation) and σ_R (Rayleigh standard deviation).

2.3. Location algorithms

We compare four different algorithms. First we have a *k-Nearest Neighbor* algorithm, as RADAR, presented in [6]. In this algorithm, the location of the MT is estimated according to the position of the nearest calibration point in the RSS space.

We also compare a statistical location algorithm, as the ones presented in [8, 9, 10]. The algorithm computes the joint probability distribution of receiving a given set of RSS points at the APs in the calibration phase, and it applies the Bayes' Theorem to find the location that maximize the likelihood of the received power signal. In order to reduce the calibration data, we have taken the assumption that the readings from the APs are statistically independent, as it is proposed in [10] and so the problem of estimating the joint *probability distribution function* (*pdf*) becomes the problem of estimating the marginal ones. As *pdfs* should usually be discretizated, Bayesian methods are also called histogram methods.

Finally, we present two new algorithms, based on the application of LDF to location estimation. They are explained in the following section.

3. APPLICATION OF LINEAR DISCRIMINANT FUNCTIONS TO LOCATIONESTIMATION

Every time a user terminal performs a measure, it obtains a RSS vector **x**

$$\mathbf{x}^T = \left(x_1, x_2, .., x_d, 1\right) \qquad (1)$$

where x_k is the RSS from antenna k. Additionally, a RSS vector obtained during the calibration phase is defined as $\mathbf{y}_i$

$$\mathbf{y}_i^T = \left(y_1, y_2, .., y_d, 1\right) \qquad (2)$$

and its position denoted as position i, i=1..c .

Location estimation can be therefore defined as obtaining the position i that corresponds to a received RSS vector **x**. It is possible to train the system to map the RSS space in c decision region, each decision corresponding to a position i. Consequently, once a RSS vector **x** is received, it is directly assigned to a physical location depending on its decision region. This decision is taken through the discriminant functions $g_i(\mathbf{x})$, so we assign **x** to an estimated position i if

$$g_i(\mathbf{x}) > g_j(\mathbf{x}) \quad j = 1..c \quad j \neq i \qquad (3)$$

An interesting particular case happens when $g_i(\mathbf{x})$ are linear (LDF)

$$g_i(\mathbf{x}) = \mathbf{a}_i^T \mathbf{x} \qquad (4)$$

and therefore the decision rule is reduced to find the maximum of c vector products. We cannot assure that the LDF are optimal for all the possible environments, but in our problem, it is usually worthy to sacrifice some performance to gain simplicity.

Minimum Square Error (MSE) procedures can be employed to calculate LDF when the calibration samples show a non separable behavior [16]. We seek a weight vector $\mathbf{a}^T_i$ that is the minimum-square-error solution to the equations

$$\left.\begin{aligned} \mathbf{a}_i^T \mathbf{y}_i &= -\lambda_{ii} \\ \mathbf{a}_i^T \mathbf{y}_j &= -\lambda_{ij} \quad j \neq i \end{aligned}\right\} \qquad (5)$$

where λ_{ij} is the loss incurred when deciding i when the true position is j. This loss should be more important as the physical distance between i and j is greater. We define n as the number of training samples $\mathbf{y}$, d' as $d+1$ and we let $\mathbf{Y}$ be the n-by-d' matrix of training samples, which we assume to be partitioned as

$$\mathbf{Y}=\begin{bmatrix}\mathbf{Y}_1\\ \mathbf{Y}_2\\ \vdots\\ \mathbf{Y}_c\end{bmatrix} \tag{6}$$

with the samples from position i comprising the rows of $\mathbf{Y}_i$. Similarly, let $\mathbf{A}$ be the d'-by-c matrix of weight vectors

$$\mathbf{A}=\begin{bmatrix}\mathbf{a}_1 & \mathbf{a}_2 & \ldots & \mathbf{a}_c\end{bmatrix} \tag{7}$$

and let $\mathbf{B}$ be the n-by-c matrix

$$\mathbf{B}=\begin{bmatrix}\mathbf{B}_1\\ \mathbf{B}_2\\ \vdots\\ \mathbf{B}_c\end{bmatrix} \tag{8}$$

where

$$\mathbf{B}_j=-\left.\overbrace{\begin{bmatrix}\lambda_{1j} & \lambda_{2j} & \cdots & \lambda_{cj}\\ \lambda_{1j} & \lambda_{2j} & \cdots & \lambda_{cj}\\ \vdots & & & \vdots\\ \lambda_{1j} & \lambda_{2j} & \cdots & \lambda_{cj}\end{bmatrix}}^{c}\right\}m \tag{9}$$

Therefore, Equation (5) can be expressed as

$$\mathbf{YA}=\mathbf{B} \tag{10}$$

and if we compute matrix $\mathbf{A}$ to minimize the square-error-vector

$$\|\mathbf{e}\|^2=\mathbf{e}^T\mathbf{e}=\left(\mathbf{YA}-\mathbf{B}\right)^T\times\left(\mathbf{YA}-\mathbf{B}\right) \tag{11}$$

then $\mathbf{A}$ results

$$\mathbf{A}=\left(\mathbf{Y}^t\mathbf{Y}\right)^{-1}\mathbf{Y}^T\mathbf{B}=\mathbf{Y}^{\dagger}\mathbf{B} \tag{12}$$

and consequently $\mathbf{A}$ is a MSE solution to (10).

It is important to notice that, as the number of samples approaches infinity, the solution (12) yields discriminant functions $g_i(\mathbf{x})$ that provide a minimum-mean-square-error approximation to the Bayes discriminant function

$$g_i(\mathbf{x})=-\sum_{j=1}^{c}\lambda_{ij}P(i\,|\,\mathbf{x}) \tag{13}$$

Equation (12) can be calculated directly or by a gradient descent procedure. The second approach has two advantages over merely computing the *pseudoinverse*: (i) it avoids the problems that arise when $\mathbf{Y}^T\mathbf{Y}$ is singular, and (ii) it avoids the need for working with large matrices. There are different gradient descent procedures suitable for a non separable behavior, such as the *LMS* rule. The problem of the LMS rule is that, although it converges whether the calibration samples are separable or not, there is no guarantee that the resulting LDF are separating functions in a separable case. To avoid this problem, we can use the *Ho-Kashyap* procedure, which works both in the separable and non separable cases [16].

The *Ho-Kashyap* is an iterative procedure where both $\mathbf{A}$ and $\mathbf{B}$ are estimated. We first initialize $\mathbf{B}_0$ with the values commented above and every step s, the calculations are

$$\begin{aligned}\mathbf{A}_s&=\mathbf{Y}^{\dagger}\mathbf{B}_s\\ \mathbf{e}_s&=\mathbf{YA}_s-\mathbf{B}_s\\ \mathbf{B}_{s+1}&=\mathbf{B}_s+\eta(s)\left(\mathbf{e}_s+Abs\left[\mathbf{e}_s\right]\right)\end{aligned} \tag{14}$$

where $\eta(s)$ is a positive scale factor or *learning rate* that sets the step size. Abs[.] is the *positive part* function.

Consequently, the use of LDF greatly simplifies the location estimation problem. Bandwidth efficiency is guaranteed by sending $\mathbf{A}$ as CIT, a d'-by-c matrix, instead of $\mathbf{Y}$, a n-by-d' as in previous methods, with $n > c$. Computation is optimized by substituting the search in the probability distribution table as in [10] (or directly in $\mathbf{Y}$ in [6]) by c products $\mathbf{a}^T_i\mathbf{x}$, especially for high dimensionality environments.

4. NUMERICAL RESULTS

We have simulated several environments in order to test the behavior of the LMS and Ho-Kashyap algorithms in comparison with a k-Nearest Neighbor (with k = 1) and a Bayesian one (with 4 *containers* in each marginal histogram). The following results are presented assuming 49 location positions, 4 APs, σ_R = -8dB, σ_N = 3dB, 2000 iterations of the Ho-Kashyap algorithm and 30 building simulations per point. The learning rate of the Ho-Kashyap algorithm is set to

$$\eta(s)=1/s \tag{15}$$

where s denotes the iteration number. Loss coefficients are set to

$$\left.\begin{aligned}\lambda_{ii}&=1\\ \lambda_{ij}&=0\quad j\neq i\end{aligned}\right\} \tag{16}$$

so now Eq. (13) becomes

$$g_i(\mathbf{x})=P(i\,|\,\mathbf{x}) \tag{17}$$

which is the *Bayes Discriminant Function*.

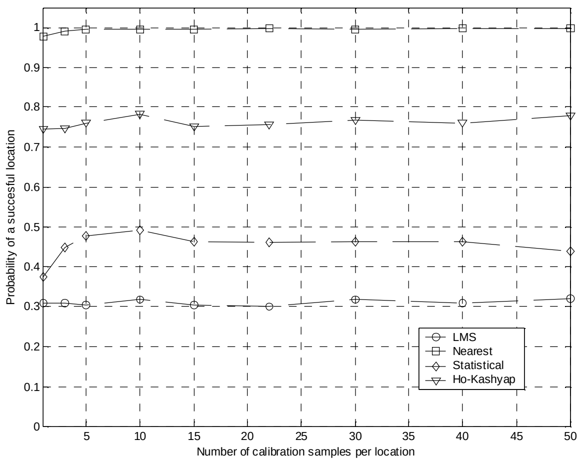


Figure 3. Probability of a successful location as a function of the number of calibration samples per location.

As we mentioned in section 2, we employ a novel approach where, instead of mapping the physical space with a fixed grid, we consider it to be constructed by the aggregation of different vital spaces, each of them with an average surface of 9 m^2, which yields a location uncertainty of around 3 m if the location estimation is successful. The accuracy parameter changes from the error distance to the *probability of a successful location*, defined as the probability of correctly estimating a MT at position *i*. This probability is calculated as the division between the number of successful estimations and the total number of samples.

In Figure 3 it can be seen how the performance increases with the number of calibration samples per location *m*, especially for the Bayesian algorithm. It should be noticed how the *k*-Nearest Neighbor and the Ho-Kashyap algorithm display the best results, even for an extremely low number of calibration samples.

In Figure 4 it is shown the computation time related to the time of the Ho-Kashyap algorithm with one calibration sample per location. It should be noticed how computation time grows linearly with *m* in *k*-Nearest Neighbor and how it is independent of the number of calibration samples for Bayesian and MSE algorithms. Notwithstanding, Bayesian computational times are more than 20 times greater than MSE ones. Consequently, MSE algorithms (LMS and Ho-Kashyap) show a superior time performance than the other algorithms, which confirms the assumption that the LDF algorithms are faster than the previous ones.

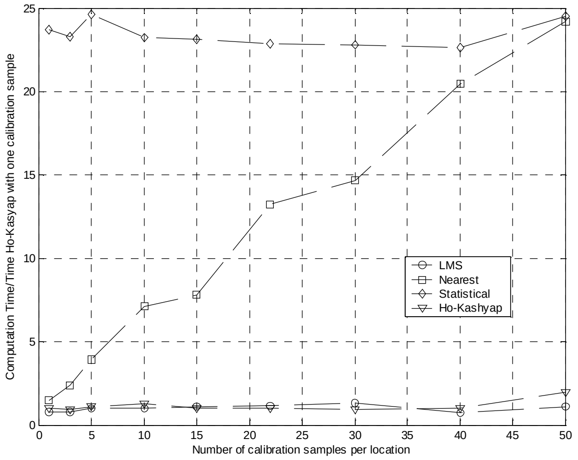


Figure 4. Computation time related to the time of the Ho-Kashyap algorithm with *m* =1 as a function of the number of calibration samples per location.

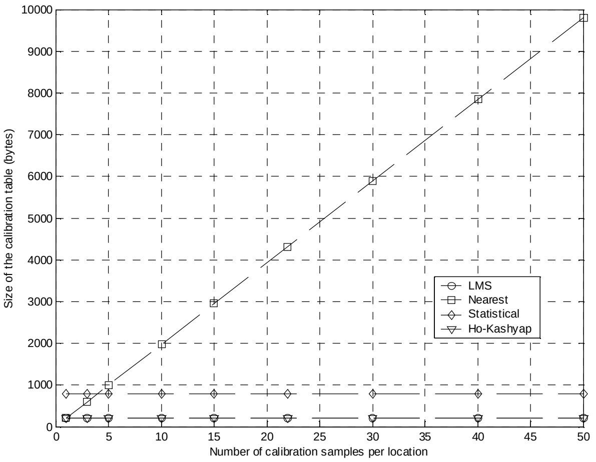


Figure 5. Size of the calibration information as a function of the number of calibration samples per location

Finally, CIT size is shown in Figure 5. It grows linearly with *m* in *k*-Nearest Neighbor algorithms as they send all the calibration samples as CIT and it is independent of *m* for the other algorithms. Once again, MSE performance is by far superior, due to the employ of LDF, so bandwidth efficiency is guaranteed. MSE methods are therefore more suitable to implement location estimation systems in terms of time and overhead, and among them, the Ho-Kashyap one shows a better location performance than the LMS.

5. CONCLUSIONS

In this paper we have presented a new location estimation system based on LDF, and we have compared it with previous algorithms presented in the literature. We can conclude that the LDF algorithms display a better performance under the considerations mentioned in Section 1, i.e. computational time, CIT size and number of calibration samples.

Accuracy is not considered to be crucial as this level because it is possible to implement a two-layer architecture based on a 1st fast and simple level as the one described here and a 2nd robust level, such as a Hidden Markov Model (HMM) [17], which allows to exploit the trade-off between computation time and accuracy. The main purpose of this paper is therefore to remark the importance of fast and simple location algorithms in terminal-based location systems.

Another contribution is the concept of a mapping based on vital spaces instead of on grids, which is more suitable for pattern recognition problems and allows the system administrator to define the position as he/she wishes, emphasizing the most important areas for the users.

In conclusion we have presented a novel approach to location estimation based on previous work. Further research is focused on the integration between LDF algorithms and HMM in WLAN positioning problems.

5. REFERENCES

[1] P. Enge and P. Misra, “Special Issue on GPS: The Global Positioning System,” *Proceedings of the IEEE*, vol. 87, no. 1, pp. 3-15, Jan. 1999.

[2] R. Want, A. Hopper, V. Falco and J. Gibbons, “The Active Badge Location System,” *ACM Transactions on Information Systems*, vol. 10, no. 1, pp. 91-102, 1992.

[3] N.B. Priyantha, A. Chakraborty and H. Balakhishnan, “The Cricket Location-Support System,” *6th ACM MOBICOM*, Boston, 2000.

[4] J. Krumm et al., “Multi-camera Multi-person Tracking for Easy Living,” *3rd IEEE International Workshop on Visual Surveillance*, Piscataway, pp. 3-10, Jul. 2000.

[5] S. Tekinay, “Special Issue on Wireless Geolocation Systems and Services,” *IEEE Communications Magazine*, 1998.

[6] P. Bahl, and V.N. Padmanabhan, “Radar: An In-building RF-Based User Location and Tracking System,” *Proc. IEEE INFOCOM 2000*, vol. 2, pp. 775-784, Tel Aviv, Israel, Mar. 2000.

[7] A. Smailagic and, F. Kogan, “Location Sensing and Privacy in a Context Aware Computing Environment,” *IEEE Wireless Communications*, vol, 9, no. 5, pp. 10-17, Oct. 2002.

[8] T. Roos, P. Myllymäki and H. Tirri, “A Statistical Modeling Approach to Location Estimation,” *IEEE Transactions on Mobile Computing*, vol. 1, no. 1, pp.59-69, Jan.-Mar. 2002.

[9] A. Schwaighofer, M. Grigoras, V. Tresp and C. Hoffmann, “GPPS: A Gaussian Process Positioning System for Cellular Networks,” *Proc. NIPS 2003*, Vancouver, Canada, Dec. 2003.

[10] M. Youssef, A. Agrawala and A.U. Shankar, “WLAN Location Determination Via Clustering and Probability Distributions,” *Proc. IEEE PERCOM 2003*, pp. 143-150, For Worth, Mar. 2003.

[11] P- Krishnan, A.S. Krishnakumar, W. Ju, C. Mallows, and S. Ganu, ”A System for LEASE: Location Estimation Assisted by Stationary Emitters for Indoor RF Wireless Networks,” *Proc. IEEE INFOCOM 2004,* vol. 2, pp. 1001-1011, Hong Kong, Mar. 2004.

[12] P. Wertz, G. Wolfle, R. Hoppe and F.M. Landstorfer, “Deterministic Propagation Models for Radio Transmission into Buildings and Enclosed Spaces,” *33rd European Microwave Conference*, Munich, vol. 3, pp. 1147-1150, Munich, Germany, Oct. 2003.

[13] T.S. Rappaport, *Wireless Communications. Principles & Practice*, Prentice Hall, New Jersey, 1996.

[14] J.B. Andersen, T.S. Rappaport, and S. Yoshida, “Propagation Measurements and Models for Wireless Communications Channels,” *IEEE Comm. Magazine*, vol. 33, no. 1, pp. 42-49, Jan. 1995.

[15] S. Y. Seidel and T. S. Rappaport, “914 MHz Path Loss Prediction Models for Indoor Wireless Communications in Multifloored Buildings,” *IEEE Trans. On Antennas and Propagation*, vol. 40, no. 2, pp. 207-217, Feb. 1992.

[16] R. O. Duda, P. E. Hart and D. G Stork, *Pattern Classification (2nd Edition)*, Wiley-Interscience, 2000.

[17] L. R. Rabiner, “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition,” *Proc. IEEE,* vol. 77, no. 2, pp. 257-286. Feb. 1989.