

A New Location Estimation System for Wireless Networks Based on Linear Discriminant Functions and Hidden Markov Models

Galo Nuño-Barrau¹ and José M. Páez-Borrillo²

¹Fundación Rafael Escalá, Universidad Politécnica de Madrid, 28040, Spain

²Centro de Domótica Integral, Universidad Politécnica de Madrid, 28040, Spain

Received 26 May 2005; Revised 14 November 2005; Accepted 8 December 2005

Location estimation is a recent interesting research area that exploits the possibilities of modern communication technology. In this paper, we present a new location system for wireless networks that is especially suitable for indoor terminal-based architectures, as it improves both the speed and the memory requirements. The algorithm is based on the application of linear discriminant functions and Markovian models and its performance has been compared with other systems presented in the literature. Simulation results show a very good performance in reducing the computing time and memory space and displaying an adequate behavior under conditions of few a priori calibration points per position.

Copyright © 2006 Hindawi Publishing Corporation. All rights reserved.

1. INTRODUCTION

Context-aware computing applications examine and react to a user's changing context in order to help promoting and mediating people's interaction with each other and their environments [1, 2]. But, what is context? In [3], it is defined as "the set of environmental states and settings that either determines an application's behavior or in which an application event occurs and is interesting to the user" and it is divided into four categories:

- (i) *computing context*, such as network connectivity and nearby resources (printers, displays, etc.);
- (ii) *user context*, such as the user's profile, location, or people nearby;
- (iii) *physical context*, such as lighting, temperature, or traffic conditions;
- (iv) *time context*, such as time of a day, week, or season of the year.

Location estimation or positioning is therefore essential information for context-aware or ubiquitous computing systems, as it can provide a lot of valuable context information. Positioning 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 an employee goes home.

There are two basic approaches for this kind of systems. The first approach is to develop a signalling system and a network infrastructure of location sensors focused primarily on positioning applications. The second approach is to use an existing wireless network infrastructure to locate the mobile terminals (MT). The advantage of the first approach is that physical specification, and consequently the quality of the location estimation results, is under control of the designer, so a high accuracy can be achieved. The advantage of the second approach is that it avoids expensive and time-consuming deployment of infrastructure: location is a value-added service that should not imply any additional hardware once the communication technology has been deployed, so no initial investment is necessary. Both approaches have their own markets but we will focus on the second one as a way to provide context-aware computing capabilities to existing wireless communication systems.

There are different promising wireless LAN (WLAN) or wireless PAN (WPAN) communication technologies to support location estimation applications such as *Bluetooth*, *Wi-Fi*, *Zigbee*, *Wi-Max*, or even *Ultra Wideband*. However, due to the commercial boom of Wi-Fi systems, we will consider the IEEE 802.11-based WLAN systems. Nevertheless, results can be easily extended to other wireless network technologies.

TABLE 1: Classification of location systems.

		Element that senses the RSS	
		Network	MT
Element that performs the location estimation	Network	Network-based	Terminal-assisted
	MT	Network-assisted	Terminal-based

Indoor WLAN positioning systems should employ at least one of the available 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 them, RSS is the only parameter that is measurable with reasonably priced currently existing commercial hardware. Previous work [4–16] has shown the feasibility of location estimation WLAN systems based on RSS measurements.

In this paper, we present a new algorithm for location estimation with WLAN systems. We first discuss the proposed system architecture and problem formulation to obtain the design parameters. Then we introduce the linear discriminant functions (LDFs) and hidden Markov models (HMMs) to develop an algorithm that improves the location performance compared to the already existing ones. In order to test our algorithm against previous systems for different environments, we have designed a software model that simulates the main system parameters.

This paper is organized as follows: in Section 2, we present the system architecture and the location stack and discuss the main characteristics of indoor location estimation systems. In Section 3, we describe the location problem in the specific environment, with an emphasis on the channel model. In Section 4, we comment previous work in RSS location estimation, and in Section 5, we present a new positioning method based on LDF and HMM. Numerical results are provided under different sets of parameters in Section 6. Finally, we present in Section 7 the main conclusions about the algorithm and further research proposals.

2. LOCATION SYSTEM ARCHITECTURE AND CHARACTERISTICS

2.1. Location system classification

Location systems can be classified according to how the location estimation process is distributed between the MT and the rest of the system components. First, the RSS can be observed by either the MT or the network access points (APs); second, the estimation can be performed by the element that senses the RSS or by another. Consequently, there are four basic configurations shown in Table 1.

In a *terminal-based* architecture, the MT estimates its position without any *uplink* communication. Nevertheless, the network can *broadcast* some data, such as calibration

information. This architecture presents two very important features: privacy and scalability, which will be commented below. If the MT needs to communicate with the network to receive the RSS information, it would be a *network assisted* architecture, and scalability would be lost. In a *network-based* system, the APs obtain the RSS and the network performs the location estimation, whereas in a *terminal assisted* system, the RSS is obtained by the MT, which sends it to the network for the estimation process.

If the network senses the RSS, two situations could arise: one is the *hearability* problem, that is, if the MT, in order to have the minimum power consumption, adjusts its signal strength to reach only the closest AP, the signal might not be received by other APs. The second problem is the *performance asymmetry*; APs are usually connected to a permanent power source and therefore their transmitted power levels are roughly constant. However, RSS coming from the MTs show more variability, as a consequence of the use of batteries and the heterogeneity among devices and manufacturers.

Additionally, terminal-based estimation offers two advantages already mentioned: it makes the system easily scalable, as the network does not perform the estimation process, and it provides users with total privacy about their positions. Privacy is a great concern in a location system, and most users ask for the control to decide whether their location is transmitted to the network or not [16, 17].

Some authors have presented network-based or assisted systems because they prefer to sacrifice some privacy and scalability to improve performance (such as the LEASE system in [18]) or because privacy is not a problem at all (as it happens in [19]). For all the reasons exposed above, and specially to ensure privacy and scalability, we have decided our architecture to be a terminal-based one.

2.2. The location stack

Intel *PlaceLab* project has presented a proposal for the stack of protocols in a location-aware computing paradigm, similar in spirit to the seven-layers *open system interconnect* (OSI) model in computer networks [20, 21]. This proposal is known as the location stack and is represented in Figure 1.

The location estimation algorithm presented in this paper should be placed at layers 2 (measurements) and 3 (fusion). Layer 2 imports the raw RSS values from the WLAN card (layer 1) and it exports estimated position, an integer from 1 to c (number of possible positions). Layer 3 imports these data and exports a more refined location estimation (related to a coordinate system) and more complex information such as derivatives (speed, acceleration), positional histories, and even user identification.

We are therefore splitting our problem into two separate ones:

- (1) *positioning*: obtaining an initial estimation from the RSS data;
- (2) *tracking*: refining the estimation and building the MT's trajectory.

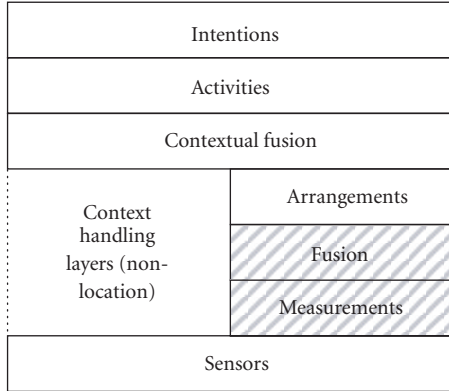


FIGURE 1: The seven-layer location stack for location-aware computing systems.

2.3. Location estimation system characteristics

Once the system architecture has been established, we should analyze how this affects its design parameters and characteristics [6, 22]. Here we briefly discuss some of them.

Granularity

The calibration points are usually collected on a grid of key-positions within the building. The spacing between grid crossings influences the granularity of the position estimate. If grid spacing is too small, RSS from adjacent points is similar, so they cannot be distinguished; if it is too large, it drastically reduces accuracy. Usual and practical grid spacing for offices ranges from 1 to 3 meters.

Accuracy

Accuracy can be measured by two parameters: the *average error distance* and the *success probability*. In this paper, we will focus on the latter, as commented in Section 6.

Fault tolerance

The system should be able to keep on operation even if some APs are disabled.

Computation time

As the location algorithms should run in the core of MTs, processor performance should not be drastically reduced. System load is therefore an important constraint to its feasibility and it is also related to the battery life.

Calibration

In order to work properly, location systems need to be previously calibrated. As manual calibration reduces the flexibility of the system (because every time a change in the

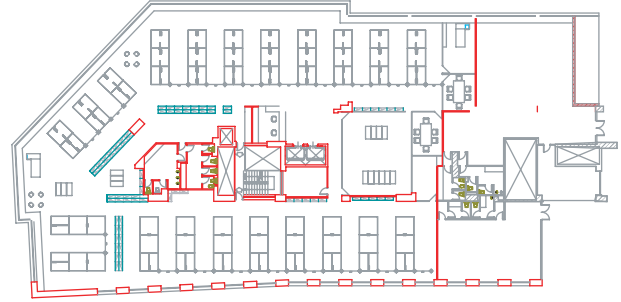


FIGURE 2: Office building floor that we have considered. Its total surface is 1200 m². We defined $c = 70$ possible locations.

environment happens, a recalibration is needed), it is desirable to find a location algorithm that can work well with a small number of calibration samples, to make the recalibration process easier and faster. It could even make possible to substitute on-site real calibration by any suitable ray-tracing technique, such as [23].

Table size

When a mobile user connects to the network, it receives the *calibration information table* (CIT), that is, the initial set of data that allow the estimation of positions in the grid. These data have been gathered in the calibration phase and pre-processed by the network according to the location algorithm, before being broadcasted to the MT. The CIT should be transmitted through the wireless link and stored at the memory of the device. Therefore, the greater the table is, the greater the transmission overhead and the memory occupation.

These are the main design parameters that determine the performance of our location algorithm: it should be fast, fault tolerant, and with acceptable error probability. Besides, it should require a small number of calibration samples and a small CIT to reduce the transmission overhead.

3. GENERAL MODEL OF THE SYSTEM

3.1. Problem description

We consider a floor in a typical office building as the one presented in Figure 2. The total surface can go from 500 to 2000 m². Employees can work either in cubicles or in separate rooms. The average surface of a worker's vital space ranges from 5 to 10 m². Assuming a 30–60% of common space, that is, space shared by all employees (like corridors, stairs, elevators, bathrooms, etc.), there would be a potential number of positions c from 20 to 280. Each position will be described by a 2-D position vector (or 3-D if location estimation is possible in different floors).

We also consider that a WLAN network has been deployed, with d APs. 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 cell phones. The location service is very simple; each user should be able to continuously have knowledge of his/her position i , which shows the office/desk where he/she is located. To ensure privacy, location estimation should be terminal based. Both the terminals and the network should have previously installed the location software. Every time a terminal connects to the network, it receives the calibration information. Terminals store that information and use it to locate themselves by analyzing the RSS from the surrounding Wi-Fi antennas.

The calibration information is obtained in the calibration phase when m calibration samples are taken per position and are stored in the CIT \mathbf{Y} . The calibration phase can be real, with measurements taken in different positions, or simulated with a ray-tracing model of the floor. Calibration should be repeated whenever a major change happens in the floor distribution.

Every time an MT performs a measurement, it obtains an RSS vector \mathbf{x} ,

$$\mathbf{x}^T = [x_1, x_2, \dots, x_d], \quad (1)$$

where x_k is the RSS from the k th AP. Additionally, RSS vectors obtained during the calibration phase are defined as \mathbf{y} ,

$$\mathbf{y}^T = [y_1, y_2, \dots, y_d], \quad (2)$$

and their position denoted as position i , $i = 1, \dots, c$. We define n as the number of training samples \mathbf{y} and let \mathbf{Y} be the n -by- d matrix of training samples, which we assume to be partitioned as

$$\mathbf{Y} = \begin{bmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_2 \\ \vdots \\ \mathbf{Y}_c \end{bmatrix}, \quad (3)$$

with the samples from position i comprising the rows of \mathbf{Y}_i . Location estimation can be therefore defined as obtaining the position i that corresponds to a received RSS vector \mathbf{x} .

In order to compare our algorithm with the previous ones, we have implemented 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 is shown in Figure 3. Each position corresponds to a position vector and it denotes a vital space of 9 m^2 . We consider that 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.

Our approach based on vital spaces is different to the usual grid-oriented one. Vital spaces are related with the physical configuration of the environment and should be defined when the software is installed. Vital spaces therefore allow a higher accuracy in the most important areas for the system administrator, but they require more human interaction than grids, which can be fully automatized.

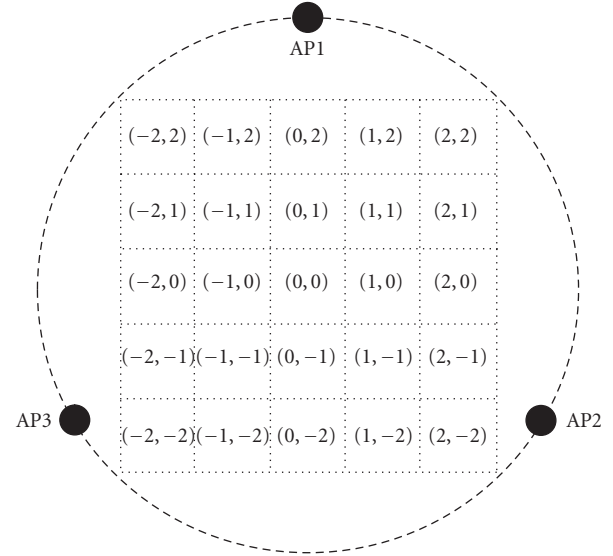


FIGURE 3: General building model for $c = 25$, $d = 3$. Each square corresponds to a vital space of 9 m^2 .

3.2. Properties of indoor RSS in WLAN system

Indoor signal propagation is difficult to predict due to the strong multipath and propagation effects such as reflection, diffraction, and scattering [6]. Multipath attenuation makes the signal fluctuate over its mean value for a given position. Received signal is usually modeled as the combination of the *large-scale* and *small-scale* fading effects [24]. Large-scale fading, that models the attenuation effects due to walls and furniture and predicts the RSS average value depending on the position, is widely accepted to follow a log-normal distribution [24, 25]. Small-scale fading reflects the signal fluctuations due to multipath attenuation; it is usually modeled as a Rician distribution if there is a line-of-sight path (LOS) and as a Rayleigh if there is no line-of-sight path (NLOS). Despite the fact that there are several small-scale fading models such as [26–28], they are mainly focused on describing signal properties from a communication perspective and they do not properly describe the RSS properties. The research carried out in [29, 30] is the most exhaustive one we have found about RSS properties.

User's orientation

Because the resonance frequency of water is at 2.4 GHz and the human body consists of 70% water, the RSS is absorbed when the user's obstructs the signal path and causes an extra attenuation. Already mentioned in [4], this effect can be a very significant source of distortions [29].

Large-scale fading

Although the signal mean value can usually be modeled as stated above, there are some conflicting results. The measurement of the large-scale fading distributions shown in

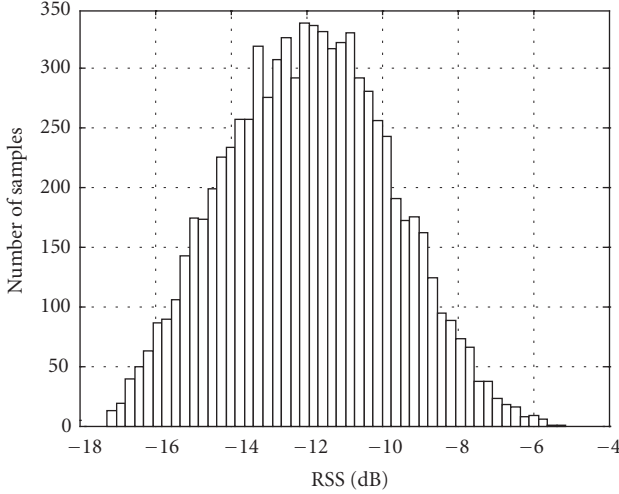


FIGURE 4: Histogram of the simulated RSS fluctuations for position $(0, 0)$, $\sigma_R = -17$ dB, $\sigma_N = 5$ dB, 8000 samples, $c = 49$, $d = 1$.

[26, 31] follow asymmetric distributions that do not fit the traditional log-normal. Additionally, their standard deviations seem to decrease with the distance between the MT and the AP.

Overlapping

RSS from two positions are grouped in different clusters. In [29], it is suggested that only two APs are sufficient to distinguish between locations for a system with small number of positions and coarse location granularity. Increasing the number of APs is one way to further separate two-location clusters.

Stationarity and independence

RSS from multiple APs can be considered uncorrelated. Stationarity can be assumed for small time scales.

Following these assumptions, our simulator models the RSS as the combination of two distributions: the mean value of the RSS between different locations is given by a log-normal of parameter σ_N , and the difference between power samples at a given location is considered to follow a Rayleigh distribution of parameter σ_R , as shown in Figure 4. We also consider that the receiver averages the received samples to reduce the impact of noise and distortion.

4. PREVIOUS WORK

4.1. k -nearest neighbor methods

In the last years, a number of different algorithms have been proposed to solve the RSS location estimation problem. One of the most important is the k -nearest neighbor (KNN) algorithm [4–6], which is based on estimating the position i depending on the average (in physical space) of the coordinates of the k closest calibration points to the received RSS vector

\mathbf{x} (in RSS space). The generalized vector distance $d(\mathbf{x}, \mathbf{y}^i)$ can be defined as

$$d(\mathbf{x}, \mathbf{y}^i) = \frac{1}{d} \left(\sum_{k=1}^d \frac{1}{w_k} |x_k - y_k^i|^p \right)^{1/p}, \quad (4)$$

where $p=2$ denotes the *euclidean* distance and $p = 1$ the *Manhattan* one. The weight w_k can be used to bias the distance by a factor that indicates how reliable the calibration sample \mathbf{y}^i is, but the improvement is not very significant [6].

The algorithm main problem is the size of the CIT, which also makes the system slower due to the search times. One possible solution is to average the calibration points from every given position, thus reducing the CIT size.

In [7] a different KNN algorithm is proposed, denoted *Weighted k -nearest neighbors*, where once we have found the k -closest calibration points, the average of coordinates is weighted by the distance in the RSS space,

$$\mathbf{l}_i = \frac{\sum_{j=1}^k (1/d(\mathbf{x}, \mathbf{y}^j) + d_0) \mathbf{l}_j}{\sum_{j=1}^k (1/d(\mathbf{x}, \mathbf{y}^j) + d_0)}, \quad (5)$$

where \mathbf{l}_j are the physical coordinates of position j (with calibration vectors \mathbf{y}^j) and d_0 is a small real constant to avoid division by zero. Traditional KNN is a special case of (5) without using distance-dependent weights.

Results show that WKNN achieves low estimation error, the size of the CIT and the computation time being their main drawbacks [7].

4.2. Bayesian decision methods

Bayesian decision algorithms employ the Bayes theorem to estimate the position [8–12]. Position i is calculated as

$$i = \arg \max_i P(i | \mathbf{x}) = \arg \max_i \frac{P(\mathbf{x} | i)P(i)}{P(\mathbf{x})}, \quad (6)$$

where $P(\mathbf{x} | i)$ is the probability of receiving a sample from position i and $P(i)$ is the probability of an MT being at this position, which initially can be considered as uniform in the location area. $P(\mathbf{x} | i)$ is calculated from the CIT \mathbf{Y} . Therefore, the location estimation problem becomes a standard maximization problem,

$$i = \arg \max_i P(\mathbf{x} | i)P(i). \quad (7)$$

The main drawback of these algorithms is the large number of calibration samples necessary to construct the distribution $P(\mathbf{x} | i)$. One possible approach to reduce the number of calibration samples is *clustering*, as proposed in [10]. Another approach is assuming that the RSS signals from different APs are independent,

$$P(\mathbf{x} | i) = \prod_{k=1}^d P(\mathbf{x}_k | i), \quad (8)$$

so the problem of estimating the joint *probability distribution function* (pdf) becomes the problem of estimating the marginal ones [11].

As pdfs are usually discretized, Bayesian methods are also called histogram methods.

4.3. Kernel methods

Kernel methods are related with Bayesian ones, as they try to simplify the $P(\mathbf{x} | i)$ estimation by assuming that it is a linear combination of m pdfs

$$P(\mathbf{x} | i) = \frac{1}{m} \sum_{i=1}^m \mathbf{K}(\mathbf{x}; \mathbf{y}_i^j), \quad (9)$$

where $K(\cdot; \mathbf{y}_i^j)$ denotes the *kernel function* [9, 13]. One widely used kernel function is the Gaussian kernel

$$K(\mathbf{x}; \mathbf{y}_i^j) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\mathbf{x} - \mathbf{y}_i^j)^2}{2\sigma^2}\right), \quad (10)$$

where σ is an adjustable parameter that determines the kernel amplitude. When σ approaches zero, this method becomes a KNN.

4.4. Support vector machines methods

A very interesting approach to location estimation is to apply support vector machines (SVM) to the RSS space, increasing the number of dimensions and employing linear discriminant functions in an optimization problem, as described in [14].

Results in [14] also show that SVM methods present a performance similar to WKNN, both in time and accuracy, outperforming the other techniques (Bayesian, KNN, and neural networks).

4.5. Neural networks methods

A multilayer perceptron (MLP) can also be applied to RSS location estimation, as discussed in [15]. The transfer function for the hidden layers is the sigmoidal function

$$f(x) = \frac{1}{1 + e^{-x}}. \quad (11)$$

Results in [15] show that the MLP is the fastest algorithm and that its accuracy is only inferior to WKNN and SVM methods. The main drawback of neural networks methods is that they require a high number of calibration samples, which is very undesirable as already commented.

4.6. Triangulation or multilateralization methods

All methods commented above are known as *fingerprinting methods*, because the system tries to find the position that best “matches” the calibration information. Triangulation however works in a different way. Instead of constructing the RSS space from the calibration samples, the MT uses the RSS to estimate its distance with the AP [16]. Once the distance has been estimated, the MT applies traditional triangulation methods to estimate its position [32].

The relationship between distance and power is usually a nonlinear one in an indoor environment and it changes depending on the position. Therefore, despite that these systems are computationally light, they are not very accurate, as commented in [6].

5. A LOCATION METHOD BASED ON LDF AND HMM

5.1. Overview

As commented in Section 2, the design parameters force our system: (i) to be fast in order to reduce as less as possible the MT performance, (ii) to use small number of calibration samples to make the system flexible, and (iii) to employ small CIT to avoid transmission overheads and memory occupation. It was also commented that our system works in two-layer architecture: layer 2 (measurements) should be fast and require few calibration samples to produce initial location estimation, whereas layer 3 (fusion) should be accurate and try not to increase too much the computational time.

Methods presented in Section 4 are potential candidates to implement in layer 2, whereas layer 3 can employ HMM or *Kalman* filters, as commented in [33]. However, we proposed here a new algorithm which combines a fast and simple *Ho-Kashyap* procedure for layer 2 combined with a robust HMM in layer 3, in order to improve the system capabilities, as described below.

5.2. Layer 2: application of LDF to positioning

As commented above, location estimation can be defined as obtaining the position i that corresponds to a received RSS vector \mathbf{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 an RSS vector \mathbf{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 \mathbf{x} to an estimated position i if

$$g_i(\mathbf{x}) > g_j(\mathbf{x}) \quad \forall j = 1, \dots, c \quad j \neq i, \quad (12)$$

or equivalently

$$i = \arg \max_i g_i(\mathbf{x}). \quad (13)$$

An interesting particular case is when $g_i(x)$ are linear (LDF)

$$g_i(\mathbf{x}) = \tilde{\mathbf{a}}_i^T + a_{0i} = \mathbf{a}_i^T \mathbf{x}', \quad (14)$$

where \mathbf{x}' is defined as the $d' \times 1$ vector ($d' = d + 1$),

$$\mathbf{x}'^T = [x_1, x_2, \dots, x_d, 1]. \quad (15)$$

To simplify notation, from now on, we will denote \mathbf{x}' as \mathbf{x} .

The decision rule is therefore reduced to find the maximum of c vector products. We cannot assure that the LDFs are optimal for all the possible environments, especially if the distributions $P(\mathbf{x} | i)$ are multimodal, which is very strange in location problems. As already commented, in layer 2 it is worth sacrificing some performance to gain simplicity, so LDFs are potential candidates to implement it.

Minimum square error (MSE) procedures can be employed to calculate the LDFs when the calibration samples show a nonseparable behavior [34]. We seek a weight vector \mathbf{a}_i^T that is the MSE solution to the equations

$$\mathbf{a}_i^T \mathbf{y}^i = 1, \quad \mathbf{a}_i^T \mathbf{y}^j = 0 \quad j \neq i. \quad (16)$$

As commented in Section 3, we let \mathbf{Y} to be the n -by- d' matrix of training samples, which we assume to be partitioned as

$$\mathbf{Y} = \begin{bmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_2 \\ \vdots \\ \mathbf{Y}_c \end{bmatrix}. \quad (17)$$

Similarly, let \mathbf{A} be the d' -by- c matrix of weight vectors

$$\mathbf{A} = [\mathbf{a}_1 \quad \mathbf{a}_2 \quad \cdots \quad \mathbf{a}_c] \quad (18)$$

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

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \\ \vdots \\ \mathbf{B}_c \end{bmatrix}, \quad (19)$$

where all of the entries of \mathbf{B}_i are zero except those in the i th column, which are unity. (16) can be expressed as

$$\mathbf{YA} = \mathbf{B}, \quad (20)$$

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

$$\|\mathbf{e}\|^2 = \mathbf{e}^T \mathbf{e} = (\mathbf{YA} - \mathbf{B})^T \times (\mathbf{YA} - \mathbf{B}), \quad (21)$$

then \mathbf{A} yields

$$\mathbf{A} = (\mathbf{Y}^T \mathbf{Y})^{-1} \mathbf{Y}^T \mathbf{B} = \mathbf{Y}^\dagger \mathbf{B}, \quad (22)$$

and consequently \mathbf{A} is an MSE solution to (16).

It is important to notice that, as the number of samples approaches infinity, the solution (22) yields discriminant functions $g_i(\mathbf{x})$ that provide a minimum MSE approximation to the Bayes discriminant function

$$g_i(\mathbf{x}) = P(i | \mathbf{x}), \quad (23)$$

and the solution of (13) would be equivalent to the Bayesian one of (6).

Equation (22) 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 nonseparable 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 LDFs 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 nonseparable cases [34].

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

$$\begin{aligned} \mathbf{A}_s &= \mathbf{Y}^\dagger \mathbf{B}_s, \\ \mathbf{e}_s &= \mathbf{YA}_s - \mathbf{B}_s, \\ \mathbf{B}_{s+1} &= \mathbf{B}_s + \eta(s)(\mathbf{e}_s + \text{Abs}[\mathbf{e}_s]), \end{aligned} \quad (24)$$

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

The use of LDF greatly simplifies the location estimation problem. Bandwidth efficiency is guaranteed by sending \mathbf{A} , a d' -by- c matrix, as the CIT instead of \mathbf{Y} , an n -by- d' as in previous methods, with $n > c$. Computation time is optimized by substituting the search in the probability distribution table (in Bayesian methods) or directly in \mathbf{Y} (in KNN ones) by c products $\mathbf{a}_i^T \mathbf{x}$, especially for high dimensionality environments.

5.3. Layer 3: application of HMM to tracking

Position accuracy can be greatly improved if a series of layer 2 estimations is available unless the MT is moving with very high speed or the time interval between measurements is very long. Such a series of estimations from layer 2 allow layer 3 to keep track of the MT as a function of time and to present derivative parameters such as speed, acceleration, or user's profile. HMM, which have been successfully applied in a wide range of applications, are convenient to model the tracking problem [33]. A very good HMM tutorial can be found in [35].

An HMM is characterized by the following.

- (1) The number of states in the model, which in our problem is equal to the number of possible positions c . We denote the individual states as $L = \{l_1, l_2, \dots, l_c\}$ and the state at time t is q_t .
- (2) The number of distinct observation symbols per state, which is also c , discrete alphabet size exported from layer 2.
- (3) The state transition probability distribution $\mathbf{P} = \{p_{ij}\}$, where

$$p_{ij} = P[q_{t+1} = l_j | q_t = l_i]. \quad (25)$$

This probability can be unknown a priori, but we can infer some of its parameters. p_{ij} are zero for nonadjacent positions or for positions separated by obstacles, such as walls. The rest of the parameters should be estimated taking into consideration the user's profile and they will be updated during the session.

- (4) The observation symbol probability distribution in state j , $P(O_t | l_j)$, is the probability of receiving at time t the estimation O_t from layer 2 of position i if the previous state was l_j :

$$P(O_t | l_j) = P(O_t = i | q_t = l_j). \quad (26)$$

This distribution will be inferred in Section 6 according to the results from layer 2.

(1) Initialization:	
	$\delta_1(i) = \pi_i P(O_1 l_i)$
	$\psi(i) = 0.$
(2) Recursion $2 \leq t \leq T$:	
	$\delta_t(j) = \max_{1 \leq i \leq c} [\delta_{t-1}(i) p_{ij}] P(O_t l_j)$
	$\psi_t = \arg \max_{1 \leq i \leq c} [\delta_{t-1}(i) p_{ij}].$
(3) Termination:	
	$q_T^* = \arg \max_{1 \leq i \leq c} [\delta_T(i)].$
(4) Path backtracking (most likely trajectory):	
	$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = T-1, T-2, \dots, 1,$

where $\delta_t(j)$ is the best score (highest probability) along a single path, at time t , which accounts for the first t observations and ends in state l_j . $\psi_t(j)$ is a matrix that contains the most probable trajectory. A more detailed description of the Viterbi algorithm can be consulted in [36].

ALGORITHM 1

(5) The initial state distribution $\pi = \{\pi_i\}$, where

$$\pi_i = P[q_1 = l_i] = P(i). \quad (27)$$

Initially we can consider this distribution to be uniform in the location area, though if possible we could include information about positions that never can be the initial ones.

These parameters are updated during the session, and they constitute the user's profile that can be stored and employed in future sessions. The updating process can be found in [35].

To obtain the most likely trajectory given a sequence of T observations O_1, O_2, \dots, O_T from layer 2, we can apply the *Viterbi algorithm* [36] (Algorithm 1).

The use of HMM in layer 3 should refine the location estimation, maintaining the system time performance, as it will be analyzed in Section 6.

6. NUMERICAL RESULTS

6.1. Layer 2: algorithm comparisons

We first compare the performance of the second layer of our system against other algorithms presented in the literature. We have selected two KNN algorithms, two Bayesian ones, and another MSE method. System parameters are defined in Table 2.

The two KNN algorithms are a simple 1-KNN and a 5-WKNN, where the preceding numbers denote the number of neighbors considered. They are supposed to display the

TABLE 2: System parameters.

Parameter	Description
c	Number of positions
d	Number of APs
m	Number of calibration samples per position
i	MT position
n	Number of training samples $n = c \times m$
\mathbf{Y}	Calibration information table (CIT)
x_k	RSS from the k th AP

best accuracy but also high computational times and transmission overheads. The two Bayesian methods are based on (8), which transforms the problem of estimating the joint pdf into the problem of estimating the marginal ones. We have decided to analyze a 4-Bayesian and a 12-Bayesian, where the numbers 4 and 12 denote the number of *containers* of each marginal histogram. The MSE is an LMS rule, with an estimation phase similar to that of the Ho-Kashyap method (16), but where the \mathbf{A} matrix is constructed according to (22). As commented before, the problem of the LMS rule is that there is no guarantee that the resulting LDFs are separating functions in a separable case. Our layer 2 is based on the Ho-Kashyap method presented in (24) where the learning rate $\eta(s) = 1/s$ and the number of iterations s_{\max} is set to 2000.

As mentioned in Section 3, 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* (P_s), defined as the probability of correctly estimating an MT at position i . This probability is calculated as the ratio between the number of successful estimations and the total number of samples.

RSS samples are considered to follow a distribution as the one described in Section 3. In Figure 5, the algorithm performance as a function of the standard deviation of the log-normal large-scale component σ_N is shown. We have simulated 10 different buildings, each with 49 possible locations, 4 APs, and 10 calibration samples per location. The number of transmitted samples in order to compute the P_s is 200. Results show how the performance is enhanced with larger σ_N , as it produces a spreading over the RSS space. It is important to notice that this property holds if large-scale distortions affect in the same way the calibration and the location samples. If not, performance would be degraded as the deviation increases.

In Figure 6, the influence of the small-scale component is shown. It can be seen how the performance is degraded as the parameter σ_R increases. As it happens in Figure 5, KNN methods show the best results, followed by the Ho-Kashyap method. Bayesian and LMS algorithms display the worst performance. 1-KNN and 5-WKNN can reach a success probability of 1 for low small-scale distortions, whereas Ho-Kashyap cannot improve the 80% of successful locations.

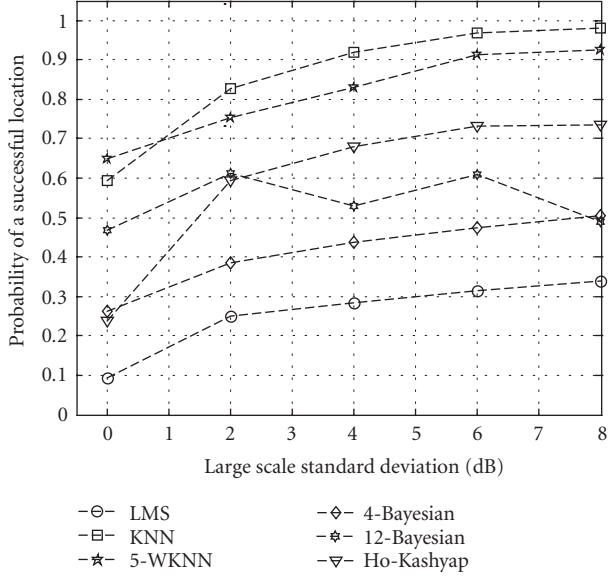


FIGURE 5: Probability of a successful location as a function of the standard deviation of the log-normal large-scale channel component. $c = 49$ locations, $d = 4$ APs, $m = 10$ calibration samples/location, $\sigma_R = -27$ dB (Rayleigh small-scale standard deviation), 200 samples/simulation, 10 simulations.

5-WKNN performs worse than 1-KNN because it sometimes takes into consideration calibration samples from locations that can be far away from the correct one, thus increasing the error probability for high small-scale distortions.

However, as it has already been mentioned, accuracy is not the main objective in layer 2. It should be fast enough and require few calibration to produce an initial location estimation that layer 3 can use to infer the right position. Following, we have analyzed the behavior of the different algorithms in terms of success probability, computational time, and transmission overhead as a function of the number of calibration samples. The first of these results is presented in Figure 7. It can be seen how the performance increases with the number of samples for all the algorithms, although calibration is more important for Bayesian and WKNN algorithms than for MSE and 1-KNN ones, which can operate without severe degradation with less than 5 samples per location.

In Figure 8, time performance is displayed, related to the computational time of the Ho-Kashyap method with $m = 1$ training sample/position. It should be noticed that time grows linearly in WKNN and KNN algorithms and that it is independent of the number of calibration samples for Bayesian and MSE ones. Nevertheless, 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, as expected.

CIT size is shown in Figure 9. It grows linearly with m in WKNN and KNN algorithms as they send all the calibration samples as CIT. It is independent of m for the other algorithms, increasing with the number of containers in the Bayesian ones (CIT is three times greater in the 12-bayesian

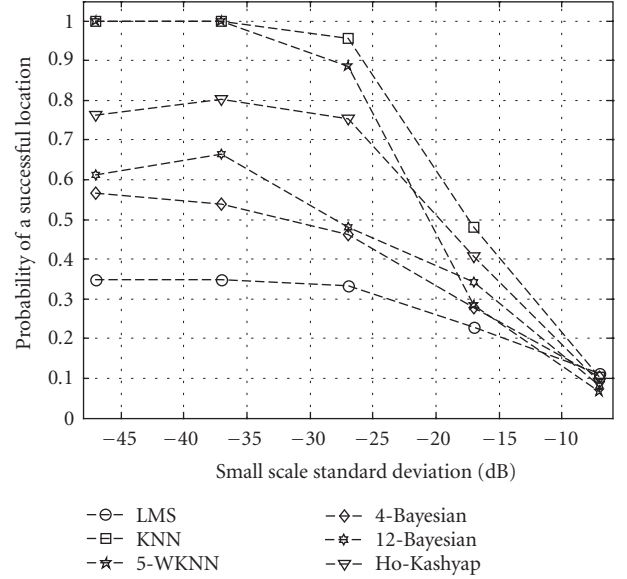


FIGURE 6: Probability of a successful location as a function of the standard deviation of the Rayleigh small-scale channel component. $c = 49$ locations, $d = 4$ APs, $m = 10$ calibration samples/location, $\sigma_N = 5$ dB (log-normal large-scale standard deviation), 200 samples/simulation, 10 simulations.

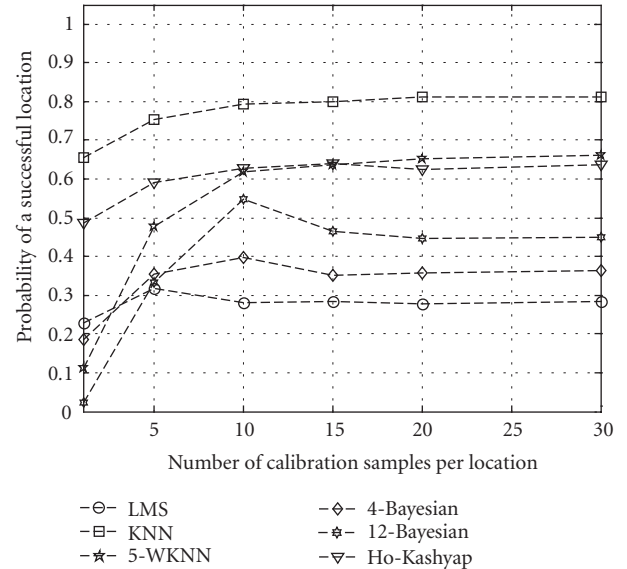


FIGURE 7: Probability of a successful location as a function of the number of calibration samples per location m . $c = 49$ locations, $d = 4$ APs, $\sigma_R = -22$ dB, $\sigma_N = 5$ dB, 200 samples/simulation, 10 simulations.

than in the 4 one). Once again, MSE performance is by far superior, due to the employ of LDF, which guarantees bandwidth efficiency. MSE methods are therefore more suitable to implement layer 2 in terms of time and overhead, and among them, the Ho-Kashyap one shows a better location performance than the LMS.

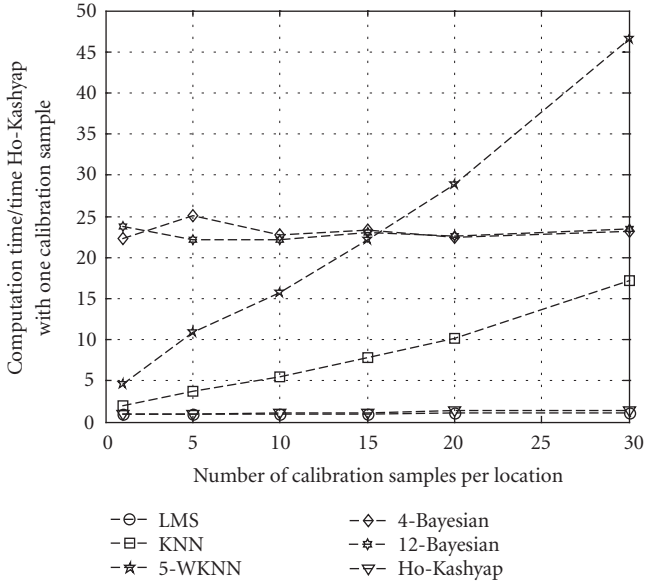


FIGURE 8: Time performance related to the time of the Ho-Kashyap with $m = 1$ as a function of the number of calibration samples per location m . $c = 49$ locations, $d = 4$ APs, $\sigma_R = -22$ dB, $\sigma_N = 5$ dB, 200 samples/simulation, 10 simulations.

It is also interesting to see how performance evolves when the physical conditions change. In Figure 10, we can see how the success probability decreases as a function of the physical location area when the number of positions c increases (as the average position area is fixed to 9 m^2). Performance reduces almost linearly with c although 1-KNN and Ho-Kashyap present smaller slopes and consequently they are less sensible to configuration changes. It is important to notice how 4 APs can theoretically manage more than 50 locations, which means that an approximate area of 500 m^2 could be covered by only 4 APs.

Another interesting result usually presented in positioning analysis is the evolution with the number of sensors (APs). It is shown in Figure 11 where it can be seen that performance improves with the number of APs, saturating when it is greater than 6–8 APs (for 49 locations). This conclusion gives us the possibility of implementing an algorithm of *smart selection* in layer 2. In this algorithm, if the number of active APs for a given MT is sufficiently high, we can discard those that show the greatest fluctuations between consecutive RSS samples in order to reduce the small-scale distortions while keeping d high enough to display a good location performance. From results in Figures 10 and 11, we propose the deployment of a *grid of APs* with a specific geometry (squares, pentagons, hexagons, etc.). This grid presents two advantages: it allows the number of APs that cover a specific area to be approximately constant, and if the number of APs is sufficiently high (e.g., hexagons for less than 49 positions), *smart selection* can be implemented, thus reducing distortions.

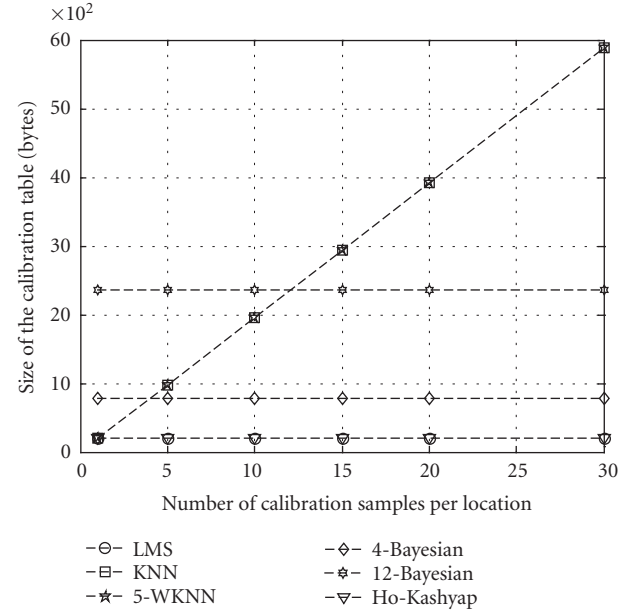


FIGURE 9: Size of the calibration in bytes as a function of the number of calibration samples per location m . $c = 49$ locations, $d = 4$ APs, $\sigma_R = -22$ dB, $\sigma_N = 5$ dB, 200 samples/simulation, 10 simulations.

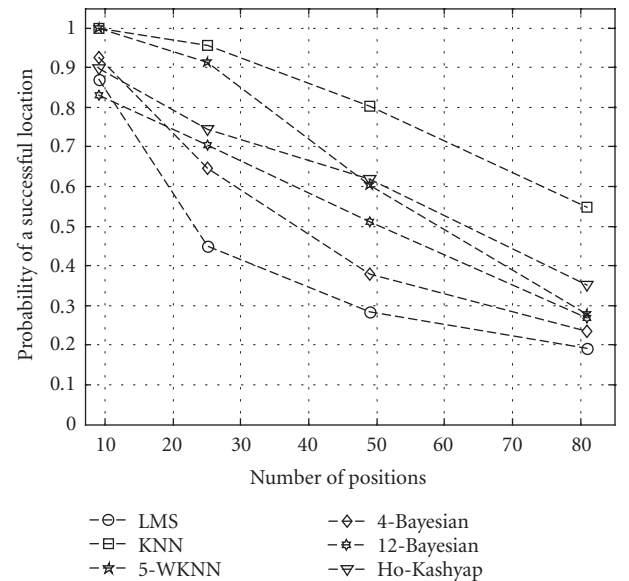


FIGURE 10: Probability of a successful location as a function of the number of possible locations c , $d = 4$ APs, $m = 10$ samples/position, $\sigma_R = -22$ dB, $\sigma_N = 5$ dB, 200 samples/simulation, 5 simulations.

It is also interesting to notice how performance decreases with a large number of APs in Bayesian algorithms, as the assumption that the RSS signals from different APs are independent does not hold when the APs are close enough.

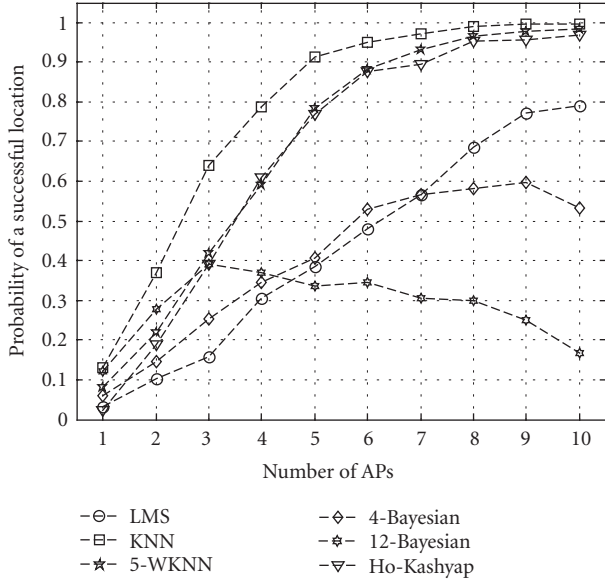


FIGURE 11: Probability of a successful location as a function of the number of APs d , $c = 49$ locations, $m = 10$ samples/position, $\sigma_R = -22$ dB, $\sigma_N = 5$ dB, 200 samples/simulation, 15 simulations.

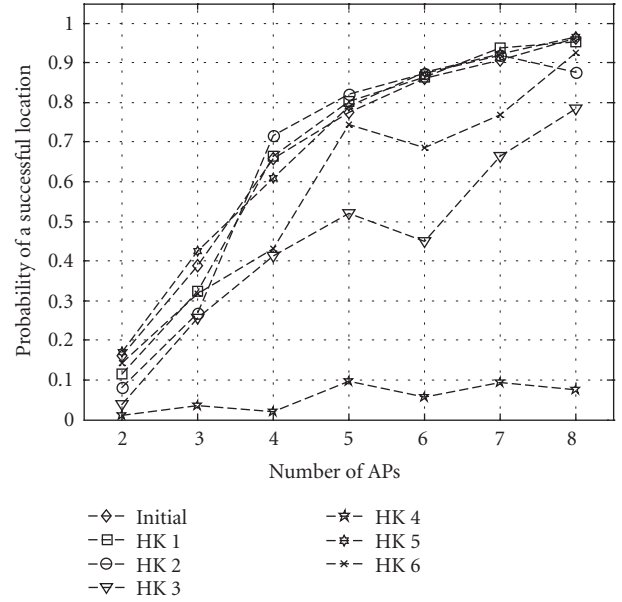


FIGURE 12: Probability of a successful location as a function of the number of APs d for the AP distributions of Table 3, $c = 49$ locations, $m = 10$ samples/position, $\sigma_R = -22$ dB, $\sigma_N = 5$ dB, 200 samples/simulation, 5 simulations.

TABLE 3: Figure 12 configurations.

Name	Radius	Angle
Initial	100%	2π
HK 1	66%	2π
HK 2	55%	2π
HK 3	33%	2π
HK 4	20%	2π
HK 5	100%	π
HK 6	100%	0.2π

6.2. Layer 2: the Ho-Kashyap method

We would like to analyze some specific properties of the Ho-Kashyap method as the potential candidate to implement the location algorithm of layer 2. We would like to see first how the performance depends on the APs distribution: APs have been considered to be equally distributed in a circumference that surrounds the location area as shown in Figure 3; we would like to see what happens when both assumptions change, and the APs are introduced inside the location area and concentrated in a given arc of circumference.

In Table 3 we present the different configurations considered, depending on their radius (denoted as a percentage of the radius in the initial case) and the angle they cover (the initial case would be 2π). Corresponding results are shown in Figure 12.

As it can be seen in Figure 12, if APs are not too close to one another, introducing them inside the location area does not impact too much on the performance, as it happens

in HK 1 (66% of radius) and HK 2 (55%), where a small improvement with respect to the initial distribution can be noticed. However, if they get too close, performance quickly degrades, as in HK 3 (33%) and HK 4 (20%). Something similar happens with the angular distribution; if APs occupy only a semicircumference as in HK 5 (angle π), performance is similar to the initial one. Notwithstanding, if they concentrate in a small angle like in HK 6 (0.2π), performance gets worse, but not so much as it could be expected. In general terms, the system is not very sensible to the APs distribution, which is a very desirable characteristic.

We are also interested in estimating the observation symbol probability distribution in state j , $P(O_t | l_j)$, which is the probability of estimating position $O_t = i$ if the previous state was l_j , as this probability is necessary for the HMM in layer 3. In order to do this, we have calculated the pdf of the observations received when the MT is at different positions, as it is shown in Figure 13 for position $(0, 0)$ and in Figure 14 for position $(0, 4)$.

In both figures, it can be seen how the pdf consists of a probability peak at the MT position and different “errors” that do not seem to follow any specific distribution. Therefore we have decided to model $P(O_t | l_j)$ as a peak at l_j , and a uniform distribution among the other positions:

$$P(O_t | l_j) = \begin{cases} p_{\text{peak}}(j) & \text{if } O_t = j, \\ \frac{1}{(1 - p_{\text{peak}}(j))} & \text{if } O_t \neq j, \end{cases} \quad (32)$$

where $p_{\text{peak}}(j)$ is the peak probability.

This probability depends on the considered MT position j , as presented in Figure 15. Here it can be seen how peak

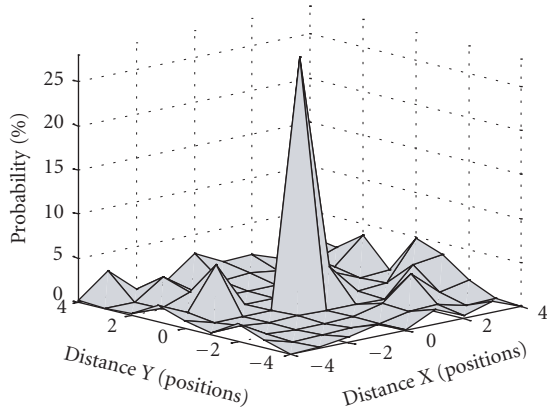


FIGURE 13: The observation symbol probability distribution in state $j = (0, 0)$, $P(O_t | I_j = (0, 0))$. $c = 81$ locations, $d = 4$ APs, $m = 10$ samples/position, $\sigma_R = -17$ dB, $\sigma_N = 5$ dB, 50000 samples/simulation, 10 simulations. The peak corresponds to the probability of a successful location (28%).

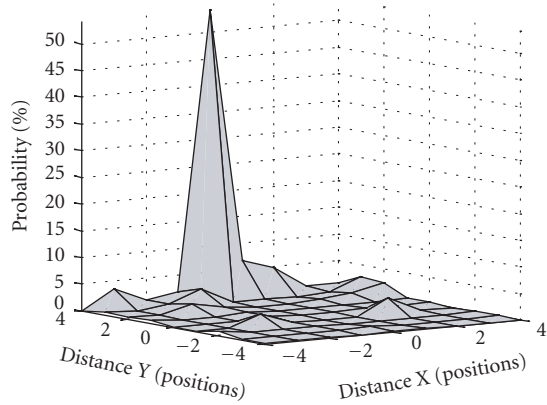


FIGURE 14: The observation symbol probability distribution in state $j = (0, 4)$, $P(O_t | I_j = (0, 4))$. $c = 81$ locations, $d = 4$ APs, $m = 10$ samples/position, $\sigma_R = -17$ dB, $\sigma_N = 5$ dB, 10000 samples/simulation, 10 simulations. The peak corresponds to the probability of a successful location (54%).

probability varies more than 0.15 between different points, degrading in the center of the location area as these positions are far from all the APs.

Notwithstanding, in order to make the system simpler and to reduce calibration, we have implemented layer 3 assuming a constant p_{peak} for all the positions, as commented below.

6.3. Layer 3: application of HMM

Layer 3 is in charge of receiving the observations O_t from layer 2 and inferring the MT position and trajectory trying to reduce as less as possible the system speed. As commented above, we have implemented this layer with an HMM of c states.

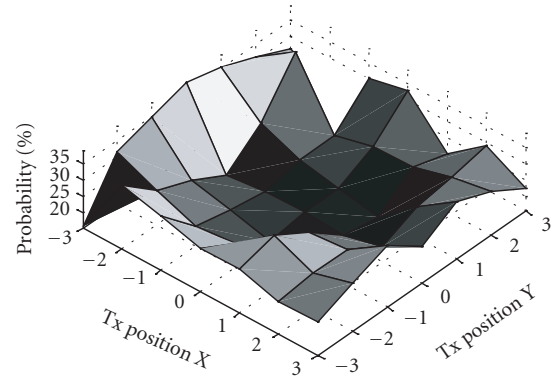


FIGURE 15: Probability of a successful location as a function of the MT position. $c = 49$ locations, $d = 4$ APs, $m = 10$ samples/position, $\sigma_R = -12$ dB, $\sigma_N = 5$ dB, 1960 samples/simulation, 20 simulations.

In order to implement it, we had to define the state transition probability distribution $\mathbf{P} = \{p_{ij}\}$ or *transition matrix*, the observation symbol probability distribution $P(O_t | I_j)$ or *observation matrix*, and the initial state distribution $\pi = \{\pi_i\}$.

We consider that the MT enters the location area through a given position (which therefore determines the initial state distribution π) and moves with a probability $p_{\text{static}}(j)$ of remaining at position j and $1 - p_{\text{static}}(j)$ of changing of position. If the MT changes its position, it can move with equal probability to all the open adjacent positions. A position i is considered to be open from an adjacent position j if there is no obstacle between them.

To make the scenario more realistic, we force the MT to move in a corridor from the entry to the opposite side, passing through the central position $(0, 0)$, where the probability of not moving is denoted as p_{Central} and it is higher than $p_{\text{static}}(j)$ as we considered that this is the user's vital space, where the MT remains most of the time. The MT trajectory is therefore fully characterized by p_{Central} for the central position and p_{static} , which we considered constant for all the other positions.

The observations matrix is modeled following (32) and we have decided to test two possible values of p_{peak} . One of them is the same for all the possible environments as it is a constant with a value $p_{\text{peak}} = 0.5$ for all the positions. The other depends on the environment and it changes depending on the system parameters presented in Table 2, so we have decided to take as p_{peak} the value of the average for all the positions of the probability of a successful location.

When trying to model the transition matrix, we find a problem as we do not know how much information does the system store about the user's profile, characterized by p_{Central} and p_{static} . Therefore we have also decided to consider two possible alternatives: in one of them the transition matrix includes perfect information about the user's profile, whereas in the other the system works with a raw estimation p'_{static} for all the positions, including $(0, 0)$. In the following simulations, we have taken $p_{\text{Central}} = 0.996$, $p_{\text{static}} = 0.6$, and $p'_{\text{static}} = 0.9$.

TABLE 4: Simulated HMM models.

		Estimation of the $P(O_t I_j)$	
		Perfect	Estimation (0.5)
Estimation of the transition matrix P	Perfect	HMM 1	HMM 2
	Estimation (0.9)	HMM 3	HMM 4

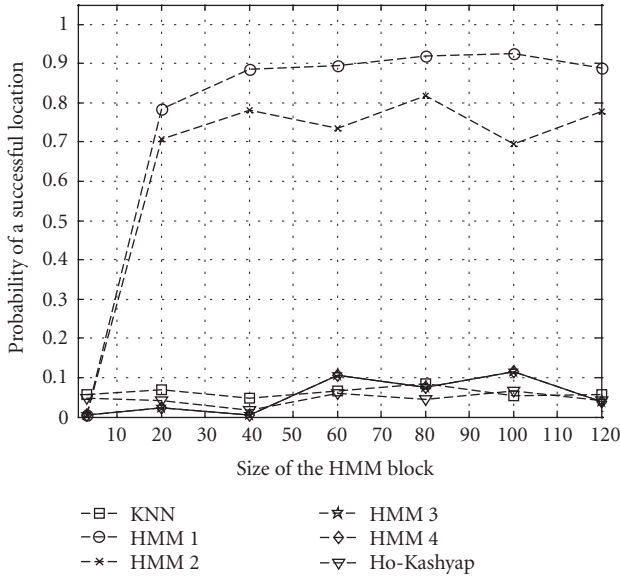


FIGURE 16: Probability of a successful location as a function of the HMM block length T . $c = 49$ locations, $d = 4$ APs, $m = 10$ samples/position, $\sigma_R = -7$ dB, $\sigma_N = 5$ dB, 10 000 samples/simulation, 15 simulations.

In Table 4, the four different HMM models that are tested can be seen. In HMM 1, both the transition and observation matrix are perfectly estimated, in HMM, $2P(O_t | I_j)$ is estimated following (32), whereas in HMM 3 it is perfectly estimated but the transition matrix is modelled as considered in the previous paragraph ($p'_{static} = 0.9$). Finally, in HMM 4 both matrices are estimated.

Results are presented in Figures 16 and 17 for high and low small-scale distortions, respectively. We are comparing the four HMM models with a “pure” layer 2 Ho-Kashyap procedure and a 1-KNN. The Ho-Kashyap can be considered as an especial case of HMM where T , the block length, is set to 1. The 1-KNN is presented in order to compare our methods with the best performance layer 2 algorithm.

In Figure 17, it is shown that, in a low-distortion environment, HMM methods can achieve a higher success probability than the other algorithms, including KNN. It should also be noticed that the performance between perfect and approximate estimations of the observation matrix is very similar (comparisons between HMM 1–2 and HMM 3–4), whereas the estimation of the transition matrix has a considerable impact on the result. Those models that estimate the transition

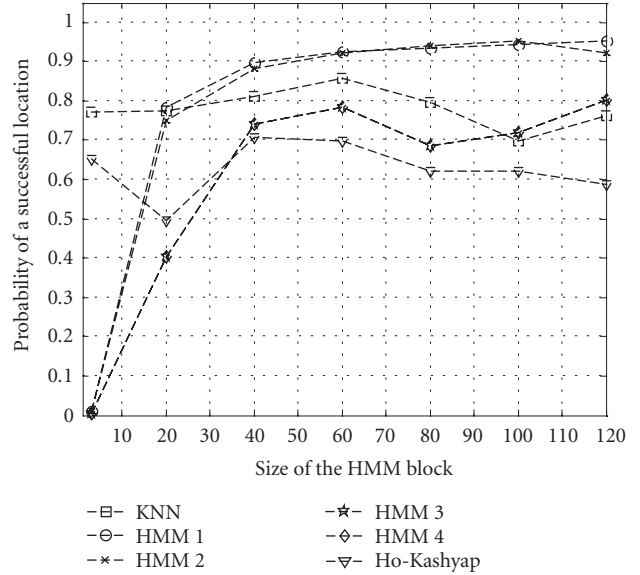


FIGURE 17: Probability of a successful location as a function of the HMM block length T . $c = 49$ locations, $d = 4$ APs, $m = 10$ samples/position, $\sigma_R = -22$ dB, $\sigma_N = 5$ dB, 10 000 samples/simulation, 15 simulations.

matrix (HMM 3 and 4) need more than 40 samples per block in order to reach a similar performance to the KNN algorithm, which is a $\sim 20\%$ worse than the probability achieved by those with a perfect user’s profile (HMM 1 and 2).

In a high-distortion environment as the one presented in Figure 16, where layer 2 algorithms totally fail to correctly estimate the MT position, models with a perfect user profiling still locate it with success probabilities beyond 80%. It is now clearly seen how although the observation matrix estimation degrades a little bit the location estimation, the substantial degradation happens if the user’s profile has not been correctly estimated, as in models 3 and 4, which show a similar behavior to layer 2 algorithms.

This analysis shows the importance of a correct user’s profiling, with a higher impact on the performance than other “traditional” parameters, such as the number of calibration points or the number of APs. A good profiling ensures a high location probability and reduces the sensibility to structural or environmental variables.

In Figure 18, time performance is displayed, related to the computational time of the Ho-Kashyap method with $m = 1$ training sample/position. Two interesting results should be highlighted: first, the reduced impact of T , HMM block size, on the computational time, which allows big blocks and therefore improved performance in HMM algorithms. Second, HMM algorithms are around 10 times slower than “pure” Ho-Kashyap, twice than KNN for this number of calibration samples, and faster than the other non-MSE algorithms already shown in Figure 8. Layer 3 therefore introduces a considerable delay on the system performance, but not so much as it could be expected, and inferior to most layer 2 algorithms, such as Bayesian or WKNN.

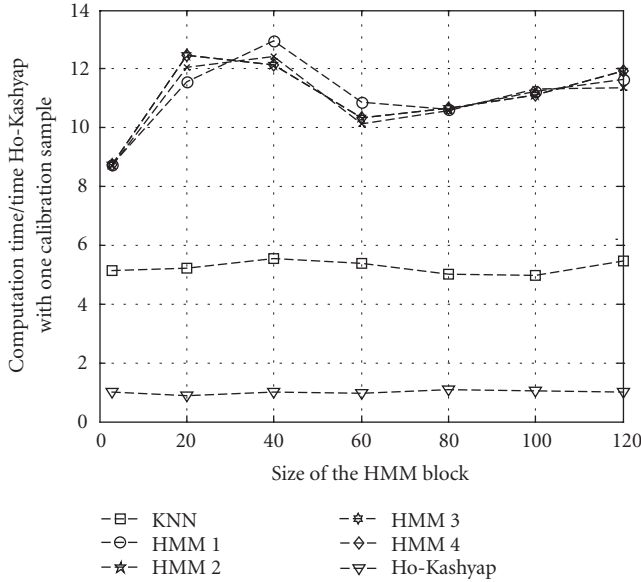


FIGURE 18: Time performance related to the time of the Ho-Kashyap with $m = 1$ as a function of the HMM block length T . $c = 49$ locations, $d = 4$ APs, $m = 10$ samples/position, $\sigma_R = -7$ dB, $\sigma_N = 5$ dB, 10 000 samples/simulation, 15 simulations.

Finally we would like to analyze the layer 3 capabilities to correctly infer the user's profile. We have therefore simulated a total estimation HMM 4 model and we have estimated p_{Central} as the probability of remaining in $(0, 0)$ and p_{Static} as the probability of remaining in $(0, -3)$, the entry position. We have followed the calibration algorithm presented in [35] and results are presented in Figure 19 for a low distortion environment. It can be noticed that convergence is relatively fast (0.95 and 0.62 with less than 500 samples), even if we cannot reach the exact transition probabilities (0.996 and 0.5) due to the inherent error probability ($\approx 20\%$).

7. CONCLUSIONS

In this paper, we have presented a new approach to location estimation systems based on LDF and HMM, and we have compared it with algorithms previously presented in the literature. Both the algorithm and the comparisons can be surely improved in the future, so we want to remark that we do not consider them to be our main contribution, but a way to highlight some of the most interesting problems in WLAN positioning systems and to present some novel approaches for them.

In the system description, our two-layer architecture and its correspondence with the work presented in [20, 21] introduces a novel approach to positioning, based on a fast and simple layer 2, instead of the complex and accurate traditional algorithms. Further research should be focused on the interface between layers in order to ensure flexibility and scalability and on the tradeoff between time and accuracy that we have achieved by implementing this architecture, which is strongly dependent on the application considered.

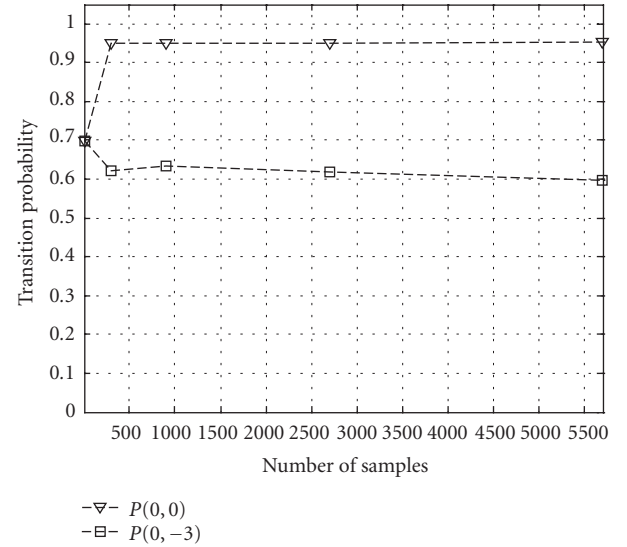


FIGURE 19: Probability of a successful location as a function of the accumulative number of samples. $c = 49$ locations, $d = 4$ APs, $m = 10$ samples/position, $T = 60$ samples/block, $\sigma_R = -22$ dB, $\sigma_N = 5$ dB, 2 simulations.

Another contribution is the concept of a mapping based on vital spaces instead of grids. Vital spaces are more suitable for pattern recognition problems and allow the system administrator to define the positions emphasizing the most important areas for the users.

A special reference should be made to channel models. Despite the exhaustive work presented in [29, 30], it is necessary to achieve a full statistical RSS characterization for different environments, in order to apply more realistic simulation models than the one we have developed. Variability between different types of building (concrete, glass, wood, etc.), different WLAN cards and antennas, and human influence should be taken into careful consideration.

In layer 2, we have presented two novel location methods based on LDF and MSE procedures. More suitable location methods will be probably developed in the near future, but we want to remark the importance of a fast and simple, almost calibration-free system if it is to be implemented in small MTs such as cell phones or PDAs. Further research should be oriented towards the study of the application of ray tracing models based on simple maps and should analyze the number of averaged RSS samples that guarantee the best tradeoff between distortion reduction and time efficiency (taking layer 3 also into consideration).

Another important point is related to the number and distribution of the APs. Our research has shown that the improvement by increasing their number quickly saturates, and that the system is robust to changes in the AP distribution. Future research in this field should be focused on the grid distribution of APs and the smart selection algorithm, so that the AP distribution allow the MTs to receive, no matter their position, a constant number of APs, discarding those with the highest RSS variability.

Although the possibility of adding an additional layer based on HMM or Kalman filters had been previously commented in literature, we have fully developed it, analyzing the importance of a good transition matrix characterization. Based on data from layer 2, we have proposed a simple statistical distribution for the observation probability that performs well in practice and we have also presented the importance of user's profiling (with a higher impact on the system performance than the calibration) and the possibility of a self-profiling system. Further work should be carried out to optimize the HMM algorithm to make it faster for location applications and to develop new theoretical models for profiling to improve the system performance.

Our analyses have yielded additional interesting conclusions about the behavior of location estimation systems.

- (i) Performance is enhanced with an increment in large-scale log-normal distortion under the assumption that it affects in the same way the calibration and the location samples. If this assumption did not hold, the effect would be the opposite.
- (ii) Small-scale distortions quickly degrade the performance, which falls sharply for σ_R greater than -25 dB.
- (iii) Performance is improved by increasing m , the number of calibration samples per position. KNN and MSE methods can cope better with small m than Bayesian methods do.
- (iv) KNN and WKNN computation time grows linearly with m , whereas it is constant for Bayesian and MSE methods. Nevertheless, Bayesian times are more than 20 times greater than the MSE ones. CIT size displays a similar behavior.
- (v) Performance falls almost linearly as the number of positions c grows, that is, when the location area gets wider.
- (vi) Performance can be improved by increasing d , the number of APs transmitting to an MP. This improvement saturates for a given number of APs; nevertheless, further increments of d can be useful if a smart selection algorithm is implemented.
- (vii) Changes in the AP distribution do not affect significantly to the system performance.
- (viii) The observation symbol probability distribution $P(O_i = i | l_j)$, that is, the probability of layer 2 estimating a sample from position j as position i , can be modelled as a peak at position j (whose height is the probability of a successful location) and a uniform distribution among the other positions. Numerical results from layer 3 show that this assumption is reasonably well suited.
- (ix) The probability of a successful location is lower in the areas that are far away from all APs.
- (x) HMM works extremely well both in high and low distortion environments. To ensure that an HMM works properly in a high distortion situation, it should have properly estimated the user's profile.
- (xi) Although layer 3 makes the system ten times slower, it is still faster than Bayesian or WKNN algorithms, and

shows a performance superior to any layer 2 method, including KNN.

- (xii) Under low distortion conditions, HMM can quickly derive the user's profile.

In conclusion, we have presented a novel approach to location estimation based on previous work. We hope it will help to increase the research in this promising field.

ACKNOWLEDGMENTS

The authors would like to thank Mrs Ivana Raos, Mr. Santiago Zazo, and Miss. Blanca Urtasun for their reviews and comments on the text. They also thank Mr. Javier Leube from the Universidad Politécnica de Madrid and Mr. Javier Quintana from Fundación Rafael Escolá for their support and encouragement. This research has been funded by the Fundación Rafael Escolá and by the Universidad Politécnica de Madrid.

REFERENCES

- [1] B. N. Schilit, D. M. Hilbert, and J. Trevor, "Context-aware communication," *IEEE Wireless Communications*, vol. 9, no. 5, pp. 46–54, 2002.
- [2] B. N. Schilit, N. I. Adams, and R. Want, "Context-aware computing applications," in *Proceedings of the Workshop on Mobile Computing Systems and Applications (WMCSA '94)*, pp. 85–90, Santa Cruz, Calif, USA, December 1994.
- [3] G. Chen and D. Kotz, "A survey of context-aware mobile computing research," Tech. Rep. TR2000-381, Department of Computer Science, Dartmouth College, Hanover, NH, USA, November 2000, pp. 1–16.
- [4] P. Bahl and V. N. Padmanabhan, "Radar: an in-building RF-based user location and tracking system," in *Proceedings of 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '00)*, vol. 2, pp. 775–784, Tel Aviv, Israel, March 2000.
- [5] P. Bahl, V. N. Padmanabhan, and A. Balachandran, "Enhancements to the RADAR user location and tracking system," Tech. Rep. MSR-TR-00-12, Microsoft Research, February 2000.
- [6] P. Prasithsangaree, P. Krishnamurthy, and P. K. Chrysanthis, "On indoor position location with wireless LANs," in *Proceedings of 13th IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC '02)*, vol. 2, pp. 720–724, Lisboa, Portugal, September 2002.
- [7] M. Brunato and C. K. Kalló, "Transparent location fingerprinting for wireless services," Tech. Rep. DIT-02-0071, September 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, 2002.
- [9] T. Roos, P. Myllymäki, H. Tirri, P. Misikangas, and J. Sievänen, "A probabilistic approach to WLAN user location estimation," *International Journal of Wireless Information Networks*, vol. 9, no. 3, pp. 155–164, 2002.
- [10] P. Kontkanen, P. Myllymäki, W. Buntine, J. Rissanen, and H. Tirri, "An MDL framework for data clustering," in *Advances in Minimum Description Length: Theory and Applications*, MIT Press, Cambridge, Mass, USA, 2004.
- [11] M. A. Youssef, A. Agrawala, and A. U. Shankar, "WLAN location determination via clustering and probability distributions," in *Proceedings of the 1st IEEE International Conference*

- on *Pervasive Computing and Communications (PerCom '03)*, pp. 143–150, Fort Worth, Tex, USA, March 2003.
- [12] M. A. Youssef and A. Agrawala, "Handling samples correlation in the horus system," in *Proceedings of 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, vol. 2, pp. 1023–1031, Hong Kong, March 2004.
- [13] A. Schwaighofer, M. Grigoras, V. Tresp, and C. Hoffmann, "GPPS: a Gaussian process positioning system for cellular networks," in *Proceedings of 17th Annual Conference on Neural Information Processing Systems (NIPS '03)*, Vancouver, BC, Canada, December 2003.
- [14] M. Brunato and R. Battiti, "Statistical learning theory for location fingerprinting in wireless LANs," *Computer Networks*, vol. 47, no. 6, pp. 825–845, 2005.
- [15] R. Battiti, T. L. Nhat, and A. Villani, "Location-aware computing: a neural network model for determining location in wireless LANs," Tech. Rep. DIT-02-0083, University of Trento, Trento, Italy, February 2002.
- [16] A. Smailagic and D. Kogan, "Location sensing and privacy in a context-aware computing environment," *IEEE Wireless Communications*, vol. 9, no. 5, pp. 10–17, 2002.
- [17] M. Spreitzer and M. Theimer, "Providing location information in a ubiquitous computing environment (panel session)," in *Proceedings of the 14th ACM Symposium on Operating Systems Principles (SOSP '93)*, pp. 270–283, Asheville, NC, USA, December 1993.
- [18] 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," in *Proceedings of 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, vol. 2, pp. 1001–1011, Hong Kong, March 2004.
- [19] T. W. Christ and P. A. Godwin, "A prison guard duress alarm location system," in *Proceedings of IEEE International Carnahan Conference on Security Technology*, pp. 106–116, Ottawa, Ontario, Canada, October 1993.
- [20] J. Hightower, B. Brumitt, and G. Borriello, "The location stack: a layered model for location in ubiquitous computing," in *Proceedings of Workshop on Mobile Computing Systems and Applications (WMCSA '02)*, pp. 22–28, Callicoon, NY, USA, June 2002.
- [21] D. Gramann, J. Hightower, W. Lara, and G. Borriello, "Real-world implementation of the location stack: the universal location framework," in *Proceedings of Workshop on Mobile Computing Systems and Applications (WMCSA '03)*, pp. 122–128, Monterey, Calif, USA, October 2003.
- [22] J. Hightower and G. Borriello, "Location systems for ubiquitous computing," *IEEE Computer*, vol. 34, no. 8, pp. 57–66, 2001.
- [23] P. Wertz, G. Wolfle, R. Hoppe, and F. M. Landstorfer, "Deterministic propagation models for radio transmission into buildings and enclosed spaces," in *Proceedings of 33rd European Microwave Conference*, vol. 3, pp. 1147–1150, Munich, Germany, October 2003.
- [24] B. Sklar, "Rayleigh fading channels in mobile digital communication systems. I. Characterization," *IEEE Communications Magazine*, vol. 35, no. 7, pp. 90–100, 1997.
- [25] T. S. Rappaport, *Wireless Communications. Principles & Practice*, Prentice-Hall, Upper Saddle River, NJ, USA, 1996.
- [26] K. Pahlavan and P. Krishnamurthy, *Principles of Wireless Networks, A Unified Approach*, Prentice-Hall, 2002.
- [27] J. B. Andersen, T. S. Rappaport, and S. Yoshida, "Propagation measurements and models for wireless communications channels," *IEEE Communications Magazine*, vol. 33, no. 1, pp. 42–49, 1995.
- [28] S. Y. Seidel and T. S. Rappaport, "914 MHz path loss prediction models for indoor wireless communications in multifloored buildings," *IEEE Transactions on Antennas and Propagation*, vol. 40, no. 2, pp. 207–217, 1992.
- [29] K. Kaemarungsi and P. Krishnamurthy, "Properties of indoor received signal strength for WLAN location fingerprinting," in *Proceedings of the 1st Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQ-uitous '04)*, pp. 14–23, Boston, Mass, USA, August 2004.
- [30] K. Kaemarungsi and P. Krishnamurthy, "Modeling of indoor positioning systems based on location fingerprinting," in *Proceedings of 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, vol. 2, pp. 1012–1022, Hong Kong, March 2004.
- [31] A. M. Ladd, K. E. Bekris, A. Rudys, G. Marceau, L. E. Kavradi, and D. S. Wallach, "Robotics-based location sensing using wireless ethernet," in *Proceedings of the 8th Annual International Conference on Mobile Computing and Networking (MobiCom '02)*, pp. 227–238, Atlanta, Ga, USA, September 2002.
- [32] K. Pahlavan, P. Krishnamurthy, and J. Beneat, "Wideband radio propagation modeling for indoor geolocation applications," *IEEE Communications Magazine*, vol. 36, no. 4, pp. 60–65, 1998.
- [33] P. Kontkanen, P. Myllymäki, T. Roos, H. Tirri, K. Valtonen, and H. Wettig, "Probabilistic methods for location estimation in wireless networks," in *Emerging Location Aware Broadband Wireless Adhoc Networks*, Kluwer Academic, Dordrecht, Germany, 2004.
- [34] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, Wiley-Interscience, New York, NY, USA, 2nd edition, 2000.
- [35] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.
- [36] G. D. Forney, "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, pp. 268–278, 1973.

Galo Nuño-Barrau received the Telecom Engineer degree (*summa cum laude*) from the Universidad Politécnica de Madrid (UPM), Spain, in 2002. He is currently working as a Research Engineer at Fundación Rafael Escolá working towards his Ph.D. He also collaborates with the Signal Processing Group of UPM. He was employed as a Research Assistant in the Department of Electrical Engineering of UPM from 2001 to 2002, working on various problems in signal processing and simulation. During 2002–2003, he was researching on substrate integrated waveguide technology under the supervision of Professor Ke Wu at Polygrammes Research Center of l'École Polytechnique de Montréal, Montréal, QC. He has been awarded a Telefónica Móviles Fellowship, a Rafael Escolá Prize, and the ETSIT-UPM Special Award to the best academic trajectory. In 2003, he joined The Boston Consulting Group as a business analyst. His research interests include UWB systems, location estimation with wireless LAN, and application of stochastic processes to economics and finance.



José M. Páez-Borrallo received the Telecom Engineer and the Doctor Engineer from Universidad Politécnica de Madrid, Spain, in 1982 and 1987, respectively. From 1982 to 1985, he was with AEG-Telefunken (Ulm, Germany, and Madrid, Spain) as a Project Engineer. From 1986 to 1987, he was with Electrónica Ensa as project manager. In both companies, he was involved with the development of radiocommunications and signal classification systems for the Spanish army. He joined the UPM as Faculty Staff in September 1987, first as Associate Professor until July 1998 and then as Full Professor. From 1999 to 2000, he was a Visiting Researcher at the International Computer Science Institute (ICSI) and Berkeley Wireless Research Center (BWRC) at Berkeley, Calif., USA. He also served as an Academic Secretary in the SSR Department from 1994 to 1998 and the Head of the Escuela Técnica Superior de Ingenieros de Telecomunicación (ETSIT) from 2002 to 2004. He is currently the Vice-Rector of the international relations at UPM. His main research activities are in the area of signal processing with special emphasis in the fields of communications and audio.

