ARTICLE IN PRESS

Future Generation Computer Systems ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Practical passive localization system based on wireless signals for fast deployment of occupancy services

Pedro E. Lopez-de-Teruel *, Felix J. Garcia, Oscar Canovas

University of Murcia, Facultad de Informatica, Murcia 30100, Spain

HIGHLIGHTS

- A WiFi passive localization system to offer occupancy services is proposed.
- The system is able to track unmodified mobile devices after a fast training phase.
- Several representation and metrics are proposed to cope with device heterogeneity.
- A complete experimental validation of the system is presented.
- A case study was conducted in a large building facility with 4000 frequent users.

ARTICLE INFO

Article history:
Received 14 November 2016
Received in revised form 2 September 2017
Accepted 7 September 2017
Available online xxxx

Keywords:
Building occupancy
WiFi signals
Passive localization
Heterogeneous devices
Machine learning

ABSTRACT

Occupancy is relevant information about key aspects, such as energy consumption or comfort management. Energy-saving and environmental quality strategies can be carried out in response to real-time facility occupancy. Some relevant solutions to measure and monitor occupancy information leverage radio-based indoor localization systems and employ Received Signal Strength (RSS) as the main source of data for location determination. However, those approaches usually require a previous training and calibration stage that involves a time-consuming and labor intensive site survey process, and which is also readily affected by environmental dynamics. In this paper, we propose a practical passive localization system for fast deployment of occupancy services able to track unmodified and heterogeneous devices after a quick and straightforward training phase. We present an experimental validation of the system that was conducted for 9 months in a lecture building of 6000 square meters with 20 classrooms and 4000 frequent users, where the existing teaching computers themselves were used as monitors to capture 802.11 traffic. In this environment, we test different representations and metrics to process the RSSI information and perform a thorough analysis of some important design parameters, which have a direct impact on both accuracy and time granularity of the localization system.

1. Introduction

Nowadays, building energy and environmental quality management is an important aspect which requires solutions and strategies that can be carried out in response to real-time changes. In this sense, and especially in dynamic environments, occupancy data represent the most relevant building information in terms of both energy consumption and overall indoor environmental quality. The presence of occupants will have a direct impact on, for example, heating, ventilation, and air conditioning (HVAC) systems, influencing variables like heat loads, system running time, heating required, cooling and distribution of conditioned air or preferred temperature set points. Occupancy information can also be beneficial in many other application areas such as safety, security or emergency response, to mention but a few.

In recent years, several solutions have been proposed to design occupancy sensing systems [1]. Due to the high density of access points in typical urban and indoor environments, many of these solutions are based on wireless localization schemes, where the Received Signal Strength (RSS) is the main source of data for location determination [2]. In these methods, the localization process is usually divided into two phases, namely, the *training phase* and the *online operation*, each with its own implementation issues. The training phase involves a site survey process in which the RSSIs at every point of interest are recorded in order to build the fingerprinting database, a manual task which is traditionally supposed to be time consuming, labor intensive, and easily affected by environmental changes. As for the online operation phase, most

* Corresponding author.
E-mail address: pedroe@um.es (P.E. Lopez-de-Teruel).

http://dx.doi.org/10.1016/j.future.2017.09.022
0167-739X/

of these systems assume that a specific software component is running on the mobile devices in order to send signal observations to a particular localization server. This is another potential drawback as, generally speaking, it is well known that users are reluctant to install apps that are battery consuming. Moreover, certain mobile operating systems present some limitations in obtaining the needed RSS information. Another important issue is that different device models tend to generate signals with very different RSSI and temporal patterns, making calibration techniques that tolerate device diversity necessary in both the training and online stages. These calibration procedures usually need device-specific data, which are not always easy to obtain.

In this paper we propose a fast deployment system for measuring building occupancy information that overcomes many of those potential drawbacks:

- First, our proposal is able to track unmodified mobile devices using monitoring equipment in the areas of interest –i.e., it performs *passive* localization – and therefore does not require the explicit collaboration of the users. Taking advantage of the fact that mobile devices periodically scan 802.11 channels for access points –which involves the transmission of probe messages – or send data frames – if they are already connected to some existing wireless network – we can, in both cases, capture the corresponding radio signals generated in order to perform the localization. Note that this does not necessarily imply the deployment of new elements, since we can make use of existing hardware in order to add the monitoring functionality.
- Second, our proposal is able to cope with the device heterogeneity problem by using different data representation methods which are mainly based on the order relationship information between RSS values, thus discarding the absolute values which require the adoption of calibration methods.
- Finally, our proposed training stage involves only a lightweight site survey based on the definition of a minimum number of points of interest and a non-exhaustive recording procedure. As we will see, this lightweight process is suitable and feasible thanks to the adapted representation methods and associated metrics that we will define.

In order to evaluate our proposal, we present an experimental validation that was conducted in a lecture building of 6000 square meters with 20 classrooms. This scenario was defined mainly for classification purposes, that is, to infer occupancy of the different classrooms during the day. During a 9-month operation period we detected more than 200,000 different MAC addresses, though a more detailed temporal analysis determined that the actual number of frequent users was only around 4000 (after eliminating those MACs simply corresponding to sporadic or nearby passing devices). These remaining devices still constitute a challenging heterogeneous dataset, with many different device models generating a widely diverse set of signal strengths and temporal patterns. We have tested several representation methods and distance metrics that, when applied to a simple k-nearest neighbors (k-NN) classifier, provide satisfying results in terms of classification accuracy, which confirms the suitability of our proposal for occupancy-based applications.

The main aim of the paper, therefore, is to get clear insights into all the practical considerations to take into account when deploying a fully operational passive localization system based on wireless signals, suitable to offer indoor occupancy information based services in a practical and agile way. There are several real life scenarios that could adopt this approach to infer occupancy information and to make use of that information for higher level services. Indeed, our testing environment has evolved and is being used in our own University as the starting point for an HVAC system based on occupancy. Additionally, other real applications are feasible with our system, like the one presented by Ruiz et al. [3] for a large hospital complex regarding people's presence, movement and roles.

The rest of the paper is structured as follows. Section 2 discusses the related work. The main elements of our system and its distinctive training and operational phases are presented in Section 3. The various representations proposed and the associated metrics are presented in Section 4, while Section 5 describes the experimental environment and Section 6 reports a thorough evaluation to illustrate the performance of our proposal. Finally, conclusions are drawn and future work is outlined in Section 7.

2. Related work

Occupancy sensing systems have become the subject of much attention recently due to the increasing number of sensors and devices with wireless connectivity. Many works follow a similar approach to the one we present here, that is, to infer information from existing infrastructure elements [4]. Kjaergaard et al. [5] provide a categorization framework for these kind of systems. According to their framework, the type of information provided by our proposal is presence-count, with a spatial granularity of room-level, a temporal coverage ranging from the past to the present time, and a sensor modality based on infrastructure.

Several works providing *passive localization* solutions to track unmodified smartphones have been described in the literature. For example, Musa and Eriksson [6] performed tests on a busy road detecting 802.11 devices to estimate trajectories. They also presented several methods to prompt passing devices to send additional messages, thus increasing detection rates. Their proposal, though, was tailored specifically to be used in places adjacent to roads, which is not our case since we aim to provide information about the behavior of users monitored passively in indoor environments. More recently, Ruiz et al. [3] proposed a WiFi monitoring system to inform facility planning. Location was estimated taking into account the location of access points (APs), using a lateration algorithm and then mapping to the location of the nearest AP. The resulting mean accuracy of 15 m is nevertheless insufficient for our targeted environment. Moreover, they relied on absolute RSS values, which, as discussed in the introduction, are clearly not suitable for heterogeneous devices.

Most of the proposed indoor positioning techniques based on Wi-Fi signals provide fine-grained device locations at the price of assuming that the device to be positioned is correctly calibrated with respect to the device employed during the training phase [7]. In fact, there is a challenging problem for all the methods based on fingerprinting, namely, that the received signal strength values at a given fixed location may widely vary if they are measured by different devices. It is impractical to manually calibrate each new device and hence some calibration-free solutions have been proposed which make use of alternative features such as RSSI relative magnitude order or hypothesizing a linear dependency between the strength of the signals in order to address the *device heterogeneity* problem [8]. One such system is Yang et al.'s Free-Loc [9]. As we will see, FreeLoc is one of the techniques in which our proposal is inspired, although we introduce some differences in the way we represent the information to make it more suitable for generic machine learning techniques.

Finally, and regarding the possible drawbacks of a potentially *cumbersome training phase*, solutions that deploy wireless localization systems avoiding an intensive site survey process have also been described in the literature [10]. In this sense, in [11] Wu et al. proposed exploiting user motions from mobile phones to crowdsource the training data. However, their technique required a specific software component installed in the mobile device, which as we have already discussed might have some inconveniences from the point of view of the final user. An alternative solution was described in [12], where a clustering method classified the rooms in an unsupervised manner. Although all these solutions avoid a previous site survey process, they generally tend to make the localization system more complex and consequently less robust. Finally, Gao and Harle [13] analyze different methods based on light path surveys instead of dense and detailed manual surveys. As we will show, our training stage follows a similar approach, although in contrast to the latter proposal, our training system does not require sub-meter ground truth geopositioning based on ultrasound techniques. Instead, we adopt a much simpler method in which an operator assisted by a training application running on a simple tablet or smartphone will be able to generate a complete fingerprinting database for a target building in a fast and easy way.

3. System description

3.1. Main elements

As in any other passive localization system, monitors are central elements in our proposal. In a broad sense, a monitor is any hardware element running software able to capture 802.11 traffic and export the relevant information of that data to a central server. The required density of monitors can be relatively low for many occupancy estimation scenarios, but this, of course, depends greatly on both the desired accuracy (i.e., the spatial granularity of the localization system) and the physical characteristics of the environment itself. As we have already explained, our monitors rely only on monitoring the frames normally transmitted by user devices as part of their usual 802.11 connections or active scanning periods, without using prompting techniques to increase the number of packets received from them (like the aforementioned system which Musa et al. proposed in [6]). Instead, in order to perform its scanning, each of our monitors simply scans the different 802.11 channels periodically, following a plain round-robin schedule. Parameters of this continuous process, such as the scan time for each channel, the set of channels to scan, or the maximum amount of time before a monitor transmits the collected information to the server, are fully configurable. For each captured packet the only information which is used is the MAC address of the emitting mobile device – which is key-hashed for privacy reasons –, the RSSI value and the corresponding time stamp.

Our occupancy sensing system has to provide a flexible characterization for all the mobile devices being monitored. We assume that they will show a wide variety of hardware, WiFi interfaces, antennas, operating systems, and the like. Consequently, they will produce signals with very different strength and temporal patterns. Since we want to use the frames normally transmitted by the devices as carried by any type of users during their daily routines, we cannot impose any restrictions on the specific device or make any assumptions on its current state (whether it is switched on or in a low-power suspended state, whether or not it is connected to any WiFi access point, etc.) This total absence of control will involve some key system design decisions that will be explained in Section 4.

Finally, a central server will be in charge of hosting the localization engine itself. A central element of this engine will be a database containing both the fingerprinting data collected during the training phase and the continuously updated information of the captures sent to it by the monitors. The server will be also in charge of running the localization software responsible for calculating occupancy and positioning information when required. In order to do this, the server provides an API that will be used by higher-level location-based services according to each specific scenario. In Section 3.3 we will provide some examples of the kind of services that can be deployed during the online phase.

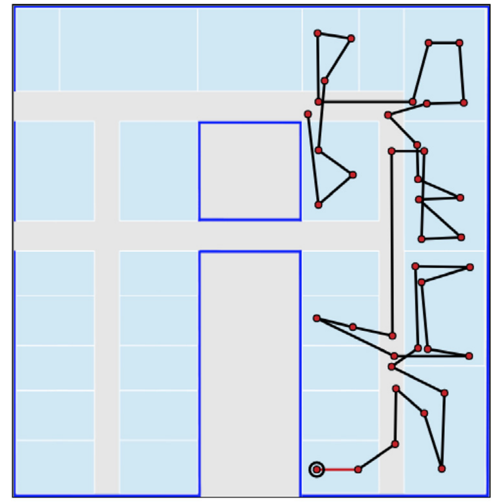

Fig. 1. User interface of the training application based on waypoints.

3.2. Training phase

Our training phase involves a site survey process to build the corresponding geopositioned fingerprinting database. This process is traditionally assumed to be time consuming, but we have used a different approach to make it faster and more straightforward. Occupancy is mainly based on a per-zone classification problem, rather than exact position regression, which alleviates the need for an otherwise typically exhaustive sampling procedure.

Our monitors can be configured to act as conventional access points (AP). This configuration is used during the training phase since we adopt a classical –*active*–approach to create a fingerprint map of radio signals for every zone of interest. Using a mobile device running a customized training application, we obtain the RSS values of the beacon frames transmitted by our monitors in AP mode. The corresponding observations are then tagged (x, y) using a locally defined coordinate system.

We do not rely on any additional localization system for location ground truth. Instead, approximate geopositions are obtained as the operator follows the indications of the training app, which provides continuous visual feedback about the required walking path for the site survey. As the example in Fig. 1 shows, a set of connected waypoints forms the path to be followed by the operator. The app is also responsible for collecting the 802.11 fingerprints that will be geo-tagged using the coordinates where the operator has to be physically at that moment (enclosing black circle shown in the example). All the operator has to do is to follow the path shown as accurately as possible and remain still at the designated waypoints (smaller red points in the figure) for the required scanning time. Given a particular scenario, our application provides the mechanisms to define these waypoint based paths, as well as how much scan time is required for each waypoint. The walking speed of the operator can also be configured. A typical path involves only a few dozen waypoints, with 5–10 s stops at each, for a total of a few minutes to cover relatively large portions of building, such as that shown in the figure.

We are aware that this method can generate some noisy observations, since the current position of the operator does not always match the exact coordinates shown by the application. The set of obtained samples also tends to be relatively sparse, due to the relatively short sampling periods. However, for occupancy purposes, the resulting fingerprinting maps offer an excellent trade-off between the accuracy subsequently obtained in the online phase and the invested training times. As we will demonstrate in Section 6,

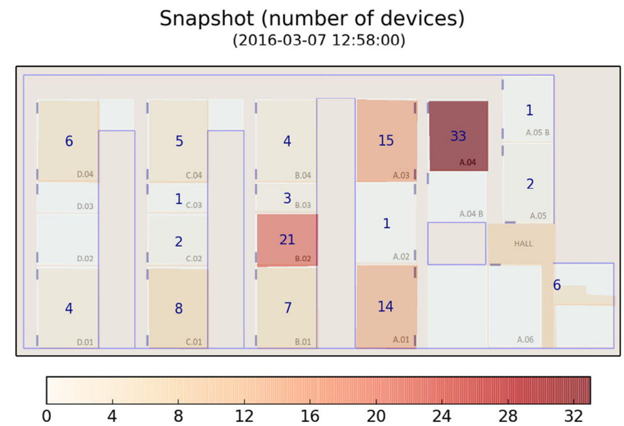

Fig. 2. Example heatmap obtained during the online operation of the system.

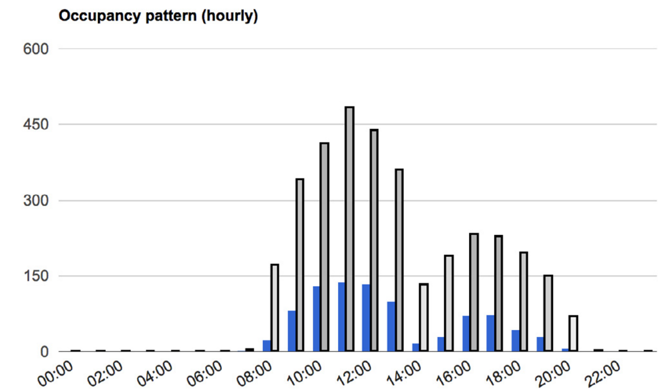

Fig. 3. Temporal occupancy analysis for a given period (one day in the example).

this light survey technique does not significantly affect the correct classification rate, and it makes the training stage clearly feasible even for relatively large scale scenarios.

3.3. Online operation

The particular services to be provided by our location engine in each scenario will be highly dependent on the targeted applications, but we present two examples that could be meaningful for illustration purposes. On the one hand, as Fig. 2 shows, it is possible to obtain occupancy heat maps, which are useful to show real-time information or to analyze occupancy during a given period of time. In the example shown, the system provides the number of occupants in every zone of the building. On the other hand, we can also analyze the temporal occupancy pattern of a building (or just a particular zone) over a given period. For example, as Fig. 3 shows, we can determine the number of devices that were present in the building for every hour of a particular day. It is worth noting that since our localization engine will be able to provide not only the location of the devices, but also the amount of time they spend at each location, we can provide two kinds of data. Gray bars in the figure refer to the total number of different devices that were present at each particular hour in the zone of interest. But, additionally, we also display a finer-grained information, shown in the corresponding blue bars, which is directly related to the amount of time that those devices remained in that zone. This information is measured in *devices*hour* units, and is calculated taking into account the total time spent by each device in the targeted zone. In terms of this measure, a mobile device which remained in the zone for a whole hour would contribute with 1 *device*hour* unit, while another that only stayed there for, say, 6 min, would contribute with 0.1. This constitutes a very useful indicator to distinguish passing areas from other zones where users tend to stay for longer periods of time.

4. Data representation and distance metrics

4.1. Raw measures

As we have already stated, our monitors are in charge of collecting the 802.11 frames emitted by the user devices, and then they send the relevant information to a central server to be processed. Using a sufficiently long sampling period Δ (typically 1 to 3 minutes; see next section for a sensible choice for this parameter), we build *raw vectors* $\mathbf{r} = (r_1, \ldots, r_M) \in R^M$ for every captured device during that sampling period, where r_i refers to the maximum RSSI value (in dBms) observed by monitor i for the different frames transmitted by that particular device in the corresponding Δ-length time interval. We use of the maximum value in order to attenuate fading and multipath effects that might affect the RSSI received, and also to minimize the impact of those values obtained when the monitors were capturing in channels which are not the central frequency used by the device to transmit the frames. If any $\mathbf{r}_i$ value is unavailable (because the corresponding monitor did not capture any frame from the corresponding device), a minimum value of -100 dBm is assigned to it, in order to get a completely defined vector.

These raw measures are then transformed into two alternative representation methods, which we call *order vectors* and *ternary vectors*. The purpose of these alternative representations is to build a vector that is well-fitted to apply different distance metrics in the k-NN classifier, while still being suitable for heterogeneous devices.

4.2. Order vectors

The idea behind *order vectors* is to represent just the magnitude relationship between the RSSI measurements of a raw vector, thus discarding the specific r_i values, which might not be very useful, due to the already discussed issue of device heterogeneity. In this case, the output vector obtained represents the relative positions of the RSSI signals perceived by each monitor when the input raw vector components are sorted into a descending order. This way, fluctuations in the RSSI values will not alter the resulting vectors as long as the relative order of the signal strengths for the different monitors is maintained, which is very suitable when dealing with heterogeneous devices. We will illustrate the idea with a very simple example. Suppose that we had only four monitors ($M = 4$) and that we obtained a raw sample $\mathbf{r} = (r_1, \ldots, r_4) = (-60, -80, -50, -62)$ (all r_i measures in dBms). The corresponding order vector $\mathbf{o} = (o_1, \ldots, o_4) \in N^4$ would be $(2, 4, 1, 3)$, reflecting the corresponding magnitude order of the signals.

In fact, *order vectors* can be computed depending also on an additional tolerance parameter δ, such that two components o_i and o_j are considered to have the same order when $|r_i - r_j| <= \delta$ dBm. The δ parameter is used here to enforce a significant difference between the RSSI values for it to be considered really relevant, just as in the cited FreeLoc system [9]. Thus, the former vector $(o_1, \ldots, o_4) = (2, 4, 1, 3)$ was computed for a value of $\delta = 0$, while it would have been $(2, 4, 1, 2)$ if we had used a value of $\delta = 5$ dBm, for example. Since it is difficult to determine a priori an optimal δ, we will obtain a reasonable value for it for our application scenario by cross-validation.

Please cite this article in press as: P.E. Lopez-de-Teruel, et al., Practical passive localization system based on wireless signals for fast deployment of occupancy services, Future Generation Computer Systems (2017), http://dx.doi.org/10.1016/j.future.2017.09.022.

4.3. Ternary vectors

Ternary vectors are just another alternative to avoid using absolute RSSI values, while still keeping the relevant magnitude order relationships among every pair of monitors. This time the ternary vector is built using all the $\binom{M}{2} = \frac{M*(M-1)}{2}$ combinations of monitors by pairs. Using again a prespecified δ parameter, each of these pairwise comparisons can give rise to three different values $+1$, -1 or 0 (thus the name of *ternary* vectors): $\forall c \in \{(i,j)\,|\,i,j \in \{1,\dots,M\}, i<j\}$, we define $t_c = +1$ if $r_i - r_j >= \delta$ (or simply monitor j did not receive any frame from the device); $t_c = -1$ if $r_i - r_j <= -\delta$ (or monitor i did not receive any frame from the device); and $t_c = 0$ when $|r_i - r_j| < \delta$ (or neither monitor i nor monitor j was able to receive any frame from the device). Again, the aim of the δ parameter is to enforce a significant difference between the RSSI values.

As an illustrating example, and using the same input raw vector as before, $\mathbf{r} = (r_1,\dots,r_4) = (-60,-80,-50,-62)$, the corresponding output vector $\mathbf{t} = (t_1,\dots,t_6)$ would be $(+1,-1,+1,-1,-1,+1)$ for $\delta = 0$, or $(+1,-1,0,-1,-1,+1)$ for $\delta = 5$, where the positions $1\dots6$ of the vector represent the $\binom{4}{2}$ possible pair comparisons $\{(1,2),(1,3),(1,4),(2,3),(2,4),(3,4)\}$, in that exact lexicographical order.

4.4. Distance metrics

Given an existing raw vector $\mathbf{r}_a$ in the training dataset, and a new input query raw vector $\mathbf{r}_b$ obtained by the passive monitoring system for a given device and time interval, we experiment with k-NN classification using the following seven different distancemetrics[1]:

1. **Euclidean distance:** This is the metric we will use when working directly with raw vectors. Given two such vectors $\mathbf{r}_a$ and $\mathbf{r}_b$, it is defined as

$$EU_{\mathbf{r}_a,\mathbf{r}_b} = \sum_{i=1}^{M} \sqrt{(r_{a,i} - r_{b,i})^2} \tag{1}$$

   where $\mathbf{r}_{a,i}$ and $\mathbf{r}_{b,i}$ are the ith component of the $\mathbf{r}_a$ and $\mathbf{r}_b$ vectors, $\forall p \in 1\dots M$. That is, the $\mathbf{r}_{a,i}$ and $\mathbf{r}_{b,i}$ values come from the raw RSSIs measures obtained for the whole set of M monitors during the training ($\mathbf{r}_a$) and the online phase ($\mathbf{r}_b$), respectively. As we will show, this metric is not suitable to work with heterogeneous environments, and will only be used as a basis for comparison with the rest of metrics.
2. **Weighted Pearson correlation distance:** This distance is a measure of the statistical dependence between two vectors. It uses order vectors $\mathbf{o}_a$ and $\mathbf{o}_b$ computed from the input raw vectors $\mathbf{r}_a$ and $\mathbf{r}_b$, and is defined as one minus the *weighted Pearson correlation coefficient similarity* [14], as shown in Eq. (2):

$$PE_{\mathbf{o}_a,\mathbf{o}_b} = 1 - \frac{\sum_{p\in I}(\mathbf{o}_{a,p} - \bar{\mathbf{o}}_a)(\mathbf{o}_{b,p} - \bar{\mathbf{o}}_b)}{\sqrt{\sum_{p\in I}(\mathbf{o}_{a,p} - \bar{\mathbf{o}}_a)^2}\sqrt{\sum_{p\in I}(\mathbf{o}_{b,p} - \bar{\mathbf{o}}_b)^2}} \cdot \frac{|I|}{M} \tag{2}$$

   where I is the set of common monitors, M the total number of them, and $\mathbf{o}_{\bullet,p}$ and $\bar{\mathbf{o}}_\bullet$ are the pth component and the mean of all the $\mathbf{o}_{\bullet,p}$ $\forall p \in 1\dots M$ components, respectively, for the corresponding $\mathbf{o}_a$ and $\mathbf{o}_b$ vectors.
3. **Levenshtein distance:** This is a string metric for measuring the difference between two sequences [15]. Again, it uses order vectors $\mathbf{o}_a$ and $\mathbf{o}_b$. Informally, this distance measures the number of single position edition operations (insertion, deletion and substitution) to transform vector $\mathbf{o}_a$ into vector $\mathbf{o}_b$, if they are considered strings, rather than vectors. Algorithmically, it is defined by Eq. (3):

$$LV_{\mathbf{o}_a,\mathbf{o}_b} = \mathrm{lev}_{\mathbf{o}_a,\mathbf{o}_b}(M,M) \tag{3}$$

   with:

$$\mathrm{lev}_{\mathbf{o}_a,\mathbf{o}_b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j)=0,\\ \min\begin{cases}\mathrm{lev}_{\mathbf{o}_a,\mathbf{o}_b}(i-1,j)+1\\ \mathrm{lev}_{\mathbf{o}_a,\mathbf{o}_b}(i,j-1)+1\\ \mathrm{lev}_{\mathbf{o}_a,\mathbf{o}_b}(i-1,j-1)\\ \quad +\mathbf{1}_{(\mathbf{o}_{a,i}\neq\mathbf{o}_{b,j})}\end{cases} & \text{otherwise.}\end{cases} \tag{4}$$

   where $\mathbf{1}_{pred}$ is the indicator function, valued 0 or 1 depending on the truth value of the Boolean predicate *pred*.
4. **Freeloc distance:** Inspired in [9], this distance works on ternary vectors $\mathbf{t}_a$ and $\mathbf{t}_b$, and is defined by

$$FL_{\mathbf{t}_a,\mathbf{t}_b} = M - |C_{\mathbf{t}_a,\mathbf{t}_b}| \tag{5}$$

   where $C_{\mathbf{t}_a,\mathbf{t}_b}$ represents the set of pairs $\{(\mathbf{t}_{a,i},\mathbf{t}_{b,i})\,|\,i \in \{1,\dots,\binom{M}{2}\}$ and $\mathbf{t}_{a,i} = \mathbf{t}_{b,i} = +1$ or $-1\}$.
5. **Only-active distance:** It again uses ternary vectors $\mathbf{t}_a$ and $\mathbf{t}_b$, but considers only active pairs. It is defined as

$$OA_{\mathbf{t}_a,\mathbf{t}_b} = \frac{|S_{\mathbf{t}_a,\mathbf{t}_b}|}{|A_{\mathbf{t}_a,\mathbf{t}_b}|} \tag{6}$$

   where $A_{\mathbf{t}_a,\mathbf{t}_b}$ is the set of pairs $(\mathbf{t}_{a,i},\mathbf{t}_{b,i})$ with at least one value $\neq 0$, and $S_{\mathbf{t}_a,\mathbf{t}_b}$ is the subset of $A_{\mathbf{t}_a,\mathbf{t}_b}$ where $\mathbf{t}_{a,i} \neq \mathbf{t}_{b,i}$.
6. **Combined Freeloc and weighted-correlation:** This seeks to improve the Freeloc distance by incorporating the statistical dependence between the two vectors. It uses both order and ternary vectors and is defined by Eq. (7):

$$FP_{\mathbf{o}_a,\mathbf{o}_b,\mathbf{t}_a,\mathbf{t}_b} = FL_{\mathbf{t}_a,\mathbf{t}_b} \cdot (1 + PE_{\mathbf{o}_a,\mathbf{o}_b}) \tag{7}$$

7. **Weighted Freeloc distance:** Finally, this distance again combines order and ternary vectors, and considers the number of common monitors detected. It is defined by Eq. (8):

$$FC_{\mathbf{o}_a,\mathbf{o}_b,\mathbf{t}_a,\mathbf{t}_b} = FL_{\mathbf{t}_a,\mathbf{t}_b} \cdot \left(1 + \frac{|I_{\mathbf{o}_a,\mathbf{o}_b}|}{M}\right) \tag{8}$$

   where $I_{\mathbf{o}_a,\mathbf{o}_b}$ represents the set of common detected monitors and M is again the total number of them.

5. Experimental environment

We conducted all our experiments in a lecture building of 6000 square meters with 20 classrooms whose floor plan is shown in Fig. 4. Every classroom, except one, is equipped with a teaching computer connected to the university intranet (represented as green circles in Fig. 4). We use this computer to install monitoring software able to capture 802.11 traffic and transmit the relevant information via Gigabit Ethernet to a central server. The use of teaching computers as monitors has the twofold advantage of avoiding the ad-hoc deployment of new equipment and saving costs. The only additional hardware needed was an inexpensive off-the-shelf WiFi card installed in each of those computers to perform the monitoring. The required density of monitors to infer occupancy in a per-classroom basis is relatively low, so one monitor in each classroom is enough for our purposes.

[1] In fact, we tested a much larger set of distance metrics in our initial experiments. Here we have shown only those that we considered more illustrative in order to compare several different strategies. Of course, the best performers in terms of both accuracy and efficiency (or a trade-off between them, as we will illustrate in the posterior experimental results section), have been included in the final version of the paper.

Please cite this article in press as: P.E. Lopez-de-Teruel, et al., Practical passive localization system based on wireless signals for fast deployment of occupancy services, Future Generation Computer Systems (2017), http://dx.doi.org/10.1016/j.future.2017.09.022.

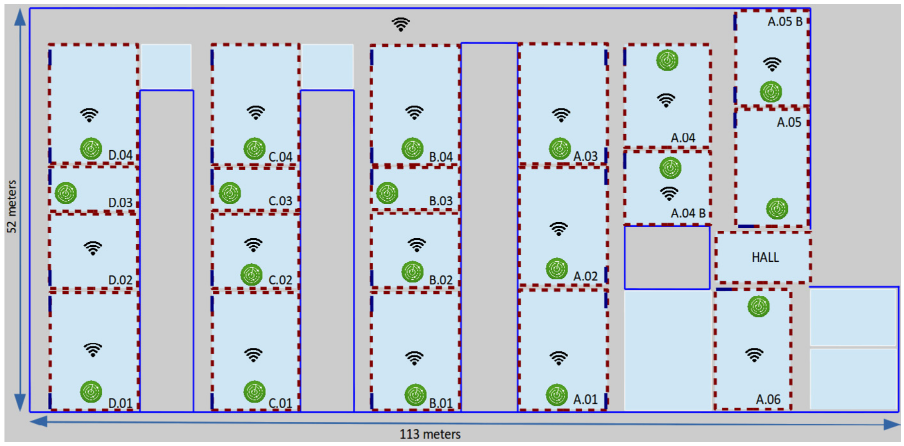

Fig. 4. Floor plan of the lecture room building used for our experiments.

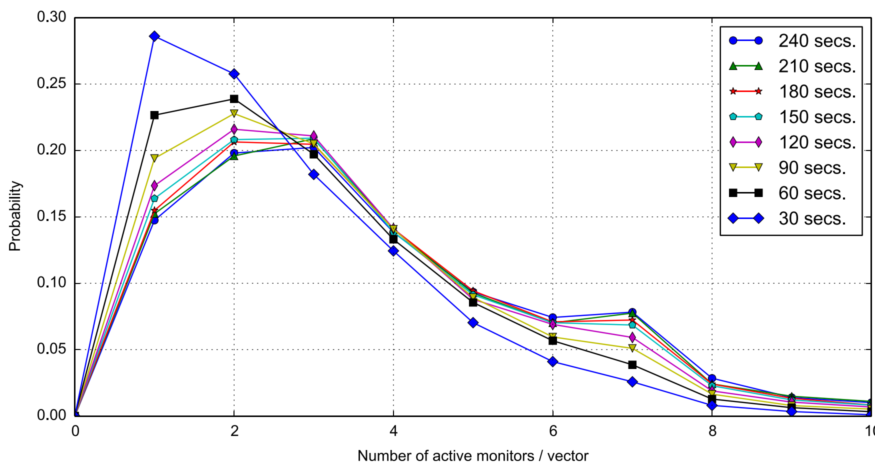

Fig. 5. Probability distributions of the number of monitors capturing signals from a device depending on the time window Δ.

We have defined zones of interest (represented by red dotted rectangles in Fig. 4) which refer to the different classrooms and the main hall. Our occupancy sensing system has to provide a characterization of the different passing users (i.e., students and professors) and their usual behavior. MAC addresses are key-hashed for privacy reasons.

5.1. *Characterization of the passive sensing*

As has been made apparent, our system is based on passive sensing, i.e. it does not require sensory information from the user devices. Instead, we rely on the data frames sent to the APs pertaining to the network infrastructure (represented as black WiFi icons in Fig. 4) or on the probe requests transmitted by the user devices. One well known issue in this kind of passive systems is that, due to both temporal and spatial sparsity of observations, it is not possible to guarantee a tracking performance similar to that of active systems. Therefore, and in order to characterize our particular environment, we conducted a statistical analysis to aid us in determining some important design and validation parameters which clearly distinguish us from typical active systems.

One of the most important design parameters of a passive system is the *time window* Δ. It should be noted that we do not have any control of the exact time when each device emits a frame to be captured by our monitors. Moreover, the monitors themselves could also be desynchronized when scanning the different channels. So, monitors just capture a set of individual raw RSSI samples per $(device, monitor)$ pair for irregularly sampled timestamps. In order to collect useful monitoring data for classification, the central server groups these individual samples by time intervals to obtain vectors including RSSIs for several monitors. Of course, there will be a clear dependence of the number of active (i.e., capturing) monitors for each vector on this Δ value. Fig. 5 shows different probability distributions of the number of monitors capturing signals from a device depending on this time window value, as obtained in our scenario. Of course, the greater the time window, the more likely a given device will be captured by more monitors, thus getting more informative vectors. On the downside, the greater the time window, the less precise will be our system for tracking moving devices. Nevertheless, people in a lecture room building tend to stay relatively static for long periods of time and Δ values of up to 2 or 3 min are assumable. We also observe that for values of $\Delta > 180$ s the number of active monitors per aggregated vector tends to stay stable.

We have also analyzed the variability of the RSSIs for different 802.11 frames captured by a monitor for a given device in the same time window interval. This variability strongly depends on the value of Δ, and clearly states how challenging the passive localization problem can be. Assuming again that the devices to be classified tend to be relatively static, and therefore the signal variability tends to be caused by occlusions, adjacent channel displacements, and uncontrolled capturing conditions, the system always takes the maximum RSSI obtained in the whole sampling interval, which should add more robustness to the order based

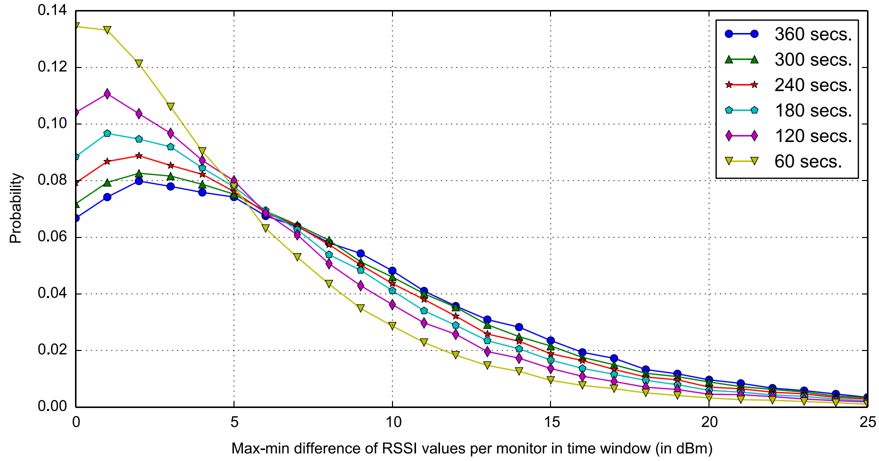

Fig. 6. Probability distributions of max–min RSSI values per monitor for a given device using several time window sizes.

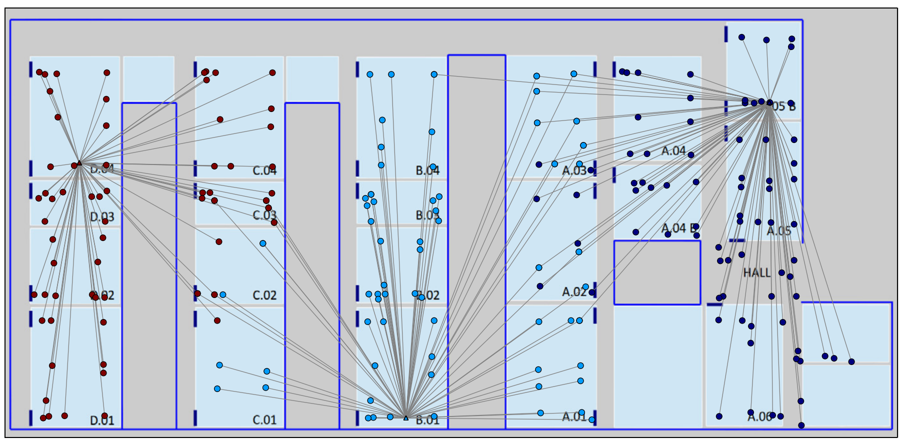

Fig. 7. Spatial sampling coverage for three arbitrary monitors (using training set).

metrics and techniques described in the next section. Fig. 6 shows some probability distributions of these max–min RSSI values per AP for several values of Δ. Again, the corresponding normalized histograms show that the variability in the max–min RSSIs values per AP tends to stabilize for time windows from 2 or 3 min on.

The spatial coverage of each monitor is another important value to take into account when designing passive localization systems. In our deployment, every monitor covers device locations up to 35–40 m from its corresponding position, or even up to 55 m in some cases (see Fig. 7, which is based on the dataset we obtained during the training phase, and that will be described in Section 6 below). Given our spatial distribution of monitors, the system has a minimum coverage of 5–6 monitors on every position of the building (and up to 12–13 on some specific, centered positions). Given an adequate Δ sampling time interval, this is more than enough to obtain quite meaningful raw measurements vectors.

Another important issue that must be evaluated is the relationship between the distance of a device to a given monitor and the corresponding received signal strength. This is illustrated in Fig. 8 (again based on the same training dataset). We observe that, though there is a clear negative correlation between RSSI and distance, the signal is in general very noisy, which translates into a relatively large variance in the direction perpendicular to the regression line. This explains the large degree of uncertainty regarding the direct use of lateration methods in unconstrained indoor environments, which justifies the need to create a finely trained fingerprinting map, which adapts much better to the peculiarities of each building.

Table 1
Training, validation and test datasets. The complete test environment had 21 zones and 19 monitors. The Galaxy, S3 and Nexus 5 are different smartphone models, Tab 2 and Infinitab are tablets, and the Asus Zenbook is an ultrabook laptop.

	Devices	*#Vectors*	*#Zones*	*#RSSIs/vec*	Δ_{base}
Training	Samsung Tab2	226	21	8.27	10 s
Validation	Samsung Tab2, Samsung Galaxy, Samsung S3	415	7	8.10	10 s
Test	Samsung Tab2 Samsung Galaxy Samsung S3 Asus ZenBook Nexus 5 Infinitab	101	8	5.35	90 s

6. Experimental results

6.1. Datasets description

We have performed a set of experiments in our environment using simple k-NN techniques based on the distance metrics presented above. The simple nature of our training process makes k-NN a good machine learning technique candidate, with a good balance between complexity, accuracy and execution time, given the relatively small size of the resulting typical training datasets [16]. More specifically, all our experiments were performed using the datasets described in Table 1. While both the

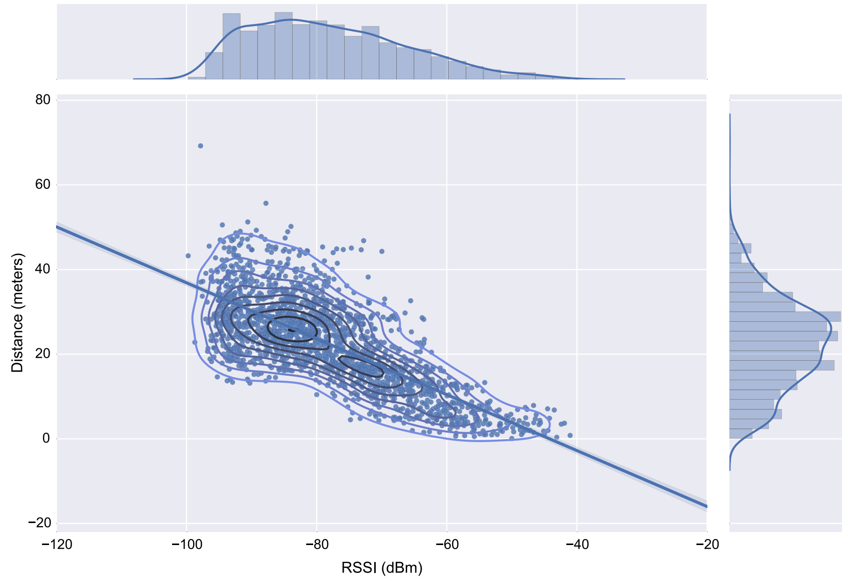

Fig. 8. Distribution of RSSI vs distance to monitor values for our testing environment (using training set). Marginal distributions of RSSI values and distances from captured devices to corresponding monitors appear on top and to the right of the figure, respectively.

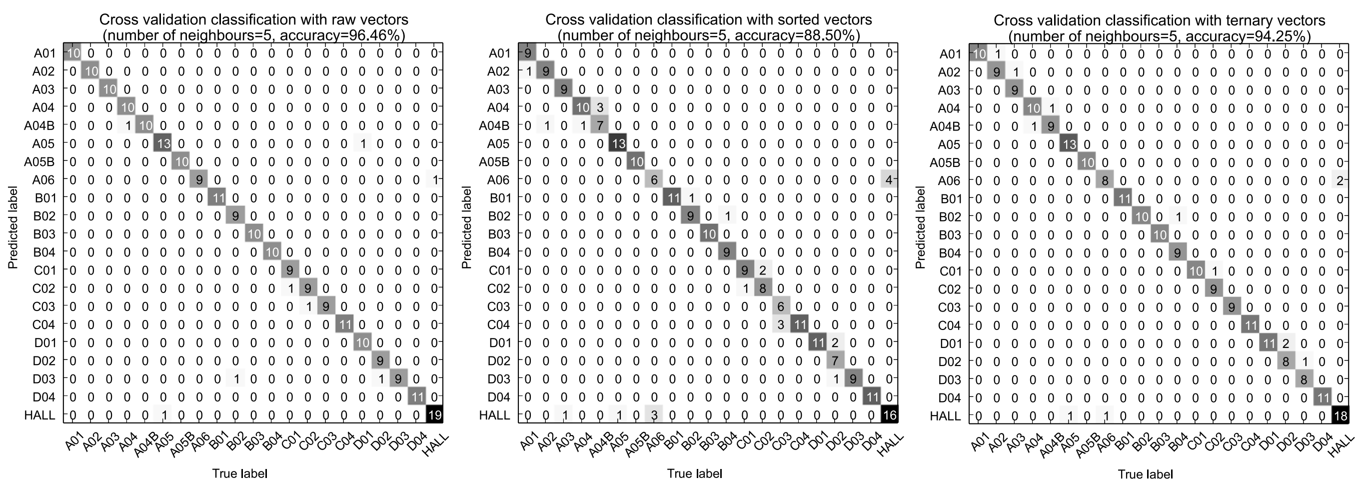

Fig. 9. Confusion matrices for classification using raw, sorted and ternary vectors when cross-validated (10-folded) on the training set.

training and *validation* datasets were obtained in active mode, using the lightweight survey process described in Section 3.2 (for example, the training dataset was composed of 5977 raw RSSI captured signals, which required about 1 h of training time to cover the whole building, and these measurements were aggregated into 226 vectors when using a sampling time period of $\Delta = 10$ s, as shown in the table), the *test* dataset was obtained from different mobile devices not running any specific localization software (just sending probe or data frames). This is a challenging set that includes six different mobile devices. The mean number of active monitors by vector (*#RSSIs/vec* column in table) is clearly lower than the training and validation datasets, due to the harder frame capturing conditions of the passive mode. The size of this dataset is also smaller, due to the need of a much larger Δ time window.

6.2. Training process evaluation

Given the lightweight nature of our training procedure, the first thing that we need to assess is the relative quality of our training dataset when cross-validated with itself. In order to evaluate this, we performed a standard k-fold cross validation [16], with $k = 10$ (that is, a random partition of the dataset in 10 subsets and using 9 of them to train a model, leaving the remaining one to test its accuracy). Although this method will, clearly, overestimate the obtained the classification accuracy that would be obtained in more realistic conditions, it is still useful to get an upper bound of the capacity of each method. Fig. 9 shows the obtained results when using raw, sorted and ternary vectors, respectively. The metrics used were Euclidean (Eq. (1)), weighted Pearson (Eq. (2)) and Freeloc (Eq. (5)), respectively. In all cases five nearest training neighbors were used to make the classification. As can be seen, any of the metrics behaves relatively well, thus ensuring that the fast training process does not imply a dramatic performance loss. It is also interesting to note that the raw vectors are those that behave the best (96.46% accuracy), given that the training and the testing conditions (i.e. device, environmental conditions, etc.) are almost identical (by definition of the k-fold evaluation). Still, ternary vectors using the Freeloc measure achieve quite a similar overall accuracy, 94.25%, which is also encouraging for us, given that these vectors throw all the absolute value RSSI information, and thus we expect them to be much more resilient to the device heterogeneity issue that we will evaluate later. Sorted vectors are

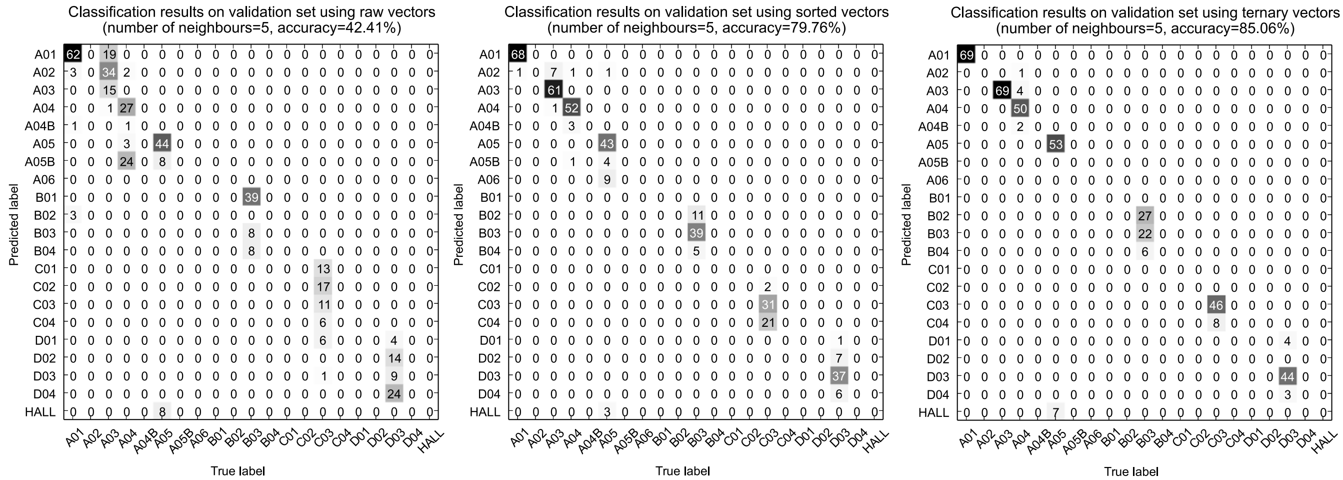

Fig. 10. Confusion matrices for classification using raw, sorted and ternary vectors when tested on the validation set.

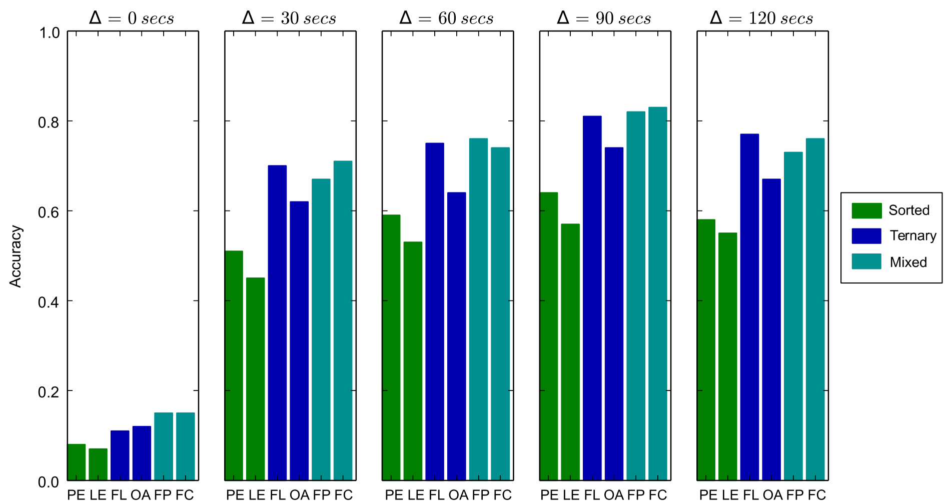

Fig. 11. Evolution of accuracy when varying Δ (in seconds), for a fixed value of $\delta = 0\ dBm$ (computed on test set). Sorted vector based metrics: weighted Pearson (PE) and Levenshtein (LE). Ternary vector based metrics: FreeLoc distance (FL) and Only active (OA). Mixed sorted and ternary vector metrics: Combined FreeLoc and weighted correlation (FP) and weighted FreeLoc distance (FC).

here the worst performers, with only 88.5% accuracy, since they are more affected than ternary vectors by the absolute information thrown away when building them from the raw measurements.

A more useful evaluation, however, should include an evaluation dataset completely different from the training dataset. In Fig. 10 we show the evaluation of the training phase when it is used to classify the validation dataset, using the same representations, metrics and number of neighbors as above. The difference here is that the validation data were obtained with three different devices, on different days and using different sampling paths than those used when performing the training. The results are again encouraging, with 85.06% accuracy for the ternary vectors, which clearly prove to be much more robust to device heterogeneity than raw vectors (which fall to a rather poor 42.41% accuracy), and again better than sorted vectors (which are still much better than raw vectors, with an overall 79.76% accuracy).

6.3. Analysis of the Δ and δ parameters of the passive system

We must nevertheless remember that the results illustrated in the above subsection are still obtained in the (easier) active conditions, that is, the located device is actively sampling the RSSIs of the beacon frames emitted by our monitors in AP mode. We must evaluate now the expected performance of the system when working in passive mode. But in order to do this, we must first determine adequate values of the main passive system parameters, that is, Δ and δ.

We will first analyze the influence of the Δ parameter. Fig. 11 shows the evolution of 5-NN classification accuracy on the test set when varying the passive time window interval (in seconds), for a fixed value of $\delta = 0$ dBm, and all the metrics described in Section 4.4 (except the Euclidean distance, which is clearly inadequate for heterogeneous devices, as shown in the previous subsection). We clearly observe how for too small values of Δ, the accuracy clearly degrades (independently of the metric), while $\Delta = 90$ s offers a good compromise between accuracy and time granularity of the resulting passive classification system. We also appreciate that the FreeLoc distance on ternary vectors (FL) and both the weighted FreeLoc distance (FC) and the combined FreeLoc and weighted correlation (FP) on mixed sorted and ternary vectors metrics are in general the best performing, with levels of accuracy of around 80%. These are very good results, taking into account that they were already obtained on the challenging test dataset of six different devices, with none of them executing any specifically dedicated software.

Another important issue was to determine a good δ parameter for our environment (see Section 4). Although when using the

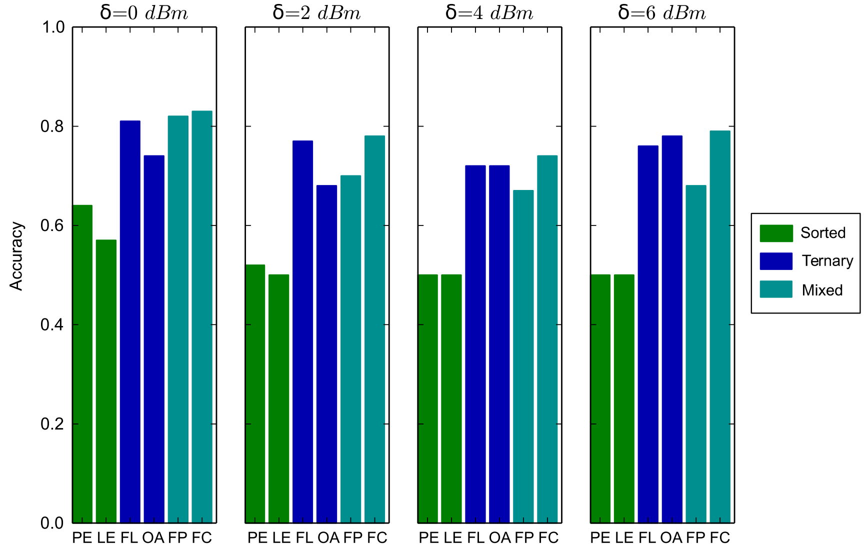

Fig. 12. Evolution of accuracy when varying δ (in *dBm*), for fixed value of $\Delta = 90$ s (computed on test set).

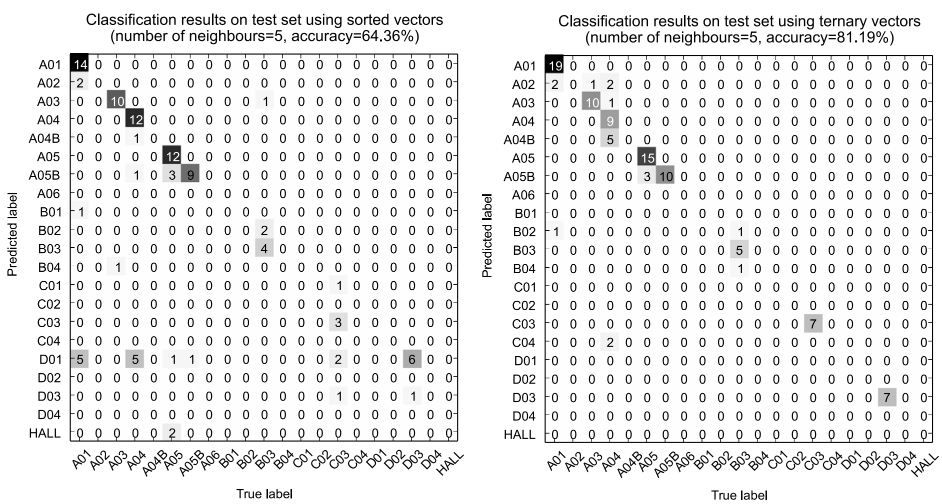

Fig. 13. Confusion matrices for classification using sorted and ternary vectors when applied to the test set (using $\Delta = 90$ s and $\delta = 0$ dBms).

independent validation set to cross-validate we obtained a best performing value of $\delta = 5$ dBm, for the test set (see Fig. 11) this parameter does not seem to have a clear influence, maybe due to the smaller number of monitors per sample, which tends to augment the difference between the available RSSIs, thus attenuating the influence of this parameter (see Fig. 12).

6.4. Passive system evaluation

Once we have determined adequate values of Δ and δ, we can now study in more detail the overall accuracy of the passive system. Given that there do not seem to be significant differences between the ternary vector representation using the FreeLoc metric (FL) and the more involved mixed representations and metrics (FP and FC), in the results shown hereinafter we will focus on the simpler (and thus more efficient) FL variant. In any case, we will also test the alternative representations of sorted vectors with Pearson metric (PE) and raw vectors with Euclidean metrics (EU), just for illustration purposes. Fig. 13 shows the detailed confusion matrices obtained when performing the classification of the test set using sorted and ternary vectors. Just as expected, the ternary vector approach proves clearly superior with 81.19% accuracy – much better than the 64.36% obtained by sorted vectors. Confusion matrix for raw vectors is not shown, as classification drops to a very poor overall accuracy of 18.81% (still well above the pure random classification rate of 1/21=4.76%, but clearly unacceptable for any practical purpose in this much more realistic scenario).

A classification accuracy around 80% is a good overall result for a passive system, but it would also be nice to get a visual idea of where exactly the remaining 20% classification errors go. Fig. 14 illustrates this. Here, we perform k-NN regression on the validation set, in order to get not just the zone, but rather the inferred (x, y) position when using the average position of the 5 nearest neighbors of each validation sample in the training set (using again the FL metric and $\delta = 5$ *dBm*). Circles represent the real device positions and triangles are the estimated positions. While most of the estimations go to the correct zone, the remaining errors almost always go to adjacent zones, thus demonstrating the relative robustness of the regression. The average distance error was 3.40 m, though we have to take into account here that this result was obtained using the actively obtained validation dataset. The reason is that, given the much larger time windows needed by the passive system, it would have been very time consuming to

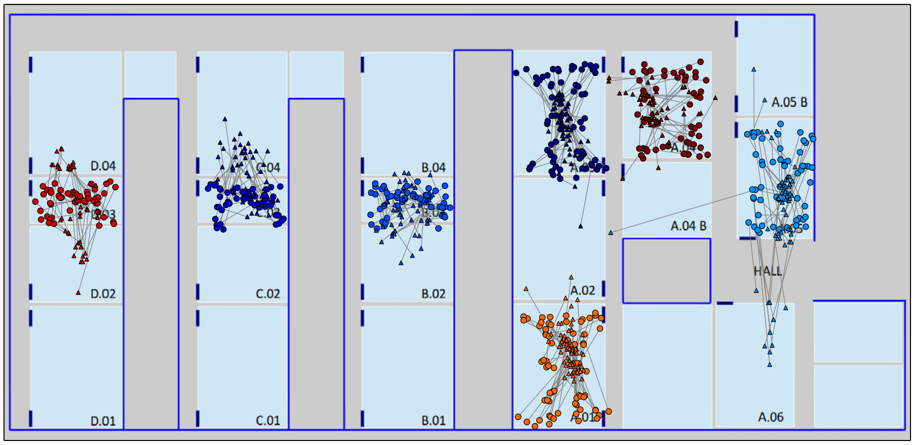

Fig. 14. Regression results on validation set, using FreeLoc metric on ternary vectors, $\delta = 5$ dBm, 5 neighbors (mean error = 3.40 m).

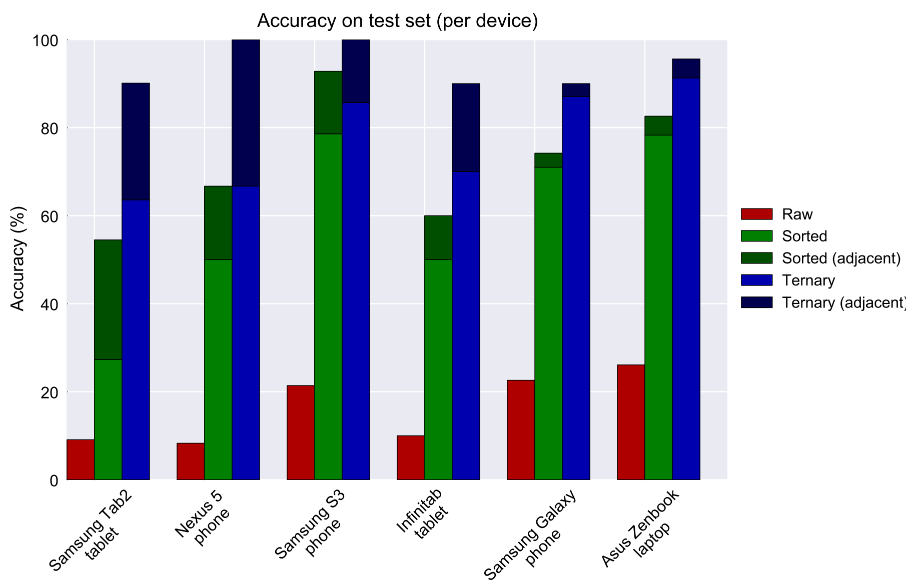

Fig. 15. Accuracy results per device in the test set (using Euclidean distance on raw vectors, weighted Pearson metric on sorted vectors, and FreeLoc metric on ternary vectors; $\delta = 5$ dBm and $k = 5$ neighbors in all cases).

obtain a passive regression ground truth test set of an acceptable size. Thus, this value of approximately 3.40 m should be only considered as a lower bound on the real error that would be obtained by the passive system when performing position regression. However, and as we have already justified throughout the paper, for occupancy purposes we are more interested in classification than in exact regression.

To end this subsection, in Fig. 15 we show again the classification results already illustrated in Fig. 13, although this time disaggregated by device model. We also show now the accuracy when we consider a location estimation wrongly assigned to an adjacent zone as approximately correct (note, of course, that this could be more or less adequate depending on the specific application of the occupancy sensing system). We observe that results are then well above ranges of 90%–95% accuracy, with slight variations depending on the specific devices. In general, we also observed that the laptop seems to be slightly better located than smartphones and tablets, and that some mobile devices (Samsung S3 and Galaxy smartphones) are better located than others (i.e. Galaxy Tab2 tablet), although in fact this could be just an artifact caused by the relatively small size of the testing dataset. A detailed study by type of device is therefore an issue that would warrant further research.

6.5. Fault tolerance to monitor failures

Finally, and given the special characteristics of our environment, where practically every relevant zone (mostly lecture rooms) has its own dedicated monitor, the reader might be wondering how a simple "zone with monitor with strongest RSSI" classification technique would perform. This is shown in Table 2. For some specific values of Δ, we can obtain even slightly better individual classification results (up to 90%). However, not only that type of classification would not be adequate for many other types of less structured environments, but also the resulting passive classification systems would be much less robust to sporadic monitor failures. Fig. 16 illustrates the resilience of our system to such events, which were simulated by removing a varying number of monitors when classifying the test set. Given that in these tests the hybrid metrics (FP and FC) tend to be slightly more robust to large number of failures than the others, in this figure we show again all the metrics described in Section 4.4 (again, except the raw vector based Euclidean metric, completely useless in the passive setting).

Table 2
Classification by zone with highest RSSI of the corresponding monitor.

Δ (s)	0	30	60	90	120	150	180	210	240
Accuracy	33.4%	81.0%	81.6%	88.1%	84.5%	89.8%	92.5%	90.7%	80.4%
#Samples	512	158	125	101	84	59	53	54	51

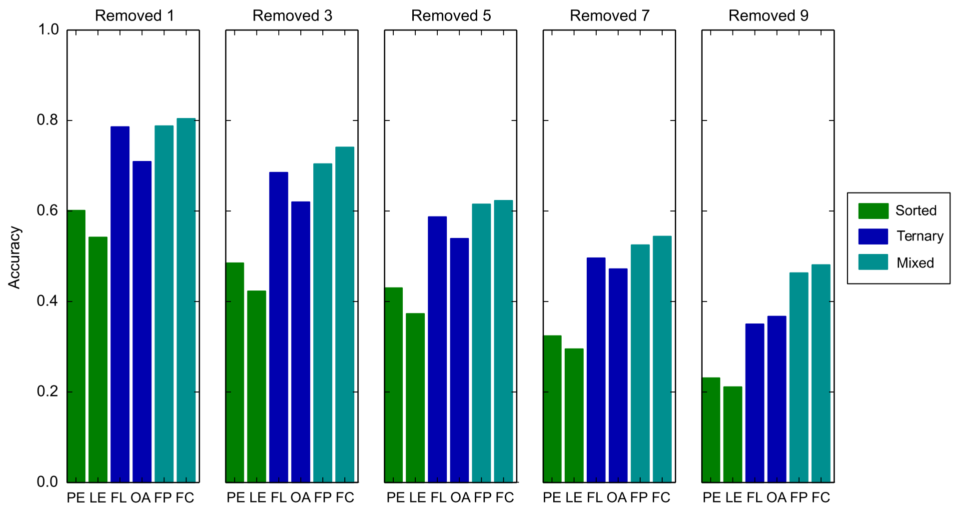

Fig. 16. Robustness to monitor removal.

7. Conclusions and future directions

We have presented a passive localization system based on wireless signals which is suitable to offer indoor occupancy information based services. Our lightweight training procedure, based on waypoints and real-time feedback, is a key stone to reduce the time required to deploy such kind of systems in a practical way. In addition, we have also performed an exhaustive experimental case study which is a valuable contribution to characterize the main features of our proposal. We studied the performance of several representation and metrics based on relative signal strength order (rather than raw measures) which are suitable to cope robustly with the device heterogeneity expected in typical unconstrained environments. We have shown that the ternary vector representation and its associated FreeLoc metric offers a well balanced solution to cope with the challenges posed by these kind of scenarios. Finally, we have tested several parameters that influence the estimation accuracy and we have also analyzed the implications of monitor failures.

Our results can be considered satisfying for occupancy estimation purposes on a per-zone classification basis. Finally, it is worth noting that the obtained classification success rates of around 80% were attained for individual vectors obtained for a time interval of just 90 s, without taking into account any additional type of temporal consistency. It is clear, therefore, that this accuracy could be easily boosted by using some simple probabilistic technique incorporating time evolution, such as a Hidden state Markov Model [16]. A thorough practical study of this additional feature, together with a higher level interpretation of the users behavior using clustering techniques will be the subject of our future research.

As an additional statement of direction, we are also analyzing whether the use of different radio technologies, like Ultra Wide Band (UWB) or Bluetooth Low Energy (BLE) or even a combination of both, may provide different results in terms of accuracy and practical deployment. More specifically, we are employing a dataset from an industry environment, which contains UWB and BLE signals obtained over 2 months from hundreds of tags carried by operators in a refinery.

Finally, we are also currently studying the application of dimensionality reduction techniques to RSSIs vectors in order to augment both the efficiency and robustness of the system [17].

Acknowledgments

This work was supported by the Spanish MINECO, as well as by European Commission FEDER funds, under grant TIN2015-66972-C5-3-R.

References

[1] J. Yang, M. Santamouris, S.E. Lee, Review of occupancy sensing systems and occupancy modeling methodologies for the application in institutional buildings, Energy Build. 121 (2016) 344–349.
[2] S. He, S.-H.G. Chan, Wi-fi fingerprint-based indoor positioning: Recent advances and comparisons, IEEE Commun. Surv. Tutor. 18 (1) (2016) 466–490.
[3] A. Ruiz-Ruiz, H. Blunck, T. Prentow, A. Stisen, M. Kjaergaard, Analysis methods for extracting knowledge from large-scale wifi monitoring to inform building facility planning, in: Proceedings of the IEEE International Conference on Pervasive Computing and Communications, PerCom, 2014, pp. 130–138.
[4] K. Christensen, R. Melfi, B. Nordman, B. Rosenblum, R. Viera, Using existing network infrastructure to estimate building occupancy and control plugged-in devices in user workspaces, Int. J. Commun. Netw. Distrib. Syst. 12 (1) (2014) 4–29.
[5] M. Kjaergaard, S. Lazarova-Molnar, M. Jradi, Towards a categorization framework for occupancy sensing systems, in: Proceedings of the 2015 ACM Sixth International Conference on Future Energy Systems, 2015, pp. 215–216.
[6] A. Musa, J. Eriksson, Tracking unmodified smartphones using Wi-Fi monitors, in: Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems, 2012, pp. 281–294.
[7] H. Koyuncu, S. Yang, A survey of indoor positioning and object locating systems, Int. J. Comput. Sci. Netw. Secur. 10 (5) (2010) 121–128.
[8] J. Park, D. Curtis, S. Teller, J. Ledlie, Implications of device diversity for organic localization, in: Proceedings of the IEEE International Conference on Computer Communications, INFOCOM, 2011.
[9] S. Yang, P. Dessai, M. Verma, M. Gerla, Freeloc: Calibration-free crowdsourced indoor localization, in: Proceedings of the IEEE International Conference on Computer Communications, INFOCOM, 2013.
[10] Z. Yang, C. Wu, Z. Zhou, X. Zhang, X. Wang, Y. Liu, Mobility increases localizability: A survey on wireless indoor localization using inertial sensors, ACM Comput. Surv. 47 (3) (2015) 54.
[11] C. Wu, Z. Yang, Y. Liu, W. Xi, WILL: Wireless indoor localization without site survey, IEEE Trans. Parallel Distrib. Syst. 24 (4) (2013) 839–848.

Please cite this article in press as: P.E. Lopez-de-Teruel, et al., Practical passive localization system based on wireless signals for fast deployment of occupancy services, Future Generation Computer Systems (2017), http://dx.doi.org/10.1016/j.future.2017.09.022.

[12] N.T. Nguyen, R. Zheng, Z. Han, UMLI: An unsupervised mobile locations extraction approach with incomplete data, in: 2013 IEEE Wireless Communications and Networking Conference, WCNC, IEEE, 2013, pp. 2119–2124.

[13] C. Gao, R. Harle, Easing the survey burden: Quantitative assessment of low-cost signal surveys for indoor positioning, in: Proceedings of the 7th International Conference on Indoor Positioning and Indoor Navigation, IPIN, 2016.

[14] K. Pearson, Notes on regression and inheritance in the case of two parents, Proc. R. Soc. London 58 (1985) 240–242.

[15] G. Navarro, A guided tour to approximate string matching, ACM Comput. Surv. 33 (1) (2001) 31–88.

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

[17] P.E. Lopez-de Teruel, O. Canovas, F.J. Garcia, Using dimensionality reduction techniques for refining passive indoor positioning systems based on radio fingerprinting, Sensors 17 (4) (2017) 871–895.

Pedro E. Lopez-de-Teruel was born in Lorca (Murcia, Spain), in 1970. He received his Ph.D. degree in Computer Science from the University of Murcia in 2003. At present, he is Associate Professor in the Computer Engineering and Technology Department at the same institution. His current research areas include computer vision and indoor localization systems, as well as machine learning algorithms and techniques applied to these fields.

Felix J. Garcia was born in San Javier (Murcia, Spain), in 1976. He is Associate Professor of Computer Networks in the Department of Computer Engineering of the University of Murcia, Spain. He received M.Sc. and Ph.D. degrees in Computer Science from the University of Murcia. His current research interests include security and management of distributed communication networks, as well as radio-based indoor localization systems.

Oscar Canovas was born in Barcelona (Spain), in 1975. He received his Ph.D. degree in Computer Science from the University of Murcia, Spain, in 2003. From 1999 to 2009 he was an Assistant Professor with the University of Murcia. Since 2009 he has been an Associate Professor at the same university. His research interests include information security, access control, payment systems, ubiquitous computing and indoor positioning.