14

Deployment of a Passive Localization System for Occupancy Services in a Lecture Building

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

FACULTY OF COMPUTER SCIENCE, UNIVERSITY OF MURCIA, MURCIA, SPAIN

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 due to the mainly dynamic nature of the presence of occupants, as referred by Yang et al. (2016). This implies a direct impact on, for example, heating, ventilation, and air conditioning (HVAC) systems. In the scenario that we present in this work, the main goal was to deploy a sensing system that would be connected to the HVAC system in order to take into account the information about occupancy in the different lecture rooms of a lecture building.

Based on our previous background and experience described in Lopez-de Teruel et al. (2017b), we developed an indoor localization system based on fingerprinting that was designed taking into account three main requirements: (a) there is no specific software component running on the mobile devices in order to scan and probably send signal observations to a particular localization server since, generally speaking, it is well known that users are reluctant to install apps that are battery consuming; (b) there are many different mobile devices to be tracked since they are personal devices belonging to university members and, therefore, they tend to generate signals with very different Received Signal Strength (RSS) and temporal patterns; and (c) training should be as quick and less time consuming as possible (Yang et al., 2015) since the proposal has to be scalable to a whole campus level when required.

Consequently, our system is characterized by some design decisions that address these requirements and that will be outlined in this chapter. 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

Geographical and Fingerprinting Data for Positioning and Navigation Systems. https://doi.org/10.1016/B978-0-12-813189-3.00014-9

of the users. Taking advantage of the fact that mobile devices periodically scan 802.11 channels for access points, which involve 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. In the scenario presented here, this did not necessarily imply the deployment of new elements, since we made use of existing computers in order to add the monitoring functionality. Second, our proposal is able to cope with the device heterogeneity problem identified by Park et al. (2011) 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 that must be covered by an operator during a nonexhaustive recording procedure performed by a training app. This lightweight process is suitable and feasible thanks to the adapted representation methods that we will define.

The practical information that we include in this work has been obtained from an already working system running in a lecture building of 6000 square meters with 20 classrooms. During a 18-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 rich dataset that is used in this chapter in order to illustrate some practical decisions that may arise when dealing with real scenarios. The main aim of the chapter, 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. In our opinion, 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. Therefore, we hope that the information provided here would be helpful for designers and developers with similar requirements.

2 Overview of the Localization System

As we previously mentioned, our system is based on passive indoor positioning. The following set of features intrinsically characterizes this kind of systems, in contrast with active ones:

- There is no need for special software installed on the mobile devices to be tracked.
- They require the deployment of special purpose devices, usually called *monitors*, whose number and specific positions are decided by the system designers.
- Estimations and other calculations are performed by an external element, instead of in the mobile devices themselves.

- Mobile devices are assumed to be heterogeneous, so a noncalibrated approach is required.
- Traffic patterns are unpredictable since there is no (or minimal) control over the mobile devices.
- The granularity of the localization tends to be coarser, based on zone classification, rather than exact position regression.

2.1 Deployment Cycle

Fig. 1 shows an overview of the deployment cycle of our passive localization system. First of all, the administrators decide a set of interest zones Z_i in the target scenario (1), and the system designers determine the number of monitors M_j to deploy and their respective initial positions (2). In a broad sense, a monitor is any hardware element running software able to capture wireless (802.11) traffic and export the relevant information of this captured data to a central server. In this particular scenario, a monitor is a standard PC, usually employed for teaching purposes, with a dedicated WiFi adapter configured to capture wireless traffic in promiscuous mode. Monitors rely only on capturing the frames transmitted by the mobile devices as part of their usual connections or active scanning

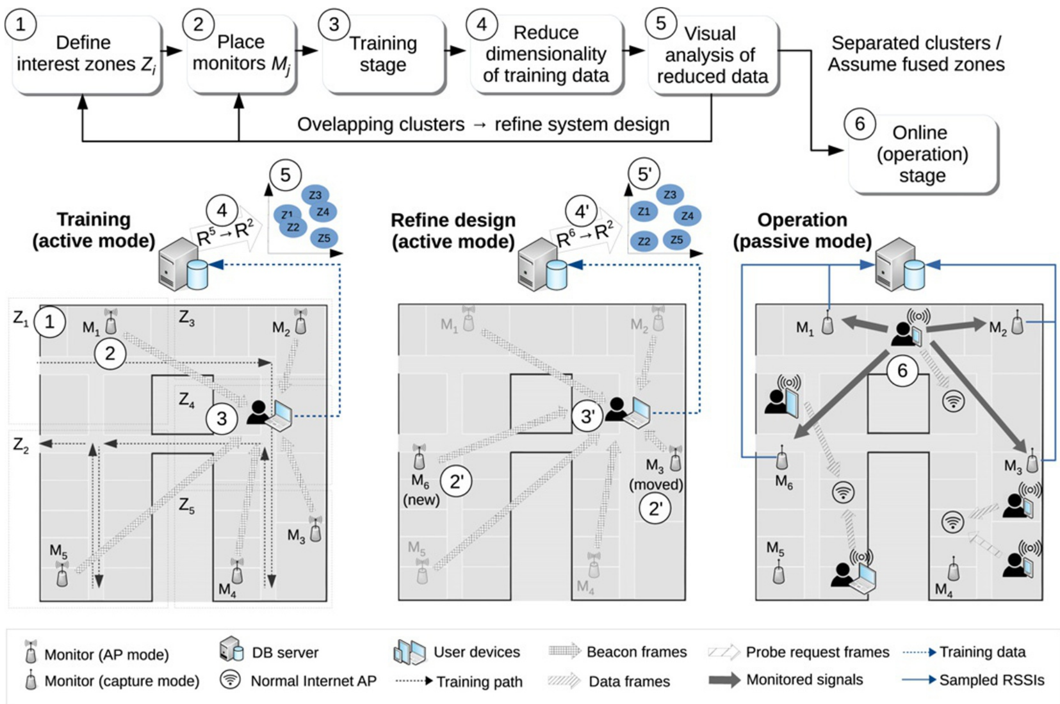

FIG. 1 Overview of the refinement process.

periods. In this scenario we do not make use of prompting techniques like Musa and Eriksson (2012) to increase the number of packets received from them.

Then, as in any fingerprinting-based technique, a training phase which involves a manual site survey process is performed (3), where an operator carrying a device running specific software follows a given walking path. The specific details of this training software are provided in Section 2.2. During this phase monitors, instead of capturing traffic, are configured to broadcast beacon frames (i.e., we set them in *access point mode*, for 802.11). The RSSIs observed by the training device for those frames are then recorded by the software, and used to save the corresponding geotagged observations. Each of these observations is thus formed by a vector of RSSI measures of dimension equal to the number of monitors, plus a ground truth (x, y) position in a coordinate system locally defined in the scenario. Using this position, the specific zone of each vector can also be easily determined.

In this particular scenario we also make use of dimensionality reduction techniques to know whether the current position of the monitors allows for correct classification of users location according to the desired zone partition. For additional information about this stage, the reader can refer to our work (Lopez-de Teruel et al., 2017a).

Once system designers have decided that the current monitor positions and distribution of zones are adequate, and using the corresponding training data as the final fingerprint map, monitors can be switched to *capture mode*, and the system can start its operation (6). We have verified that the user mobile devices (smartphones, tablets, laptops, etc.) being monitored show a wide variety of hardware, WiFi interfaces, antennas, operating systems, and the like. Moreover, some of them are connected to local available Internet access points, and thus generating normal data frames traffic, while others are not currently connected, and thus just generating sporadic probe frames in order to request information from available access points. Consequently, they produce signals with very different strength and temporal patterns. Note also that training vectors are based on RSSI of frames emitted from our monitors and captured by the training device, while in the operational phase just the opposite occurs, that is, RSSIs are obtained for frames emitted by user devices and measured by the monitoring elements. Though the respective RSSIs will be clearly correlated, as determined by the device-monitor distance, in general these measures will not have to be exactly the same. This asymmetry adds another source of heterogeneity to the related RSSIs fingerprints. For all these reasons, heterogeneity had to be addressed in our system, by means of some design decisions regarding data representation that will be explained later.

2.2 Training Approach

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.

Training observations are tagged (x, y) using a locally defined coordinate system, and those 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. 2 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 geotagged using the coordinates where the operator has to be physically at that moment. All the operator has to do is to follow the path shown as accurately as possible and remain still at the designated waypoints (smaller 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. In the particular scenario of the building being described here, the required time for training was around 1 hour for the whole building (approximately 4000 square meters of the total 6000 square meters were mapped). The training was divided in five different walking paths of approximately 10 minutes each, one of them shown as example in Fig. 2.

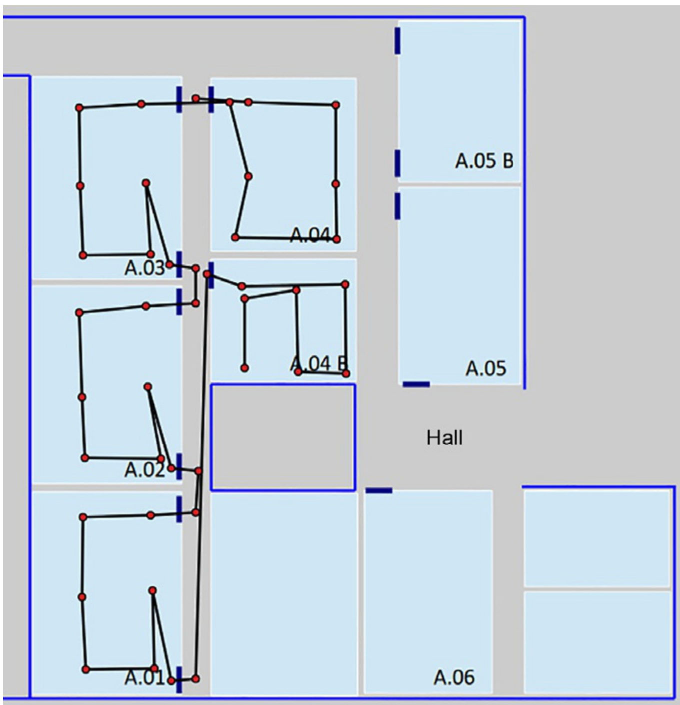

FIG. 2 User interface of the training application based on waypoints.

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 see, 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.

2.3 Data Representation

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 1–3 minutes), 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 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 (Nearest Neighbor) classifier, while still being suitable for heterogeneous devices.

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.

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. Additional information about these representation methods and their related distance metrics can be found in our work (Lopez-de Teruel et al., 2017b).

3 Real Scenario: Occupancy for a Lecture Building

3.1 Overview

Our system is deployed in a lecture building of 6000 square meters with 20 classrooms whose floor plan is shown in Fig. 3. Every classroom, except one, is equipped with a teaching computer connected to the university intranet (represented as circles in Fig. 3). We use these computers 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 dual-band WiFi card installed in each of those computers to perform the monitoring. According to our study based on dimensionality reduction techniques published in Lopez-de Teruel et al. (2017a), one monitor in each classroom is enough for our purposes.

Several zones of interest (represented by dotted rectangles in Fig. 3) were defined, which refer to the different classrooms and the main hall. Our occupancy sensing system provides a characterization of the different passing users (i.e., students and professors) and their usual behavior.

In order to perform its scanning, each of our monitors simply scans the different 802.11 channels (both 2.4 GHz and 5 GHz) 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

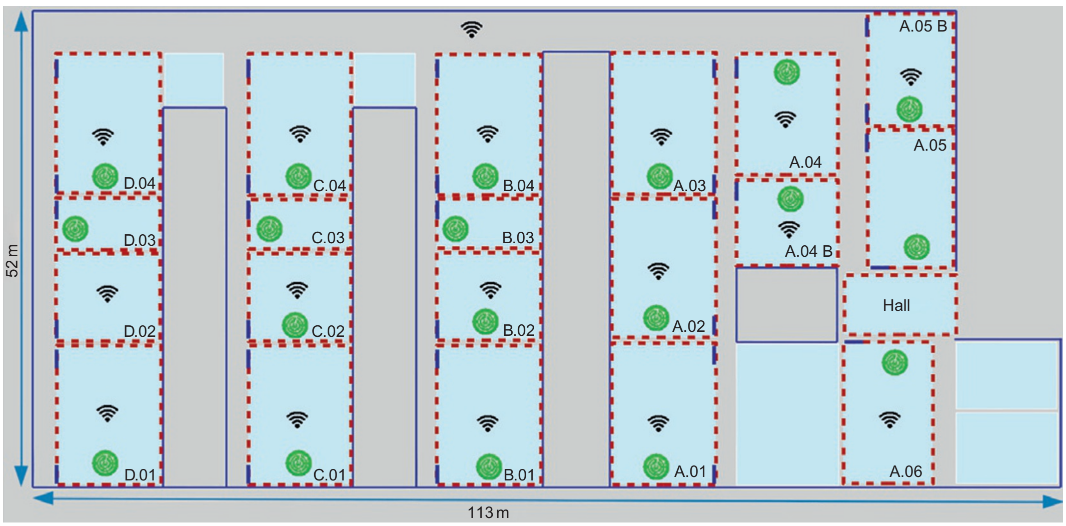

FIG. 3 Floor plan of the lecture room building.

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.

A central server is in charge of hosting the localization engine itself. Another central element of this engine is 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 is also in charge of running the localization software responsible for calculating occupancy and positioning information when required. The server provides a RESTful API which is used by higher-level occupancy services.

3.2 Characterization of the Passive Sensing

As has been made apparent, we rely on the data frames sent to the APs pertaining to the network infrastructure (represented as black WiFi icons in Fig. 3) 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 deployment and validation parameters which clearly distinguish us from typical active systems.

One of the most important deployment 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. 4 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, as we have verified, 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 minutes are assumable. We also observe that for values of $\Delta > 180$ seconds the number of active monitors per aggregated vector tends to stay stable.

We have also verified the influence of the Δ parameter in the accuracy of the location estimations. Using data representations based on order and a 5-NN (Nearest Neighbors) classifier, we analyzed the classification accuracy on a test set composed by samples obtained with six different devices when varying the passive time window interval. We clearly observed how for too small values of Δ, the accuracy clearly degrades, while $\Delta = 90$ seconds offers a good compromise between accuracy and time granularity of

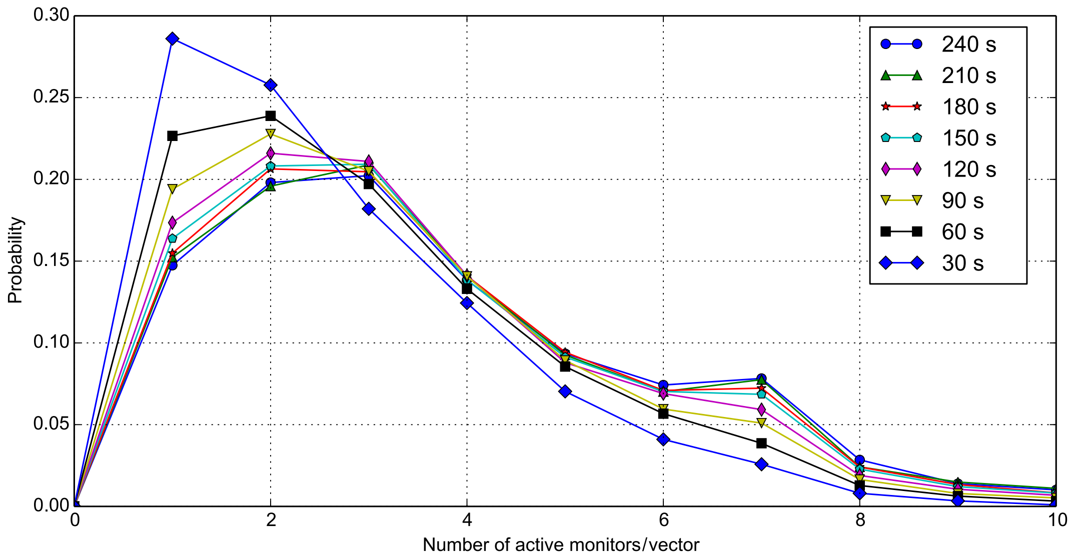

FIG. 4 Probability distributions of the number of monitors capturing signals from a device depending on the time window Δ.

the resulting passive classification system. Accuracy level is around 80%. These are very good results, taking into account that they are obtained on a challenging test dataset of six different devices, with none of them executing any specifically dedicated software.

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. 5, which is based on the dataset we obtained during the training phase). 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 was enough to obtain meaningful raw measurements vectors.

3.3 Considerations About Accuracy

Fig. 6 shows classification results in relation to the test set, this time disaggregated by device model. We also show 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 testing dataset.

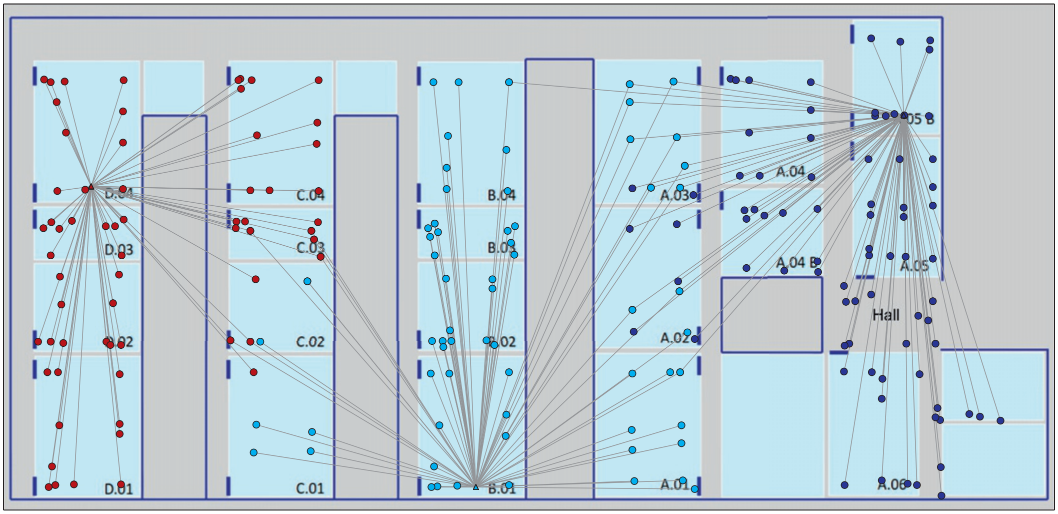

FIG. 5 Spatial sampling coverage for three arbitrary monitors (using training set).

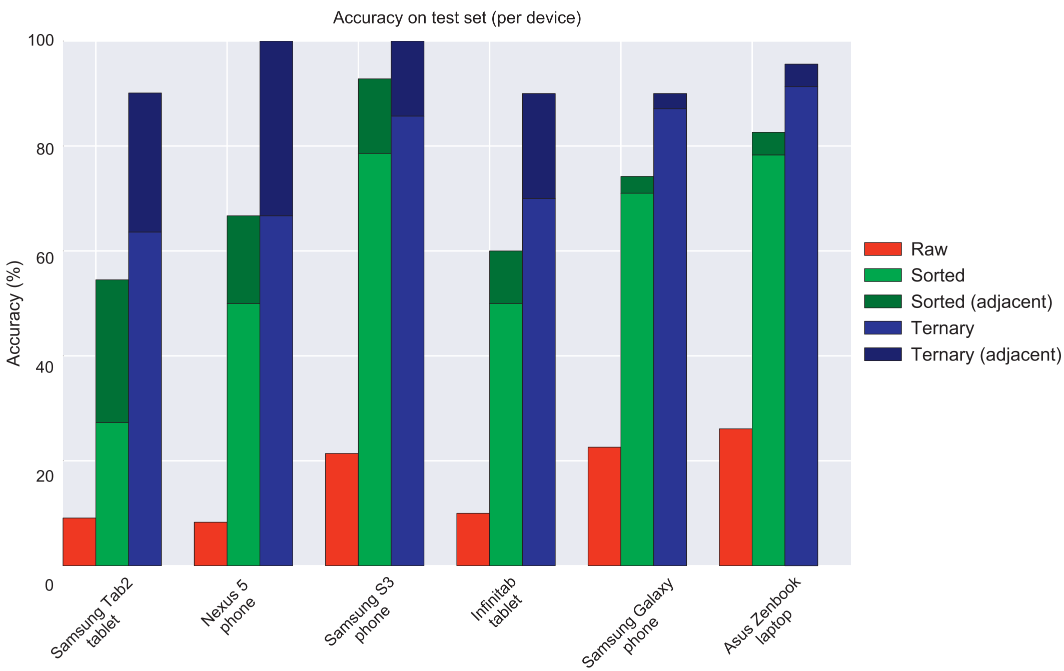

FIG. 6 Accuracy results per device in the test set. Note that overall accuracy gets slightly increased if we consider adjacent zones' errors as also correct, which could be acceptable or not depending on the final application of the occupancy system.

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. Our analysis determined that, 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. We have tested the resilience of our system to such events, by removing a varying number of monitors when classifying the test set. Using ternary vectors, it is possible to obtain up to 70% classification accuracy even after removing three monitors.

3.4 Occupancy Services

As we have already commented, the localization system provides a RESTful API that can be used to develop specific location-based services. In this particular scenario there are several services for occupancy purposes. We present two examples that could be meaningful for illustration purposes. On the one hand, as Fig. 7 shows, it is possible to obtain occupancy heat maps, which are useful to show real-time information, to analyze occupancy during a given period of time, and, in our case, to provide information to the HVAC system. In the example shown, the system provides the number of devices in every

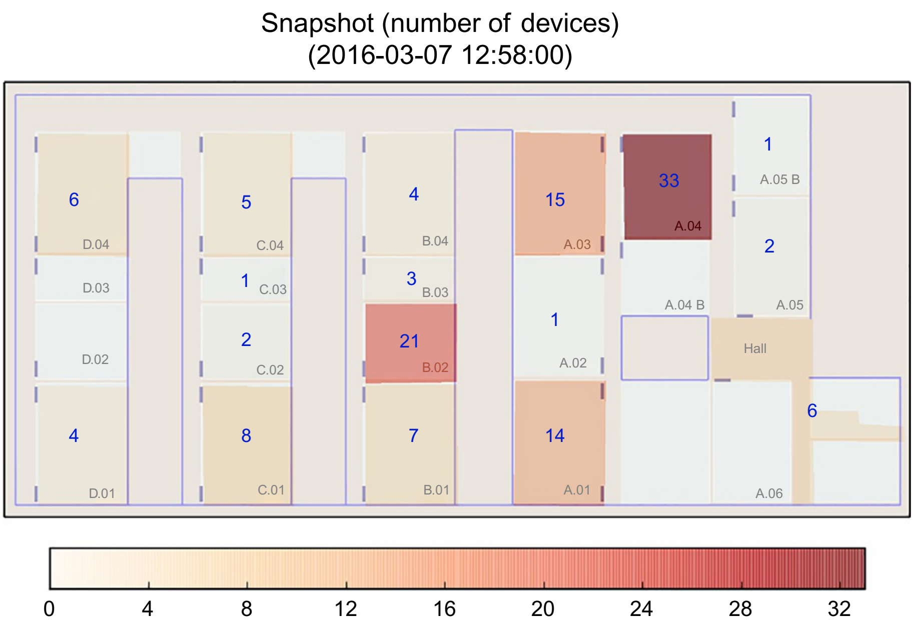

FIG. 7 Example heatmap obtained during the online operation of the system.

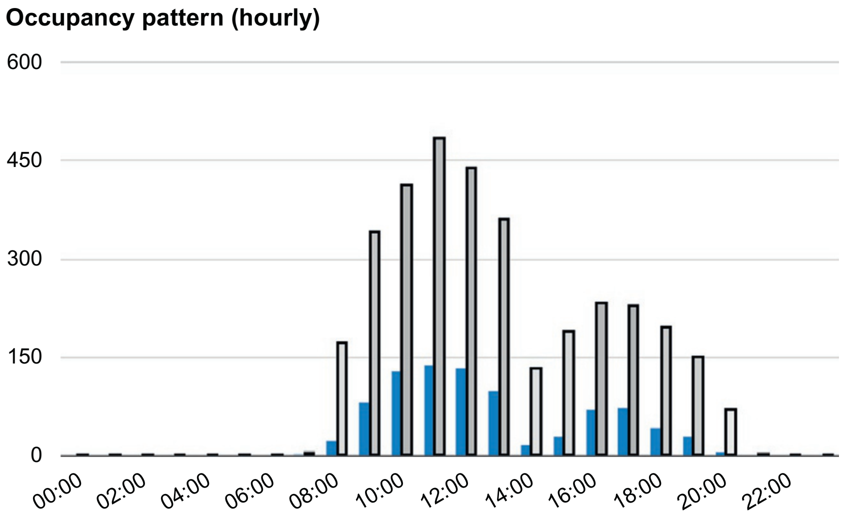

FIG. 8 Temporal occupancy analysis for a given period (one day in the example).

zone of the building (this is related with the potential number of occupants if we assume that every user has a single device). 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. 8 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 is 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. For every time slot we display two different bars. Right 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 left bars, which is directly related to the amount of time that those devices remained in that zone. This information is measured in $devices \times hour$ 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 $device \times hour$ unit, while another that only stayed there for, say, 6 minutes, 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 Conclusions

Occupancy sensing systems have become the subject of much attention recently due to the increasing number of sensors and devices with wireless connectivity. As Christensen et al. (2014) showed, many scenarios follow a similar approach to the one we present here, that is, to infer information from existing infrastructure elements.

As we have shown in this work, our passive localization system based on wireless signals is suitable to offer indoor occupancy information-based services and it addresses the requirements imposed for the deployment. First, 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. Second, we use data representations based on relative signal strength order (rather than raw measures) to cope robustly with the existing device heterogeneity in an environment like the one present here. Finally, we have verified several parameters that influence the estimation accuracy and we have presented some existing location-based services already running for occupancy purposes.

As a statement of direction, we have already initiated a research line focused on a more detailed characterization of the environment and we are applying clustering techniques to identify sets of users and behaviors that might provide additional information for occupancy systems. More information is available in Lopez-de Teruel et al. (2017c).

References

Christensen, K., Melfi, R., Nordman, B., Rosenblum, B., Viera, R., 2014. Using existing network infrastructure to estimate building occupancy and control plugged-in devices in user workspaces. Int. J. Commun. Netw. Distrib. Syst. 12 (1), 4–29.

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

Lopez-de Teruel, P.E., Garcia, F.J., Canovas, O., 2017b. Practical passive localization system based on wireless signals for fast deployment of occupancy services. Futur. Gener. Comput. Syst. https://doi.org/10.1016/j.future.2017.09.022.

Lopez-de Teruel, P.E., Garcia, F.J., Canovas, O., Gonzalez, R., Carrasco, J.A., 2017c. Human behavior monitoring using a passive indoor positioning system: a case study in a SME. Proc. Comput. Sci. 110, 182–189. 14th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2017)/12th International Conference on Future Networks and Communications (FNC 2017)/Affiliated Workshops.

Musa, A., Eriksson, J., 2012. Tracking unmodified smartphones using Wi-Fi monitors. In: Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems, pp. 281–294.

Park, J., Curtis, D., Teller, S., Ledlie, J., 2011. Implications of device diversity for organic localization. In: Proceedings of the IEEE International Conference on Computer Communications (INFOCOM).

Yang, Z., Wu, C., Zhou, Z., Zhang, X., Wang, X., Liu, Y., 2015. Mobility increases localizability: a survey on wireless indoor localization using inertial sensors. ACM Comput. Surv. 47 (3), 54.

Yang, J., Santamouris, M., Lee, S.E., 2016. Review of occupancy sensing systems and occupancy modeling methodologies for the application in institutional buildings. Energy Buildings 121, 344–349.