

Improving Wi-Fi based localization using external constraints

Gloria Ciavarrini, Francesco Marcelloni, Alessio Vecchio

Dip. di Ingegneria dell'Informazione, Università di Pisa, Pisa, Italy

gloria.ciavarrini@for.unipi.it, francesco.marcelloni@unipi.it, alessio.vecchio@unipi.it

Abstract—Wi-Fi based localization enables detection of users' position in indoor spaces by means of wireless networking infrastructure. The positive aspects of this solution include the reuse of already deployed systems and thus its reduced costs. On the negative side, Wi-Fi based localization is not particularly accurate, because the common operating conditions are far from the ideal ones. We propose to use external constraints for improving the accuracy of Wi-Fi based localization. A set of known schedules is used to restrict the estimated position of the user to a single room. The schedule for a given user is automatically selected from a set of possible ones by observing user's movements with coarse-grained resolution.

Keywords-Indoor localization, Wi-Fi, smartphone.

I. INTRODUCTION

Indoor localization of users within large buildings has been an active research area for the last decade. In fact, knowing the position of users in indoor spaces enables a wide range of applications, like providing navigation towards a given target or sending suggestions/advertisements based on users' positions [1]. In general, position is considered one of the key elements of context as it provides information useful to deliver customized services and to infer other properties of the environment where applications are executed. More recently, indoor localization started being considered as a fundamental component of smart building solutions. If users' positions are known, it is possible to tune heating and conditioning systems based on the occupancy levels, thus saving energy. Indeed, for instance, it is useless to heat a room that is not currently in use; on the other hand, crowded areas have to be conditioned more intensively to maintain the temperature within the comfort zone.

The majority of indoor localization solutions proposed so far are smartphone-based and rely on Wi-Fi: the smartphone operates as a sensing and computing element, whereas visible access points (APs) can be used to determine the position of the user [2]. In some cases the signal strength of APs in the nearby is used to increase localization accuracy [3], [4]. Other solutions integrate Wi-Fi based information together with data collected by means of other sensors commonly available on smartphones. For example, some systems use the barometer to detect floor changes [5]. Others implement dead-reckoning mechanisms using the accelerometer and/or gyroscope [6], [7], [8], [9]. Many Wi-Fi based localization solutions require a preliminary setup phase: the wireless fingerprint of the considered building is collected and stored within a database; then, in the operational phase, the wireless properties observed by a

user are compared to stored information for computing his/her position. Unfortunately, this approach is rather expensive as fingerprinting is labor intensive and must be repeated periodically to detect possible changes in the infrastructure. Moreover the wireless fingerprint is influenced by the presence/absence of crowds, thus capturing all the details is almost impossible.

Several Wi-Fi based localization systems operate according to a simplified model of the environment. For example, one of the APs within the communication range of the user's smartphone is selected as reference using some criteria (such as the AP with strongest signal) and the user is then co-located with the reference AP. In other cases, there is a one-to-one mapping between APs and rooms: the AP that provides the strongest signal in a given room is associated to that room. Subsequently, when a user is connected to an AP, then the associated room is inferred as the user's location. On the negative side, these assumptions make the localization system prone to errors. We observed that a single AP can be able to cover a relatively large number of rooms, thus one-to-one mapping cannot be considered as a viable option. In particular, in our experimental analysis we found that the signals of several APs get received in 10-20 rooms (this obviously depends on the structure and topology of the considered building).

In this paper we propose the use of external constraints for removing localization ambiguities. Our technique relies on the presence of a well-known schedule about the possible movements of users within a building (or a set of buildings) to determine the room where a user is located. In detail, our reference scenario is an academic campus where students move from one classroom to another for attending lectures. The schedule of lectures is known and can be used to remove localization ambiguities of a Wi-Fi based system. The output of the localization system could be used to efficiently tune the conditioning system or to understand the real occupancy levels of rooms. We believe that the proposed approach, with some marginal changes, can be applied in a number of other scenarios, e.g. in a hospital where personnel and patients move according to a schedule that can be derived from the medical information system.

The remaining of this paper is organized as follows: in Section II the most relevant work related to indoor localization in smart building solutions and constraint-based localization is summarized; in Section III the principle of operation and the main steps of the proposed technique are described; Section IV illustrates a case study where the method has been tested;

conclusions are drawn in Section V.

II. RELATED WORK

Heating, ventilation, and air conditioning (HVAC) systems are responsible for a large fraction of the energy consumption in commercial and residential buildings. Reports highlight that commercial buildings, which account for approximately 20% of USA global energy demands, could significantly reduce their footprint through adoption of power-saving strategies [10], [11]. In [12], the relation between building occupancy in the MIT campus, estimated by counting the number of accesses to Wi-Fi networks, and energy demand is studied. This information is then used to tune the HVAC system according to the number of users in the monitored spaces. Detailed information about the number of occupants in a room can also be used, according to the authors, to “weight” the thermostat demand, allowing larger groups of people to change the room temperature with greater control with respect to small groups.

Sentinel is a system for HVAC actuation based on occupancy information collected via existing Wi-Fi infrastructure [13]. Authentication logs of users’ smartphones are used to coarsely localize them within the considered buildings. As highlighted, a single AP generally covers an area that includes several rooms and/or offices, thus it is difficult to understand which is the actual location of the user. In Sentinel some spaces are marked as “personal” (e.g. an office), whereas the remaining spaces are tagged as “shared”. This external information is used to increase localization accuracy: a personal space cannot be occupied unless the owner is present in the space; similarly, whenever a user is detected by an AP that covers the user’s personal space, then the user is supposed to be located in his/her personal space. The authors demonstrate that large energy savings can be achieved by using occupancy information, even in the presence of some inaccuracies.

The use of constraints to increase the accuracy of Wi-Fi based localization has been discussed also in [14]. In particular, the trajectory of the user in an urban area is estimated using a number of passive Wi-Fi monitors, and taking into account the constraints introduced by road segments, intersections, and buildings. The method is inspired by Viterbi map-matching techniques, which are used to find the maximum probability path as a sequence of hidden states in the corresponding Markov model.

In [15], a method for extracting knowledge from a large Wi-Fi monitoring experiment is presented. The aim of the described system is to provide information that can be useful for managing large buildings, such as the position of users, and density and flow of crowds. The analysis is contextualized in a large hospital complex composed by 22 buildings. As far as localization is concerned, the proposed approach estimates the position of users using the centroid iteration algorithm; then the estimated position is snapped to the location of the nearest AP. Interestingly, the authors also classify users in a number of behavioral roles that are relevant to the considered scenario (patients, employees, visitors) using features extracted from Wi-Fi traces. This is somehow the converse of our procedure:

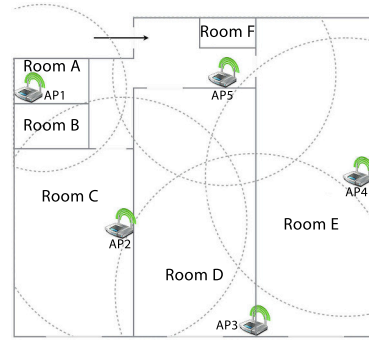


Fig. 1: Example of map

we use external constraints, inferred from the classification of users, to restrict the uncertainty level of localization. Also in our case the classification of users is dynamical.

ARIEL is a Wi-Fi based localization system that operates at room level [16]. Room fingerprints are automatically learned based on occupants’ indoor movements. The accelerometers commonly available on smartphones are used to detect users’ motion (sampling at low frequency to reduce the consumption of energy). Wi-Fi signals collected in stationary sessions are then aggregated using a zone-based clustering technique.

The problem of indoor localization with room-level granularity has been studied also in [17], where received signal strength of visible APs is used. The system requires a training phase to collect the Wi-Fi fingerprints of considered spaces. To reduce the effort, also the training phase is organized with room-level granularity and does not require operators to move according to a grid (as done in other systems).

III. METHOD

In this paper the use of external constraints for improving Wi-Fi based localization is proposed. In particular, a set of constraints derived from a well-known schedule is used to restrict the position of the user, roughly calculated from Wi-Fi infrastructure, to a single room.

The proposed method assumes that the users within a building (or a set of buildings) move according to a well-known schedule (as it happens with the employee of a hospital or the students in campuses). The identity of single users is not known, thus it is not possible to directly infer from Wi-Fi data the schedule for the currently observed user. According to the proposed method, first a possible schedule for a given user is selected by observing his/her movements, then such schedule is used to restrict the position of the user to a single room from a set of possible rooms.

A. Principle of operation

The use of Wi-Fi as mechanism for locating users in indoor environments has been extensively studied during the last years as a low-cost solution, since Wi-Fi infrastructure is already available in almost all buildings and thus there is no need to deploy additional equipment.

The strawman approach, for Wi-Fi based localization, can be described as follows. Initially, a Wi-Fi map of the environment is created off-line. Such map reports the visible APs

for a number of sampled positions (sampling can be carried out with different granularity levels, depending on the required precision and the effort that can be allocated for creating the map). The minimum effort is achieved by collecting a single sample for every room of the building; in such case the map basically reduces to a list of visible APs for every room. Then, when the system is operational, localization of a given user takes place by comparing the list of APs in proximity of the user to information contained in the map. In particular, the user could be located in one of the previously sampled positions where currently visible APs have been reported. If the map has room-level granularity, the result is the set P of possible rooms that are compatible with the APs surrounding the user. For example, let us consider the scenario depicted in Figure 1: if the user is within communication range of AP2 and AP3, then such user could be located either in room C or D ($P = \{C, D\}$); if the only visible AP is AP1, then the user could be located in room A, B, or C ($P = \{A, B, C\}$).

Obviously, the smaller the cardinality ($|P|$) of the set of possible rooms is, the better the localization accuracy is. Unfortunately, in some circumstances, this set can be quite large: we experimentally verified that a single AP can cover more than 20 rooms (this clearly depends on building structure, layout, Wi-Fi technology, etc.). As a consequence, the localization accuracy of an approach like the one described above can be quite unsatisfactory.

We devised a technique that is able to reduce the ambiguity of the strawman approach by including external constraints in the localization process. In particular, the technique excludes from the set of possible rooms those rooms that are not compatible with the schedule of the user at a given time. The schedule of the user is automatically learned by observing his/her movements during a period of time. Let P_1, P_2, \dots, P_n the sets of possible rooms computed at different times, and let $S = \{s_1, s_2, \dots, s_m\}$ be the set of known schedules. Each schedule s_l defines the position of users who follow such schedule as a list of time intervals and associated rooms. The inferred user's schedule \hat{s} , with $\hat{s} \in S$, is computed by selecting the "best match" as follows:

$$\hat{s} = \arg \min_{s_l \in S} \text{dist}(s_l, [P_1, \dots, P_n])$$

where dist is a distance function (more details are provided in the following). The automatically learned schedule is then used to remove from P those elements that are not in \hat{s} at that time.

These principles of operation have been contextualized for an academic scenario: users are students frequenting the buildings of a University campus for attending lectures according to a well-known schedule.

B. Creating the map

The AP map is created by moving across all the rooms of the considered buildings and registering the set of visible APs for every room. Results are then sent to a remote server where they are stored on persistent memory.

During scanning the following information is collected: i) BSSID: each Basic Service Set is identified by a Basic Service

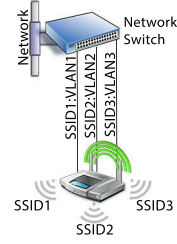


Fig. 2: Multiple SSIDs

Set ID (BSSID) which is derived from the MAC address of the AP; ii) ESSID: the Extended Service Set Identifier (ESSID) is a human-readable string that differentiates one WLAN from another. ESSID is essentially a name that identifies a wireless network. Using multiple ESSIDs allows users to access different networks through a single access point (Figure 2). The ESSIDs are sent in broadcast for advertising the presence of APs to the users.

ESSID is used to discard APs that do not belong to the academic/enterprise network to be mapped (APs located in close buildings are sometimes visible and have to be ignored). BSSIDs are used to identify single APs (adjacent BSSIDs are managed by a single AP and are coalesced into a single AP identifier). Collected data is processed to obtain the set of APs that are visible from every room; dual information, i.e. the set of rooms covered by every AP, is also determined.

Creating a map is always a cumbersome process. However the approach here proposed requires a reduced effort, as it needs a single sample per room. Moreover the map does not register signal strength information, which is more likely to change in presence of crowds than mere visibility of APs.

C. Processing user traces

A raw user trace is a sequence of samples collected with period T . Each sample consists of the set of BSSIDs visible at that time. Such traces are first processed to convert BSSIDs into AP identifiers using the AP map data structure. The result is a time-annotated sequence of AP sets:

$$u = A_{t_1}, A_{t_2}, \dots, A_{t_n}$$

where A_t is the set of AP visible at time t ($A_t \subseteq I$, where I is the set of all APs). Trace u is then filtered using a time window to remove those APs that appear rarely (e.g. those collected during movements from a room to another room). Let W be the size of the time window, and let w_i be the time interval that goes from $W \cdot (i - 1)$ to $W \cdot i$. The filtered trace is

$$\hat{u} = \hat{A}_{w_1}, \hat{A}_{w_2}, \dots, \hat{A}_{w_k}$$

where $a \in \hat{A}_{w_i}$ if $a \in \bigcup A_t$ with $t \in w_i$ and the number of occurrences of a in w_i is greater than a given threshold q . Finally, a room-based trace \bar{u} is generated from \hat{u} by converting AP identifiers in room identifiers:

$$\bar{u} = P_{w_1}, P_{w_2}, \dots, P_{w_k}$$

where P_{w_i} is the set of rooms for the i -th time window ($p \in P_{w_i}$ if p is covered by a with $a \in A_{w_i}$).

Finally, \bar{u} is compared with known schedules by computing a distance function. Then the schedule with the smallest distance is selected and used to restrict the position of the user.

D. Computing the distance with schedules

Each schedule is basically a list of rooms where the user is supposed to be if he/she follows that schedule. For each room the time interval is also specified. A schedule s_l is defined as:

$$s_l = r_{t'_1, t''_1}, r_{t'_2, t''_2}, \dots, r_{t'_m, t''_m}$$

where $r_{t'_j, t''_j}$ is a room identifier and t'_j, t''_j is the j -th time interval (from t'_j to t''_j). To compare schedules with user traces, the former are converted using time windows having the same size of the latter ones. This conversion can be easily performed selecting W as a submultiple of the typical time interval (t'_j, t''_j).

Thus, obtained schedules can be expressed as

$$\bar{s}_l = r_{w_1}, r_{w_2}, \dots, r_{w_m}.$$

Distance d between \bar{u} and \bar{s}_l is computed as follows:

$d \leftarrow 0$

for $i \leftarrow 1$ to k **do**

$x \leftarrow r_{w_i}$

if $x \notin P_{w_i}$ **then**

$d \leftarrow d + 1$

end if

end for

Distance is calculated for all known schedules ($\forall s_l$), then the schedule with the smallest value of distance is selected. k defines the amount of time the user is under observation, with $k \leq m$. In practice, the value of k can be determined as the smallest value that makes a schedule clearly distinguishable from the others.

It is worthwhile to notice that according to the proposed method the set of possible rooms is determined as the union of the sets of rooms covered by all visible APs. This is somehow counterintuitive: if the user is within the communication range of a set of APs, then he/she should be reasonably located in a room that belongs to the intersection set. Nevertheless, from preliminary experiments, we observed that the use of intersection makes the method scarcely tolerant to differences between coverage information acquired when the map is created and coverage information registered during the operational phase. In other words, the use of union makes the method more robust to possible inaccuracies in map data and interferences.

IV. A CASE STUDY

A preliminary evaluation has been carried out in a real environment.

A. Operational scenario

A Wi-Fi coverage map of the campus of the Faculty of Engineering, University of Pisa, has been collected. The map includes four buildings, named A, B, C, and F, as shown

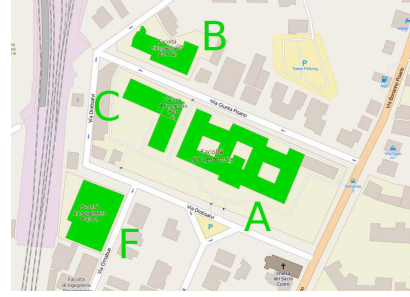


Fig. 3: The four buildings considered (campus of the Faculty of Engineering, University of Pisa, image data © OpenStreetMap contributors).

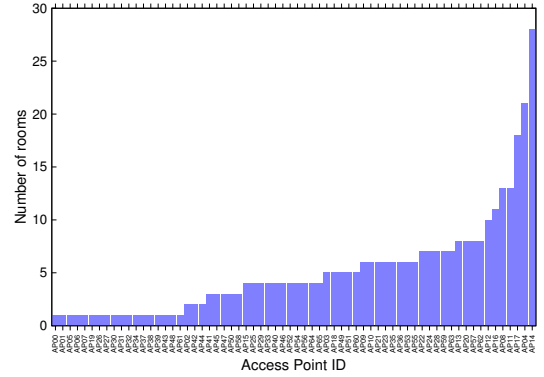


Fig. 4: Number of rooms covered by APs

in Figure 3. Such buildings host approximately 10000 students and 400+ members of the academic, technical, and administrative staffs. The buildings include several large rooms (auditoriums with 300 seats) and many medium and small classrooms, as well as many small offices.

B. Statistics

The four buildings are equipped with 65 APs. The number of rooms is equal to 48, which can be divided in 41 classrooms and 7 “spaces” dedicated to other activities. The map has been analyzed to better understand the operational scenario. In particular, we focused on the relationship between the coverage area of APs and rooms, as it directly affects the localization ambiguity when using the strawman approach.

First, we analyzed how many rooms are in each AP coverage area. Figure 4 shows that over 33% of APs in our scenario cover more than five rooms. Then we investigated how many APs are visible from each room. Figure 5 shows that for all the rooms in the considered buildings at least two distinct access points are visible. The same information is aggregated and shown in Figure 6, in terms of frequency (number of rooms where the amount of visible APs is the one specified by the abscissas). Complete coverage information for the considered area is available in Appendix.

In this scenario we observe that some APs that are located in a building are visible in other buildings as well. This would make even more ineffective a method like the strawman approach, since not even the exact localization of a user, at building level, would be feasible.

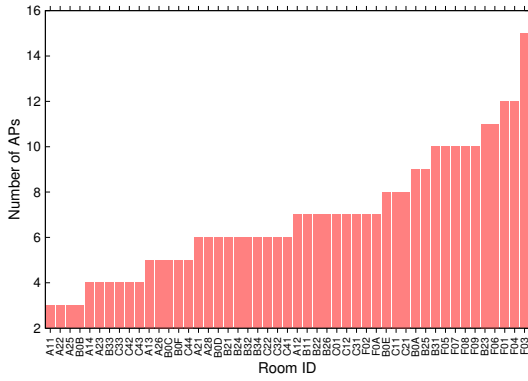


Fig. 5: Number of APs visible from every room

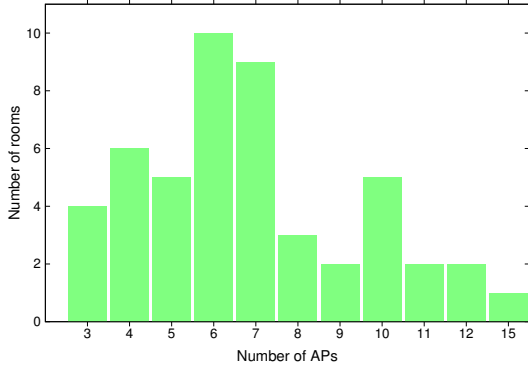


Fig. 6: Number of rooms vs number of visible APs

C. Results

Ten traces, corresponding to ten users moving in the considered area, have been collected approximately from 8am to 7pm. Such students attend ten different undergraduate/graduate courses and thus they have different schedules.

Table I reports the minimum, maximum, and average value of $|P|$ for the considered students. Global values for the ten users are also reported. Note that the registered values are rather large and this means that the ambiguity of the strawman approach is considerable. For this analysis raw user traces have been collected with 1 minute period, the value of q and W have been set to 15 and 1 hour respectively.

The proposed method is successful if the schedule with smallest distance is the real schedule of the users and all the other schedules have larger distance values (to make the real schedule distinguishable from the others).

For the ten considered users, in 90% of cases identification of user's schedule is successful and the position of the user can be restricted to a single room. In one case the method has not been able to uniquely identify the schedule of the user: despite the distance for the real schedule of the user is zero, there are two other schedules with the same distance. Results are shown in Table II. The first column reports the real schedule identifier for the ten users. The other columns contain the identifiers of the schedules with the smallest distances (in order of increasing distance). It can be noticed that for s_3 there are three schedules (s_3, s_{11}, s_{12}) with a distance value equal to zero.

TABLE I: Cardinality of the set of possible rooms for ten users

User ID	Min	Max	Avg
1	12	33	27.2
2	37	37	37.0
3	12	25	19.8
4	7	24	13.8
5	33	33	33.0
6	33	39	34.7
7	33	33	33.0
8	37	39	37.8
9	37	37	37.0
10	9	39	32.1
Global	7	39	30.54

TABLE II: Results of the application of the proposed method on ten users

Schedule	Distances		
s_1	s_1 0	s_{45}, s_{46} 1	s_{13}, s_{14}, s_{15} 2
s_2	s_2 0	$s_{16}, s_{17}, s_{18}, s_{19}$ 2	s_4 3
s_3	s_3, s_{11}, s_{12} 0	s_{22}, s_{23}, s_{24} 1	s_{34}, s_{46} 3
s_4	s_4 0	$s_{22}, s_{25}, s_{26}, s_{27}, s_{28}$ 1	s_{45}, s_{47}, s_{48} 2
s_5	s_5 0	s_{29}, s_{45} 2	s_{30}, s_{31} 3
s_6	s_6 0	s_{33}, s_{34} 2	s_{16}, s_{35}, s_2 3
s_7	s_7 0	s_{36} 1	$s_{18}, s_{19}, s_{37}, s_{46}$ 2
s_8	s_8 0	s_{12} 2	s_{38} 3
s_9	s_9 0	s_{14}, s_{37} 3	s_{29}, s_{41}, s_{42} 4
s_{10}	s_{10} 0	s_{24}, s_{43} 3	s_{14}, s_{23}, s_{44} 5

In general, a user can be observed until his/her schedule becomes clearly distinguishable from the others. Conversely, the observation period can be stopped as soon as a clear schedule emerges from the user's coarse-grained trajectory. For the ten considered users the observation period was fixed and equal to one working day, and this amount of time has been sufficient to successfully infer their schedule in 90% cases.

To operate properly the method should avoid generating "false positives": users who move without following a pattern should not be recognized as belonging to any schedule. We performed a preliminary analysis by collecting ten traces of users who move according to a randomly generated pattern, then we calculated the minimum distance of these traces with respect to the known schedules. Each trace is 8 hour long and users remain in rooms using the same time window of the schedules (W). This makes random traces similar to those produced by users following schedules (to test the method in the worst conditions). On average, the minimum distance between random traces and the closest schedule is ~ 3 . Thus, considering that for users following schedules such value is always zero, we can reasonably state that the two classes can be distinguished by means of a threshold.

