GPS/HPS-and Wi-Fi Fingerprint-Based Location Recognition for Check-In Applications Over Smartphones in Cloud-Based LBSs

Igor Bisio, *Member, IEEE*, Fabio Lavagetto, Mario Marchese, *Senior Member, IEEE*, and Andrea Sciarrone

***Abstract*—This paper proposes a new location recognition algorithm for automatic check-in applications (LRACI), suited to be implemented within Smartphones, integrated in the Cloud platform and representing a service for Cloud end users. The algorithm, the performance of which is independent of the employed device, uses both global and hybrid positioning systems (GPS/HPS) and, in an opportunistic way, the presence of Wi-Fi access points (APs), through a new definition of Wi-Fi FingerPrint (FP), which is proposed in this paper. This FP definition considers the order relation among the received signal strength (RSS) rather than the absolute values. This is one of the main contributions of this paper. LRACI is designed to be employed where traditional approaches, usually based only on GPS/HPS, fail, and is aimed at finding user location, with a room-level resolution, in order to estimate the overall time spent in the location, called Permanence, instead of the simple presence. LRACI allows automatic check-in in a given location only if the users' Permanence is larger than a minimum amount of time, called Stay Length (SL), and may be exploited in the Cloud. For example, if many people check-in in a particular location (e.g., a supermarket or a post office), it means that the location is crowded. Using LRACI-based data, collected by smartphones in the Cloud and made available in the Cloud itself, end users can manage their daily activities (e.g., buying food or paying a bill) in a more efficient way. The proposal, practically implemented over Android operating system-based Smartphones, has been extensively tested. Experimental results have shown a location recognition accuracy of about 90%, opening the door to real LRACI employments. In this sense, a preliminary study of its application in the Cloud, obtained through simulation, has been provided to highlight the advantages of the LRACI features.**

***Index Terms*—Check-in applications, cloud computing, GPS/HPS receivers, smartphone terminals, Wi-Fi fingerprint.**

I. Introduction

A Location-Based Service (LBS) is an information service, accessible through mobile devices, such as Smartphones, which provides the identification of people and objects location. LBS can be used in many applicative scenarios, such as health, object search, entertainment, work and personal life. LBS applications may include parcel and vehicle tracking services and mobile commerce when taking the form of advertising directed at customers and based on their current location. One of the most popular LBS applications concerns *Check-In*, whose aim is allowing people to *Check-In* at specific locations such as pubs, supermarkets, and post offices. Two well-known *Check-In* applications are *Foursquare* [1] and *Gowalla* [2], which have spread rapidly. Using these applications, users can *Check-In* at a location, sharing information with other people, leaving comments and votes, retrieving suggestions and enjoying benefits dedicated to "regulars" that spend some time in the location. On the other hand, the increasing popularity of these applications has allowed revealing some of their weaknesses. For example, it is difficult to guarantee the owner of a pub (the location where to *Check-In*) that a customer has actually stayed in the location for a given amount of time. Some users could be tempted to *Check-In* when they simply pass near the location without really staying, just to obtain possible commercial benefits dedicated to accustomed people. To avoid this possibility *Check-Ins* should be validated by considering not only the correct user location but also a minimum period of time spent by a user in a given location. This period is called Stay Length (SL) and it is usually set by a business owner. In practice, a *Check-In* request is considered valid only if the user permanence in the location (i.e., the overall time spent by that user in the location) is larger (or equal) than the SL.

This is the direction taken by the Location Recognition Algorithm for Automatic Check-In applications (LRACI) introduced in this paper. LRACI is implemented over Smartphones, is independent of the employed device, uses GPS and HPS positioning information together with data received by WI-FI access points, when available, exploited through a new definition of Wi-Fi FingerPrint (FP), and uses the concept of Stay Length (SL) to validate Check-Ins.

The success of Cloud Computing (CC) offers further opportunities for LBSs, which can be exploited in the Cloud and give origin to cloud-based LBSs. CC paradigm is clearly defined in [3]. The cloud model is composed of: 1) five essential characteristics: on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service; 2) three service models: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS); 3) four deployment models: private clouds, community cloud, public cloud, and hybrid cloud. The detail of service models is important to evidence the exploitation of LBS through the Cloud. From [3]: SaaS represents the capability provided to the consumer to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices

Manuscript received March 02, 2012; revised July 12, 2012; accepted November 13, 2012. Date of publication January 11, 2013; date of current version May 13, 2013. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Joel Rodrigues.

The authors are with the Department of Telecommunications, Electronic, Electrical Engineering and Naval Architectures (DITEN), University of Genoa, 16145, Genoa, Italy (e-mail: igor.bisio@unige.it; mario.marchese@unige.it; fabio.lavagetto@unige.it; andrea.sciarrone@unige.it).

Digital Object Identifier 10.1109/TMM.2013.2239631

through client interfaces, such as web browsers and/or program interfaces. PaaS focuses on the capability provided to the consumer to deploy onto the cloud infrastructure consumer-created or acquired applications developed by using programming tools supported by the provider. IaaS evidences the capability provided to the consumer concerning processing, storage, networks, and other computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. Another model, detailed in [4], is important to mention: Data as a Service (DaaS), which focuses on the capability provided to the consumer to access shared Data in the Cloud.

The full and efficient utilization of LBS, and of LRACI in particular, is strictly connected with the evolution of Cloud Computing, as also certified by several commercial initiatives such as, among the others, LocAid [5] and AT&T [6]. In detail: A) Smartphones, GPS/HPS, and Wi-Fi access points are part of the Cloud Platform and Infrastructure and are tools used by LBS applications: Platform and Infrastructure are Services for LBS applications and this matches PaaS and IaaS models. B) LRACI implementation, as said, is independent of the specific Smartphone technology: it is implemented in the Cloud, not within a part of it; so it is a Software Service for the Cloud following SaaS model. C) Location data (or, more specifically for this paper, Check-In data) represent a service for Cloud Users (as in DaaS model); this feature is characteristic for Cloud-based LBS, as clearly evidenced in [4], and is a clear distinction factor with respect to traditional LBS. In this view may be interpreted not only LRACI but also the Platform functions, oriented to floating-car-traffic-data-based traffic information, to Point-of-Interest (POI) search, and to Path finding, presented in [7], where location data are a service available in the Cloud. D) LRACI Location/Check-In data are shared among Cloud users and represent a resource pool to access.

The paper is structured as follows: Section II surveys the State of the Art in the field and highlights the main differences between the existing solutions and the proposed approach. Section III contains the main contribution of this work: it describes the analytical details of the opportunistic location recognition method. The computation of the *Permanence* in a location, the LRACI feature that allows providing robust Check-Ins, is detailed in Section IV. Section V contains a performance evaluation of LRACI both in terms of location recognition accuracy (through real measurements) and as a tool in a Crowdedness Monitoring Application operating over the Cloud (through simulations). Conclusions are drawn in Section VI.

II. State of the Art

In recent years, many indoor and outdoor location recognition methods have been developed. For indoor environments, infrared, ultrasonic, GSM, Wi-Fi and RFID are commonly used technologies while, in case of outdoor scenarios, GPS and Cell Tower Localization are the most employed [8], [9], although also Wi-Fi is used [10] at metropolitan-scale.

GPS is the most popular and widely used positioning system, it is maintained by the United States government and provides location information obtained by signals sent from a group of satellites. GPS can provide users' locations very accurately but its signals are often blocked and absorbed by walls or other obstacles [11]. Therefore GPS is not suitable for indoor environments. *Rosum's TV-GPS* is an enhanced positioning technique, which works both indoor and outdoor. It uses the Time Difference Of Arrival (TDOA) approach applied to TV signals to estimate the position. As said in [12], it needs additional hardware for television transmitter towers to achieve precise time synchronization. The achieved positioning error is in the range 3.2–23.3 [m]. Another interesting localization approach is Japan's Indoor Messaging System (IMES), which is an important part of the regional Quasi-Zenith Satellite System (QZSS) project. It uses GPS signals and provides precise positioning because it employs terrestrial transmitter equipments and beacons to assist the whole localization process [13].

All the above mentioned localization systems are not suitable for Smartphone platforms integrated in the Cloud, which is the reference technological environment of this paper. The motivation is linked to two main factors: *i)* the high cost of the network infrastructure for a metropolitan-scale coverage; *ii)* the necessity of extra modules for mobile devices, which increases Smartphone implementation costs. In the last years, the need of localization techniques that provide good user position without requiring extra-hardware and high costs, and so that can be efficiently implemented over Smartphones, is satisfied by methods (called Hybrid Positioning Systems—HPS) that jointly use GPS, Wi-Fi Access Points, Bluetooth devices and Cell Towers signal strengths. An example of HPS is the Intel's Place Lab [8] method, which employs radio beacons (802.11 APs, GSM, and Bluetooth) that already exist in the environment and have unique or semi-unique IDs such as, for example, MAC addresses. Mobile devices compute their own location by detecting one or more IDs, looking up the associated beacons' positions in a locally cached map, and estimating their own position with respect to the beacons' positions [8]. Always within the family of HPSs, another method, applied by Skyhook, is the X Positioning System (XPS) [14]. It is the first commercial metropolitan-scale positioning system and employs a wardriving collection to build a reference database containing the locations of Wi-Fi APs. In order to improve the localization accuracy, XPS detects Wi-Fi APs through a scanning algorithm, uses GPS positioning information about Wi-Fi APs to (reverse) triangulate the position of detected APs, and stores the position in a reference database. The precision of Wi-Fi APs positions is affected by the scanning scheme employed by the mobile device, which can have an error larger than 10 [m]. Similar to XPS, PlaceEngine [15] is a Wi-Fi-based location platform, jointly developed by Sony Computer Science Laboratory and University of Tokyo, which covers many cities in Japan, including the metropolitan area of Tokyo, and uses a database that contains data of about 500,000 APs' [15].

Another interesting technique, applied to estimate the position of mobile users, is Count-Of-Beacon (COB) [16], [17]. COB is a radiolocation technique where the existence of Non-Line Of Sight (NLOS) propagation does not degrade the precision of the estimated position significantly. In more technical words, a sort of probability density function, built by using the received power of some beacons, is associated to the possible user position and employed to estimate the current position.

Wi-Fi *FingerPrint* (FP)-based location recognition methodologies, originally designed and employed for indoor positioning purposes by using Wi-Fi [18], [19] and Bluetooth [11], [20], are a very interesting approach also for outdoor positioning. There are many scientific papers in the literature aimed at recognizing locations by using Wi-Fi FPs that can be either built manually by trained experts and so available in advance, or, more practically, built automatically. [21] proposes a *crowdsourcing* radiomap building method for location recognition in urban environment, whose main idea is based on the fact that it is not always feasible to manually build a collection of Wi-Fi FPs (i.e., a radiomap) for each location/Point Of Interest (POI), especially in large-scale urban environments. In order to solve this problem, [21] proposes an algorithm in which the users of Wi-Fi enabled mobile devices contribute to build the FP in a collective and automatic way. FPs may be also collected, computed, and compared each other dynamically over time transparently to users by using the Received Signal Strength (RSS). Among FP-based approaches, a well-known platform, described in [19], [22], is LifeMap. It is based on the autonomous construction of a personalized POI map, which provides location information for advanced mobile services. The key concept is to use an accelerometer to track user locations and to identify the POIs. The solution incrementally builds user's POIs through a personalized radiomap generated from the properties of Wi-Fi APs (e.g., from the RSS). [18], [23] describe other interesting location estimation approaches, implemented within Smartphone platforms, which employ the RSS received by the APs in the surrounding even if they do not build any FP.

The common point, among many of the techniques mentioned above is the employment of the absolute values of the measured RSS. FPs are built by measuring the RSS, sensed during a first step, called training phase. RSS absolute values are employed also in the recognition phase. This action, independently of the robustness of the employed method, presents some drawbacks. Measured RSS absolute values: a) are sensitive to multipath fading, to device orientation, and to other important factors deeply studied in [24]–[26]; b) are strongly dependent on the employed device (i.e., two different Smartphones, in the same position with the same orientation, often provide different RSS measures). These drawbacks have an impact not only on the performance, but, the latter in particular, on the practical applicability of the location recognition solutions as SaaS in the Cloud, and on the Cloud Computing deployment model, mentioned in the Introduction. A particular, not obvious, implication of the device dependability is that, if during the recognition phase the employed device is different from the device used during the training, the accuracy of the location recognition approaches decreases. This paper proposes also a possible solution of this problem based on "relative" RSS measures, i.e., on the employment of the order relation among the measured RSS sent from different APs, rather than on RSS absolute value.

In short, we introduce a new Location Recognition Algorithm for Check-In applications, whose acronym is LRACI, where the recognition action is based on: i) exploitation of GPS-HPS information opportunely filtered and weighted (Error Correction Filter with Wi-Fi stability condition); ii) new automatic, opportunistic and device-independent FP building and matching method.

III. Location Recognition Algorithm for Automatic Check-In Applications (LRACI)

A. LRACI Action and Flowchart

LRACI employs positioning information, provided by Smartphones Operating System(s), acquired from GPS/HPS, and a new definition of FP detailed below. LRACI is based on a sliding time window of T seconds during which positioning data (latitude and longitude) and Wi-Fi scans, used to define the FP, are acquired simultaneously. They represent the available information elements of the window. Z^{ζ} elements are stored in the generic ζ-th window. The number of stored elements can be different for each window because positioning data and Wi-Fi scans are provided by the Smartphone OS at irregular time intervals independent of time window T.

LRACI defines a generic location l, $l \in [1, L]$ by three different features: *i)* the coordinates of the location centre represented by $C^l = \left[C^l_{lat}, C^l_{lon}\right]$ where C^l_{lat} indicates the latitude and C^l_{lon} is the longitude; *ii)* the radius R^l expressed in [m]; *iii)* the Wi-Fi FP of the location indicated through F^l. $\mathbf{C}^l$ and R^l completely identify the location from the geometrical viewpoint as a simple circle and are supposed known *a priori*, stored in a reference Database (DB) that is always accessible and available directly on the Smartphone when needed. *FingerPrint* F^l is not always available within the DB because either it cannot be computed because no Wi-Fi signals cover the location or because it is under computation locally to a Smartphone (through the algorithm explained later in the paper) and not yet uploaded in the DB. LRACI does not need the FP mandatorily, nevertheless, practical experiments, whose results have been reported in the performance evaluation, have pointed out that FP exploitation considerably improves the accuracy of location recognition.

Fig. 1 represents the flowchart of LRACI. For each generic ζ-th time window of T seconds, a location may be recognized by using dedicated procedures (detailed below), whose employment depends on the availability of the F^l of a given location within the reference DB. In particular, if F^l is not available in the DB, the location recognition and the consequent Check-In is mainly based on GPS/HPS data. The acquired Wi-Fi signals, scanned during T are employed, if a Smartphone checks-in, to compute a FP that will be uploaded in the DB. It means that for successive recognition/Check-In in that location F^l will be available in the DB. If the FP is available the location recognition is based on it. The two procedures are called *Unavailable FP* and *Available FP*, respectively.

It is worth noting that, even if F^l is available, a FP is locally computed by a Smartphone to recognize the location and, after the Check-In, to upgrade the stored FP in the reference DB. In practice, LRACI offers a continuous and opportunistic learning process of the locations' FP.

LRACI realizes the so called *Stay Detection*, which provides robust *Check-In*. T^l_P, called *Permanence*, is the time during which a given location is continuously recognized by a Smartphone (i.e., the time spent by a user in l). It can be measured by exploiting the sliding window mechanism of LRACI (as detailed in Section IV). If T^l_P is longer than a predefined *Stay*

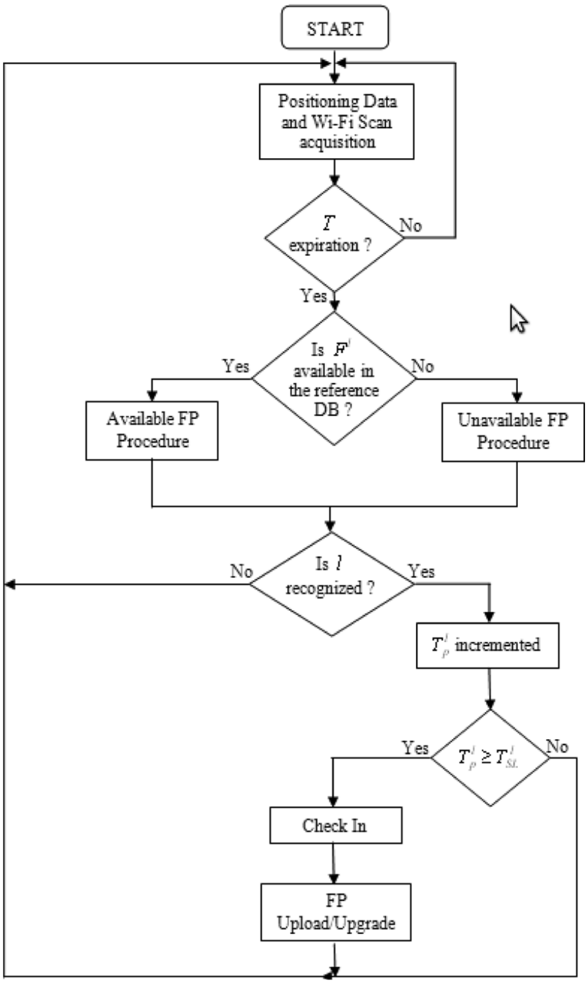

Fig. 1. LRACI flowchart.

Length T_{SL}^{l}, for a location l, the conclusion is that a user has really stayed in l and is not just passed near it. In this case, the user is considered automatically checked-in.

B. Wi-Fi Scan Definition

Some preliminarily definitions are necessary to describe LRACI in more detail. As said in the previous sub-section, LRACI uses a method where a Smartphone cyclically radio scans the surroundings of a location looking for Wi-Fi signals. Three important features for each detected AP are acquired during each scan: *i)* APs *MAC addresses*, *ii)* Service Set IDentifier *SSID* and *iii)* measured Received Signal Strength (*RSS*) in [dBm]. A matrix $\hat{S}^k$, whose elements are $\hat{s}^k_{m\hat{n}}$, is computed for each generic k-th Wi-Fi scan:

$$\hat{S}^k = \begin{pmatrix} \hat{s}^k_{11} & \cdots & \hat{s}^k_{m1} & \cdots & \hat{s}^k_{M1} \\ \vdots & & \vdots & & \vdots \\ \hat{s}^k_{1\hat{n}} & \cdots & \hat{s}^k_{m\hat{n}} & \cdots & \hat{s}^k_{M\hat{n}} \\ \vdots & & \vdots & & \vdots \\ \hat{s}^k_{1N} & \cdots & \hat{s}^k_{mN} & \cdots & \hat{s}^k_{MN} \end{pmatrix}$$

$m \in [1, M]$ identifies the features (concerning this paper $m = 1$ is the *MAC address*, $m = 2$ the *SSID* and $m = 3$ is the *RSS*) and $\hat{n} \in [1, N]$ identifies the sensed AP during scan $k \in [1, K]$. K is the total number of scans performed to define the FP. The total number N of sensed APs N obviously varies and can change for different scans.

LRACI uses a matrix S^k (s^k_{mn} are the elements) defined as the down sorting of the rows of the matrix $\hat{S}^k$. This action is carried out by supposing that the elements of the last column of $\hat{S}^k$ are always the measured *RSS* (i.e., M always identifies *RSS*s). In practice, the matrix S^k is obtained by reordering the rows of $\hat{S}^k$ so that $s_{Mi} \geq s_{Mj}, \forall i < j$ and n is a mere index in the ordered matrix ($\hat{n}$ is the only AP identifier). This preliminary down-sorting step is the basic action to derive a device-independent FP definition. The key idea is that different Smartphones provides different *RSS* absolute values, due to distinct hardware and/or Operating Systems (OSs) versions, but the order within S^k is the same (or very similar). In consequence, the proposed FP definition uses a function that assigns a weight to each AP by considering the order positions taken in the S^k $\forall k \in [1, K]$ matrixes.

C. GPS/HPS Positioning and Error Correction Filter (ECF)

In order to improve the possible position error of the HPS and to obtain a more robust position data, in this paper we propose a correction filtering approach that exploits the time-based sliding window mentioned above. The main idea is to store the positioning information, composed of latitude and longitude, within a filter during a temporal window of T seconds. T is in the order of minutes. This is compatible with Check-In applications. In practice, we collect data (i.e., the elements in the window) until the difference between the acquisition time of the last and the first element is longer than the window size T. When the filter is completely filled, the weighted mean position $P = [P_{lat}, P_{lon}]$, obtained from all acquired data, is computed by using the following equation:

$$P_{lat} = \frac{\sum_{z=1}^{Z^\varsigma} \omega^z \hat{P}^z_{lat}}{\sum_{z=1}^{Z^\varsigma} \omega^z}; P_{lon} = \frac{\sum_{z=1}^{Z^\varsigma} \omega^z \hat{P}^z_{lon}}{\sum_{z=1}^{Z^\varsigma} \omega^z} \qquad (1)$$

where, Z^ζ is the number of positioning data stored during the ζ-th time window, $\hat{P}^z_{lat}$ and $\hat{P}^z_{lon}$ represent the latitude and longitude of the z-th generic position, respectively, and ω^z is a weight assigned to each data. Since GPS is often more accurate than HPS, we set a higher weight for the GPS positioning data with respect to the HPS ones. Practical experiences allow concluding that the following values can be employed:$\omega^z = 0.8$ if the positioning data is provided by GPS, $\omega^z = 0.2$ if the positioning is obtained from HPS. The process is repeated by sliding the time window T of one element. In practice, the first element of the windows is removed and new positioning data can enter; when the window is filled again, equations in (1) are applied. The filtered positioning data $P = [P_{lat}, P_{lon}]$ are used to recognize a location if the FP of that location is not available, as described at the beginning of this Section and, in more detail, in Section III-E.

D. New FP Definition

The proposed FP definition, used to recognize a given location, is based on three processing steps: *Merging*, *Sorting* and *Filtering*. *Merging* and *Sorting* are iterated over the total amount of K scans of the available Wi-Fi signals, while *Filtering* is applied only once. In general, the FP is computed after K scans performed in a given location. Empirical experiments have shown that $K = 15$ represents a good trade-off between FP robustness and processing time. Before introducing the steps, we define the FP of location l at the step 0 as the empty set, thus $F^{l,0} = \emptyset$.

1) Merging Step: When a Wi-Fi scan is available (i.e., S^k, $k \geq 1$) the merging step starts. $\tilde{F}^{l,k}$ is a column vector and represents the FP of location l, after the merging step, at k-th scan. Vector $F^{l,k-1}$ is the FP of location l computed after the previous $(k-1)$-th sorting step, detailed below. For each scan, the following quantity is computed:

$$\tilde{F}^{l,k} = F^{l,k-1} \cup col_1 S^k, \forall k \in [1, K] \tag{2}$$

where $col_1 S^k$ represents the column that contains the *MAC Address* in the matrix S^k (the first column, in this paper). In practice, $\tilde{F}^{l,k}$ is the union of $F^{l,k-1}$ and $col_1 S^k$ and, as a consequence, contains all *MAC addresses* of $F^{l,k-1}$ and $col_1 S^k$. For the next steps, we define the *MAC Address* $\tilde{f}_n^{l,k}$ as the *n-th* element of the set $\tilde{F}^{l,k}$.

2) Sorting Step: The output of the k*-th* sorting step is a new FP, $F^{l,k}$, obtained from the partial FP $\tilde{F}^{l,k}$, computed as in (2), down-sorted by using the weighting function defined below.

For the sake of simplicity, we preliminarily define the following quantity:

$$\Omega(I, x) = p \quad x \in I, p \in [1, card(I)] \tag{3}$$

where I is a generic not-void ordered set, x is a generic element, and $card(I)$ represents the cardinality of the set I (i.e., the number of elements of the set I), $p \in [1, card(I)]$ is the index of the element x within I. $\Omega(\cdot,\cdot)$ has been defined to extract the index of an element within an ordered set (i.e., within $F^{l,k-1}$ and $col_1 S^k$). Starting from the $\Omega(\cdot,\cdot)$ operator, the key idea of this procedure is that, if a *MAC Address* is in the top part of vectors $F^{l,k-1}$ and $col_1 S^k$, then a higher weight value is assigned. In practice, for each *MAC Address* a weight is defined in (4) and $\tilde{F}^{l,k}$ is down-sorted consequently. The weighting function $\hat{W}^k$, computed for all the elements of the vector $\tilde{F}^{l,k}$, is:

$$\hat{W}^k\left(i^k, j^k\right) = \frac{1}{|i^k - j^k| + \left(\frac{i^k + j^k}{2}\right)} \tag{4}$$

where $i^k, j^k \in [1, N]$ are:

$$\begin{cases} i^k = \Omega(F^{l,k-1}, \tilde{f}_n^{l,k}) \\ j^k = \Omega\left(col_1 S^k, \tilde{f}_n^{l,k}\right) \end{cases} \quad n \in [1, N]. \tag{5}$$

Equation (4) and (5) are used if $\tilde{f}_n^{l,k} \in F^{l,k-1}$ and $\tilde{f}_n^{l,k} \in col_1 S^k$. Alternatively, $\hat{W}^k = 0$. In other words, i^k is the position of the n-th *MAC Address* of vector $F^{l,k-1}$ and j^k is the position of the same *MAC Address* in $col_1 S^k$, $n \in [1, N]$. The weighting function $\hat{W}^k$ takes into account the position of *MAC Addresses* $(\tilde{f}_n^{l,k})$ in the vector obtained, after the *Merging Step* $(\tilde{F}^{l,k})$ within $F^{l,k-1}$ and within $col_1 S^k$, from the difference between the *MAC Address* positions within these vectors. In practice: if a *MAC Address* is in the top positions in both the column-vectors $F^{l,k-1}$ and $col_1 S^k$ the function provides a large weight; if a *MAC Address* is in the highest positions of the vectors $F^{l,k-1}$ and in the lowest one of the vector $col_1 S^k$, the weight is low; if a *MAC Address* is in the lowest positions in both vectors $F^{l,k-1}$ and $col_1 S^k$, the weight is very low. If a *MAC Address* is not shared by $F^{l,k-1}$ and $col_1 S^k$, the weight is zero. The weighting function $\hat{W}^k$ does not consider the weights computed during previous scans, but, operatively, it would be opportune to consider them. A simple way to perform this action is to employ weighting values, at the k-th step, averaged over the k scans, as indicated in the following recursive formula (6), adopted in this paper. The *Sorting step* concludes after the K scans.

$$W^k = \frac{1}{k}\left[(k-1)\, W^{k-1} + \hat{W}^k\right]. \tag{6}$$

3) Filtering Step: To avoid the inclusion of APs rarely detected during the K scans, the *Filtering step* has been introduced. The main reason behind this step is that, performing the *Merging step*, each *MAC Address* that has been detected at least once during the K scans would be listed in the final FP F^l of location l.

Operatively, this step is aimed at eliminating the *MAC Addresses* with low weighting values. A threshold γ_W has been defined and the final FP vector includes only AP *MAC Addresses* with $W^K \geq \gamma_W$. As a positive side effect, the filtering step allows limiting the fingerprint size so saving time and memory resources of Smartphones. Big values of γ_W imply dropping *MAC Addresses* that may represent significant elements of the final FP but, on the other hand, low γ_W values filter only few APs. Experimental tests have shown that $\gamma_W = 0.01$ represents a good trade-off value.

E. Location Recognition

LRACI employs positioning data collected and filtered as described in Section III-B. and, whenever possible, the FP defined in Section III-D. In practice, during the ζ-th window of T [s], both positioning data and Wi-Fi scans are acquired simultaneously. So, operatively, the total number of scans K is equal to the number Z^{ζ} of the positioning data. Among the locations considered in the decision process, whose definitions (i.e., centre, radius and, if available on the reference DB, FP) are known *a priori*, LRACI recognizes the locations with no FPs available in the DB through the "*Unavailable FP*"*Procedure* (Section III-E1); for the locations that have an available FP, the "*Available FP*" *Procedure* is applied (Section III-E2).

1) "Unavailable FP" Procedure: In case a location is characterized only by centre and radius, the Location Recognition step is based on two sub-mechanisms: Position Stability Detection (PSD) and Wi-Fi Stability Detection (WSD) whose details are described below.

The Location Recognition based only on positioning data, in some cases, is not precise. Supposing that a user is inside a small room, to determinate the real room centre is not a simple task because different locations can be adjacent or overlapped. For example, it is common to consider as locations a specific room (e.g., a laboratory) within a building and the building itself. Both of them may have the same centre. In this case, if a user is moving, passing from room to room or from floor to floor, and its position falls within the building, the recognized location can be the room (i.e., the mentioned laboratory) even if the user is not actually there.

In order to avoid the described problem, we designed and employed a Wi-Fi stability detection mechanism. The key idea is to recognize a specific location only if the positioning data provided by the Error Correction Filter (ECF) are inside the room circle (see PSD) and if either the radio environment does not present significant variations or there is a dominant AP (see WSD).

The Position Stability Detection (PSD): PSD is a simple mechanism used to verify if the position $P = [P_{lat}, P_{lon}]$, provided by ECF, is inside the circle of centre $\mathbf{C}^l$ and radius R^l for a generic location l. It happens if the distance d^l between $P = [P_{lat}, P_{lon}]$ and the location centre $\mathbf{C}^l$ is lower than the radius R^l. Analytically, the distance d^l is computed by exploiting the definition of orthodromic distance (approximated by the Haversine formula) as reported below:

$$d^l = 2r_e \arcsin \sqrt{\sin^2 \Delta P_{lat}^l + \cos C_{lat}^l \cos P_{lat} \sin^2 \frac{\Delta P_{lon}^l}{2}} \tag{7}$$

where $r_e \cong 6367.38 \cdot 10^3$ [m] is the Earth's radius, $\Delta P_{lat}^l = \left|C_{lat}^l - P_{lat}\right|$ and $\Delta P_{lon}^l = \left|C_{lon}^l - P_{lon}\right|$.

The position is defined as stable if, for a generic location l, the condition $d^l \leq R^l$ is verified at the end of the ζ-th sliding window of T [s].

Wi-Fi Stability Detection (WSD): The stability of the radio environment is detected by verifying if at least one of the following conditions A and B is true.

Condition A: Wi-Fi Environment Does Not Present Significative Variations: In order to understand if the radio environment is significantly changed, a Utility Value (UV) is computed at the end of each ζ-th sliding window of T [s]. uv^k is the single utility value, determined every time that a wireless scan S^k is available, within a sliding window, for $k \geq 2$.

$$uv^k = \frac{card\left(S^k \cap S^{k-1}\right)}{\max\left[card\left(S^k\right), card\left(S^{k-1}\right)\right]}, k \geq 2. \tag{8}$$

As a consequence, we obtain $Z^\zeta - 1$ single utility values for each window, where Z^ζ is the number of scans in the ζ-th sliding window. It is worth noticing that uv^k is proportional to the cardinality of the intersection between S^k and S^{k-1}, hence more APs two consecutive wireless scans have in common, higher uv^k value is. Once that the ζ-th ECF window of T [s] is completely filled, the mean of all single utility values is computed through (9).

$$UV^\zeta = \frac{1}{Z^\zeta - 1} \sum_{k=2}^{Z^\zeta} uv^k. \tag{9}$$

As a consequence, we assume that the Condition A is satisfied, for the window ζ if

$$UV^\zeta = \gamma_{_{UV}} \tag{10}$$

where $\gamma_{_{UV}}$ is a predefined threshold.

Condition B: Existence of A Dominant AP: This condition has been added since it allows improving WSD robustness. In case of locations with very few APs, UV^ζ computed in (9) is not sufficient. For example, if a generic location has only very few APs and some wireless scans do not detect all of them, the computed mean Utility Value UV^ζ will be close to zero (i.e., the cardinality of the intersection between consecutive scans will be very low). Hence Wi-Fi stability, tested through *Condition A*, will not be guaranteed, even if the radio environment is not changed so much. In order to solve this problem, the concept of *Dominant AP* has been introduced. The main idea behind this condition is that, even if the user is inside a location with very few APs, until the dominant one (defined below) is present, the stability condition is guaranteed. We define the *Intersection Set* (IS) as in (11).

$$IS^\zeta = \bigsqcup_{k=1}^{Z^\zeta} S^k \quad k \in [1, K] \tag{11}$$

where k represents the scan index and ζ is the window index. IS^ζ is the set defined by the intersection among all the Z^ζ wireless scans, for the ζ-th window. Symbol $\sqcup$ indicates the disjoint set union. This kind of mathematical operator has been used instead of the traditional set union since, in the disjoint union, the cardinality of each element in the final set is the sum of the cardinalities of the same elements in all united sets (i.e., including multiple elements). Hence, the final set IS^ζ contains all the elements of all wireless scans S^k and its cardinality is $card\left(IS^\zeta\right)$. Being the operator $\varphi(I, x)$ a function providing the number of occurrences of the element x in the set I, we define the *Dominant AP* (AP_D) according to the following equation:

$$AP_D = \begin{cases} is_s^\zeta & \text{if } \exists! is_s^\zeta : \underset{is_s^\zeta}{\operatorname{argmax}} \frac{\varphi\left(IS^\zeta, is_s^\zeta\right)}{card\left(IS^\zeta\right)} \\ \varnothing & \text{otherwise} \end{cases} \tag{12}$$

where is_s^ζ is the s-th element of IS^ζ and $\varnothing$ stands for the empty set. Equation (12) means that the *Dominant AP* exists only if one AP is detected for a number of times larger than the number of times all the others APs have been detected during all the Z^ζ wireless scans. Summarizing the entire procedure in more mathematical detail, we recognize the location l if the logical condition reported in (13) is true.

$$\left(d^l \leq R^l\right) \wedge \left[\left(AP_D \neq \varnothing\right) \vee \left(UV^\zeta \geq \gamma_{_{UV}}\right)\right]. \tag{13}$$

It is straightforward that, if the generic location has no APs in its radio surroundings, this approach is not applicable and the Location Recognition is based only on the positioning data provided by the ECF without the WSD mechanism.

2) "Available FP" Procedure: During a (sliding) window of T [s], when a Wi-Fi scan is performed, a S^k matrix (as defined in Sec. III.C.2) is obtained. Given the FP (F^l) of a generic

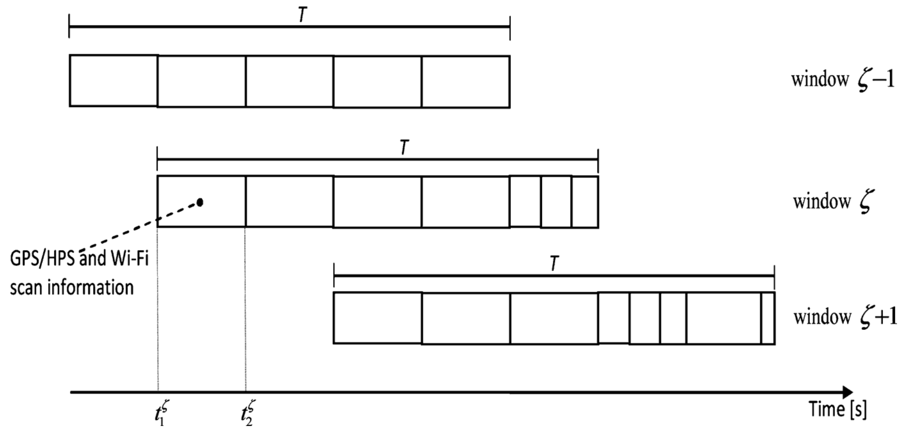

Fig. 2. Temporal window evolution.

location l available in the reference DB, the set of Common Access Points (CAPs), in terms of APs' *MAC Address*, between the sets $col_1 S^k$ and F^l, is a set represented by their intersection. Formally:

$$\Pi^{lk} = \{F^l \cap col_1 S^k\} \quad (14)$$

whose t-th element is π_t^{lk} and its cardinality is $card\left(\Pi^{lk}\right)$. Thus, for each k-th scan and l-th location, whose FP is available, we compute a match score value $\Im^{lk}$ by using the following equation:

$$\Im^{lk} = \sum_{t=1}^{card\left(\Pi^{lk}\right)} \hat{W}^k\left(\Omega\left(col_1 S^k, \pi_t^{lk}\right), \Omega\left(F^l, \pi_t^{lk}\right)\right) \quad (15)$$

where functions $\Omega(\cdot,\cdot)$ and $\hat{W}^k(\cdot,\cdot)$ are defined as in (3) and (4). Considering (15), the value of the match score is directly proportional to the number of CAPs. This is coherent with the fact that the more APs are in common among scans, the higher the matching value is. Considering that a time window of T [s] contains Z^ζ scans, when the window finishes and the location recognition process ends, the considered overall match score value, for a given location l, is:

$$\Im^l = \frac{1}{Z^\zeta}\sum_{k=1}^{Z^\zeta} \Im^{lk}, \forall k \in \left[1, Z^\zeta\right]. \quad (16)$$

In order to decide in which location the Smartphone is located, we do not select the location with highest $\Im^l$ but the location(s) with match score(s) above a given threshold $\gamma_\Im$. This approach, on one hand, avoids considering locations with very low match score values but, in the same time, implies the possible physical overlapping of locations. If $\gamma_\Im$ is too low then the procedure cannot distinguish the location(s) where the Smartphone is among all the considered ones. On the contrary a too high $\gamma_\Im$ often does not allow recognizing any location. Practical experiments have shown that $\gamma_\Im = 0.1$ is a good trade-off.

IV. Check-In Application With the Stay Detection

As schematized in Fig. 1, LRACI is a *Location Recognition* method that includes the concept of *Stay Detection*, so to obtain a robust *Check-In* algorithm. A key point is the sliding window *Permanence* mechanism of LRACI, which is managed over time as reported in (17).

$$\begin{cases} T_P^{l,\zeta} = T & \zeta = 1 \\ T_P^{l,\zeta} = T_P^{l,\zeta-1} + \left(t_2^\zeta - t_1^\zeta\right) & \zeta > 1 \end{cases} \quad (17)$$

ζ represents the window's number and T is the duration of the temporal window. t_1^ζ is the time instant in which the ζ-th window begins. It coincides with the acquisition time of the first positioning/Wi-Fi data for the ζ-th window. t_2^ζ is the time in which the second element of the ζ-th windows is available. In the successive iteration $(\zeta + 1)$ the window is shifted forward of one element. The quantity $T_P^{l,\zeta}$ is the *Permanence* in the generic location l, computed when the window has reached the temporal dimension T, at the end of the $\zeta - th$ iteration. If the same location l is recognized for Ψ successive windows the *Permanence* is $T_P^l = T_P^{l,\Psi}$. Fig. 2 schematizes the computation in (17). If T_P^l is longer than the predefined *Stay Length* T_{SL}^l, for the location l, the conclusion is that a user has really stayed in a location and is not just passed near it. In this case, the user is considered automatically checked-in.

All the operations, analytically detailed in Section III, are schematically reported in Fig. 3 where the overall architecture of LRACI is shown.

V. Performance Investigation

The proposed performance investigation has two aims. The first aim concerns the evaluation of the LRACI algorithm by showing results about the performance of the correct location recognition and of the correct Check-In obtained through real experiments carried out through different Smartphones in which the LRACI has been implemented. The second aim is related to the LRACI-based applications within the Cloud. In practice, Check-In information, acquired by employing LRACI, are used as a service in the Cloud itself. The results, in this case, have been carried out by realizing an *ad hoc* simulator in which LRACI is used in the Cloud to allow people avoiding crowded locations. It will provide, as a practical result, an efficient management of the daily-activities of the Cloud's users.

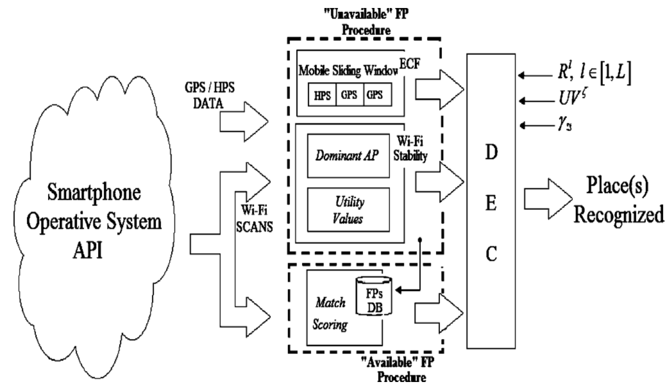

Fig. 3. Location Recognition Scheme.

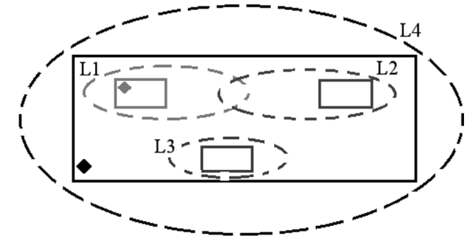

Fig. 4. Radio and physical representation of the considered locations.

A. LRACI Evaluation

All data used in this work have been acquired and processed by using two different Smartphones: *HTC Dream* (Terminal1) and *Samsung Galaxy S* (Terminal2), both equipped with Android OS, in which the LRACI procedures previously described have been implemented as a software application. Four different locations have been considered. Each location has different characteristics, described below, and is situated inside a single building (a Department) of the University of Genoa. Each location has its own radio signals coverage defined below. These signals, coming from the overall department, even if not dedicated to LRACI tests may be captured, with different *RSS* s, also in the surroundings of the locations.

1) Implemented Test Application: LRACI has been tested by using a *Check-In* application written for the aim. This application starts retrieving user's favorite locations from the DB (mentioned in Section III). The application allows the user to indicate its behavior, i.e., to say when he enters or leaves a location, which is defined as *Ground Truth* (GT). The GT will be used to perform a comparison with the location computation carried out by LRACI. The *Stay Length* has been fixed to 10 minutes for each location in our test. Every time a user indicates he is entering a location, a countdown with duration equal to the expected *Stay Length* is started. When the *Location Recognition* method detects the stay in a location (as described in Section IV), a feedback popup is shown to the user that may confirm the correct detection or to report a wrong behavior. Every user action has been logged and the logs have been processed offline to generate the results.

2) Locations Description: The first location considered $(L1)$ is a laboratory situated at the third floor of the building. Its area is approximately 70 $[\mathrm{m}^2]$ and there is one AP directly within it. The second considered location $(L2)$ is another laboratory, with similar area, situated at the same floor of the first one. No AP is located directly within it. $L2$ is located close to $L1$ even if the locations are physically distinguished. The closeness causes a partial overlapping between the two locations in terms of radio coverage. The radio coverage referred to a location l is defined as the physical area where the same Wi-Fi signals that are received within the location perimeter are detected, even if the Smartphone is outside the location. It includes the location area and its surroundings. In practice, the mentioned overlap means that some Wi-Fi signals received in $L1$ may be captured also in $L2$, with different *RSS*. The consequent inconvenient is represented by possible misunderstandings during the recognition phase. The third location $(L3)$ is the canteen situated at the ground floor. Similarly to $L2$, no AP is located directly in it but it is far from $L1$ and $L2$. This implies that $L3$ is well distinguished also in terms of radio coverage (i.e., Wi-Fi signals received at the third floor are not, or weakly, received at the ground floor of the mentioned building). The last considered location $(L4)$ is the biggest one. It is the whole building composed of four floors and, obviously, contains, from the physical and Wi-Fi radio coverage viewpoint, all the other locations. For the sake of simplicity, all locations have the same physical centre $\mathbf{C}^l = [44.4035, 8.9584]\,, \forall\, l \in [1, 4]$; this assumption is justified because they are located within the same building. Locations have the following radius: $R^l = 20\ [\mathrm{m}]\ \forall\, l \in [1, 3]$ and $R^4 = 50\ [\mathrm{m}]$. Fig. 4 clarifies radio and physical relationships, without reference to real sizes, among the considered locations. In particular, the dotted lines represent the Wi-Fi radio coverage, defined above, while the continuous line represents the physical size of each single location. The rhomboid symbol, within $L1$, means that this location directly contains an AP.

3) Numerical Results: The analysis is focused on $L1$. The aim is to show the behavior of the following schemes over time: GPS/HPS, GPS/HPS with ECF and FP-based approach. The first two schemes provide a position estimation based on latitude and longitude and may be analyzed by showing the distance in [m] of the estimated position with respect to $L1$ centre, as done in Fig. 5. The results are shown for a single realization when a user enters $L1$. The Smartphone employed is the *HTC Dream*, which is held by the user entering location $L1$ during the entire duration of the test. The continuous line in Fig. 5 represents the radius of the location (20 [m]). The dash-dot line represents the distance from the location centre estimated by using GPS/HPS information without applying the ECF (*Unfiltered Distance*). The dashed line represents the ECF filtered distance. The filtered distance is less affected by errors, typically due to the heterogeneity of the positioning information, but the location recognition based on this approach is not sufficiently reliable. The distance from the centre of the position estimated by GPS/HPS is always above the radius and, as a consequence, the user is considered out of location $L1$ and is not checked-in. If ECF correction is used, the result slightly improves because, for a limited time of about 2 minutes between minutes 13.5 and 15.5, the distance from the centre of the position estimated by GPS/HPS corrected by ECF is below the radius. This is clear in Fig. 6, which reports the performance in terms of location recognition with respect to the GT (manually set by the Smartphone

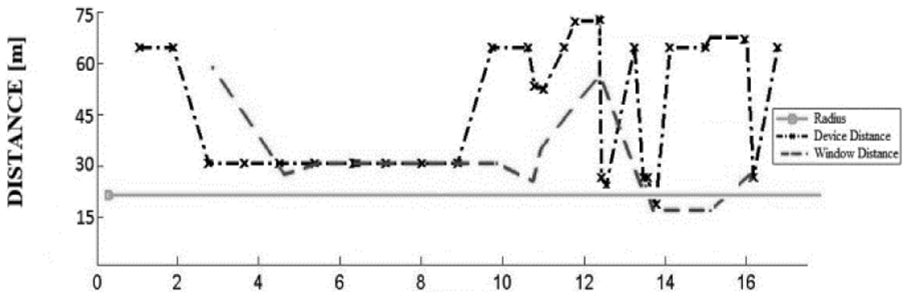

Fig. 5. Distance from the location centre of the estimated position—$L1$.

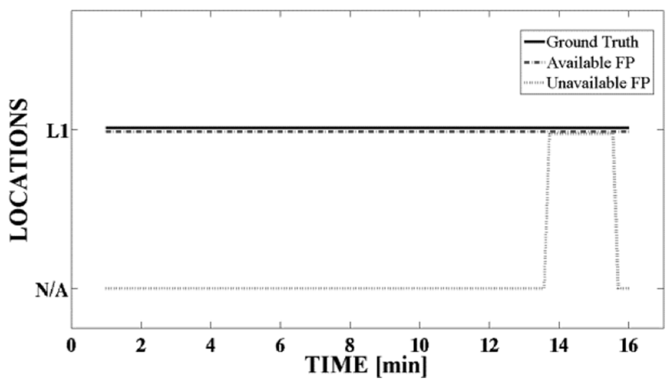

Fig. 6. Location recognition result over time—$L1$.

user). GT is shown through a continuous line and the GPS/HPS corrected by ECF through a dotted line, identified as *"Unavailable FP" Procedure* (i.e., the FP of $L1$ is supposed unavailable in the reference DB). As a consequence, applied to the *Check-In* applications (described above), the real stay (i.e., the *Permanence*) has been wrongly estimated. It is however worth noting that if the proposed ECF were not applied, the *Location Recognition* process would have been completely ineffective. Fig. 6 includes the location recognition performance of the proposed FP based method. It is shown through a dashed line and is called *"Available FP" Procedure* (i.e., in this case, the FP of $L1$ is supposed available in the reference DB). The location is correctly recognized, except for the first two minutes after the real entrance in the location (the beginning of the continuous line), which is the time needed to fulfill the first time window T (set to 120 [s]). Considering that the T^1_{SL} has been fixed to 10 minutes, a robust Check-In is guaranteed if the $L1$ FP is available.

Similar results are shown in Figs. 7 and 8 for location $L3$ that, differently from L1, does not contain a dedicated AP in the considered location. In this case LRACI operates in an opportunistic way by using radio signals transmitted by other APs, which characterize, from the radio coverage viewpoint, the area. Again the *"Available FP" Procedure* offers satisfactory performance.

The overall LRACI performance may be seen in Table I, which shows the percentage of correct Check-In in the locations during tests. In practice, it is the confusion matrix of the locations considered in this performance evaluation. In the table also a " *No Location* " case has been considered. In this case, data collections have been acquired randomly in locations different from $L1$, $L2$, $L3$ and $L4$.

FPs of locations have been obtained by using the procedure described in Section III and, in order to build them, for each considered locations, 50 data collections, taken in different points of the locations, of 10 minutes each, have been acquired. 25 collections have been realized with Terminal1 and 25 with Terminal2. During the mentioned data collections, a certain number of Wi-Fi scans have been carried out and employed to compute the locations' FPs. Scans results are stored in the DB server where other data referred to locations are contained (centre and radius). When the Check-In application described at the beginning of this section starts, it sends a query to the DB server and obtains FP and other data about the considered locations. In this experiment just one location, $L4$, does not have any FP. This setting has been used to highlight the effect of the joint action of the GPS/HPS based approach and of the FP-based one.

The percentage values reported in Table I have been computed by averaging the results obtained by 50 data collections, not the same ones used to build the locations' FPs but realized with the same criteria.

In general the overall LRACI performance is very good. It has, on average, an accuracy of 89.8%. If $L1$ is considered, the accuracy is high also thanks to the presence of an AP in the location. In general, when a location contains an AP, its radio signal dominates the others. It characterizes the FP of the location and enables an efficient recognition. The absence of a dedicated AP causes a degradation of the location recognition accuracy. If a location radio coverage is partially shared with an adjacent location, it may create possible confusion. It is the case of $L2$: about 11% of recognitions are not correct because confused with $L1$. This problem has a lower impact if the locations are not adjacent (i.e., if they do not share Wi-Fi signals). It is the case of $L3$. The accuracy obtained for $L4$ is high because, in this specific case, the GPS/HPS positioning information filtered by the proposed ECF is efficient due to the very large $L4$ radius (with respect to the other locations).

B. LRACI Applied in the Cloud

1) Crowdedness Monitoring Service With LRACI: Check-In information is used to generate information about people density in a specific location. The service is called Crowdedness Monitoring Service (CMS). People density, measured in $[\mathrm{customer}/\mathrm{m}^2]$, is identified as Estimated Mean Crowdedness (EMC). CMS is inspired on a similar service, aimed at monitoring the urban traffic, for navigation applications described in [7] and has been simulated as detailed in the sub-Section below to highlight the impact of the LRACI algorithm on such

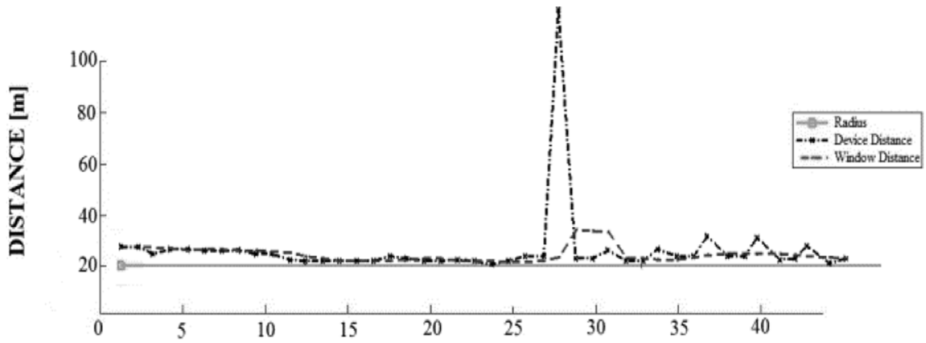

Fig. 7. Distance from the location centre of the estimated position—$L3$.

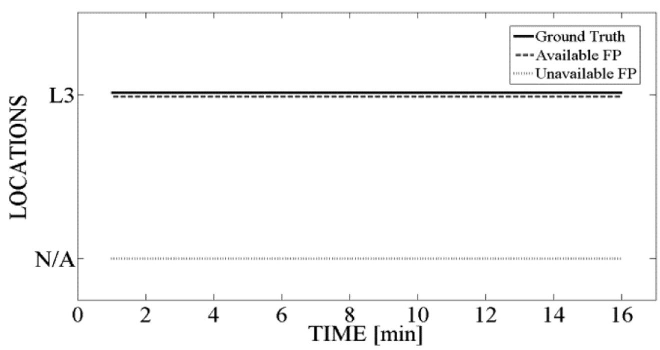

Fig. 8. Location recognition result over time—$L3$.

TABLE I
RECOGNITION PERCENTAGE ACCURACY (CONFUSION MATRIX)

Real Location	*Recognized Places (%)*				
	L1	*L2*	*L3*	*L4*	*No Location*
L1	**94.5**	0	0	0	5.5
L2	10.8	**81.1**	0	0	8.1
L3	0	0	**91.3**	0	8.7
L4	0	0	0	**98.0**	2.0
No Location	14.4	2.3	0	0	**83.9**

a service. Simulated CMS is based on Check-In information obtained from a large number of Smartphones, which are components of the Cloud, employed in a wide area. Obviously, Smartphones are supposed equipped with the LRACI algorithm and send Check-In data to a CMS processing centre. In practice, Check-In data of Smartphones are gathered to find the EMC in a given location (such as a shop, a post office or a restaurant). The described service is completed with the *less-crowded location* finding function of the Crowdedness Monitoring platform. It is used to suggest the less-crowded among the locations of interest (*less-crowded suggestion*) in the user surroundings to Cloud users. In more detail, it can suggest the location (among equivalent alternatives) in the surroundings, which allows wasting less user time. In the case simulated below, users/customers entering a commercial area can choose between two equivalent shops at different distances (and travel time) from the entrance of the commercial area. CMS provides the *less-crowded suggestion* to the users by indicating the less crowded shop so allowing users' shopping in the shortest time. Check-In information provided by LRACI plays a crucial role. Conventionally, information about the crowdedness of a location is unavailable or, even if available, the area of information provision is limited to location proximity and practically useless to make choices.

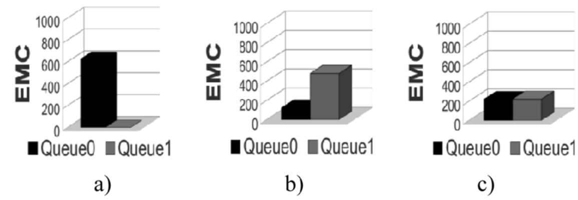

Fig. 9. Estimated Mean Crowdedness with and without Check-In information.

From the practical viewpoint, the realized *ad hoc* tool simulates the described scenario where customers enter a commercial area with a random inter-arrival time $(1/\lambda = 1\ [\mathrm{min}])$ exponentially distributed. Both mentioned locations (i.e., the shops) are modeled as a single queue with one server with constant service time $1/\mu = 10\ [\mathrm{min}]$. As previously said, the two considered locations, called *Queue0* and *Queue1*, are differently located with respect to the commercial area entrance: *Queue0* is nearer than *Queue1* to the entrance. Time needed to reach *Queue0* from the entrance (i.e., the travel time) is $T_0 = 1\ \mathrm{minute}$ while to reach *Queue1* is $T_0 = 2\ \mathrm{minutes}$.

2) Simulation Results: The first set of results is referred to three situations:

a) customers do not use the CMS and simply choose the nearest queue (*Queue0*);
b) as in situation a) but if the number of customers queued in *Queue0* is larger (or equal) to 50 the new entering customers prefer the farthest queue (*Queue1*);
c) customers employ the CMS, know the estimated crowdedness of each queue and choose the less crowded location.

In the simulations, whose duration is always 1000 minutes of simulated time, possible queue changes (i.e., a user switches from one queue to another one) have not been considered for the sake of simplicity. Queues' crowdedness has been estimated by employing the number of Check-Ins, computed by using LRACI, whose accuracy has been supposed to be 100% for the first set of results in Fig. 9.

In particular, Fig. 9(a) shows the EMC in the a) case and Fig. 9(b) shows the ECM in the b) case. These results are obvious and represent comparison references for the case c) whose ECM is shown in Fig. 9(c). The estimated mean crowdedness

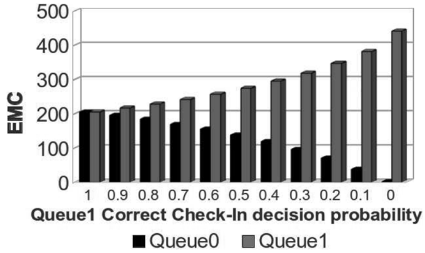

Fig. 10. Estimated mean crowdedness with different correct Check-In detection probability.

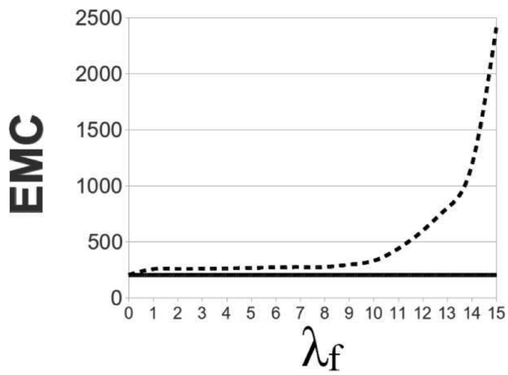

Fig. 11. Estimated mean crowdedness with fake Check-Ins.

knowledge has a clear impact: customers' distribution in the two queues is equalized in Fig. 9(c). It represents a "global" advantage: all customers spend a fair quantity of time in the queues.

The impact of the Check-In algorithm accuracy on the CMS has been reported in Fig. 10 where the EMC for both queues is shown in dependence on the correct Check-In probability (i.e., the accuracy) of *Queue1*. In the simulations we assumed constant correct Check-In and, as a consequence, crowdedness information for *Queue0* and probabilistic correct Check-In detection for *Queue1*. The obtained results show that the equilibrium is obtained only if all users apply Check-In algorithms with good accuracy performance. Actually, in the case of the proposed LRACI, whose accuracy is around 90% (i.e., correct Check-In detection is about 0.9), the two queues are almost in equilibrium.

The last part of the simulations has been dedicated to the *robustness* of the Check-Ins. The effectiveness of the Stay Detection mechanism, which represents a peculiar feature of the proposed LRACI, has been tested. In more detail, possible fake Check-Ins provided by customers that are not queued (e.g., they are just passing near the queue) have been considered in the simulations with an inter-arrival time with average $1/\lambda_f$ [min], exponentially distributed. The effect of the Stay Detection, set to 5 minutes in the reported results, has been reported only for *Queue0* because the results for the other queue are similar. Fig. 11 shows the estimated mean crowdedness by varying λ_f: the dashed line represents the case in which the Stay Detection is inactive; the continuous line represents the case in which it is active, as in LRACI. The estimated mean crowdedness grows significantly with λ_f if the Stay Detection mechanism is inactive. It implies erroneous and misleading information for customers and inefficient CMS in the Cloud. If LRACI Stay Detection feature is used, CMS is not influenced by faked Check-Ins. It helps to efficiently manage daily-activities of Cloud's users by reducing wasted time.

VI. Conclusions

The paper introduces a new Location Recognition algorithm for Automatic Check-In applications called LRACI. LRACI is implemented over Smartphones and integrated in the modern Cloud Computing platform so representing a service for Cloud end-users. The proposed *Location Recognition* method is based on the joint exploitation of GPS/HPS positioning information, corrected by using a simple sliding window filtering (ECF), and of a novel Wi-Fi *FingerPrint* (FP) definition. The proposed FP definition is independent of the Received Signal Strengths (*RSS* s) measured absolute values because it considers only the order relation among them. As a consequence, the proposed method, tested through real experiments, can be employed with heterogeneous Smartphone platforms, which sense different AP RSS values from the same positions and orientations, without any impact on the location recognition accuracy which is about 90%. Simulative results about the employment of LRACI in the Cloud to provide a Crowdedness Monitoring Service, aimed at suggesting less crowded locations to users/customers, have shown that LRACI helps to efficiently manage daily-activities of Cloud users by correctly estimating the presence of people in a given location and reducing, as a consequence, wasted time.

From a more theoretical viewpoint, the idea of determining a fingerprint which is not based on absolute values but on the order relation among the measures has a more general meaning. In this view, location recognition is an application field, but the idea may be applied to other scenarios, where the measures are device-dependent.

References

[1] FourSquare. Sep. 2011 [Online]. Available: http://www.foursquare.com
[2] Gowalla. Sep. 2011 [Online]. Available: http://gowalla.com
[3] T. Grance and P. Mell, "The NIST definition of cloud computing," NIST, U.S. Dept. Commerce, Special Pub., 2011, pp. 145–800.
[4] Y. Zheng, "location-based services on the Cloud," Microsoft Research [Online]. Available: http://research.microsoft.com/apps/pubs/?id=102318
[5] LOC-AID. 2012 [Online]. Available: http://www.loc-aid.com/
[6] Mobile Commerce Daily. Jan. 2011 [Online]. Available: http://www.mobilecommercedaily.com/2011/01/06/att-to-launch-cloud-based-lbs-mobility-data-offering
[7] K. Tamai and A. Shinagawa, "Platform for location-based services," *FUJITSU Sci. Tech. J.*, vol. 47, no. 4, pp. 426–433, Oct. 2011.
[8] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, J. Tabert, P. Powledge, G. Borriello, and B. Schilit, "Place lab: Device positioning using radio beacons in the wild," in *Proc. Pervasive Computing Conf. Lecture Notes in Computer Science*, Munich, Germany, May 2005, pp. 116–133.
[9] F. Alizadet-Shabdiz and E. J. Morgan, "System and Method for Estimating Positioning Error Within a Wlan-Based Positioning System," U.S. Patent 7856234, Dec. 21, 2010, Primary Class 455/456.1.
[10] H. Tian, L. Xia, and E. Mok, "A novel method for metropolitan-scale Wi-Fi localization based on public telephone booths," in *Proc. IEEE/ION PLANS*, Indian Wells, CA, USA, May 2010, pp. 357–364.
[11] E. Jung and S. Kim, "Location-awarness for mobile users in the pervasive network environment," in *Proc. Int. Conf. Inf. Networking*, Busan, South Korea, Jan. 2008, pp. 1–5.
[12] M. Rabinowitz and J. J. Spilker , Jr, "A new positioning system using television synchronization signals," *IEEE Trans. Broadcasting*, vol. 51, no. 1, pp. 51–61, Mar. 2005.

[13] D. Manandhar, S. Kawaguchi, M. Uchida, M. Ishii, and H. Torimoto, "IMES for mobile users social implementation and experiments based on existing cellular phones for seamless positioning," in *Proc. Int. Symp. GPS/GNSS*, Tokyo, Japan, Nov. 2008, pp. 1–6.
[14] Skyhook, The World Leader in Location Information, Context and Intelligence. 2011 [Online]. Available: http://www.skyhookwireless.com
[15] PlaceEngine. 2011 [Online]. Available: http://www.placeengine.com
[16] M. P. Wylie and J. Holtzman, "The non-line of sight problem in mobile location estimation," in *Proc. Int. Conf. Universal Personal Commun.*, Cambridge, MA, USA, Aug. 2002, pp. 827–831.
[17] H. Miao, K. Yu, and M. J. Juntti, "Positioning for NLOS propagation: Algorithm derivations and cramer-rao bounds," *IEEE Trans. Veh. Technol.*, vol. 56, no. 5, pp. 2568–2580, Sep. 2007.
[18] B. J. Shin, K. W. Lee, S. H. Choi, J. Y. Kim, W. J. Lee, and H. S. Kim, "Indoor WiFi positioning system for Android-based smartphone," in *Proc. Int. Conf. Inf. Commun. Technol. Convergence*, Jeju, South Korea, Nov. 2010, pp. 319–320.
[19] Y. Chon and H. Cha, "Lifemap: A smartphone-based context provider for location-based services," *IEEE Pervasive Computing*, vol. 10, no. 2, pp. 58–67, Feb. 2011.
[20] L. Vu, Q. Do, and K. Nahrstedt, "Jyotish: A novel framework for constructing predictive model of people movement from joint WiFi/Bluetooth trace," in *Proc. IEEE Int. Conf. Pervasive Computing Commun.*, Seattle, WA, USA, Mar. 2011, pp. 54–62.
[21] M. Lee, H. Yang, D. Han, and C. Yu, "Crowdsourced radiomap for room-level place recognition in urban environment," in *Proc. IEEE Int. Conf. Pervasive Computing Commun. Workshops*, Mannheim, Germany, Mar. 2011, pp. 648–653.
[22] Y. Chon, E. Talipov, and H. Cha, "Autonomous management of everyday places for a personalized location provider," *IEEE Trans. Syst., Man, Cybern. C*, vol. 42, no. 4, pp. 518–531, Jul. 2012.
[23] D. Kelly, R. Behan, R. Villing, and S. McLoone, "Computationally tractable location estimation on wifi enabled mobile phones," in *Proc. Int. Signal Syst. Conf.*, Dublin, Ireland, Jun. 2009, pp. 1–6.
[24] R. A. Valenzuela, O. Landron, and D. L. Jacobs, "Estimating local mean signal strength of indoor multipath propagation," *IEEE Trans. Veh. Technol.*, vol. 46, no. 1, pp. 203–212, Feb. 1997.
[25] K. Kaemarungsi and P. Krishnamurthy, "Properties of indoor received signal strength for WLAN location fingerprinting," in *Proc. 1st Annu. Int. Conf. Mobile and Ubiquitous Syst.: Networking and Services*, Boston, MA, USA, Aug. 2004, pp. 14–23.
[26] M. J. Johnson, J. S. King, R. S. Shryock, J. Kiviniemi, and T. Heinonen, "Analysis of a signal strength based positioning system for commercial environments," in *Proc. 2nd IEEE Consumer Commun. Networking Conf.*, Las Vegas, NV, USA, Jan. 2005, pp. 533–538.

Igor Bisio was born in Novi Ligure (Alessandria), Italy in 1978. He got his "Laurea" degree in Telecommunication Engineering at the University of Genoa, Italy in 2002 and his Ph.D. degree in 2006. He is currently Assistant Professor and he is member of the Digital Signal Processing (DSP) and Satellite Communications and Networking (SCNL) Laboratories research staffs at the University of Genoa. He is author of about 70 scientific papers including international journals, international conferences and book chapters. His main research activities concern Resource Allocation and Management for Satellite and Space Communication systems, Signal Processing over Portable Devices such as Smartphones, Context and Location Awareness, Adaptive Coding Mechanisms, Indoor Localization, Security and e-health Applications.

Mario Marchese was born in Genoa, Italy, in 1967. He received the "Laurea" degree (cum laude) in electronic engineering and Ph.D. degree in telecommunications from the University of Genoa, Genova, Italy, in 1992 and 1996, respectively. He is currently an Associate Professor with the DITEN Department, University of Genoa. His main research activity concerns satellite and radio networks, transport layer over satellite and wireless networks, quality of service and data transport over heterogeneous networks, and applications for smartphones.

Fabio Lavagetto was born in Genoa on August 6th, 1962. He is currently full professor in Telecommunications at the DITEN Department of the University of Genoa. Since 2008 he is Vice-Chancellor with responsibility for Research and Technology Transfer at the University of Genoa. Since 2005 he is Vice-Chair of the Institute for Advanced Studies in Information Technology and Communication. Since 1995, he is the head of research of the Digital Signal Processing Laboratory of the University of Genoa. He was General Chair of several international scientific conferences and has authored over 100 scientific publications in international journals and conferences.

Andrea Sciarrone was born in Livorno, Italy in 1984. He got his Bachelor Degree in Telecommunication Engineering at the University of Genoa in 2007; in 2009 he achieved a Master Degree Cum Laude in Telecommunication Engineering. He currently is a Ph.D. Student at the University of Genoa. His main research activity concerns: Signal Processing over Portable Devices such as Smartphones, Context and Location Awareness, Indoor Localization, Security and e-health Applications, Android Operative System.