SignalSLAM: Simultaneous Localization and Mapping with Mixed WiFi, Bluetooth, LTE and Magnetic Signals

Piotr Mirowski, Tin Kam Ho, Saehoon Yi
Statistics and Learning Research Department, ECT
Bell Laboratories, Alcatel-Lucent
600 Mountain Avenue, Murray Hill, NJ 07974, USA
Email: piotr.mirowski@computer.org

Michael MacDonald
Wireless Communications Research Department
Bell Laboratories, Alcatel-Lucent
600 Mountain Avenue, Murray Hill, NJ 07974, USA

Abstract—**Indoor localization typically relies on measuring a collection of RF signals, such as Received Signal Strength (RSS) from WiFi, in conjunction with spatial maps of signal fingerprints. A new technology for localization could arise with the use of 4G LTE telephony small cells, with limited range but with rich signal strength information, namely Reference Signal Received Power (RSRP). In this paper, we propose to combine an ensemble of available sources of RF signals to build multi-modal signal maps that can be used for localization or for network deployment optimization. We primarily rely on Simultaneous Localization and Mapping (SLAM), which provides a solution to the challenge of building a map of observations without knowing the location of the observer. SLAM has recently been extended to incorporate signal strength from WiFi in the so-called WiFi-SLAM. In parallel to WiFi-SLAM, other localization algorithms have been developed that exploit the inertial motion sensors and a known map of either WiFi RSS or of magnetic field magnitude. In our study, we use all the measurements that can be acquired by an off-the-shelf smartphone and crowd-source the data collection from several experimenters walking freely through a building, collecting time-stamped WiFi and Bluetooth RSS, 4G LTE RSRP, magnetic field magnitude, GPS reference points when outdoors, Near-Field Communication (NFC) readings at specific landmarks and pedestrian dead reckoning based on inertial data. We resolve the location of all the users using a modified version of Graph-SLAM optimization of the users poses with a collection of absolute location and pairwise constraints that incorporates multi-modal signal similarity. We demonstrate that we can recover the user positions and thus simultaneously generate dense signal maps for each WiFi access point and 4G LTE small cell, "from the pocket". Finally, we demonstrate the localization performance using selected single modalities, such as only WiFi and the WiFi signal maps that we generated.**

Keywords—WiFi; LTE; localization; SLAM; crowd-sourcing; kernel methods

I. Introduction

Indoor localization and mapping is a key enabler for pervasive computing. While location-based services, that can for instance accurately localize a smart phone in an indoor GPS-denied environment, are used on daily basis, another need has arisen with the challenge of optimizing the deployment of telecommunication networks. For instance, precise localization is a recurring hurdle when deploying networks of 4G Long Term Evolution (LTE) small cells. These small cells have limited range (comparable to the typical WiFi access point coverage) but can be installed at multiple locations throughout office spaces, residential areas or public spaces. Cost and quality of service are important considerations when planning the placement of such small cells and thus generate the need to build precise 4G LTE signal coverage maps.

Other than providing communication links with additional bandwidth, another advantage of the 4G LTE small cells is that they could be used, in combination with WiFi, as inputs to a localization system.

Both location-based services and network deployment optimization typically rely on measuring a collection of radio-frequency (RF) signals, characterized for instance by their Received Signal Strength (RSS), for WiFi, or by their Reference Signal Received Power (RSRP), for LTE, along with spatial location. The time and effort to manually collect these signal *fingerprints* can be prohibitive and has prompted research either on automated methods using mobile robots [30], [34] or on methods for reconstructing RF maps from (potentially crowd-sourced) pedestrian trajectories. The latter fall into the general category of Simultaneous Localization and Mapping (SLAM) and unsupervised mapping for RF signals and are reviewed in section I-B.

While using autonomous WiFi mapping robots may prove impractical in some buildings as it involves a deployment cost [30], it is worth noting that the pedestrian position could be recovered thanks to a wearable system consisting of a color and depth (RGBD) camera and a computer running real-time vision-based odometry [44] or even a full SLAM algorithm using the RGBD camera combined with a laser range [11]. In our research, we have decided not to rely on any vision systems and to run a cruder version of SLAM from the user's pocket, using only the sensors available on a smartphone.

A. Objective: Crowd-sourced RF Mapping from the Pocket

In our study, we propose to build multi-modal RF signal maps, that can include WiFi, 4G LTE or Bluetooth, without any additional effort from the experimenter, simply by exploiting all of the sensor signals recorded on one or multiple off-the-shelf smartphones. Our goal is to enable "crowd-sourced RF mapping from the pocket". In a preferred scenario the user(s) would simply collect RF signals while walking freely through a building, taking care of their daily activities. In the meanwhile, they would be collecting time-stamped WiFi and Bluetooth RSS, 4G LTE RSRP, magnetic field magnitude, GPS reference points when outdoors, Near-Field Communication (NFC) tag or QR code readings at specific landmarks and Pedestrian Dead Reckoning (PDR) based on inertial data.

Our system relies on a state-space model, where the trajectory of the user is unknown and depends both on the *dynamics* coming from a pedestrian motion model and on multi-sensor *observations*, including WiFi or LTE signal. Unlike existing sensor fusion algorithms for tracking a user indoors using a motion model and WiFi [5], [15], [24], we do not know the RF signal map in advance: our objective is to reconstruct it from the data acquired by a freely moving pedestrian wearing a commercial grade smartphone in her pocket, with limited or no human intervention.

The two building blocks of our system are the pedestrian dead reckoning with position fixes (see the next section for the limitation of PDR on smartphones) and the SLAM algorithm adapted for RF signal data. SLAM provides a solution to the challenge of building a map of observations without knowing the location of the (moving) observer.

The two innovations in our research are the use of pose-invariant PDR that lets the user place the phone in any pocket of a trouser, and our modified version of SLAM, called SignalSLAM, which optimizes the users' poses thanks to a collection of absolute location and pairwise constraints that incorporate multi-modal signal similarity, including the WiFi RSS, Bluetooth RSS, LTE RSRP or even the magnitude of the magnetic field.

1) Pedestrian Dead Reckoning and its Limitations: Pedestrian Dead Reckoning (PDR) [22] has been presented as a possible solution for localization in GPS-deprived areas [27]. It requires an Inertial Measurement Unit (IMU) constituted of at least a 3-axis accelerometer and 3-axis compass. In its simplest form, PDR consists in a step counter that detects the peaks in the vertical component of acceleration (i.e., every time that the foot hits the floor) and reads the heading of the smartphone from the compass.

A recent survey of the numerous PDR methods [19] investigated so far listed a large collection of equipment, such as foot-mounted Inertial Measurement Units (IMU) or smartphones, of techniques for step detection, heading estimation or inertial navigation as well as for their integration into hybrid systems with absolute position fixes in order to correct the dead reckoning output. It highlighted the need for using position fixes to cope with long-term drift, or to use additional sources of information such as RF signal strength measures and a known map of RF signal fingerprints.

Among the recently-published hybrid localization techniques using PDR on smartphones, we notice that they typically require the user to hand-held the smartphone during the walk [15], [17], [25]. For better accuracy in the estimation of the step length or even the heading direction, it is preferable to use foot-mounted sensors [37], [38].

B. Review of Collaborative RF Localization and Mapping

1) Unsupervised and Semi-Supervised Mapping: [45] introduced 2D map building without localization and without a motion model, using only visibility, bearing or distance to 2D landmarks and manifold learning (dimensionality reduction through Multi-Dimensional Scaling) over vectors of observations. This technique was extended in [8] to vectors of WiFi RSSI observations, assuming an RF propagation model with RSSI monotonously decreasing with distance from the access point. The manifold learning method for building RF maps was further refined in [35] into an iterative, incremental, scheme where RF localization alternates with manifold learning.

2) RF Simultaneous Localization and Mapping: Simultaneous Localization and Mapping (SLAM) [10] is the standard mathematical framework for iteratively optimizing 1) the trajectory (sequence of poses) or *dynamics* of a user (robot) based on the predictions of her motion model as well as on the *observations* such as laser range, visibility or position of landmarks and 2) the position of the landmarks and the map itself. SLAM has first been adapted to WiFi signals as the WiFiSLAM algorithm [12], where the state space model is modeled by a Gaussian Process Latent Variable Model. The main weakness of this algorithm is the cubic dependence on the number of time steps.

Probabilistic Graphical Models, where the optimization of state-space models is done using particle filters [9], can be used to implement WiFi SLAM models, following [6], [41]. In these cases, each particle carries not only the position and orientation of the user, but also a map of the WiFi access points. An extension of particle filter models to include other multi-modal RF signals would be non-trivial, though, due to the need to specify conditional dependencies across multiple types of variables.

Our approach to RF SLAM is based on the extension to WiFi [20] of the GraphSLAM algorithm [16]. The latter is typically used in robotics to do *bundle adjustment* (i.e. loop closure) on a graph of robot poses and is explained in section II-C. One of the contributions of [20] was to bring the computational complexity from cubic to quadratic in the length of the optimized sequence. The main difference between our algorithm and the one by Huang et al. [20] is that their model assumes that two nearby positions of the user, $\mathbf{x}_a$ and $\mathbf{y}_b$, should be subject to similar signal strength observations S_a and S_b according to a Gaussian process. In our approach, we take the reverse approach, claiming that it is the similarity, in signal space, that conditions the proximity in physical space. As detailed in section II-D, we can pre-compute the *kernel* similarity between the RF signal at each time interval during the trajectory and we can easily combine multiple sources of RF signal by multiplying multiple kernel matrices.

3) RF SLAM with Building Blueprints: An alternative solution to the SLAM problem relies on existing maps [21], [25], building blueprints (which can be obtained from evacuation plans [36]) or even assumptions about the architecture of the indoor space [37]. Here, a state-space model such as a particle filter [9], extended Kalman filter or even dynamic programming can be used to track the hidden trajectory of the user in a semantic (e.g., traversability) 2D map of obstacles. We wanted however to achieve maximal flexibility and not

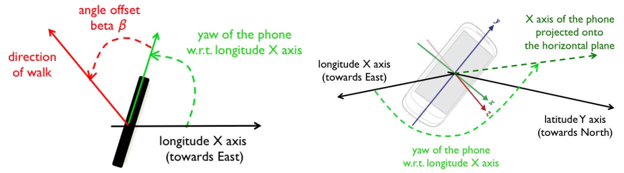

Fig. 1. Orientation and yaw angle of the smartphone with arbitrary pose. Image credit: Android API (see http://developer.android.com)

to have to rely on blueprint constraints. In our solution, we are able to merely use a geo-referenced satellite image of the building that can easily be obtained on a map search engine.

In the rest of this paper, section II introduces our modification to the GraphSLAM algorithm that optimizes the users poses with a collection of absolute location and pairwise constraints that incorporates multi-modal signal similarity. Section III explains our data acquisition app running on commercial Android OS smartphones placed in the user's pocket. In the Results section IV, we demonstrate that we can recover the user positions and thus simultaneously generate dense signal maps for the WiFi access points and for 4G LTE cells, and we illustrate the localization performance using WiFi fingerprints generated while walking.

II. Methods

A. Pedestrian Dead Reckoning with Position Fixes

We give here a simple overview of a motion model provided by our pedestrian dead reckoning (PDR) system that is invariant to the phone pose. The specific implementation of the phone orientation estimation, step counting and step heading estimation are detailed in sections III-A1, III-A2 and III-A3, respectively.

First, the 3D orientation angle of the phone is estimated at every time point, including the yaw of the phone, noted θ_t (see Figure 1). We call yaw the angle between the longitude axis and the projection of the X axis of the phone coordinate system onto the ground plane. This convention means that when the user is looking at the phone screen with her arm extended towards the North, the yaw will be 0deg, and if extended towards the East, the yaw will be -90deg.

We then assume that the walking motion is always forwards and that the phone is immobile in the user's pocket. The angle between the yaw axis and the front direction of the walk is noted β_t. For simplicity and despite potential shifts in the trouser's pocket, we fix β_t to a constant value β within the time interval between two position fixes.

The last angle that we need to consider is the offset ξ between the arbitrary map coordinate system and the coordinate system defined by the longitude (towards East) and latitude (towards North) axes. We decided to ignore the curvature of the Earth at the scale of a building (at most a few hundred meters) and to consider the (longitude, latitude) coordinate system as orthogonal within that radius, enabling affine transforms.

By noting d_t the length of the stride (corresponding to two steps), $x_{1,t}$ the X coordinate of the user, $x_{2,t}$ the Y coordinate of the user and ϕ_t the increment in the yaw after one stride, the motion model becomes:

$$\begin{aligned} x_{1,t+1} &= x_{1,t} + d_t \cos(\theta_t + \xi + \beta_t) && (1) \\ x_{2,t+1} &= x_{2,t} + d_t \sin(\theta_t + \xi + \beta_t) && (2) \\ \theta_{t+1} &= \theta_t + \phi_t && (3) \end{aligned}$$

As explained in section III-A2, the stride d_t is not calculated at every step but assumed to remain constant between two landmarks.

During the walking trajectory, the PDR can be frequently reset thanks to a collection of landmarks that either encode their own position (e.g., GPS readings, Near Field Communication (NFC) tags or QR codes) or whose position can be known in advance, such as Bluetooth dongles. The position $(x_{1,t}, x_{2,t})$ is simply assigned the k-th landmark's coordinates $(y_{1,k}, y_{2,k})$.

B. Least Squares Calibration of PDR Trajectories

Because of long-term shifts of the phone within the user's pocket or changes of the user stride and because our experimental data protocol involves taking the phone in and out of the pocket to read the position landmarks (see sections III-B and IV-A), the values of β_t and d_t are re-calibrated in each segment of time $I = [t_i, t_b]$ between two landmarks I and J, of respective coordinates $\mathbf{y}_a = (y_{1,i}, y_{2,i})$ and $\mathbf{y}_b = (y_{1,j}, y_{2,j})$.

The calibration consists in minimizing the Least Square Error $||\hat{\mathbf{x}}_j - \mathbf{y}_j||_2$, where $\hat{\mathbf{x}}_j = (\hat{x}_{1,j}, \hat{x}_{2,j})$ is the PDR trajectory re-estimated by starting from $\hat{\mathbf{x}}_i = (x_{1,i}, x_{2,i})$ and iterating the PDR motion model (1), (2) and (3), using stride length $\hat{d}$ and offset angle $\hat{\beta}$:

$$(d, \beta) = \arg \min_{(\hat{d}, \hat{\beta})} ||\hat{\mathbf{x}}_j - \mathbf{y}_j||_2 \qquad (4)$$

Because of the small range of admissible values for the stride (typically between 1m and 2m) and because of the magnitude of the errors introduced by our simplistic motion model, the values of d and β in Eq. (4) are found by grid search with small step increments (e.g., 0.1m and 5deg, respectively).

C. GraphSLAM

The GraphSLAM algorithm is explained in details in [16], [33], with pseudo-code for the 2D version available in [16], therefore we focus here only on its main ideas.

GraphSLAM considers the trajectory of a mobile as a sequence of *poses* $\mathbf{x} = \{\mathbf{x}_t\}$; in the 2D case, each pose $\mathbf{x}_t = (x_{1,t}, x_{2,t}, x_{3,t})$ is a 2D position $(x_{1,t}, x_{2,t})$ and an orientation angle $x_{3,t}$. The sequence of poses can be represented by a chain graph, where each pose is associated to a vertex and each *motion edge* corresponds to a known motion increment (in the case of pedestrian dead reckoning, each edge is a step with stride d_t and yaw θ_t). In the case of PDR, the sequence $\{\mathbf{x}_t\}$ is produced by iterating the equations of the motion model (1), (2) and (3).

Whenever the mobile "comes back", at time t_j, near a location previously traversed at time t_i, and if it is capable of recognizing that similarity, then a new *loop closure edge* is added to the graph, linking $\mathbf{x}_j$ to $\mathbf{x}_i$. In our work, such constraint edges can be added for instance if at time t_j, the mobile is next to the same landmark as at time t_i (e.g., touching the same Bluetooth or NFC tag or seeing the same QR code): the constraint edge has a displacement assumed to be equal to 0 and an unknown change of orientation. If there is a gap in the PDR trajectory (or if one switches to a different mobile), a constraint edge can be inserted between the disjoint ends.

One can also define global landmarks, such as position fixes given by GPS or readings from an NFC tag or QR code with a known landmark position $(y_{1,t}, y_{2,t})$. In that latter case, a global vertex can be added, for instance at the origin of the coordinate system, and the displacement in the constraint *landmark edge* is equal to the landmark's coordinates $(y_{1,t}, y_{2,t})$.

GraphSLAM is initialized using positions and orientations derived from pedestrian dead reckoning. This means that the motion edge constraints are, initially, all satisfied. However, the constraints in the loop closure and landmark edges are initially violated, because of the cumulated drift due to noisy PDR. The algorithm then iteratively optimizes the poses $\{\mathbf{x}_t\}$ in the graph by aiming at minimizing the errors at all edges, including the violations of the loop closure and landmark constraints.

Formally, we note, for any two vertices i and j in the pose graph that have an edge linking them:

- $\mathbf{z}_{ij}$ as the vector of constraints: it is the observed relative displacement (2D translation and change of orientation angle) between pose i and pose j, expressed from the viewpoint of pose i,
- $\hat{\mathbf{z}}_{ij}(\mathbf{x}_i, \mathbf{x}_j)$ as the calculated displacement given the current values of poses $\mathbf{x}_i$ and $\mathbf{x}_j$, calculated from the viewpoint of pose i,
- $\mathbf{e}_{ij} = \mathbf{z}_{ij} - \hat{\mathbf{z}}_{ij}(\mathbf{x}_i, \mathbf{x}_j)$ as the error between the current configuration of poses i and j and the observed constraint,
- $\mathbf{\Omega}_{ij}$ as the 3×3 *information matrix* (inverse covariance matrix) of the constraint between pose i and j.

This information matrix expresses the noise in the observation of the (PDR, loop closure, landmark) constraint but in practice is generally kept diagonal. The term $\omega_{3,3}$ at row, column $(3, 3)$ in the matrix expresses the inverse covariance of the orientation angle constraint. In the experiments reported in this paper, we set Ω to be equal to the identity matrix with the exception of the term $\omega_{3,3}$ which was equal to 4 (motion edges and loop-closure edges) or to 0 when the angle constraint was unknown (landmark edges).

GraphSLAM aims at minimizing the negative log likelihood $\mathbf{F}$ of all the observations over the set C of edges:

$$\mathbf{F}(\mathbf{x}) = \sum_{(i,j)\in C} \mathbf{e}_{ij}^T \mathbf{\Omega}_{ij} \mathbf{e}_i j \quad (5)$$

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \mathbf{F}(\mathbf{x}) \quad (6)$$

The solution to (Eq. 6) is obtained by iterative local linearization [16], [33]. At each step, a new, optimal sequence of poses $\mathbf{x}'$ is recomputed given the current edge constraints and the constraints on all edges are then re-evaluated given the new poses. The new sequence of poses after each step's update is decomposed as $\mathbf{x}' = \mathbf{x} + \mathbf{\Delta x}$.

The error $\mathbf{e}'_{ij}$ after the update is linearized as a sum of the current error $\mathbf{e}_{ij}$ and a term linear in the pose update $\mathbf{\Delta x}_{ij}$. As detailed in [16], the minimization of (Eq. 6) in terms of $\mathbf{x}'$ can be expressed as a minimization of a quadratic function in terms of $\mathbf{Deltax}$ and therefore the global sequence of pose update $\mathbf{\Delta x}$ becomes the solution of a linear system. The locally optimal solution to the sequence of poses is reached by iteratively recalculating the error functions $\mathbf{e}_{ij}$ at each edge and solving for the pose update $\mathbf{\Delta x}$, until convergence.

As expressed above, the GraphSLAM optimization can be directly applied to "close the gaps and the loops" in the PDR trajectory; in other words, it can ensure that the trajectory of the phone (which may be made discontinuous because of the repeated position resets, explained in section II-A, at the landmarks encountered on the trajectory) becomes a smooth curve that goes through all the landmarks while obeying the noisy pedestrian dead reckoning. In practice, and as shown in the Results section, this optimization is redundant with the simpler PDR recalibration explained in section II-B.

In the following sections II-D and II-E, we introduce our contribution to GraphSLAM.

D. Signal Similarity using Kernels

As the user walks around the building, a large collection of RF signals from many access points is collected. Under the assumption that the RF signal does not change significantly over a short distance of a few meters, which corresponds, at a walking speed, to about 5s to 10s, we can use a measure of signal similarity. We propose to rely on kernel functions to compute the similarity, in signal space, between two segments of a trajectory.

For WiFi or Bluetooth data, where there can be hundreds of different access points (AP) scattered around the building, we decide to use the Kullback-Leibler [23] divergence between multinomials of AP visibility, as suggested in [28]. For the LTE RSRP or the magnitude of the magnetic field (as measured by the magnetometer of the smartphone), we can simply use the Euclidean distance. The KL divergence or the Euclidean distance can be input into a kernel function $k(S_{t_i}, S_{t_j})$, where S_{t_i} is the multivariate signal recorded around time point t_i (for a specific phone). The kernel function is either the Gaussian or the KL-divergence kernel [28] and takes the following form: $e^{-\alpha D}$ where D is either the squared Euclidean distance or the symmetrized KL divergence $KL(S_{t_i}||S_{t_j}) + KL(S_{t_i}||S_{t_j})$.

Figure 2 illustrates the kernel matrix for the WiFi AP visibility, from WiFi data acquired by 3 different phones. The data, along the rows and columns, are arranged by phone, then by time. The presence of off-diagonal blocks shows that there are cross-phone similarity in the signal space.

In the presence of multiple sources of RF data, such as WiFi RSS, LTE RSRP, Bluetooth RSS or the magnitude of the magnetic field, the kernel matrices K_1, K_2, etc... for each signal modality can be multiplied, element by element, provided that the rows and columns in each matrix correspond to the same time intervals on the same phone.

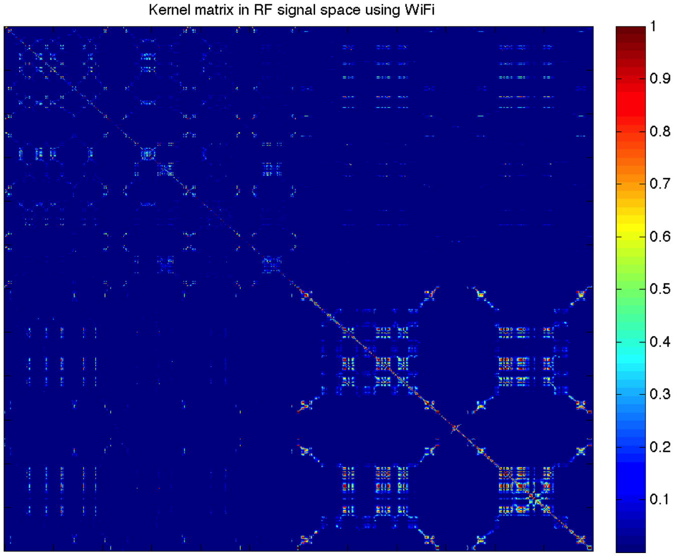

Fig. 2. RF kernel computed using the measure of signal similarity between any two points of the trajectory of any 3 phones. The kernel function is the Kullback-Leibler divergence over histograms of WiFi access point visibility.

E. SignalSLAM

Once the kernel similarity, explained in the previous section, is computed for all the time segments of the trajectory, it can used in the following way in the original GraphSLAM.

First, let us restrict the set of poses to time points $t \in T$ that correspond to time windows when signal strength was collected and between which signal similarity $\{k(S_t, S_{t'})\} \forall t, t' \in T$ was computed. In our experiments, we used windows of duration 10s, overlapped every second.

At a given iteration of the GraphSLAM algorithm and for the current configuration of poses $\mathbf{x}$, one can use weighted kernel regression to predict the expected pose $\mathbf{x}_t$ based on the poses of its immediate neighbors $S_{t'}$ in the signal space (where $t' \in T$ and $t \neq t'$). Such neighbors in the signal space are poses $\mathbf{x}_{t'}$ whose kernel function $k(S_t, S_{t'})$ evaluates the highest. We note N_t the neighborhood, in signal space, of pose $\mathbf{x}_t$ at time t, and typically use $n = 10$ neighbors. The weighted kernel regression predicts the following position for $\mathbf{x}_t$:

$$\bar{\mathbf{x}}_t = \frac{\sum_{t' \in N} \mathbf{x}_{t'} k(S_t, S_{t'})}{\sum_{t' \in N} k(S_t, S_{t'})} \tag{7}$$

That prediction $\bar{\mathbf{x}}_t$ is then simply used as a temporary "absolute" landmark for the GraphSLAM algorithm. This means that at that iteration of GraphSLAM, an additional *signal edge* is added from pose/vertex $\mathbf{x_t}$ to a new pose/vertex $\bar{\mathbf{x}}_\mathbf{t}$. The information matrix $\Omega_{t,t'}$ specific to signal edge constraints is unit diagonal with the exception of the orientation term $\omega_{3,3} = 0$.

After each update of $\mathbf{x}$, the signal similarity based prediction of each pose $\mathbf{x}_t$ needs to be re-evaluated by re-computing (Eq. 7). However, the signal similarity kernel $k(S_t, S_{t'})$ does not change, neither do the neighbors N_t of $\mathbf{x}_t$ in signal space. As a consequence, the overhead introduced by our algorithm is negligible.

Because our simple modification to GraphSLAM enables to handles any kind of similarity between any two poses at times t and t' that merely relies on sampling signals S_t and $S_{t'}$, we called it SignalSLAM. Note that SignalSLAM can operate on any kind of signal similarity that is stationary (i.e. time independent), which includes for instance WiFi RSSI, LTE, Bluetooth from fixed beacons or, under some circumstances, magnetic field.

III. Data Acquisition from the Pocket

In this section, we explain the details of our system for logging smartphone data that can be supplied to the GraphSLAM and SignalSLAM algorithms for recovering the RF signal maps. We give an overview of the smartphone app in section III-C and provide further details about the Pedestrian Dead Reckoning (PDR), namely the phone orientation estimation (section III-A1), the step counting (section III-A2) and heading estimation (section III-A3) as well as the landmark acquisition (section III-B).

A. Pedestrian Dead Reckoning from the Pocket

1) Orientation Estimation: We propose to use all the inertial sensors available on modern smartphones, including the 3-axis accelerometer, 3-axis gyroscope and 3-axis magnetometer that are integrated within the Inertial Measurement Unit (IMU) chip, in order to track the phone's 3D orientation (yaw, pitch and roll angles) at high frequency. Orientation estimation is done using the Madgwick filter [26], which is a low-computational-complexity information filter relying on the quaternion representation of angles and updated using gradient descent. A Java version of the code (available at `http://www.x-io.co.uk`), implemented on a smartphone running the Android OS, can run in negligible time and sample accelerometer and gyroscope measurements at 50Hz.

This orientation can be represented by a 4-by-4 rotation matrix $\mathbf{R}_t$ or by the (yaw, pitch, roll) angles. This enables us both to recover the vertical component of the acceleration by rotating the raw accelerometer readings $\mathbf{a}_t$ in the phone coordinate system to the accelerometer readings $\mathbf{h}_t$ in the so-called *human* or local ground coordinates defined by the longitude, latitude and upright axes: $[\mathbf{h}_t 1]^T = \mathbf{R}_t [\mathbf{a}_t 1]^T$.

2) Step Counting: The step detection and counting in our system follows the "PDR from the Pocket" method explained in [42]. Namely, the vertical component of acceleration is extracted from the rotated acceleration vector $\mathbf{h}_t$ in the human coordinate system. After median-filtering (step size 5, corresponding to 0.1s at 50Hz), a sliding-window variance is computed on a window of length 7 (0.14s at 50Hz). This variance is then thresholded to find peaks. A stride detection corresponds to two consecutive peaks of the variance of acceleration, within a time window comprised between 0.5s and 1.5s.

We decided not to estimate the stride length d using acceleration data or user height [15], [17] and only assume it to be a constant during a segment of the trajectory. As explained in section II-B, the stride length can be re-calibrated for each segment between any two consecutive landmark readings.

Note that foot-mounted IMUs [3], [38] with the Zero Velocity Update Method typically achieve a far better pedestrian dead reckoning accuracy, because of both better hardware and reduced noise due, as the IMU is attached to the foot that hits the ground as opposed to being stashed loose in a pocket, but they lack the convenience of consumer-grade smartphones casually worn in the pocket.

3) Step Heading Estimation: Most PDR systems on smartphones rely on the compass only for estimating the heading, and the phone would typically need to be held vertically, screen facing the user and pointing towards the direction of the walk [15], [17], [25]. This conflicts with our target of PDR from the pocket.

An alternative to heading estimation during walking motion, that is able to operate on a phone placed in a pocket or on a clip belt, is to rely on the accelerometer data $\mathbf{h}_t$ in the human coordinate system, by extracting the direction of the third principal component of the acceleration points [14], [42]. Initial tests on several smartphones such as the Samsung Nexus S however showed that the orientation estimation was often erroneous. Moreover, this method only provides with the direction modulo 180deg.

For these reasons, we ultimately decided to simply rely on the yaw of the phone, computed in real time by the Madgwick filter (see section III-A1). The yaw values are median filtered over 5 time points and averaged within the time interval corresponding to one stride. This filtered yaw θ_t is then incremented by the "pocket" offset angle β_t, as illustrated on Fig. 1.

B. Low-cost Landmarks

The ultimate objective of our line of research is to enable RF mapping entirely "from the pocket". Because the PDR needs periodic corrections (see section II-A) and because we perform re-calibration of the stride length and pocket angle offset (see section II-B), landmarks with known position are however needed.

In the current version of our system, we use the following absolute landmarks that bear latitude and longitude coordinates and whose commonality is to be readable from a recent model commercial smartphone running the Android OS. These landmarks consist in:

- User-validated Global Positioning System (GPS) position fixes.
- Self-describing visual 2D bar codes (QR codes) which are encoding their own position (similarly to [30], [40]).
- Programmable, self-describing Near-Field Communication (NFC) tags with their position similarly encoded. Our novel idea of using NFC tags for localization is based on similar research using RFID tags [1].

An example of the text string encoded in an NFC tag or QR code landmark, that we can define, that bears the latitude, longitude and altitude ($z = 0$) information along with a tag ID, and that can be easily parsed by a program, is:

```
#6e1ebc 40.684492,-74.401406 z0
```

Inside the building, NFC tags or QR codes could be substituted by short-range (e.g., 2m) Bluetooth beacons, thus alleviating the need for taking the smartphone out of the pocket.

An alternative step towards our goal of "RF mapping from the pocket" would be to exploit the *organic* activity landmarks (i.e., stairs or elevators along with position matching on a map) by activity classification based on accelerometer readings [13], [17], [18].

Note that currently, our method handles the optimization on a single floor, with constant altitude. In a multi floor building, changes of floors can be detected either by a combination of activity classification (stair and elevator detection) with barometer readings, or by writing the floor or altitude in the NFC tag or QR code landmarks.

C. Android App for Data Acquisition

In order to conduct the experiments using SignalSLAM and detailed in the next section, we have implemented a data acquisition and logging tool, running on the Android operating system. It acquires the following timestamped sensor data:

- LTE 4G received signal strength (RSS), reference signal received power (RSRP) and reference signal received quality (RSRQ), from the LTE cell onto which the phone is currently connected. The signal is sampled every 1s to 3s, depending on the phone model.
- WiFi received signal strength (RSS), in dBm, from each WiFi access point, sampled every 1s to 3s, depending on the phone model.
- Bluetooth signal strength (RSS), in dBm, from each Bluetooth access point, sampled every 10s.
- GPS location fixes and their accuracy, sampled every 1s when satellite information is available.
- NFC tag readings from custom landmarks.
- QR code tag readings from custom landmarks.
- Accelerometer, gyroscope and magnetometer readings, sampled at 50Hz (10Hz for the magnetometer).
- Orientation estimation using the Madgwick filter (yaw, pitch and roll), implementing the algorithm in section III-A1 and sampled at 50Hz.
- Pedestrian dead reckoning, implementing the algorithm in sections III-A2 and III-A3, with on-demand position resets in presence of landmarks (see section III-B).

The data logging tool plots the current GPS and pedestrian dead reckoning position overlaid on a geo-referenced map. This map specifies the affine transform to go from the (lat, long) coordinate system to a local map system, assuming that area of interest has negligible curvature. Multiple maps can be loaded, depending on the current (latitude, longitude) coordinates

All the data logging functionalities rely on the standard Android API. In order to make the app compatible with the largest number of Android smartphones, we used the version

10 of the Android Software Development Toolkit, for Android version 2.3.3 and above (released mid-2011).

As explained in the next Results section, we have run our experiments on three different models of Android smartphones: a Samsung Nexus S (without 4G LTE), a Samsung Galaxy Note 2 (with a Verizon 4G LTE subscription) and a Samsung Galaxy S4, reading a proprietary 4G LTE small cell.

IV. Results

A. Experimental Protocol

In line with our goal of RF mapping from the pocket, the experimental protocol essentially consisted in letting two experimenters repeatedly walk around, back and forth, a large office building covering an area of approximately 200m×160m, carrying one or two smartphones in the trouser's pocket that were running the Android app detailed in section III-C. The experiments were carried over the course of several weeks and we report in this paper the results for two sets of data acquisition: 1) data acquisition over 1 day, with one experimenter carrying a Samsung Nexus S in one pocket and a Samsung Galaxy S4 in another while a second experimenter carries a Samsung Galaxy Note 2 and 2) the same experiment as in 1) but repeated 4 days later.

In order to provide landmarks, we placed NFC tags at 12 strategic locations in the building, which correspond to one of the following: entrances, staircases, elevators, the reception desk with badge reader, the cafe counter and two meeting rooms. We contend that such areas could be recognized using activity recognition based on inertial data, for example using the systems implemented in [13], [17], [18], or simply using GPS (for the entrances). These landmarks correspond to *organic* landmarks.

The conditions for pedestrian dead reckoning in this building were challenging because of the metallic walls throughout the building that created perturbations to the compass readings.

B. Crowd-sourced Multi-phone SignalSLAM

In a first step, we reconstructed the trajectories of all the three phones independently. Figures 3, 4 and 5 show successive steps in the reconstruction, starting from raw PDR (in blue), followed by calibrated PDR (in red) using the heuristic described in section II-B, then the GraphSLAM optimization with landmark and loop closure constraints (in green) and finally SignalSLAM using a product of WiFi and Bluetooth similarity kernels (in black).

The difference in accuracy of PDR between three successive models of the Samsung phones is striking and can be explained by the increasing quality and precision of IMU chips mounted on commercial smartphones. The low quality of the IMU on the Nexus S and the magnetic perturbations seem to pose a challenge for PDR. We then notice that PDR calibration succeeds in closing the gaps in the trajectory (after each position reset) while keeping the trajectory reasonably straight along the corridors, notably for the Galaxy S4 (Figs. 5). Because PDR calibration already closes those gaps and is constrained by landmarks, the pure GraphSLAM algorithm does not substantially change the shape of the trajectory. Finally, the largest improvement comes from running SignalSLAM. In particular on the Galaxy Note 2 and on the Galaxy S4 data, that algorithm succeeds in closing the gap between two trajectories along the same corridor.

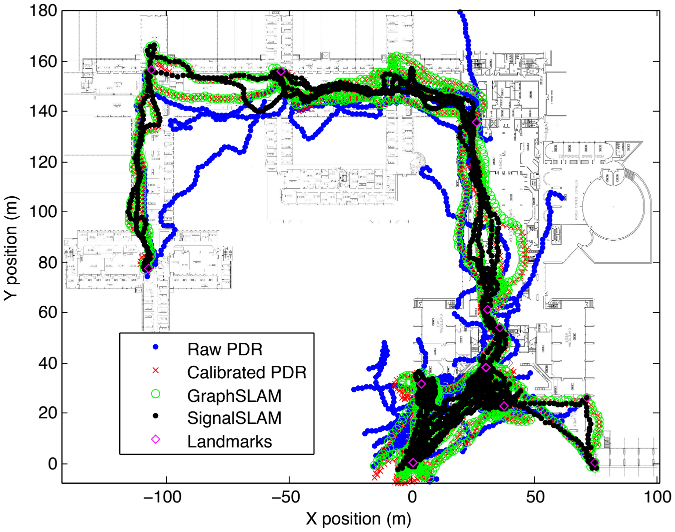

Fig. 3. Reconstructed trajectories for the Samsung Nexus S, using several iterations (raw pedestrian dead reckoning, re-calibrated PDR using landmarks, landmark-based GraphSLAM and WiFi- and Bluetooth-based SignalSLAM).

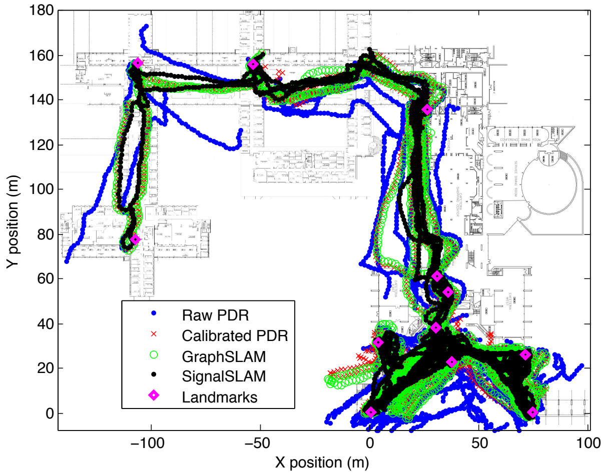

Fig. 4. Reconstructed trajectories for the Samsung Note 2, using the same procedure as illustrated in Fig. 3.

In a second step, we run the joint optimization of the trajectories of all the three phones. Although the three trajectories are distinct, the SignalSLAM algorithm uses a WiFi and Bluetooth similarity kernel (shown on Figure 2) that establishes similarity, in RF signal space, between specific points of the trajectory of two different phones, in the same way as it establishes similarity for the same phone.

C. Step-by-step Comparison of Trajectories

A numerical evaluation of the reconstruction of the trajectories can be provided as following. After running the three optimizations for each of the phones independently, we extract the time segment when the two experimenters carrying the three phones were walking alongside. We then compute the

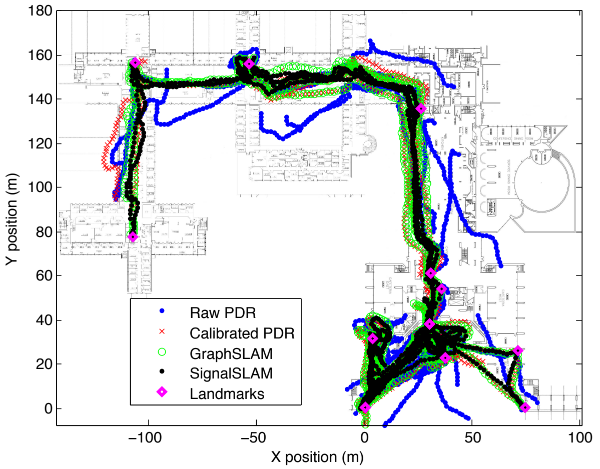

Fig. 5. Reconstructed trajectories for the Samsung Galaxy S4, using the same procedure as illustrated in Fig. 3.

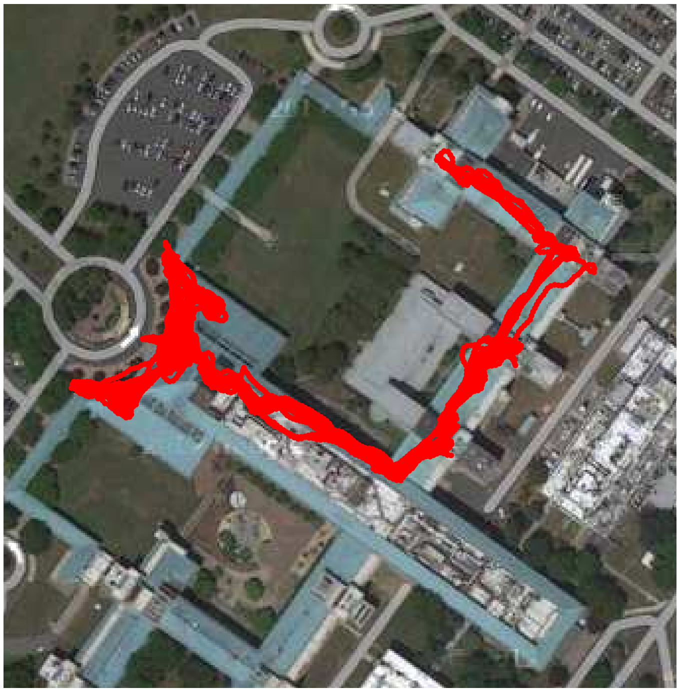

Fig. 6. Jointly reconstructed trajectories of three different phones (from Figs. 3, 4 and 5). Joint SignalSLAM was run with a WiFi- and Bluetooth-based kernel computed using measurements from all the phones at once.

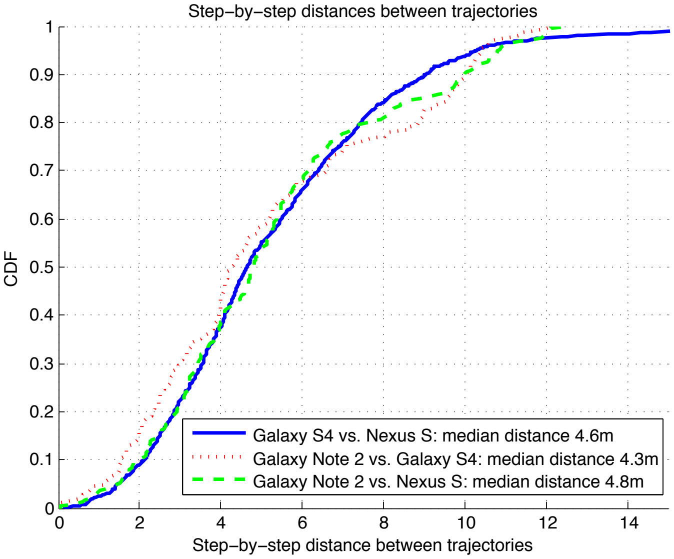

Fig. 7. Step-by-step distances between the inferred trajectories of 3 different smart phones (we used WiFi- and landmark-based Signal-SLAM and the trajectories for each phone were inferred independently of the other phones). Two phones (Samsung Galaxy S4 and Samsung Nexus S) were carried by the same experimenter while the third phone, a Samsung Galaxy Note 2, was carried by a second person walking alongside.

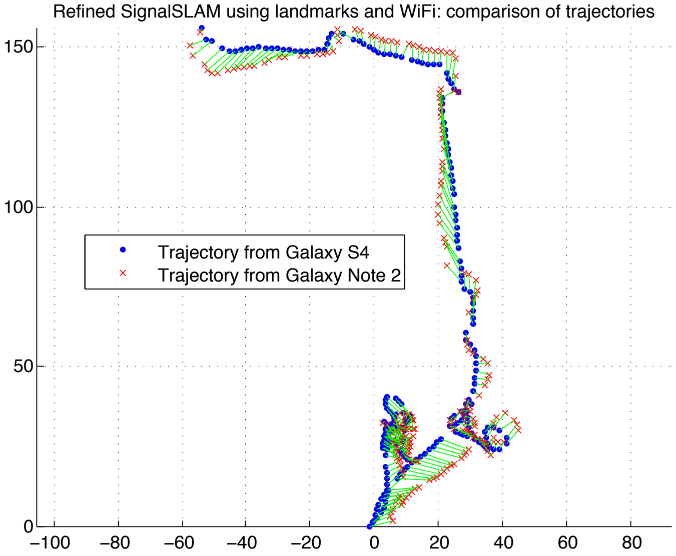

Fig. 8. Side-by-side comparison of the inferred trajectories (see Fig. 7) of 2 different smart phones carried by 2 experimenters walking together (Samsung Galaxy S4 and Samsung Galaxy Note 2).

distance, at each second within that selected time interval, between the three trajectories (we use linear interpolation to estimate the position between two strides). As Figure 7 shows, the median step-by-step distance between three trajectories remains under 5m (under 10m at 90% percentile). Figure 8 illustrates the step-by-step proximity of the trajectory of the Galaxy Note 2 and of the Galaxy S4.

D. Generation of Dense LTE and WiFi Signal Maps

Using directly the trajectories recovered in the previous two sections, we can plot the map of LTE RSRP for two different phones: the map of the Verizon 4G LTE RSRP at the level of the building, on Figure 9, and that of the small cell 4G LTE on Figure 10.

E. WiFi Tracking Using Crowd-sourced RF Maps

Finally, we acquire a second set of WiFi data logs and trajectories 4 days after the dataset illustrated in the previous section to try WiFi-based geo-localization [2].

We use the WiFi data from day 1, along with the reconstructed trajectories, to build a map of fingerprints. To build the map, we simple use a 5m grid overlaid on the 3 trajectories, and select, for each fingerprint cell, the time points in the

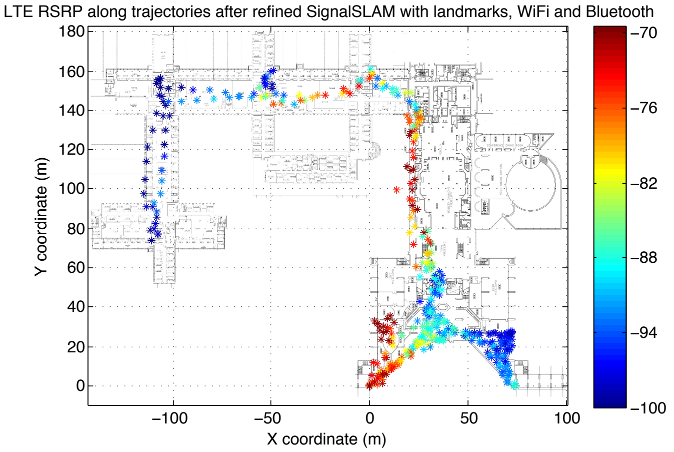

Fig. 9. Reference Signal Received Power (RSRP) for the 4G LTE Verizon network, as mapped on a Samsung Galaxy Note 2 phone using the inferred trajectory from Fig. 4.

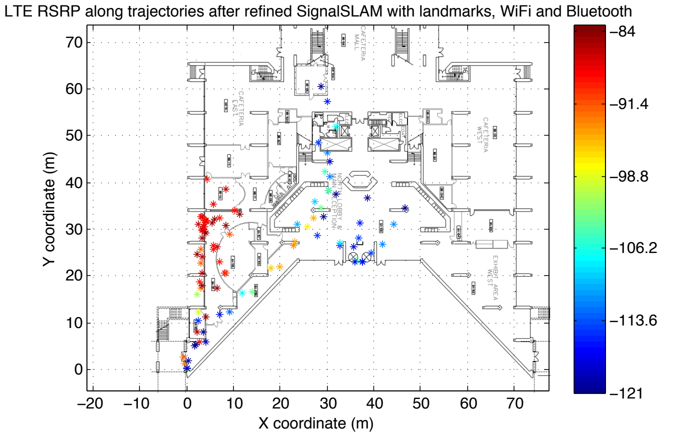

Fig. 10. Reference Signal Received Power (RSRP) for a small cell LTE, as mapped on a Samsung Galaxy S4 phone using the inferred trajectory from Fig. 4.

trajectories of each phone that fall into that cell. We then retrieve the WiFi readings at those time points. We either estimate the empirical distribution of the RSSI, following [28] (for the Nexus S phone which samples WiFi every 1s) or simply estimate their mean value of the RSSI (for the two other phones that collect data only every 3s).

For the WiFi tracking of the WiFi data from day 2, assuming that the actual trajectory is the one that was inferred using SignalSLAM, we consider a sliding window of 10s to estimate the mean or the empirical distribution of WiFi RSSI from all the access points. The tracking algorithm is respectively Weighted Kernel Regression using Kullback-Leibler Divergence kernels [28] for the Nexus S, or weighted K-nearest neighbors [2] for the Galaxy Note 2 or the Galaxy S4.

As Tables I and II prove, the geo-localization performance is rather weak (between 11m and 16m median error) when one compares the positions on the trajectory inferred by SignalSLAM vs. the positions inferred using WiFi only and a crowd-sourced fingerprint map. One however needs to consider several challenges in this exercise. First, the ground truth positions were actually never known in this experiment, only the inferred ones. Second, the data were collected while walking, without any special infrastructure, only the existing WiFi access points whose positions were unknown. Finally, the experimenters did not stop at any location but simply quickly passed through the corridors.

TABLE I. TRACKING RESULTS USING PHONE-SPECIFIC WIFI FINGERPRINTS FROM INDEPENDENTLY-INFERRED TRAJECTORIES

Tracked phone	Median accuracy	Accuracy at 90%
Nexus S	11.1m	30.5m
Galaxy Note 2	13.3m	48.2m
Galaxy S4	13.9m	67.9m

TABLE II. TRACKING RESULTS USING WIFI FINGERPRINTS FROM 3 DIFFERENT PHONES WITH JOINTLY-INFERRED TRAJECTORIES

Tracked phone	Median accuracy	Accuracy at 90%
Nexus S	14.8m	31.3m
Galaxy Note 2	12.9m	40.1m
Galaxy S4	16.5m	53.8m

In our future experiments, we will evaluate how increasing the number of individuals and the variety of smartphones involved in the crowd-sourcing of the RF map can bring the localization error down.

V. CONCLUSION

We described a method for automatically generating and updating an RF signal map in buildings while determining the location of the measuring device, namely, a smart phone loaded with sensors. The method uses an adaptation of the GraphSLAM technique to synthesize the sensor measurements that include those from inertial measurement units as well as available RF signals, and thereby to infer the unknown smart phone trajectory. The pedestrian dead reckoning estimates are robust to phone pose. Thus the method avoids some common restrictions such as requiring the users to hold the smart phone in hand. We tested the method using several Android phones of different models, and showed that the method can accommodate multiple users participating in the measurements. We believe that crowd-sourcing with more users and over repeated days, would enable to easily maintain a WiFi/LTE signal map based geo-localization system.

ACKNOWLEDGMENT

The authors would like to thank Chun-Nam Yu for help with data collection as well as Byron Chen, Nida Chatwattanasiri and Johannes Rosch for useful discussions and feedback.

REFERENCES

[1] M.R. Andrews, T.K. Ho, G.P. Kochanski, L.J. Lanzerotti and D.J. Thomson, “Method and Apparatus for Location Determination Based on Dispersed Radio Frequency Tags”, *US Patent 6,900,762 B2*, 2005.

[2] P. Bahl and V.N. Padmanabhan, “An In-Building RF-based User Location and Tracking System”, *IEEE Infocom*, vol.2, pp.775–784, 2000.

[3] J.B. Bancroft, D. Garrett and G. Lachapelle, “Activity and Environment Classification using Foot Mounted Navigation Sensors”, *International Conference on Indoor Positioning and Indoor Navigation*, 2012.

[4] M.S. Bargh and R. de Groote, “Indoor Localization Based on Response Rate of Bluetooth Inquiries”, *International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments*, pp.48–54, 2008.

[5] A. Brajdic and R. Harle, “Scalable indoor pedestrian localisation using inertial sensing and parallel particle filters”, *International Conference on Indoor Positioning and Indoor Navigation*, 2012.

[6] L. Bruno and P. Robertson, “WiSLAM: improving FootSLAM with WiFi”, *International Conference on Indoor Positioning and Indoor Navigation*, 2011.

[7] P. Castro, P. Chiu, T. Kremenek and R. Muntz, “A Probabilistic Location Service for Wireless Network Environments”, *Ubiquitous Computing*, 2001.

[8] K. Chintalapudi, A. Padmanabha Iyer and V.N. Padmanabhan, “Indoor Localization Without the Pain”, *MobiCom*, 2010.

[9] F. Dellaert, D. Fox, W. Burgard and S. Thrun, “Monte Carlo Localization for Mobile Robots”, *International Conference on Robots and Automation*, 1999.

[10] M.W.M.G. Dissanayake, P.M. Newman, S. Clark, H.F. Durrant-Whyte and M. Csorba, “A solution to the simultaneous localization and map building (SLAM) problem”, *IEEE Transactions on Robotics and Automation*, vol.17n n.3, 2001.

[11] M.F. Fallon, H. Johannsson, J. Brookshire, S. Teller and J.J. Leonard, “Sensor Fusion for Flexible Human-Portable Building-Scale Mapping”, *IEEE International Conference on Intelligent Robots and Systems*, 2012.

[12] B. Ferris, D. Fox and N. Lawrence, “WiFi-SLAM Using Gaussian Process Latent Variable Models”, *International Joint Conference on Artificial Intelligence*, 2007.

[13] A. Fialho, A.M. Cavalcante, A. Costa and J. Ledlie, “Classifying and using motion in organic indoor positioning”, *International Conference on Indoor Positioning and Indoor Navigation*, 2012.

[14] T. Gadeke, J. Schmid, W. Stork and K.D. Muller-Glaser, “Pedestrian Dead Reckoning for Person Localization in a Wireless Sensor Network”, *International Conference on Indoor Positioning and Indoor Navigation*, 2011.

[15] T. Gallagher, E. Wise, B. Li, A.G. Dempster and C. Rizos, “Indoor Positioning System based on Sensor Fusion for the Blind and Visually Impaired”, *International Conference on Indoor Positioning and Indoor Navigation*, 2012.

[16] G. Grisetti, R. Kummerle, C. Stachniss and W. Burgard, “A tutorial on graph-based SLAM”, *IEEE Intelligent Transportation Systems Magazine*, vol.2, n.4, 2010.

[17] D. Gusenbauer, C. Isert and J. Krotsche, “Selft-Contained Indoor Positioning on Off-The-Shelf Mobile Devices”, *International Conference on Indoor Positioning and Indoor Navigation*, 2010.

[18] M. Hardegger, D. Roggen, S. Mazilu and G. Troster, “ActionSLAM: Using location-related actions as landmarks in pedestrian SLAM”, *International Conference on Indoor Positioning and Indoor Navigation*, 2012.

[19] R. Harle, “A Survey of Indoor Inertial Positioning Systems for Pedestrians”, *IEEE Communications Surveys & Tutorials*, n.99, 2013.

[20] J. Huang, D. Millman, M. Quigley, D. Stavens, S. Thrun and A. Aggarwal, “Efficient, Generalized Indoor WiFi GraphSLAM”, *IEEE International Conference on Robotics and Automation*, 2011.

[21] M. Kessel and M. Werner, “Automated WLAN calibration with a backtracking particle filter”, *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2012.

[22] M. Kourogi and T. Kurata, “Personal positioning based on walking locomotion analysis with self-contained sensors and a wearable camera”, *Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality*, 2003.

[23] S. Kullback and R.A. Leibler, “On Information and Sufficiency”, *Annals of Mathematical Statistics*, vol.22, n.1, pp.79–86, 1951.

[24] A.M. Ladd, K.E. Bekris, A. Rudys, L.E. Kavraki and D.S. Wallach, “Robotics-Based Location Sensing Using Wireless Ethernet”, *Wireless Networks*, vol.11, 2005.

[25] J.A.B. Link, P. Smith, N. Viol and K. Wehrle, “Footpath: Accurate map-based indoor navigation using smartphones”, *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2011.

[26] S. Madgwick, A. Harrison, R. Vaidyanathan, 2011, “Estimation of IMU and MARG orientation using a gradient descent algorithm”, *IEEE International Conference on Rehabilitation Robotics*.

[27] O. Mezentsev, J. Collin and G. Lachapelle, “Pedestrian Dead Reckoning–A Solution to Navigation in GPS Signal Degraded Areas?”, *Geomatica*, vol.59, n.2, 2005.

[28] P. Mirowski, H. Steck, P. Whiting, R. Palaniappan, M. MacDonald and T.K. Ho, 2011, “KL-divergence kernel regression for non-Gaussian fingerprint based localization”, *International Conference on Indoor Positioning and Indoor Navigation*.

[29] P. Mirowski, P. Whiting, H. Steck, R. Palaniappan, M. MacDonald, D. Hartmann and T.K. Ho, 2012, “Probability kernel regression for WiFi localisation”, *Journal of Location Based Services*, 6 (2), 81-100.

[30] P. Mirowski, R. Palaniappan and T.K. Ho, 2012, “Depth Camera SLAM on a Low-cost WiFi Mapping Robot”, *International Conference on Technologies for Practical Robot Applications* (TePRA).

[31] P.J. Moreno, P.P. Ho and N. Vasconcelos, “A Kullback-Leibler divergence based kernel for SVM classification in multimedia applications”, *Neural Information Processing Systems*, 2002.

[32] E. Nadaraya, “On estimating regression”, *Theory of Probability and Applications*, vol.9, pp.141–142, 1964.

[33] E. Olson, J. Leonard and S. Teller, “Fast iterative alignment of pose graphs with poor initial estimates”, *IEEE International Conference on Robotics and Automation*, 2006.

[34] R. Palaniappan, P. Mirowski, T.K. Ho, H. Steck, P. Whiting and M. MacDonald, 2011, “Autonomous RF surveying robot for indoor localization and tracking”, *International Conference on Indoor Positioning and Indoor Navigation*.

[35] J.J. Pan, S.J. Pan, J. Yin, L.M. Ni, Q. Yang, “Tracking Mobile Users in Wireless Networks via Semi-Supervised Colocalization”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.

[36] M. Peter, D. Fritsch, B. Schafer, A. Kleusberg, J.A.B. Link and K. Wehrle, “Versatile Geo-Referenced Maps for Indoor Navigation of Pedestrians”, *International Conference on Indoor Positioning and Indoor Navigation*, 2012.

[37] J. Pinchin, C. Hide and T. Moore, “A Particle Filter Approach to Indoor Navigation Using a Foot Mounted Inertial Navigation System and Heuristic Heading Information”, *International Conference on Indoor Positioning and Indoor Navigation*, 2012.

[38] M. Romanovas, V. Goridko, A. Al-Jawad, M. Schwaab, L. Klingbeil, M. Traechtler and Y. Manoli, “A Study on Indoor Pedestrian Localization Algorithms with Foot-Mounted Sensors”, *International Conference on Indoor Positioning and Indoor Navigation*, 2012.

[39] T. Roos, P. Myllymaki, H. Tirri, P. Misikangas and J. Sievanen, “A Probabilistic Approach to WLAN User Location Estimation”, *International Journal of Wireless Information Networks*, vol.7, n.3, 2002.

[40] S. Alberto, T. Dessi, D. Carboni, V. Popescu and L. Atzori, “Inertial navigation systems for user-centric indoor applications”, *Networked and Electronic Media Summit*, Barcelona, 2010.

[41] H. Shin, Y. Chon and H. Cha, “SmartSLAM: Constructing an Indoor Floor Plan using Smartphone”, *Yonsei University, Technical Report*, MOBED-TR-2010-2, 2010.

[42] U. Steinhoff and B. Schiele, “Dead reckoning from the pocket-an experimental study”, *IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2010.

[43] L. Vincent, “Crowd-sourced information for interior localization and navigation”, *US Patent 8,320,939 B1*, 2012.

[44] T. Whelan, H. Johannsson, M. Kaess, J.J. Leonard and J. McDonald, “Robust Real-Time Visual Odometry for Dense RGB-D Mapping”, *IEEE International Conference on Robotics and Automation*, 2013.

[45] T. Yairi, “Map building without localization by dimensionality reduction techniques”, *International Conference on Machine Learning*, 2007.

[46] M.A. Youssef, A. Agrawala and U. Shankar, 2003, “WLAN location determination via clustering and probability distributions”, *First International Conference on Pervasive Computing and Communications*, 143–150.