International Conference on Indoor Positioning and Indoor Navigation (IPIN), Sapporo, Japan, September 18-21, 2017

Autocalibration of a wireless positioning network with a FastSLAM algorithm

Fernando Seco and Antonio R. Jiménez
Center for Automation and Robotics, Spanish National Research Council (CSIC-UPM)
Ctra. de Campo Real km 0,200 Arganda del Rey, Madrid, Spain
Email: fernando.seco@csic.es

***Abstract*—The calibration (measurement of the position) of a network of wireless nodes used for indoor localization purposes is a tedious process and prone to errors if done manually. This paper presents a method for the autocalibration of that network, using the measurement of the received signal strength (RSS) of RF signals coming from the nodes, and captured while a person is taking a random walk in the environment. The calibration method is adapted from a Simultaneous Localization and Mapping (SLAM) technique from Robotics, and is based on a Bayesian particle filter modeling the unknown position of the user and the location of the beacons. Information coming from RSS measurements is incorporated to the filter using a rather generic measurement model (the path loss law), producing a sequence of beacon nodes position estimates with decreasing uncertainty over time. The accuracy and convergence of the method can be further enhanced by using pedestrian dead-reckoning (PDR) techniques from the handheld smartphone used to capture the RF data. The method is demonstrated with a deployment of 60 unknown position active RFID tags (and 4 known position tags) in an indoor environment, and a trajectory lasting 1054 s. The results are a median beacon positioning error of 4.9 m using only the RSS information, and 3.4 m if PDR information is incorporated to the particle filter. This error can be further decreased by adding the results of more calibration routes.**

I. Introduction

Most indoor location systems are based on signal transmissions from a set of static beacon nodes to the user's current position; then an estimate of the user location can be obtained by processing some physical measurements such as time-of-arrival, received signal strength, etc [1]. The location of the beacon nodes themselves is assumed to be known and its obtention a part of the positioning system deployment process. This calibration stage, however, is tedious and prone to human error [2]. Considering that, in some circumstances, the beacon network might be altered without warning (for example, wifi access points might be installed or removed), or the network be modified frequently (for example, nodes embedded in mobile stands in an exhibition fair, which can be displaced daily), it would be convenient to have at our disposal an *autocalibration* method which permits quick and accurate retrieval of the current node positions (referred to a map) from in situ measurements and some additional information. In particular, we are interested in crowdsourcing techniques which demand minimum attention from the users. Such a procedure is presented in this work.

We point out that location systems calibration is usually associated with machine learning techniques for localization (such as fingerprint), where the exact location of the beacons is not required, and the calibration process consists in the generation of a radio map instead [3]. In the current work, however, we refer to the calibration of the beacon network as the computation of the beacon locations as physical coordinates in a map, for later use in parametric localization methods.

A. Related work

In the event that all nodes can communicate among them, such as in wireless sensor networks (WSN), the calibration problem is addressed with cooperative localization techniques [4]. This is not feasible in the wireless networks designed for one direction communication (beacons to user or viceversa) commonly employed for indoor individual localization. In ultrasonic [5], [6] or ultrawideband radio positioning systems [7], where direct range estimation from beacons to user is available, the calibration of the network can be posed as problem of inverted localization, and beacon positions are computed with the same algebraic equations used for localization of the mobile nodes. Applying a similar strategy to signal strength-based indoor positioning systems is more challenging due to the large variance of measured values for a given physical range between emitter and receiver, and the usual problem of non existence of line-of-sight between them.

Bayesian techniques, such as those previously developed by the Robotics community for the Simultaneous Location and Mapping (SLAM) problem are well suited for handling measurements with large uncertainties [8]. Some researchers have translated these methods to the indoor localization field. One example is the "war-driving" technique used to map the position of unknown wifi access points as they are detected [9]. However, war-driving relies on the user being able to access an external positioning system (usually GNSS) to localize himself and then infer the location of the detected wifi access points, which is not be practical in indoor environments with no satellite coverage.

Mapping techniques not requiring any external device are based in pedestrian dead reckoning (PDR) estimates. For example, the FootSLAM method, introduced in [10], uses a SLAM technique to correct the drift of PDR trajectories and generate a map of the environment, with no additional sensors. The WiSLAM system [11] further extends this technique to estimate the location of wifi access points; however, only two

APs are positioned experimentally, and a full evaluation of the calibration performance of this technique is not given. Additionally, FootSLAM methods require installation of a foot-mounted inertial motion unit, which is not always feasible in practical situations.

Crowdsourcing methods use the data collected (perhaps unknowingly) by users in an environment to generate information useful for navigation. The system presented in [12] is very similar to the work in this paper; however, it is oriented to generation of a radio map of the environment (as usable by fingerprinting methods), and not specifically to the location of the RF emitters. More closely related to our work is the system described in [13], in which beacon locations (and additionally, propagation parameters) are estimated from unknown trajectories of people. The approach followed by the authors is based on least squares minimization of a cost function related to beacon locations, unlike the current contribution, which is based in Bayesian techniques.

The novelty of the present work is introducing a Bayesian-based crowdsourcing technique which permits the calibration (defined as the determination of their physical locations in a floorplan) of static RF emitting beacons, from the RSS measurements gathered by users which move freely in the environment, without requisites on their displacement trajectories or any special action from their part, or the installation of additional sensors such as inertial motion units (although it can benefit from them).

The remainder of this paper is organized in the following way. Section II introduces the general approach of the Bayesian method for autocalibration of a wireless positioning network, and then a step-by-step description of the adaptation of the FastSLAM technique to such goal. Section III presents the experimental device for the demonstration of the performance of the methods, and section IV the experimental results along with an analysis of the obtained positioning accuracy. Finally, we expose some conclusions and point to future lines to expand the current work.

II. Theory

A. The Bayesian approach to calibration

This paper presents a calibration method based upon the FastSLAM algorithm developed in Robotics for the classic SLAM (Simultaneous Localization and Mapping) problem, in which a robot navigates an unknown environment, simultaneously localizing itself and creating a map of its environment. Translating this scheme to the calibration of a wireless positioning network, the idea is that, during a short, random walk taken by a person covering the displacement area, an automatic method can recover the position of all unknown nodes (that have a unique identification code), and additionally provide an estimate of his own path (see Figure 1). Some additional information such as measurement parameters and the location of a few anchor nodes is provided before calibration.

In the Bayesian framework, the beacon locations and the user's trajectory are formulated as probability distributions.

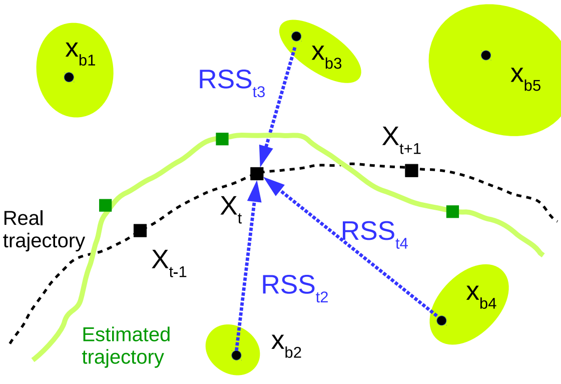

Fig. 1. Autocalibration process. The real trajectory ($\mathbf{x}_t$) and the actual beacon locations ($\mathbf{x}_{bj}$) are shown in black, while signal strength measurements (RSS$_{jt}$) received at time t are represented by blue dashed lines. Green filled ellipses denote beacon position estimates with mean μ_{jt} and covariance error Σ_{jt}, and the green continuous line the estimate of the user's trajectory.

Then the objective of the autocalibration process is computing the following posterior probability [14]:

$$p(\mathbf{x}_{0:t}, \mathbf{x}_b \,|\, \mathbf{z}_{0:t}, \mathbf{u}_{0:t}), \tag{1}$$

where $\mathbf{x}_{0:t}$ is the trajectory followed by the person from time 0 to time t, $\mathbf{x}_b = \{\mathbf{x}_{bj}, j = 1, \ldots, N_b\}$ is the position of the beacon nodes to be calibrated, $\mathbf{z}_{0:t}$ is the set of signal strength measurements received from time 0 to time t, and (optionally) $\mathbf{u}_{0:t}$ corresponds to displacement information provided by some inner sensor (in our case, the inertial motion unit contained in the smartphone) in the same time interval. We are interested in a recursive solution of Equation 1, which refines the beacon position estimates as more measurements are acquired.

The FastSLAM method, introduced in [15], decomposes the problem into tracking of the user position and individual estimation of the beacon positions, which are conditionally independent once the user's trajectory is known - or estimated. The posterior probability of Equation 1 is factorized in the following way:

$$p(\mathbf{x}_{0:t}, \mathbf{x}_b \,|\, \mathbf{z}_{0:t}, \mathbf{u}_{0:t}) = p(\mathbf{x}_b \,|\, \mathbf{x}_{0:t}, \mathbf{z}_{0:t}) p(\mathbf{x}_{0:t} \,|\, \mathbf{z}_{0:t}, \mathbf{u}_{0:t}), \tag{2}$$

where, due to the conditional independence of the beacon locations,

$$p(\mathbf{x}_b \,|\, \mathbf{x}_{0:t}, \mathbf{z}_{0:t}) = \prod_{j=1}^{N_b} p(\mathbf{x}_{bj} \,|\, \mathbf{x}_{0:t}, \mathbf{z}_{0:t}). \tag{3}$$

Due to the partition in Equation 2 (Rao-Blackwellization), we can represent the beacon locations analytically, and we only have to sample the user's trajectory. This is the key property of the FastSLAM approach which allows to keep the complexity of the problem linear with the number of beacons.

To sample the user's location we use a particle filter (PF), which contains multiple hypotheses on the shape of the posterior of Equation 1, with each particle of the filter having the following structure:

$$X_t^k = \{x_t^k, y_t^k, \mu_{1t}^k, \Sigma_{1t}^k, \mu_{2t}^k, \Sigma_{2t}^k, \ldots, \mu_{N_b t}^k, \Sigma_{N_b t}^k, w_t^k\}, \quad (4)$$

where $k = 1, \ldots, N_p$ is the particle index, N_p is the number of particles, (x_t^k, y_t^k) is the user's position at time t, and $(\mu_{jt}^k, \Sigma_{jt}^k)$ are the estimated mean position and covariance error of the j-th beacon. The last element of Equation 4 is the weight w_t^k, which stands for the relative probability of the hypothesis represented by that particle. Weights are normalized such that $\sum_k w_t^k = 1$.

Two further relationships are needed to process the sensorial information with Bayesian filters. The first is the observation or measurement model, relating the received measurement with the current user and beacon location:

$$p(z_j \mid \mathbf{x}, \mathbf{x}_{bj}). \quad (5)$$

The second is the motion model which describes the state transition:

$$p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{u}_t), \quad (6)$$

where displacement $\mathbf{u}_t$ is measured with a motion sensor carried by the user. Suitable observation and motion models for our problem are given in the following section.

B. Adaptation of FastSLAM method to the autocalibration of a sensor network

The FastSLAM particle filter adapted to the calibration process is shown in Figure 2, and is explained below step by step. The goal is estimating the two-dimensional location of N_b beacons, from which we can receive signal strength readings (RSS) emitted periodically. The identity of the emitting beacon is contained in the received RF message and is known unambiguously. We also assume that there is also a small set of anchor beacons at known locations; with them we can refer the estimated coordinates of the unknown beacons to the floorplan of the building.

The PF updates sequentially the different states of X_t^k for each particle as RSS readings or motion information are produced. Discrete index t stands for time instants where new information is available for the PF.

Initialization. To accommodate the particle structure in Equation 4, let $\mathbf{x}$ be a $2 \times N_p$ array containing the user location estimate, and μ (size $2 \times N_b \times N_p$) and Σ (size $2 \times 2 \times N_b \times N_p$) be two arrays representing the mean and covariance of the location of the beacons for each particle. Indices $j = 1, \ldots, N_b$ and $k = 1, \ldots, N_p$ stand respectively for the beacon and particle index. We initialize μ_{j0}^k to random values in our environment, and Σ_{j0}^k to the identity matrix multiplied by a large constant the size of our building or larger. The initial user position, $\mathbf{x}_0^k$, can be taken anywhere in our building, or in the correct start position, say with room accuracy (the issue of initialization of the PF is discussed in section IV-C). Particle weights are initially assigned to a constant value, $w_0^k = 1/N_p$.

These initial values define a proposal distribution for X_0^k. The PF next measures how this proposal fits the experimental RSS measurements from the detected beacons, and updates the particle weights and beacon location estimates.

RSS measurement from an anchor beacon. Upon reception of signal strength measurement z_t (associated to an anchor beacon), we correct particle weights depending on the difference between the expected and the actual measurements. For particle k, the predicted measurement is:

$$\hat{z}_t^k = h(\mathbf{x}_t^k, \mathbf{x}_a),$$

where $\mathbf{x}_a$ is the (known) anchor beacon location, and h is the measurement function which relates the positions of the beacon and user with the measurement. As measurement model we will use the standard path loss law [16]:

$$z = h(\mathbf{x}, \mathbf{x}_a) + e_z = \mathrm{RSS}_0 - 10\alpha \log_{10} \frac{\|\mathbf{x} - \mathbf{x}_a\|}{d_0} + e_{\mathrm{RSS}}, \quad (7)$$

where d_0 is a reference distance, RSS_0 is the signal strength at distance d_0, α is the path loss exponent, and e_{RSS} represents fading noise that will be modeled as Gaussian distributed, $e_{\mathrm{RSS}} \sim \mathcal{N}(0, \sigma_{\mathrm{RSS}}^2)$. The particle weights are updated in order to increase the probability of particles whose location better matches the experimental RSS reading:

$$w_t^k = w_t^k \cdot \frac{1}{\sqrt{2\pi\sigma_{\mathrm{RSS}}^2}} \exp(-\frac{(z_t - \hat{z}_t^k)^2}{2\sigma_{\mathrm{RSS}}^2}). \quad (8)$$

RSS measurement from an unknown position beacon. Upon reception of signal strength measurement z_t (associated to beacon j at an unknown location), both the user position and the estimated beacon location need to be updated in the PF. The predicted RSS value for the k-th particle:

$$\hat{z}_t^k = h(\mathbf{x}_t^k, \mu_{j,t-1}^k),$$

where h is again the path loss law (Equation 7), and we use the mean estimate of the beacon location, $\mu_{j,t-1}^k$ for each particle.

Update beacon location. This is done analytically with the Extended Kalman Filter (EKF) equations. Let $H(\mathbf{x}, \mathbf{x}_{bj}) = \frac{\partial h(\mathbf{x}, \mathbf{x}_{bj})}{\partial \mathbf{x}_b j}$ be the Jacobian of the nonlinear function h with respect to the position of the j-th emitting beacon.

Using Equation 7, we evaluate this Jacobian at positions $(\mathbf{x}_t^k, \mu_{j,t-1}^k)$:

$$H = \frac{10\alpha}{\|\mathbf{x}_t^k - \mu_{j,t-1}^k\|^2} (\mathbf{x}_t^k - \mu_{j,t-1}^k)^T.$$

The Kalman gain is then computed as:

$$K = \Sigma_{j,t-1}^k H^T (H \Sigma_{j,t-1}^k H^T + Q_t)^{-1},$$

where Q_t is a scalar covariance value. The mean and covariance of the observed $j-$th beacon are updated as:

$$\mu_{j,t}^k = \mu_{j,t-1}^k + K(z_t - \hat{z}_t^k) \quad (9)$$

$$\Sigma_{j,t}^k = (1 - KH)\Sigma_{j,t-1}^k, \quad (10)$$

for all particles (k), while the estimates for the remaining beacons maintain undisturbed.

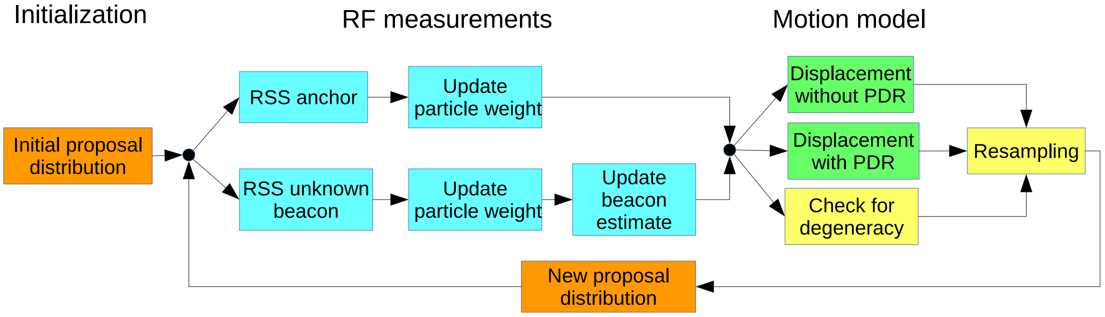

Fig. 2. Block diagram of the different stages of the autocalibration method, as explained in section II-B.

Particle weight update The particle weights are updated with the actual RSS readings in a similar way as Equation 8:

$$w_t^k = w_t^k \cdot \frac{1}{\sqrt{2\pi|Q|}} \exp(-\frac{(z_t - \hat{z}_t^k)^2}{2Q}), \tag{11}$$

with $Q = H\Sigma_{j,t-1}^k H^T + Q_t$.

Generation of a proposal distribution. Periodically, a new proposal distribution for X_t must be produced from the previous X_{t-1}, by resampling the particles and displacing the coordinates of the user location, so his motion can be correctly tracked. In our implementation, this new proposal is generated in any of these three events: (a) a given time has elapsed since the last resampling, (b) a step taken by the user has been detected (if PDR information is available), (c) a degeneracy condition is detected. We explain each circumstance now.

Particle displacement and resampling (without PDR) In the case where no motion information is available, or if a given time T_{max} has passed without detecting a step, all particles are displaced from their current position by:

$$\begin{aligned} x_t^k &= x_{t-1}^{k'} + \delta_x \\ y_t^k &= y_{t-1}^{k'} + \delta_y, \end{aligned} \tag{12}$$

where (δ_x, δ_y) are sampled from a uniform distribution in the circle defined by $\delta_x^2 + \delta_y^2 \leq v_{\text{max}}^2 T_{\text{max}}^2$, and particle index k' is drawn from the set $k' \in \{1, \ldots, N_p\}$ with probability w_{t-1}^k.

Particle displacement and resampling (with PDR) Particle displacement can be done more efficiently if we have direct motion measurements. Pedestrian Dead Reckoning (PDR) algorithms use data provided by the inertial motion unit contained in a conventional smartphone, and produce a sequence of steps of the form [17]: $\{l_t, \theta_t\}$, where t is the time where a step is completed, l_t is the step length and θ_t the orientation of the user (or trajectory heading). For each detected step, the k-th particle's position is updated from the previously estimated position (x_{t-1}, y_{t-1}) as:

$$\begin{aligned} x_t^k &= x_{t-1}^{k'} + (l_t + \delta_l)\cos(\theta_t + \delta_\theta) \\ y_t^k &= y_{t-1}^{k'} + (l_t + \delta_l)\sin(\theta_t + \delta_\theta), \end{aligned} \tag{13}$$

where δ_l and $\delta_{\Delta\theta}$ are Gaussian modelled disturbances on both the step length and estimated orientation, which are distributed according to $\delta_l \sim \mathcal{N}(0, \sigma_{l\text{PDR}}^2)$ and $\delta_\theta \sim \mathcal{N}(0, \sigma_{\theta\text{PDR}}^2)$. The index k' is chosen in the same way as in Equation 12.

Particle degeneracy. This condition happens when the number of particles with significant weights is greatly reduced, leading to particle impoverishment. In our implementation, we compute the number of effective particles (as defined in [18]), and perform a resampling step (without motion), if it falls below threshold $N_p/10$. This event occurs rarely in our experiments.

In any of the three instances of particle resampling, the particle states corresponding to the beacons mean position and covariance error are resampled from the same probability distributions as the user coordinates: $\mu_{jt}^k = \mu_{jt}^{k'}$, $\Sigma_{jt}^k = \Sigma_{jt}^{k'}$. Additionally, all weights are reassigned to a constant value, $w_t^k = 1/N_p$.

After resampling, a new proposal distribution for X_t is produced, which will be weighted as new RF signals arrive. This process is repeated recursively until all the data obtained during the trajectory has been processed. It is expected that the particle ensemble mean $\{\mu_{jt}^k\}$ converge to the true beacon locations $\{\mathbf{x}_{bj}\}$, while the covariances $\{\Sigma_{jt}^k\}$ decrease as more RF readings are acquired.

III. Experimental device

We use RFID technology provided by RFCode (see Figure 3), consisting in a set of active tags (model M100) placed at fixed positions in our research building, and a portable reader (model M220) attached to the user's hip, equipped with two 1/4 wave articulated helical antennas. The reader decodes the RF signal transmitted at 433 MHz by the tags and reports the tag ID, the measured RSS at both antennas and a timestamp to an Android-based mobile phone through a Bluetooth link. Additionally, the accelerometer, gyroscope and magnetometer signals from the inertial motion unit (IMU) of the phone are sampled at an average rate of 50 Hz, which is high enough for PDR algorithms. GNSS data (when available) is also acquired by the program, but not used in our experiments, a difference with the war-driving techniques for network calibration. RFID technology is used for convenience, given our previous experience with the system; however the autocalibration technique is applicable to any RF-based indoor positioning system which provides RSS measurements, such as wifi or Bluetooth.

The tag detection range in our indoor environment is 20 m for 50% percent of emissions. We have used a setup with

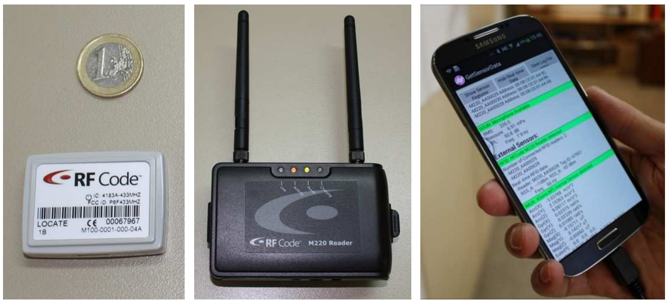

Fig. 3. Experimental device for demonstration of the autolocalization method, consisting of several active RFID tags, an RFID reader, and a smartphone.

TABLE I
PARAMETERS FOR THE AUTOCALIBRATION ALGORITHM

Path loss law (Equation 7)		
Emitted RSS at ref. distance	RSS_0	-61.5 dBm
Loss exponent	α	2.30 dBm
Reference distance	d_0	1 m
Particle filter		
Number of particles	N_p	10,000
RSS noise variance (Equation 8)	σ^2_{RSS}	10^2 dBm^2
Kalman covariance (Equation 11)	Q_t	20^2 dBm^2
Motion model (Equations 12 and 13)		
Resampling time	$T_{\max}$	2 s
Maximum speed	$v_{\max}$	2 m/s
Noise on step length	$\sigma_{l\mathrm{PDR}}$	0.3 m
Noise on heading angle	$\sigma_{\theta\mathrm{PDR}}$	0.3 rad

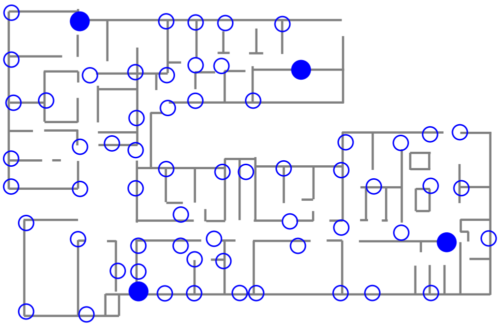

Fig. 4. Floorplan. The objective of the autocalibration method is finding the location of 60 unknown position tags (hollow circles); additionally, there are 4 known location tags or anchors (full circles). The dimensions of the building are 65 × 40 m; its area 1600 m^2.

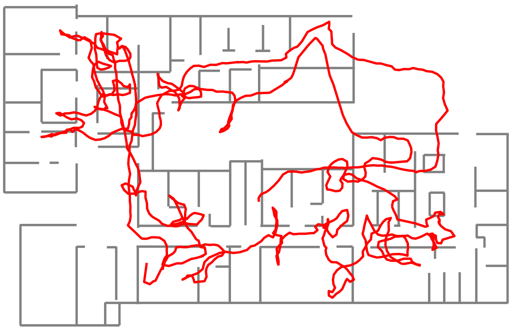

Fig. 5. Approximate actual trajectory followed by the user for the calibration of the sensor network in Figure 4, reconstructed from PDR information and RF beacon corrections.

64 tags distributed in our building (shown in Figure 4), of which only 4 are taken as anchor nodes with known positions, and the remaining 60 are beacons whose locations must be estimated. Anchor beacons are used for convenience to refer all positions to a common frame of coordinates; however the autocalibration procedure can be carried out with all nodes having unknown positions, although the positioned nodes would not be referred to a particular coordinate system.

SLAM algorithms exist in two versions: with and without known correspondence, meaning that the received measurements can be associated unambiguously with a emitting beacon in the environment or not. In our experiments, this correspondence is obtained from the unique identification number encoded in the RF signal emitted by each RFID tag. This would be the case too for wifi access points or Bluetooth tags, using their respective Media Access Control (MAC) addresses.

A. Algorithm parameters

The parameters for the FastSLAM algorithm applied to autocalibration of the RFID network are summarized in Table I. Values for RSS_0, α and σ_{RSS} were obtained in previous works with the same setup [19], and are average values valid for both line of sight (LOS) and non-line of sight (NLOS) situations in our building. Our general experience with Equation 7 as a measurement model in indoor localization problems is that it is convenient to use a conservative (i.e., large) value for the measurement noise variance, so the PF does not become overconfident on some RSS values.

We have used $N_p = 10,000$ particles for the autocalibration PF; an increase in the number of particles does not improve the performance of the method significantly.

IV. Experimental Results and Discussion

In the experiments carried out to assess the performance of the autocalibration method, a person walks in our building (remaining mostly indoors) carrying the smartphone and the RFID reader. This person is not required to follow a predefined trajectory or pass through determinate landmarks (as usual in fingerprinting calibration), but instead performs a natural, roughly circular route, from which most of the tags can be detected by the reader at one instant.

One such trajectory is shown in Figure 5, consisting of two loops through the interior part of our building. The length of this sample trajectory is 714 m and its duration 1054 s. This “groundtruth” trajectory has been reconstructed approximately from the PDR deduced displacement with corrections from the RFID beacons (however, the trajectory itself is not available for the autocalibration methods evaluated next). During the time taken to traverse this route, we received 1500 readings from the 4 anchor tags, and 25,700 readings from the 60 unknown location tags.

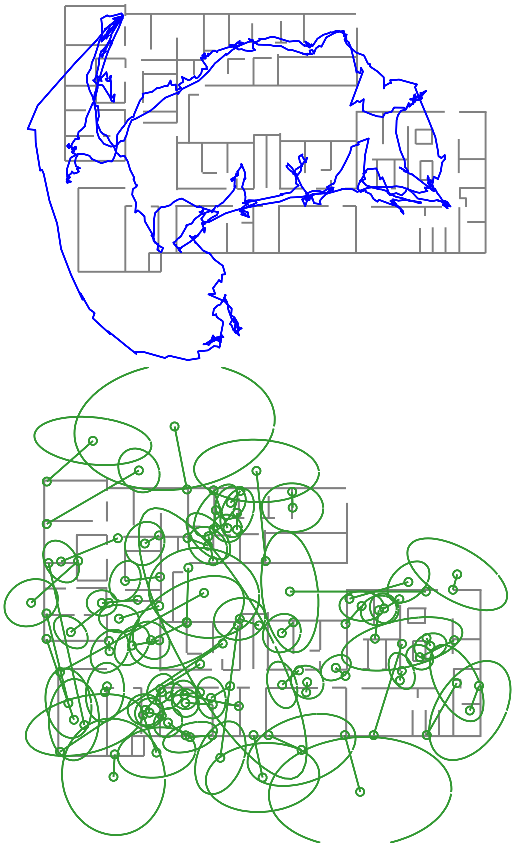

Fig. 6. Results of the FastSLAM-based autocalibration algorithm, using only RSS data. Top: estimated trajectory; bottom: estimated positions for the 60 unknown location RFID tags, along with the 90% confidence ellipse for each.

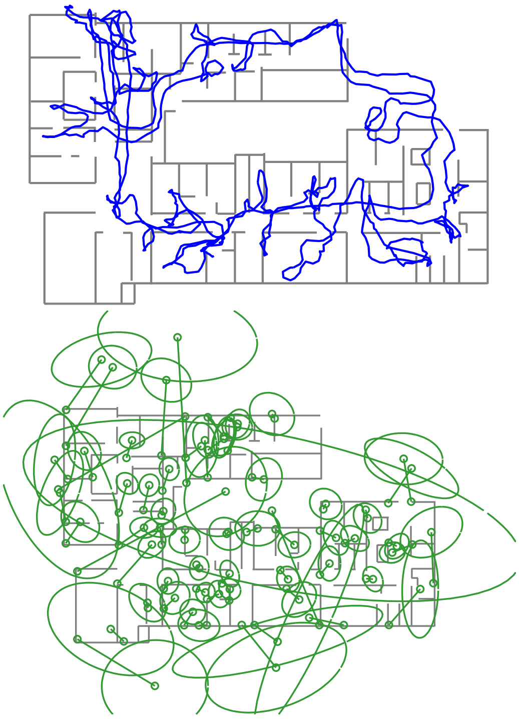

Fig. 7. Results of the FastSLAM-based autocalibration algorithm, using RSS and PDR data. Top: estimated trajectory; bottom: estimated positions for the 60 unknown location RFID tags, along with the 90% confidence ellipse for each.

A. *Beacon positioning with RSS-only measurements*

We first show the autocalibration results obtained when only RSS readings are used, and no PDR information is available (the motion model of the PF takes the simple form in Equation 12) in Figure 6, both for the reconstructed trajectory and the estimated position of the unknown beacons. We have initialized the user's position with particles distributed at random in the correct room. The obtained median positioning error for the beacons is 4.9 m. We can see that, while in the interior part of the building most beacons are correctly located, those in the periphery have larger errors; in many cases the 90% confidence ellipse does not contain the true beacon position. The estimated user trajectory is also largely deviated from the groundtruth shown previously in Figure 5.

B. *Enhancement with PDR*

Use of Pedestrian Dead Reckoning data in the particle displacement stage (Equation 13) improves the estimation of the user's trajectory and, consequently, the estimates of the beacon locations. The IMU signals are processed with a step and heading system (SHS) [17] which produces a sequence of step lengths with orientation (heading) for the motion model of Equation 13. The step length l_t of a walking person is estimated from the vertical motion (bounce) of the upper body, measured by the phone accelerometer, while the heading θ_t at each step is measured with the magnetometer. This provides reasonably accurate results in our building, although it may exhibit problems in buildings with metallic parts [20]. Step detection is not perfect: some steps are missed, and extraneous ones can be detected, for example when a door is opened with an arm, and its motion is coupled to the opposite arm holding the phone.

The results of the RSS+PDR autocalibration are shown in Figure 7. The median positioning error for the beacons is reduced to 3.4 m, and the estimated trajectory is also improved, more closely resembling the groundtruth in Figure 5.

C. *Analysis of results*

Positioning with the autocalibrated network. Figure 8 show positioning examples obtained with a particle filter (without using PDR) using the calibrated beacon positions estimated in the last section, for a simple C-shaped trajectory along the main corridor in our building. The top trajectory is obtained using only RSS-signals from the 4 anchor nodes; the results are poor due to the low number of nodes. The bottom one is computed with the RSS-signals of the 60-node network (without the anchor nodes) with the node positions given in Figure 6, showing how the trajectory can be successfully tracked with the calibrated network. Although it may seem that the positioned nodes obtained through autocalibration are very approximate, they produce sensible positioning results in practice (albeit with dense networks).

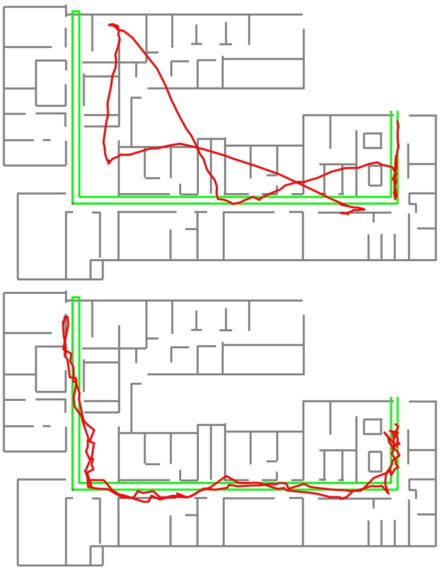

Fig. 8. Tracking of a simple C-shaped trajectory (green trace) by a particle filter with the RSS signals (without using PDR information) obtained from a network with only 4 anchor nodes (top); the calibrated network of Figure 6 (60 beacons, no anchor nodes) (bottom). Similar tracking performance is obtained with the calibrated network of Figure 7.

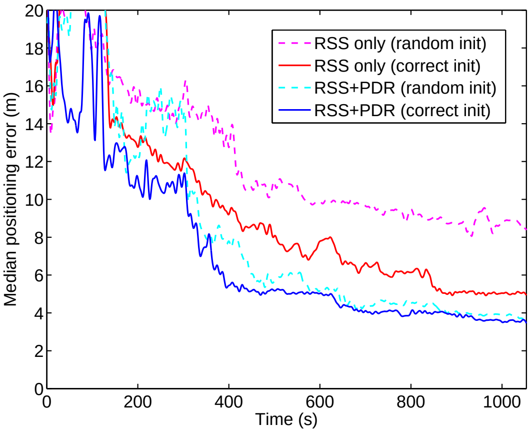

Fig. 9. Effect of the user position's initialization (at random points in the building or in the correct room) on the convergence speed and obtained beacon location accuracy, for the RSS-based and RSS+PDR-based versions of the autocalibration method.

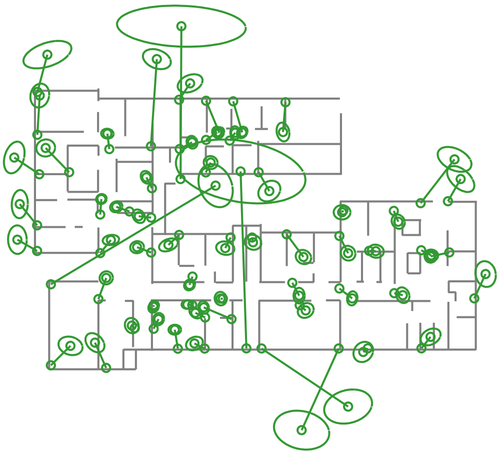

Fig. 10. Estimated beacon locations, accumulating the measurements obtained with 5 different routes. The median positioning error is 2.3 m, although a few beacon estimates remain largely deviated from their true locations.

The effect of initialization. The user's position part of the particle filter (coordinates (x_0^k, y_0^k) in Equation 4) can be initialized at random locations in the building or at the true position with, for example, room accuracy (however, the coordinates for the beacon locations, $\{\mu_{j0}^k\}$, are *always* initialized to random values within the building). In practice, this depends on whether the user will start the autocalibration process from a predetermined location, or completely at random. The impact of initialization on the convergence and final accuracy of the autocalibration method is shown in Figure 9. We see that, while both the RSS-based and the RSS+PDR-based filters converge more slowly when initialized at random locations, the effect is relatively minor for the second case, since PDR data helps the estimate of the user's position to get "back in track" relatively quickly.

Crowdsourcing measurements. From Figure 9 it is apparent that the beacons positioning error decreases with traversed distance, implying that better accuracy can be obtained with sufficiently long trajectories. One positive feature of the autocalibration technique presented in this work is that it is accumulative: data from consecutive experiments can be used to progressively improve the estimation of the beacons. Since the calibration routes can be performed by different persons, the method lends itself to crowdsourcing measurement campaigns.

In Figure 10, we show the beacon position estimation obtained by combining five different experiments, totaling a traveled distance of 4470 m in 6110 s. The median positioning error is reduced to 2.3 m. A few RFID tags still have abnormally large location errors, which can be probably attributed to physical problems with them (for example, low charge in their batteries or being surrounded by materials with high attenuation). We are investigating this issue to obtain a better explanation of the behaviour of these beacons.

Histogram with beacon positioning results. A histogram showing the calibrated beacons position error is shown in Figure 11, for one experiment using RSS signals only (section IV-A), RSS signals combined with PDR (section IV-B), and for the accumulated run of five different routes.

V. Conclusions and Future Work

In this work, we have demonstrated that a Simultaneous Localization and Mapping (SLAM) technique from Robotics can be used for the autocalibration (determination of the physical position) of a beacon network such as those used for indoor local positioning systems. The technique is computationally efficient and converges to the true location of the beacons for most of them, in spite of using noisy signal strength measurements from the received RF signals. Unlike inverse localization or fingerprinting calibration, the method does not

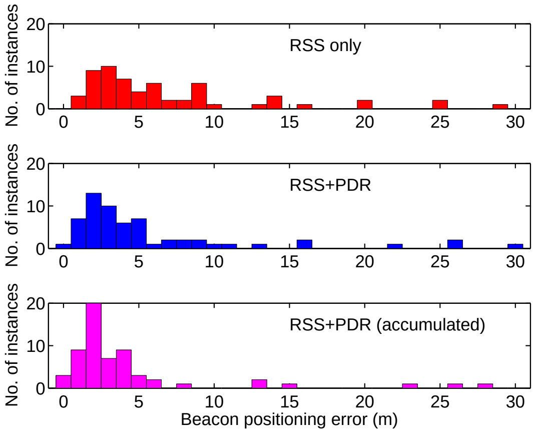

Fig. 11. Histogram with the beacon positioning error after calibration. The median positioning errors are 4.9 m (RSS only), 3.4 m (RSS and PDR combined), and 2.3 m (RSS+PDR, accumulated for several routes).

require the user to follow a predefined course, to position himself at given coordinate points, or to record his location periodically. Instead, the user walks along an arbitrary route while the calibration is performed automatically.

The autocalibration is improved if a simple PDR algorithm is used to process the signals acquired from a low grade IMU sensor of a conventional smartphone carried by hand. In one demonstration trajectory of length 714 m, we achieved a median positioning error of 3.4 m for a network of 60 beacons; this accuracy can be further increased with longer routes.

This work can be extended in several ways. Incorporating the building map to the particle filter's motion model would improve the estimated trajectory and in principle reduce the beacon positioning error. As a second possibility, we could adapt the FastSLAM 2.0 method [21] to the autocalibration problem; this method uses a proposal distribution taking into account both the motion model and the current observations, and compensates for a motion model with large errors compared to the measurement model. However, it is possible that at this stage, the limitations of the autocalibration process arise not from the estimated trajectory but from the large variance of RSS measurements, and not much improvement can be obtained from these methods. For this purpose, the beacon update estimates of the autocalibration could be processed with the Unscented Kalman Filter (UKF) version of FastSLAM [22], which is better suited to model the nonlinearities of the path loss law equation than the EKF filter used in our work.

Finally, it is possible that the beacon positioning errors are ultimately due to individual RFID tag variations (like their transmitting power). The autocalibration algorithm could be modified to account for this situation, estimating the beacon individual characteristics during the autocalibration process.

Acknowledgements

Financial support for this work comes from projects TARSIUS (ref. TIN2015-71564-C4-2-R) and REPNIN (ref TEC2015-71426-REDT).

References

[1] R. Zekavat and R. M. Buehrer, *Handbook of position location: Theory, practice and advances*. John Wiley & Sons, 2011.

[2] R. Casas, D. Cuartielles, A. Marco, H. J. Gracia, and J. L. Falco, "Hidden issues in deploying an indoor location system," *IEEE Pervasive Computing*, vol. 6, no. 2, pp. 62–69, 2007.

[3] M. Brunato and R. Battiti, "Statistical learning theory for location fingerprinting in wireless LANs," *Computer Networks*, vol. 47, no. 6, pp. 825 – 845, 2005.

[4] N. Patwari, J. Ash, S. Kyperountas, A. Hero, R. Moses, and N. Correal, "Locating the nodes: cooperative localization in wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 54–69, July 2005.

[5] J. Guevara, A. R. Jiménez, J. C. Prieto, and F. Seco, "Auto-localization algorithm for local positioning systems," *Ad Hoc Networks*, vol. 10, no. 6, pp. 1090–1100, 2012.

[6] J. Ureña, D. Ruiz, J. C. García, J. J. García, A. Hernández, and M. C. Pérez, "LPS self-calibration method using a mobile robot," in *IEEE International Instrumentation and Measurement Technology Conference*, May 2011, pp. 1–6.

[7] M. Kok, J. D. Hol, and T. B. Schön, "Indoor positioning using ultra-wideband and inertial measurements," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, pp. 1293–1303, 2015.

[8] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.

[9] A. LaMarca, J. Hightower, I. Smith, and S. Consolvo, "Self-mapping in 802.11 location systems," *Ubiquitous Computing (UbiComp)*, pp. 903–903, 2005.

[10] P. Robertson, M. Angermann, and B. Krach, "Simultaneous localization and mapping for pedestrians using only foot-mounted inertial sensors," in *Proceedings of the 11th international conference on Ubiquitous computing*. ACM, 2009, pp. 93–96.

[11] L. Bruno and P. Robertson, "Wislam: Improving footSLAM with wifi," in *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2011, pp. 1–10.

[12] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: Zero-effort crowdsourcing for indoor localization," in *Proceedings of the 18th annual international conference on Mobile computing and networking*. ACM, 2012, pp. 293–304.

[13] Y. Zhuang, Y. Li, H. Lan, Z. Syed, and N. El-Sheimy, "Smartphone-based wifi access point localisation and propagation parameter estimation using crowdsourcing," *Electronics Letters*, vol. 51, no. 17, pp. 1380–1382, 2015.

[14] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," *IEEE robotics & automation magazine*, vol. 13, no. 2, pp. 99–110, 2006.

[15] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit *et al.*, "Fastslam: A factored solution to the simultaneous localization and mapping problem," in *Proceedings of the 18th Conference on Artificial Intelligence (AAAI)*, 2002, pp. 593–598.

[16] S. Y. Seidel and T. S. Rappaport, "914 MHz path loss prediction models for indoor wireless communications in multifloored buildings," *IEEE Trans. on Antennas and Propagation*, vol. 40, no. 2, pp. 207–217, 1992.

[17] R. Harle, "A survey of indoor inertial positioning systems for pedestrians," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1281–1293, 2013.

[18] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman filter: Particle filters for tracking applications*. Artech House, London, 2004.

[19] F. Seco, A. R. Jiménez, and X. Zheng, "RFID-based centralized cooperative localization," in *IEEE International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Oct 2016, pp. 1–7.

[20] A. R. Jiménez, F. Seco, C. Prieto, and J. Guevara, "A comparison of pedestrian dead-reckoning algorithms using a low-cost MEMS IMU," in *IEEE International Symposium on Intelligent Signal Processing (WISP)*, Aug 2009, pp. 37–42.

[21] M. Montemerlo and S. Thrun, "Fastslam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges," in *Proceedings International Joint Conference on Artificial Intelligence*, 2003, pp. 1151–1156.

[22] C. Kim, R. Sakthivel, and W. K. Chung, "Unscented FastSLAM: A robust and efficient solution to the SLAM problem," *IEEE Transactions on Robotics*, vol. 24, no. 4, pp. 808–820, 2008.