Int. J. Electron. Commun. (AEÜ) 70 (2016) 1105–1113

Contents lists available at ScienceDirect

International Journal of Electronics and Communications (AEÜ)

journal homepage: www.elsevier.com/locate/aeue

Regular Paper

Intelligent fusion of information derived from received signal strength and inertial measurements for indoor wireless localization

Li Li [a], Wang Yang [a,*], Guojun Wang [b,a]

[a] School of Information Science and Engineering, Central South University, Changsha 410083, China
[b] School of Computer Science and Educational Software, Guangzhou University, Guangzhou 510006, China

ARTICLE INFO

Article history:
Received 25 September 2015
Accepted 28 April 2016

Keywords:
Hierarchical Bayesian hidden Markov model
Unsupervised learning
Wireless localization
Inertial measurement unit

ABSTRACT

In this paper, we focus on improving the accuracy of wireless localization in wireless sensor networks using information derived from the inertial measurement unit (IMU) in a smartphone and the received signal strength (RSS). We propose an algorithm to relate the RSSs and measurements obtained from the IMU to the coordinates of an indoor robot. To deal with the dynamic nature of fingerprint information in an indoor radio environment, we first use the hierarchical Bayesian hidden Markov model (HB-HMM) to process a time series of RSSs. Unlike other HMM-based methods, the HB-HMM depends only on a single initial hyper-parameter for global optimization. Next, we evaluate the measurements obtained from the IMU to identify the robot's state, which includes the rotating, moving, and bumping states. We used the IMU accelerometers to estimate the velocity. Lastly, a method based on the particle filter (PF) was used to fuse the results obtained from RSS and IMU. Experiments show that our algorithm can achieve better accuracy than related algorithms.

1. Introduction

Nowadays, users are demanding an increasing diversity of applications from wireless sensor networks (WSNs) [1]. Among these applications, indoor localization has emerged as a key function of wireless sensor networks for tasks including target tracking, robotics and navigation, smart space, intrusion detection, and inventory management [2]. Presently, most commercial WSNs are based on the IEEE 802.15.4 infrastructure, which can be incorporated into an Internet of Things (IoT) architecture with internet protocol version 6 over low power wireless personal area networks (6LoWPAN) [3]. It is widely believed that this will develop into one of the more pervasive wireless technologies of the future.

In particular, indoor wireless localization based on received signal strength (RSS) in an IEEE 802.15.4 infrastructure has long been a challenging problem [4,5]. On the one hand, RSS information can be acquired internally through the use of a WSN communication device, such that indoor localization based on RSS is an off-the-shelf commodity for WSNs. On the other hand, in an indoor wireless environment characterized by non-line of sight (NLOS) and multipath conditions, the distance-RSS conversion model becomes inaccurate.

Within an IoT environment, a smartphone can get data from the WSN [3]. Moreover, smartphones currently incorporate a number of inertial sensors, including accelerometers, gyroscopes, and magnetic compasses, within an inertial measurement unit (IMU). The fusion of data derived from RSS and IMU signals can then be employed to avoid the performance limitations associated with RSS alone, and represents a tangible solution for indoor localization in WSNs.

However, the performance of inertial sensors in smartphones is limited. Many studies have focused on using the peak-detection algorithm to analyze pedestrian gait in indoor environments [6]. Less attention has been paid to the more challenging problem of locating and tracking mobile robots or vehicles in an indoor environment using the IMU signals of a smartphone. In this paper, our goal is to enhance the value of WSNs in an IoT environment to provide accurate robot/vehicle location and tracking capabilities, which are of paramount importance for many applications such as smart space and inventory management.

In the field of indoor localization using WSNs, fingerprinting [7], trilateration [8], close-proximity [9], dead reckoning [10], and Bayesian filters [11] are commonly used techniques. Of these techniques, the close-proximity technique is the simplest, where a mobile target is assumed to be located in an area nearest the reference node transmitting the strongest signal. Systems employing this technique can provide only a coarse location and require a

* Corresponding author.
E-mail address: yangwang@csu.edu.cn (W. Yang).

http://dx.doi.org/10.1016/j.aeue.2016.04.017
1434-8411/

high density of reference nodes. In fact, RSS can provide a wider range of information based on logarithmic propagation models. According to theoretical and empirical models, the RSS can be used to establish the distance between the transmitter and the receiver [12]. Then, using trilateration or another range-based technique such as weighted least squares (WLS) [13], Semidefinite programming (SDP) [14] and Multidimensional scaling (MDS) [15], the locations of mobile devices within the WSN can be estimated.

However, the unpredictability of NLOS signal propagation and multipath propagation through indoor environments are major challenges in indoor localization. To mitigate the influence of NLOS and multipath errors, fingerprinting techniques have recently been examined. In these techniques, RSS values at specific locations are used as fingerprint information. This is also referred to as a range-free algorithm. Because RSS patterns are directly mapped out according to specific locations, the measurement errors caused by NLOS and the multipath environment are avoided. However, a significant investment of human labor is typically required to build a detailed map-database. Additionally, some authors have proposed Bayesian filter techniques to accumulate information from consecutive measurements. Among Bayesian filters, the extended Kalman filter (EKF) and the particle filter (PF) have been widely used in the field of indoor localization. However, the performance of Bayesian filter techniques depends on a priori knowledge of the statistical distribution of the measurement noise. The dead reckoning technique incorporates measurements derived from inertial sensors, and has also been used in indoor localization. This technique uses an accelerometer to perform distance detection, and then uses a compass for heading direction estimation. While the technique requires no infrastructure assistance, the location accuracy is limited by the performance of the inertial sensors.

In this paper, we propose an unsupervised learning algorithm, namely HPFRI (HB-HMM and Particle Filter for the Fusion of RSS and Inertial Measurements), which employs the hierarchical Bayesian hidden Markov model (HB-HMM) for the RSS time series analysis and the PF for fusion of the RSS and inertial measurements obtained from the IMU in a smartphone, for improved accuracy of wireless localization in WSNs. With respect to other localization methods employed in WSNs, the contributions of the present study are as follows:

- Because we generate the HB-HMM model of RSS values to estimate the location, and use the information obtained from the IMU which is completely irrelevant to signal propagation in the indoor environment, measurement errors caused by NLOS and multi-path is mitigated.
- Our algorithm is based on unsupervised learning that accumulates information through consecutive observation, and, thus, the RSS value is automatically collected.
- Novel methods are employed to identify the robot state based on IMU information.
- A PF method is employed for the fusion of RSS and IMU data.

The remainder of this paper is organized as follows. Section 2 provides an overview of related localization systems. The problem statement regarding indoor wireless localization is introduced in Section 3. Subsequently, Section 4 presents the HPFRI algorithm. Section 5 discusses the experiments run in indoor scenarios. The paper closes with conclusions in Section 6.

2. Related work

A number of localization techniques based on RSS and IMU data have been developed for WSNs. For instance, in response to an initiative for the development of a real-time indoor localization system for tracking people, D'Souza et al. designed the Indoor People Tracker, an approach that employed the proximity to static nodes in the WSN in conjunction with data obtained from onboard inertial sensors [16]. A multi-hypothesis estimation algorithm based on dead reckoning and Monte Carlo approaches was applied to the Indoor People Tracker, which attained an indoor resolution of 2.5 m. In the field of indoor localization based on inertial sensors, Bayesian filter techniques have been most commonly used. Correa et al. proposed an enhanced filtering method (Power Threshold Covariance Matrix Tuning, PT-CMT) for indoor localization using a WSN that employed the inertial measurements obtained from a nine-degrees of freedom IMU. The experimental results verified that the system correctly estimated the location of a mobile terminal indoors within an area of around 1 m × 1 m [17]. Hur et al. proposed a discrete-time H_∞ filter for indoor robot localization in WSNs based on IMU data and chirp spread spectrum (CSS) ranging [18]. This method improved the ranging precision accuracy to 1 m. Lee et al. proposed an indoor pedestrian localization system equipped both with inertial sensors and the IEEE 802.15.4a CSS radio [19]. Experimental results demonstrated a mean error of approximately 1.5 m with five position-unknown beacons. Aimed at monitoring groups of mobile nodes, Franco et al. proposed a data fusion method using information derived from an RSS indicator (RSSI) and an IMU [20]. The Kalman filter and multidimensional scaling were employed to estimate robot dynamics. The error distribution of the system was determined to demonstrate a mean error = 60.3 mm and a standard deviation = 56 mm. In another study, Tarro et al. proposed a framework based on RSS and IMU data for reducing energy consumption in localization systems [21]. For a given target accuracy, the proposed framework exhibited a lower energy consumption than conventional methods.

However, these approaches did not use unsupervised learning techniques. As such, their performance depended on prior information, which limited their applicability to their initial environments. Wang et al. proposed UnLoc, a self-learning method for indoor wireless localization [22]. By combining dead reckoning, urban sensing, and partitioning based on WiFi into a framework for unsupervised localization, they established landmarks from various sensors (e.g., accelerometer, magnetometer, sound, and ambient light) to substantiate the position estimate. However, the experiment did not employ time series data information, and this limited the validation because the transition probability is an important contribution for determining accuracy.

3. Problem statement

We consider a WSN deployed in an indoor environment with a robot equipped with a mobile node and a smartphone. Given this scenario, we seek to determine the robot's position based on RSS and IMU data. Specifically, a vector representative of the mobile node's signal strength received from M reference nodes, namely the fingerprint, is defined as (1).

$$S_i = \{s_{i,1}, s_{i,2}, \ldots, s_{i,j}, \ldots s_{i,M}\} \quad i = 1,2,3,\ldots N \quad and \quad j = 1,2,\ldots,M \tag{1}$$

Here, i denotes the ith signal strength sample in the time sequence, as received from the jth reference node, and N is the total number of samples. For convenience, we divide the indoor environment into grids. If an S_i is given, our goal is to locate the grid that matches the mobile node's position. In our unsupervised fingerprint method, a priori knowledge regarding the positions of the reference nodes is not essential. To save time and labor, in each of K grids we investigate only a single labeled fingerprint as an initial representative, which is defined as

$$F_k = \{f_{k,1}, f_{k,2}, \ldots, f_{k,j}, \ldots, f_{k,M}\} \quad k = 1,2,3,\ldots,K \quad and \quad j = 1,2,\ldots,M \tag{2}$$

where $f_{k,j}$ denotes the labeled fingerprint of the kth grid from the jth reference node.

To estimate the coordinates of the indoor wireless mobile sensor nodes, we also depend on measurements derived from the IMU, which have been widely and variously implemented in wireless networks. The limitations of the IMU in a smartphone are that the magnetic compass is influenced by nearby electronic devices, and the gyroscope is subject to the accumulation of drift error over time. Furthermore, performing double integrations on the accelerometer to evaluate position can cause significant errors in long-term estimations. One approach is relating IMU measurements to distance by extracting peaks that represent user steps [23,24].

In practice, it is difficult to calculate an accurate moving distance for a robot using only the IMU in a smartphone. As such, it is necessary to incorporate RSS, IMU, and time series data. However, an interesting question arises. Given an increasing number of unsupervised RSS and IMU samples, how do we select a model that best matches the actual observations? The problem we seek to address in this paper is the unsupervised improvement of localization accuracy. We have addressed this question with a learning and tracing algorithm based on unsupervised learning.

4. Proposed algorithm

4.1. Hierarchical Bayesian hidden Markov models

In the unsupervised learning algorithm, collecting a large amount of training data is not difficult because it is necessary to label every fingerprint. With decision stumps, all learning algorithms achieve higher accuracies with larger datasets. In the field of unsupervised learning, k-means clustering, self-organizing maps (SOM), and HMM are commonly used [25–27]. In this paper, we have proposed HB-HMM for the RSS time series analysis. HB-HMM has the following advantages. Unlike conventional k-means and SOM models, which only operate on the measurement model, HMM provides an extra motion model that contributes to accuracy. In addition, while HMM-based algorithms generally depend on a set of initial parameters for global optimization, HB-HMM depends only on a single hyper-parameter β, simplifying the global optimization problem of unsupervised learning. Lastly, HB-HMM works well with the PF method.

In our HB-HMM approach, the hidden states correspond to the grids in the indoor environment. The probability of a location is conditioned by the location that preceded it, and the probability of an RSS vector is conditioned by its location. The former is referred to as the transition probability, and the latter is known as the likelihood probability (also known as emission probability). More specifically, this is given by the following expressions:

$$\pi(k) = Pr(X_1 = k) \tag{3}$$

is the probability that a sequence begins at state k,

$$\varphi(k|l) = Pr(X_i = k|X_{i-1} = l) \quad 1 \leqslant i \leqslant N \tag{4}$$

is the transition probability from state l to state k, and

$$\upsilon(S_i|X_i) = Pr(S_i|X_i) \tag{5}$$

is the emission probability which represents how likely a mobile node in state X_i is to receive a certain fingerprinting. Here $\{X_i\}_{i=1\ldots N}$ is hidden state variable which draws value from 1 to K, and k, l are integers $(1 \leqslant k, l \leqslant K)$. We focus on estimating $\{X_i\}_{i=1\ldots N}$ from the observation sequence $\{S_i\}_{i=1\ldots N}$ where i is the time index.

This standard HMM model depend on sufficiently good initial conditions $(\pi(k), \varphi(k|l), \upsilon(S_i|X_i))$ to yield highly competitive results. For simplifying the global optimization problem of unsupervised learning, we propose the HB-HMM.

In HB-HMM, let $Q = (Q_1, \ldots, Q_l, \ldots Q_k, \ldots, Q_K)$, *where* $Q_l = \sum_{k=1}^{K} Q_k \varphi(l|k)$. The vector Q represents a stationary distribution. An irreducible Markov chain has a stationary distribution if and only if all of its states are positive recurrent [28]. The probability of the time series is then :

$$P(X_1, \ldots, X_N|Q) = \prod_{k=1}^{K} Q_k^{n_k}, \tag{6}$$

where n_k is the number of times that $X_i = k$. Because the hidden state variables follow a multinomial distribution in this case and the Dirichlet distribution is the conjugate prior of the multinomial distribution, the Dirichlet distribution is a natural choice for the prior [29]. We have assumed that the probabilities Q follow the Dirichlet distribution with a positive concentration hyper-parameter β [30]:

$$P(Q|\beta) \sim Dirichlet(\beta/K, \ldots, \beta/K) = \frac{\Gamma(\beta)}{\Gamma(\beta/K)^K} \prod_{k=1}^{K} Q_k^{\beta/K-1}, \tag{7}$$

where $\Gamma()$ is a gamma function. The hyperparameter β in the Dirichlet distribution can be viewed as pseudocounts, which represent the process of adding some observations for each state priorly. The value of β can bias the estimator toward more skewed distributions. When $\beta = 1$, the Dirichlet distribution is equivalent to a uniform distribution. When the value of beta above 1, all the states values in a sample are similar to each other. For β below 1, most of the values of states in a sample will be close to 0. The value of β also can avoid zero being assigned to the probabilities of unseen states, which causes trouble when multiplying probabilities, and can avoid a reduction in hidden states. Next, we integrate out the probabilities Q under the hyper-parameter β (see Appendix A).

$$\begin{aligned} P(X_1, \ldots, X_N|\beta) &= \int P(X_1, \ldots, X_N|Q) P(Q|\beta) dQ \\ &= \int \frac{\Gamma(\beta)}{\Gamma(\beta/K^K)} \prod_{k=1}^{K} Q_k^{n_k+\beta/K-1} dQ_1 \ldots dQ_K \\ &= \frac{\Gamma(\beta)}{\Gamma(n+\beta)} \prod_{k=1}^{K} \frac{\Gamma(n_k+\beta/K)}{\Gamma(\beta/K)} \end{aligned} \tag{8}$$

Given all the other state values are known, the probability that X_i takes value k is easily obtained (see Appendix B):

$$P(X_i = k|X_{-i}, \beta) = \frac{n_{-i,k} + \beta/K}{N - 1 + \beta} \tag{9}$$

where X_{-i} indicates all sequence variables except X_i and $n_{-i,k}$ is the number of observations, excluding X_i, that take value k. This conditional probability is useful for Gibbs sampling [31].

The likelihood function $P(S_i|X_i, X_{-i})$ is calculated using the Gaussian kernel [32]. It is denoted as kernel density estimation (KDE) which is a non-parametric method for estimating the probability density function of a random variable [33]. By KDE, we can approximate the probability density function (pdf) of $P(s_{i,j}|X_i)$ as

$$P(s_{i,j}|X_i = k) = \frac{1}{\sqrt{2\pi} n_k h} \cdot \sum_{l=1}^{N} exp\left(-\frac{(s_{i,j} - s_{l,j})^2}{2h^2}\right) \delta(X_l, k), \tag{10}$$

where $\delta(a, b)$ is the Kronecker-delta function (1 if $a = b$, and 0 otherwise), n_k is the number of times that $X_i = k$, h is bandwidth (The bandwidth h strongly influences the result. We defines it as 15 according to Silverman's rule of thumb [34]). If we viewed $s_{i,j}$ as a continuous variable, this function has a guarantee of integration to 1. If we assume $\overline{z_{k,j}}$ approximately equals $f_{k,j}$ which denotes the labeled fingerprint of the kth grid from the jth reference node, we

can approximate the probability density function of $P(s_{i,j}|X_i = k)$ based on the Gaussian kernel function as follows:

$$P(s_{i,j}|X_i = k) = \frac{1}{\sqrt{2\pi}n_k h} \cdot \sum_{l=1}^{N} exp\left(-\frac{(s_{i,j} - s_{l,j} + \overline{z_{k,j}} - f_{k,j})^2}{2h^2}\right)\delta(X_l, k),$$
$$where \quad \overline{z_{k,j}} = \frac{\sum_{l=1}^{N} s_{l,j}\delta(X_i, k)}{n_k}, \tag{11}$$

If $\overline{z_{k,j}}$ is not approximately equal to $f_{k,j}$ and $s_{l,j}\delta(X_l, k)$ approximately equals $\overline{z_{k,j}}$, then the Eq. (11) is approximately a density estimation for a Gauss distribution, the mean of which is $f_{k,j}$. Otherwise, the formulation can at least make the $s_{l,j}\delta(X_l, k)$ closer to $\overline{z_{k,j}}$ in clustering. Combined with Dirichlet prior, the function is effective in our case.

We use (11) in the initial phase of the training iteration. Considering the high computational complexity of (11), the remaining iterations employ the following expression:

$$P(s_{i,j}|X_i = k) = \frac{1}{\sqrt{2\pi}n_k h} exp\left(-\frac{(s_{i,j} - \overline{z_{k,j}})^2}{2h^2}\right) \tag{12}$$

We use (11) to get the function which describe the corresponding histograms. The Eq. (10) and (11) is different from (12). Using (12), we only get a coarse density estimation which use simple normal distribution. The histogram is more accurate.

We employ KDE to transfer some boundary data determined by k-means clustering to another category for greater clustering density. Use of this method is reasonable because different terrain have different robot trace densities. The Fig. 1 shows this effect.

Supposing that S_i is independent of X_{-i} or β, we obtained:

$$P(S_i|X_i, X_{-i}) = P(S_i|X_i = k) = \prod_{j=1}^{M} P(s_{i,j}|X_i = k) \tag{13}$$

Next, we can produce hidden state from the posterior distribution:

$$P(X_i|X_{-i}, S, \beta) = P(X_i|X_{-i}, S_i, \beta) = \frac{P(X_i|X_{-i}, \beta)P(S_i|X_i, X_{-i}, \beta)}{P(S_i|X_{-i}, \beta)}$$
$$\propto P(X_i|X_{-i}, \beta)P(S_i|X_i, X_{-i}) \tag{14}$$

For the inference problems of the hidden state sequence, we employ Gibbs sampling. The Eqs. (13) and (14) are calculated for Gibbs sampling. With Gibbs sampling the sample sequences eventually converge to the true joint posterior distribution $P(X_{1:N}, S_{1:N}|\beta)$ [31].

The Gibbs sampling produces state sequences X according to (14). We first randomly initialize a hidden state sequence, and then iteratively resample each hidden state according to (14). We use Gibbs sampling as a gradient search step for maximizing an evaluation function of the hidden state sequence. The evaluation function for the gradient search step is

$$e_t = P(S_{1:N}|X_{1:N}) = \prod_{i=1}^{N} P(S_i|X_i) \tag{15}$$

where the subscript t indicates the tth iteration. Using an evaluation function, we select a single sample which represents an optimal result in maximizing clustering density, and pushes the center of clustering to reside closer to fingerprinting data. Next, we build the transition probability of the tth iteration:

$$P(X_i = k|X_{i-1} = l, \beta) = \frac{n_{lk} + \beta/K}{\sum_{j=1}^{K} n_{lj} + \beta} \quad where \ n_{lj}$$
$$= \sum_{i=1}^{N-1} \delta(X_i, l)\delta(X_{i+1}, j) \tag{16}$$

Including pseudocounts, this simple maximum likelihood estimation (MLE) consists in computing observation relative frequencies. Because the training data is extensive, the transition probability is good for most events, and can filter noise and avoid unreasonable transitions. Theoretically, employing transition probability can improve the posterior Cramer-Rao lower bound (PCRLB) for a tracking algorithm [35]. Gibbs sampling is not suited for real time tracking, therefore, we use PF. In this PF, we use the transition probability to predict the next state, and use the likelihood probability to resample the particles.

4.2. Inertial measurement unit

When employing IMU in a smartphone, our main challenges are that the step detection cannot be used for a robot in an indoor environment, and that double-integration solutions cause estimation error growths with time due to the typical drift of the inertial

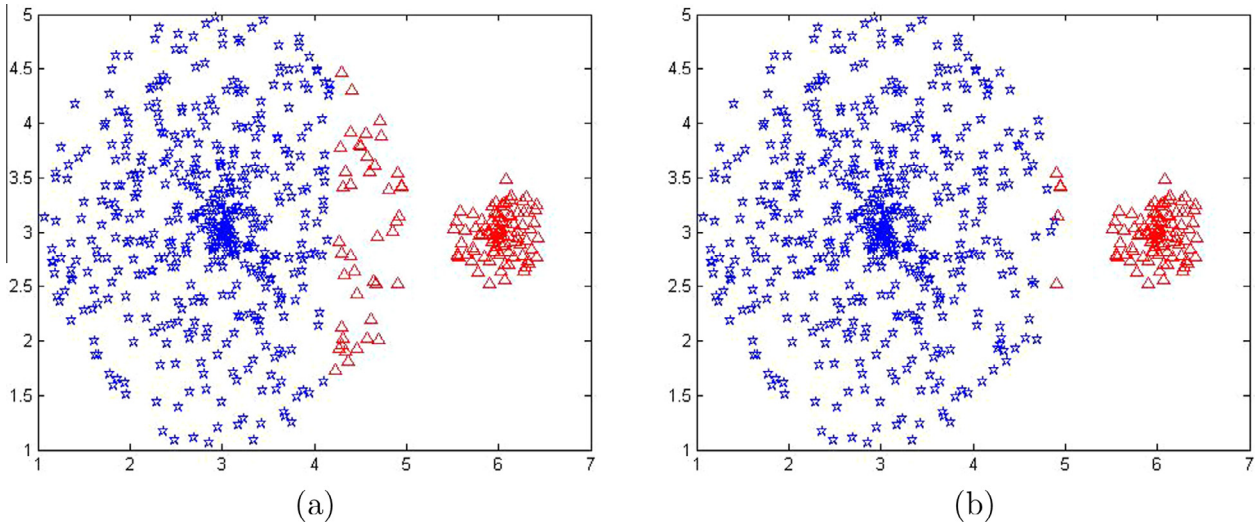

Fig. 1. The comparison of k-means clustering and KDE clustering. (The points and circles represent different clusters.) (a) The effect of K-means clustering on an artificial sample. (b) The effect of KDE clustering on the same sample.

measurements. For the localization problem based on RSS and IMU, we use the following techniques.

Based on the accelerometer, we first focused on detecting when the robot began moving. The accelerometers employed in the IMU measure robot acceleration including the force of gravity g. This measurement therefore requires isolation with a low-pass filter, whereupon g is then subtracted from the accelerometer readings as follows:

$$\hat{a} = a - g \tag{17}$$

where $\hat{a}$ is the linear acceleration and a is the accelerometer reading. Next, we check whether the Euclidian norm of the vector is greater than a threshold. If so, the point when the robot began moving is identified. We only determined when the robot begins to move, because, once the robot begins moving, it does not stop until it is turned off. Therefor we need not consider a stationary state. As shown in [17], the heading of a mobile robot at a time instant t' is computed using a combination of the magnetometer and gyroscope measurements as follows:

$$\hat{\theta}_{t'} = (1 - \hat{W})(\hat{\theta}_{t'-1} + \omega_{t'}T) + \hat{W}\gamma_{t'} \tag{18}$$

where $\gamma_{t'}$ is the heading estimation from the magnetometer , $\hat{W}$ is the weighting factor, $\hat{\theta}_{t'}$ is the heading estimation at time instant t', and $\omega_{t'}$ is the integration of the gyroscope signal at the t'th period, and T is sample period. Examining of the $\hat{\theta}_{t'}$ during a given period can also identify whether the robot is rotating. If it is determined to be rotating, we identify the robot in our case as being located at its last position for its standard activity. To exploit the position information, we detect bumps that occur occasionally over the period of robot motion. A bump which means velocity is zero causes vibration that can be sensed by the accelerometers according to the z-axis acceleration [36]. More specifically, a z-axis acceleration signal $\hat{X}$ is identified as a bump signal when the absolute difference between $\hat{X}$ and its Gaussian distribution mean μ is greater than the standard deviation σ multiplied by a predefined threshold $\hat{V}$:

$$|\hat{X} - \mu| \geqslant \hat{V}\sigma \tag{19}$$

Otherwise, $\hat{X}$ is regarded as a smooth-road signal, and the mean and variance are respectively updated as follows:

$$\lambda(\hat{X}) = \frac{1}{\sqrt{2\pi}\sigma} exp\left(-\frac{(\hat{X} - \mu)^2}{2\sigma^2}\right) \tag{20}$$

$$\mu = (1 - \lambda(\hat{X}))\mu + \lambda(\hat{X})\hat{X} \tag{21}$$

$$\sigma = \sqrt{(1 - \lambda(\hat{X}))\sigma^2 + \lambda(\hat{X})(\hat{X} - \mu)^2} \tag{22}$$

Here $\lambda(.)$ is the probability density function extracted from past samples. In fact, the method used here as a low-pass filter to calculate moving average. Bump state is different from moving state since moving state is on smooth-road. The expressions in (20)–(22) can also be seen in the paper [36]. The bump helps the localization. In our method, if the bump occurred the velocity of a robot was zero. So the bump heavily influenced the moving distance. Then, we can assume that the robot moves at an approximately constant velocity between the sampling period when the robot is not rotating or bumping. we used accelerometers only to adjust the velocity value as follows:

$$V_{t'} = (1 - \alpha)V_0 + (\alpha)(V_{t'-1} + \hat{a}T) \tag{23}$$

where V_0 is the prior knowledge regarding the robot's velocity, and α is a scale factor. We merely suggest that $0 \leqslant \alpha \leqslant 1$. Smaller values of α indicates that the results reside closer to V_0. Otherwise the results are affected by V_0 little. Finally, in this subsection, we propose another PF for tracing the robot and determining its coordinates. In this PF, we predict the robot's next state at a time instant $t' + 1$ using the following motion model:

$$\xi_{t'+1} = L_{t'}\xi_{t'} + \rho_{t'}$$

$$L_{t'} = \begin{bmatrix} 1 & 0 & T\cos(\hat{\theta}_{t'}) & 0 \\ 0 & 1 & T\sin(\hat{\theta}_{t'}) & 0 \\ 0 & 0 & (1-\alpha)V_0 + (\alpha)(V_{t'-1} + \hat{a}T) & 0 \\ 0 & 0 & 0 & (1-\hat{W})(\hat{\theta}_{t'-1} + \omega_{t'}T) + \hat{W}\gamma_{t'} \end{bmatrix} \tag{24}$$

Here $\xi_{t'} = [x_{t'}, y_{t'}, V_{t'}, \hat{\theta}_{t'}]^T$ is the transposed vector representing the state of the mobile node, and $L_{t'}$ models the position dynamics, $\rho_{t'}$ follows $N\{0, \hat{Q}_{t'}\}$, where $x_{t'}$ and $y_{t'}$ represent the robot position in Cartesian coordinates, $V_{t'}$ is its speed and $\hat{\theta}_{t'}$ is its heading at time-instant t'. According the first PF, we can obtain the number of particles in every grid $\hat{n}_k$. In the second PF, we resample the particles using the following weights :

$$W(\hat{i}) = \sum_{k=1}^{K} \frac{\hat{n}_k}{N'} \frac{1}{\sqrt{2\pi\hat{\sigma}^2}} exp\left(-\frac{(x_{t'}^{\hat{i}} - c_k)^2 + (y_{t'}^{\hat{i}} - d_k)^2}{2\hat{\sigma}^2}\right) \tag{25}$$

where $W(\hat{i})$ is the weight of the $\hat{i}$th particle, $x_{t'}^{\hat{i}}$ and $y_{t'}^{\hat{i}}$ are $\hat{i}$th particle's position in the t'th time instant; (c_k, d_k) are the center position of the kth grid in Cartesian coordinates, $\hat{n}_k$ is the number of particles in grid k that were calculated from the first particle filter, N' is the number of all particles in the first PF, and $\hat{\sigma}$ is bandwidth. The Eq. (25) is used by us to fuse RSS measurements with IMU measurements. The value of $\hat{n}_k$ derives from RSS measurements. The $x_{t'}^{\hat{i}}, y_{t'}^{\hat{i}}$ in the Eq. (25) derives from IMU measurements and the motion model (24). We fuse these information derived from these sources with the second PF in which the particles are given more weight when $\hat{n}_k$ is great and the current position is near (c_k, d_k).

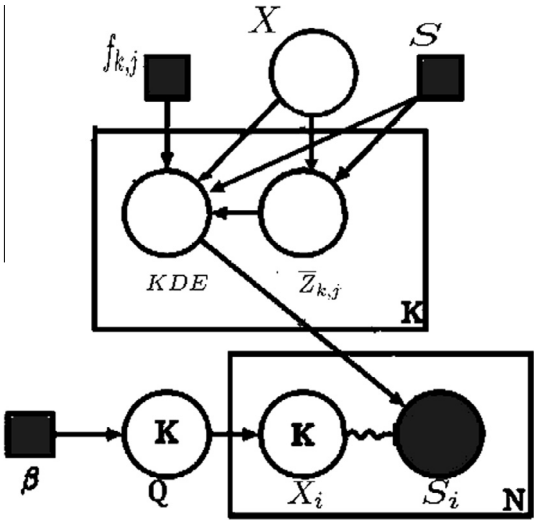

Fig. 2. The HB-HMM model. Smaller squares indicate fixed parameters; larger circles indicate random variables. Filled-in shapes indicate known values.

4.3. Proposed algorithm

Based on unsupervised learning, HPFRI is summarized in Algorithm 1. The Fig. 2 shows the structure of HB-HMM, and Fig. 3 shows the structure of training algorithm and tracking algorithm.

Overall, our model can be expressed as follows. Given a model

$$\begin{aligned} &f_{k,j} = as \quad above \\ &P(Q|\beta) \sim Dirichlet(\beta/K, \ldots, \beta/K) \\ &X_i \sim multinomial \quad distribution. \\ &S_i \sim the\ distribution\ calculated\ by\ (11). \end{aligned} \tag{26}$$

The model can be seen in Fig. 2. In our TRAINING phase, the Eq. (11) is employed to approximate the probability density function of S in every grid. Then the evaluation function in the Eq. (15) is used here to maximize clustering density of S. Because with Gibbs sampling method which uses the Eq. (14) the sample sequences eventually converge to the true joint posterior distribution $P(X_{1:N}, S_{1:N}|\beta)$, we used the Gibbs sampling as a gradient search step to find an optimal result to maximize e_t in the Eq. (15). Lastly, from the result we calculate the transition probability according to the Eq. (16) and calculate the likelihood probability according to the Eq. (11). In the TRACKING phase, we used the first PF to inference X_i in realtime. The particle filter algorithm used in this paper is sequential importance resampling (SIR) [37] which estimates X_i from the transition probability given by the Eq. (16) and resamples by the Eq. (11). Next, we identified the robot's state by IMU. Finally, we used these results within the second PF which estimates the position by the Eq. (24) and resamples by the Eq. (25). The proposed TRAINING algorithm and TRACKING algorithm can be seen in Fig. 3.

In wireless localization system, multipath propagation may lead to ranging outliers. Our kernel-based methods estimate the density distribution of the position space and identify outliers as lying in regions of low density. In our system, outliers have low likelihood probability. However, outliers caused by malfunctioning anchors is quiet different. In [38], increasing maximum communication range of anchors is suggested to mitigate the impact of malfunctioning anchors.

Algorithm 1 HPFRI: HB-HMM and particle filter for the fusion of RSS and inertial measurements

Input: collected RSS database ,hyper-parameter β, real time RSS and IMU measurements;
Output: Coordinates;
if TRAINING **then**
  Instantiate a random hidden state sequence;
  repeat
    Compute count matrices of $n_k, n_{-i,k}$;
    Gibbs sampling by applying (14);
    Compute the Eq. (11) or (12);
    Compute evaluation function e_t by applying the Eq. (15);
    if $e_t \geqslant e_{t-1}$ **then**
      The acceptance probability = 1.
    else
      The acceptance probability = $exp(-4(e_t - e_{t-1})/e_t)$.
    end if
    if any $n_k \leqslant \frac{N}{10K}$ **then**
      The acceptance probability=0.
    end if
    until convergence or iteration limit reached
    Compute the likelihood probability by applying (11)
    Compute the transition probability by applying (16)
end if
if TRACKING **then**
  Estimate the hidden state by Particle Filter based on the Eq. (11) and (16).
  Estimate the Robot state by IMU measurements.
  Estimate the coordinates of the robot by another Particle Filter based on the Eq. (24) and (25).
end if
return Coordinates;

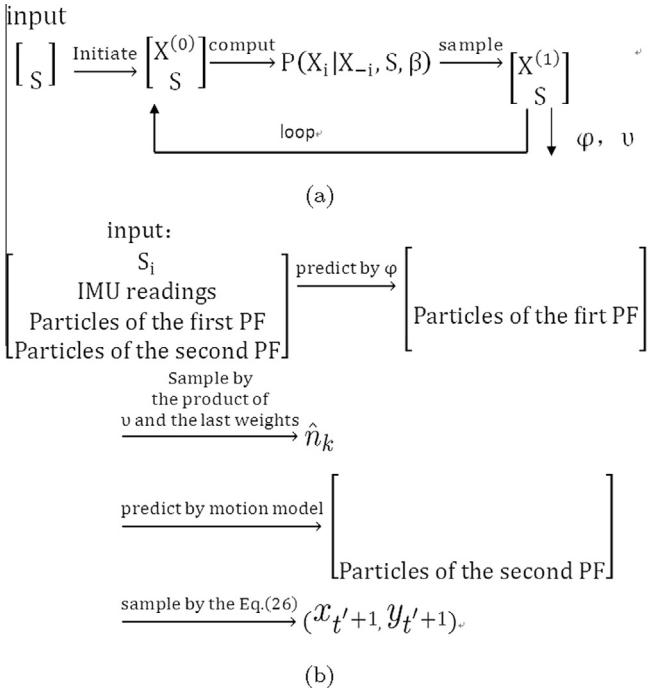

Fig. 3. The proposed training and tracking algorithm. (a) The structure of the proposed training algorithm. (b) The structure of the proposed tracking algorithm.

5. Evaluation

We evaluated of the performance of the proposed unsupervised learning localization algorithm through a series of experiments in an actual indoor environment comprised of a single room and a single mobile robot. As shown in Fig. 4, the room is a rectangular room 4.5 m × 7 m in size. The room is divided into six grids, which are marked by encircled numbers in the figure. A table and three sofas are placed in the center of the room. Six reference nodes were positioned in the room, as marked on the map by small circles.

The experimental system was built upon the IEEE 802.15.4 wireless network. We employed 8 nodes, which included 6 reference nodes, one blind node, and one gateway. The reference nodes and WLAN interface were equipped with an IEEE 802.15.4-compliant 2.4 GHz Chipcon CC2430 radio. The blind node employed a Chipcon CC2431 radio. The robot was a automatic vacuum cleaner engaged in its standard activity, and the smartphone was a Huawei Honor 6. At 1s intervals, the mobile node received six RSS values from the six reference nodes. The time series data was collected for approximately seventy minutes, and twenty minutes of this data was employed for real time tracking.

The results presented in Fig. 5 show the estimated location distribution over a single trial, which displayed the expected Gaussian distribution. In our experiment in NLOS environment, the best

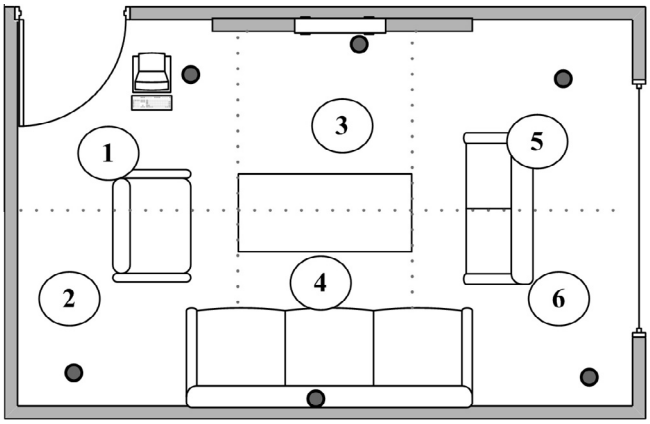

Fig. 4. The room plan employed in the experiments, comprised of six grids given by encircled numbers and six reference nodes marked by small circles.

Fig. 5. The estimated location distribution over a single trial, which follows the expected Gaussian distribution.

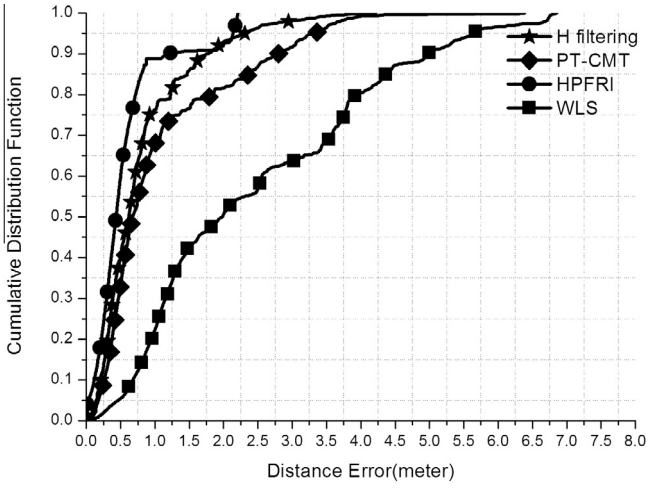

Fig. 6. The cumulative distribution function (CDF) comparison of different algorithms.

accuracy rate of our localization algorithm was 90.1% within 1.7 m. For $600 \leqslant \beta \leqslant 3000$, the accuracy rate varied from 80.5% to 90.1%. Meanwhile, for $0.1 \leqslant \beta \leqslant 600$, the accuracy rate varied from 40.5%

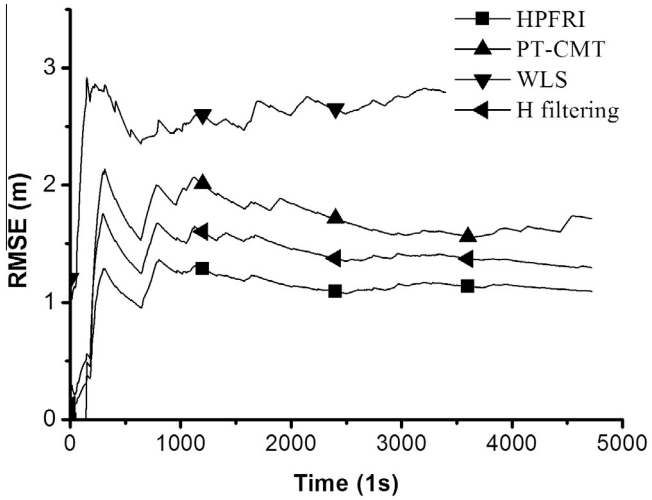

Fig. 7. The root mean squared error (RMSE) comparison of different algorithms.

to 70.1%. Therefore, we propose that $600 \leqslant \beta \leqslant 3000$. The cumulative distribution function (CDF) was used to analyze the distribution of location errors. When comparing these results with those obtained for the WLS algorithm [13], the PT-CMT algorithm [17], and the H_∞ filtering algorithm [18], the HPFRI algorithm achieved at least a 20% improvement on the average error distance. The Fig. 6 shows the CDFs of the location errors for the algorithms tested. Based on the root mean squared error (RMSE) results shown in Fig. 7, the HPFRI algorithm exhibited an RMSE of 1.15 m, PT-CMT was 1.5 m (RMSE), WLS was 3 m (RMSE), and H_∞ was 1.3 m (RMSE). Therefore, the test results indicate that the proposed HPFRI algorithm is the most accurate among these algorithms.

6. Conclusion

To address the challenging problem of indoor localization in WSNs, we proposed an unsupervised algorithm based on the fusion of RSS and IMU measurements. We improved the RSS-based algorithm with the help of IMU measurements that employed a novel method to identify the robot's state. First, unlike conventional fingerprint algorithms, our unsupervised algorithm did not require the typical a priori information, and was thus achieved at low cost. Moreover, we used HB-HMM to simplify the problem of global optimization. Extensive experiments were conducted to evaluate and validate our proposed algorithm. Our experimental results showed that the proposed algorithm demonstrated an improved accuracy relative to the WLS, PT-CMT, and H_∞ filtering algorithms.

Acknowledgements

This work is supported by the National Natural Science Foundation of China under Grant Nos. 61272151, 61073037 and 61309025, the Hunan Provincial Natural Science Foundation of China under Grant No. 13JJ4016, International Science and Technology Cooperation Program of China under Grant No. 2013DFB10070 and the National Key Technology R&D Program under Grant No. 2015BAH05F02.

Appendix A. Proof of the Eq. (8)

we can integrate out the probabilities Q under the hyperparameter β. To do so, we makes a change of variables to rewrite the integral in a more "comfortable" region. we adapted the formulae to the spherical coordinates:

$$
\begin{aligned}
&Q_1 = r\cos^2(\phi_1)\\
&Q_2 = r\sin^2(\phi_1)\cos^2(\phi_2)\\
&\dots\\
&Q_{K-3} = r\sin^2(\phi_1)\dots\sin^2(\phi_{K-2}\cos^2(\phi_{K-2}))\\
&Q_{K-2} = r\sin^2(\phi_1)\dots\sin^2(\phi_{K-2}\sin^2(\phi_{K-2}))\\
&\textbf{Jacobian} = (2r)^{K-2}\sin^{2(K-2)-1}(\phi_1)\\
&\cos(\phi_1)\sin^{2(K-3)-1}(\phi_2)\cos(\phi_2)\dots\\
&\sin(\phi_{K-2})\cos(\phi_{K-2})
\end{aligned}
\tag{A.1}
$$

The integral can be described in simpler gamma function.

$$
\begin{aligned}
&\int P(X_1,\dots,X_n|Q)P(Q|\beta)dQ\\
&=\int_0^{\pi/2}\int_0^{\pi/2}\dots\int_0^{1}\sin^{2(t_2+\dots+t_{K-2})+2(K-2)-1}(\phi_1)\cos^{2t_1+1}(\phi_1)\\
&\quad\times\sin^{2(t_3+\dots+t_{K-2})+2(K-3)-1}(\phi_2)\cos^{2t_2+1}(\phi_2)\dots\sin^{2t_{K-2}+1}(\phi_{K-2})\\
&\quad\times\cos^{2t_{K-2}+1}(\phi_{K-2})r^{t_1+\dots+t_{K-1}}(1-r)^{t_K}drd\phi_1\dots d\phi_{K-2}\\
&=\frac{\Gamma(\beta)}{\Gamma(n+\beta)}\prod_{i=1}^{K}\frac{\Gamma(n_i+\beta/K)}{\Gamma(\beta/K)}\quad(t_i=n_i+\beta/k-1)
\end{aligned}
\tag{A.2}
$$

Thus,

$$
\begin{aligned}
P(X_1,\dots,X_n|\beta) &= \int P(Y_1,\dots,Y_n|Q)P(Q|\beta)dQ\\
&= \int\frac{\Gamma(\beta)}{\Gamma(\beta/K^K)}\prod_{i=1}^{K}Q_i^{n_i+\beta/K-1}dQ_1\dots dQ_K\\
&= \frac{\Gamma(\beta)}{\Gamma(n+\beta)}\prod_{i=1}^{K}\frac{\Gamma(n_i+\beta/k)}{\Gamma(\beta/k)}
\end{aligned}
\tag{A.3}
$$

where $\Gamma()$ is a gamma function.

Appendix B. Proof of the conditional probability for Gibbs sampling

We note the following.

$$
n_k=\begin{cases} n_{-i,k} & if \quad i\neq k\\ n_{-i,k}+1 & if \quad i=k\end{cases}
\tag{B.1}
$$

$$
\begin{aligned}
P(X_i=k|X_{-i},\beta)\propto P(X_i=k,X_{-i}|\beta) &= \frac{\Gamma(\beta)}{\Gamma(n+\beta)}\prod_{j=1}^{K}\frac{\Gamma(n_j+\beta/K)}{\Gamma(\beta/K)}\\
&\propto\prod_{j=1}^{K}\Gamma(n_j+\beta/K)=\Gamma(n_k+\beta/K)\prod_{j\neq k}\Gamma(n_j+\beta/K)\\
&=\Gamma(n_{-i,k}+1+\beta/K)\prod_{j\neq k}\Gamma(n_{-i,j}+\beta/K)\\
&=(n_{-i,k}+\beta/K)\prod_{j=1}^{K}\Gamma(n_{-i,j}+\beta/K)\propto(n_{-i,k}+\beta/K)
\end{aligned}
\tag{B.2}
$$

References

[1] Cui H, Wang Y. Four-mobile-beacon assisted localization in three-dimensional wireless sensor networks. Comput Electr Eng 2012;38:652–61. http://dx.doi.org/10.1016/j.compeleceng.2011.10.012.

[2] Wang G, Bhuiyan M, Alam Z, Zhang L. Two-level cooperative and energy-efficient tracking algorithm in wireless sensor networks. Concurr Comput Pract Exp 2010;22(4):518–37.

[3] Bouaziz M, Rachedi A. A survey on mobility management protocols in wireless sensor networks based on 6lowpan technology. Comput Commun 2014. http://dx.doi.org/10.1016/j.comcom.2014.10.004.

[4] Liu H, Darabi H, Banerjee P, Liu J. Survey of wireless indoor positioning techniques and systems. IEEE Trans Syst Man Cybern Part C Appl Rev 2007;37(6):1067–80. http://dx.doi.org/10.1109/TSMCC.2007.905750.

[5] Miah MS, Gueaieb W. Mobile robot trajectory tracking using noisy RSS measurements: an RFID approach. ISA Trans 2014;53(2):433–43. http://dx.doi.org/10.1016/j.isatra.2013.09.016.

[6] Harle R. A survey of indoor inertial positioning systems for pedestrians. IEEE Commun Surv Tutor 2013;15:1281–93. http://dx.doi.org/10.1109/SURV.2012.121912.00075.

[7] Subbu K, Zhang C, Luo J, Vasilakos A. Analysis and status quo of smartphone-based indoor localization systems. IEEE Wireless Commun 2014;21:106–12. http://dx.doi.org/10.1109/MWC.2014.6882302.

[8] Thomas F, Ros L. Revisiting trilateration for robot localization. IEEE Trans Rob 2005;21:93–101. http://dx.doi.org/10.1109/TRO.2004.833793.

[9] Blumrosen G, Hod B, Anker T, Dolev D, Rubinsky B. Continuous close-proximity RSSI-based tracking in wireless sensor networks. In: 2010 International conference on body sensor networks (BSN). p. 234–9. http://dx.doi.org/10.1109/BSN.2010.36.

[10] Kang W, Han Y. Smartpdr: smartphone-based pedestrian dead reckoning for indoor localization. IEEE Sens J 2015;15(5):2906–16. http://dx.doi.org/10.1109/JSEN.2014.2382568.

[11] Dhital A, Closas P, Fernandez-Prades C. Bayesian filters for indoor localization using wireless sensor networks. In: 2010 5th ESA workshop on satellite navigation technologies and European workshop on GNSS signals and signal processing (NAVITEC). p. 1–7. http://dx.doi.org/10.1109/NAVITEC.2010.5707982.

[12] Zheng J, Bhuiyan MZA, Liang S, Xing X, Wang G. Auction-based adaptive sensor activation algorithm for target tracking in wireless sensor networks. Future Gener Comput Syst 2014;39:88–99.

[13] Tarrio P, Bernardos AM, Casar JR. Weighted least squares techniques for improved received signal strength based localization. Sensors 2011;11(9):8569. http://dx.doi.org/10.3390/s110908569.

[14] Ouyang R, Wong A-S, Lea C-T. Received signal strength-based wireless localization via semidefinite programming: Noncooperative and cooperative schemes. IEEE Trans Veh Technol 2010;59(3):1307–18. http://dx.doi.org/10.1109/TVT.2010.2040096.

[15] Shang Y, Ruml W. Improved MDS-based localization. INFOCOM 2004. Twenty-third annual joint conference of the IEEE computer and communications societies, vol. 4. p. 2640–51. http://dx.doi.org/10.1109/INFCOM.2004.1354683.

[16] D'souza M, Wark T, Karunanithi M, Ros M. Evaluation of realtime people tracking for indoor environments using ubiquitous motion sensors and limited wireless network infrastructure. Pervasive Mob Comput 2013;9(4):498–515. http://dx.doi.org/10.1016/j.pmcj.2012.03.007.

[17] Correa A, Barcelo M, Morell A, Lopez Vicario J. Enhanced inertial-aided indoor tracking system for wireless sensor networks: A review. IEEE Sens J 2014;14(9):2921–9. http://dx.doi.org/10.1109/JSEN.2014.2325775.

[18] Hur H, Ahn H-S. Discrete-time h filtering for mobile robot localization using wireless sensor network. IEEE Sens J 2013;13(1):245–52. http://dx.doi.org/10.1109/JSEN.2012.2213337.

[19] Lee S, Kim B, Kim H, Ha R, Cha H. Inertial sensor-based indoor pedestrian localization with minimum 802.15.4a configuration. IEEE Trans Ind Inf 2011;7(3):455–66. http://dx.doi.org/10.1109/TII.2011.2158832.

[20] Di Franco C, Franchino G, Marinoni M. Data fusion for relative localization of wireless mobile nodes. In: 2014 9th IEEE international symposium on industrial embedded systems (SIES). p. 58–65. http://dx.doi.org/10.1109/SIES.2014.6871187.

[21] Tarro P, Cesana M, Redondi A. Energy-accuracy trade-offs for hybrid localization using RSS and inertial measurements in wireless sensor networks. Ad Hoc Netw 2013;11(6):1874–89. http://dx.doi.org/10.1016/j.adhoc.2013.04.012.

[22] Wang H, Sen S, Elgohary A, Farid M, Youssef M, Choudhury RR. No need to war-drive: unsupervised indoor localization. In: Proceedings of the 10th international conference on mobile systems, applications, and services. ACM; 2012. p. 197–210. http://dx.doi.org/10.1145/2307636.2307655.

[23] Ngo TT, Makihara Y, Nagahara H, Mukaigawa Y, Yagi Y. Similar gait action recognition using an inertial sensor. Pattern Recognit 2015;48(4):1289–301. http://dx.doi.org/10.1016/j.patcog.2014.10.012.

[24] Harle R. A survey of indoor inertial positioning systems for pedestrians. IEEE Commun Surv Tutor 2013;15(3):1281–93. http://dx.doi.org/10.1109/SURV.2012.121912.00075.

[25] Macqueen J. Some methods for classification and analysis of multivariate observations. In: 5-th Berkeley symposium on mathematical statistics and probability. p. 281–97.

[26] Kohonen T, Honkela T. Kohonen network. Scholarpedia 2007;2(1):1568.

[27] Rabiner LR. A tutorial on hidden markov models and selected applications in speech recognition. Proc IEEE 1989;77(2):257–86.

[28] Serfozo R. Basics of applied stochastic processes, probability and its applications. Berlin: Springer-Verlag; 2009.

[29] Beal MJ, Ghahramani Z, Rasmussen CE. The infinite hidden markov model. In: Machine learning. MIT Press; 2002. p. 29–245.

[30] Antoniak CE. Mixtures of Dirichlet processes with applications to bayesian nonparametric problems. Ann Stat 1974:1152–74. <http://www.jstor.org/stable/2958336>.

[31] Geman S, Geman D. Stochastic relaxation, gibbs distributions, and the Bayesian restoration of images PAMI-6 1984;6:721–41.

[32] Prieto J, Mazuelas S, Bahillo A, Fernandez P, Lorenzo R, Abril E. Adaptive data fusion for wireless localization in harsh environments. 2012;60(4):1585–96. doi:http://dx.doi.org/10.1109/TSP.2012.2183126.

[33] Kushki A, Plataniotis KN, Venetsanopoulos AN. Kernel-based positioning in wireless local area networks. IEEE Trans Mob Comput 2007;6(6):689–705. http://dx.doi.org/10.1109/TMC.2007.1017.
[34] Silverman B. Density estimation for statistics and data analysis. Biometrical J 1986.
[35] Tharmarasa R, Kirubarajan T, Hernandez M, Sinha A. Pcrlb-based multisensor array management for multitarget tracking. IEEE Trans Aerosp Electron Syst 2007;43(2):539–55.
[36] Tan G, Lu M, Jiang F, Chen K, Huang X, Wu J. Bumping: a bump-aided inertial navigation method for indoor vehicles using smartphones. IEEE Trans Parallel Distri Syst 2014;25(7):1670–80. http://dx.doi.org/10.1109/TPDS.2013.194.
[37] Doucet A, Johansen AM. A tutorial on particle filtering and smoothing: Fifteen years later. Handbook of Nonlinear Filtering, 12, 2009. p. 656–704.
[38] Wu N, Xiong Y, Wang H, Kuang J. A performance limit of TOA-based location-aware wireless networks with ranging outliers. IEEE Commun Lett 2015;19(8). 1-1.