A Graphical Model Approach for Efficient Geomagnetism-Pedometer Indoor Localization

Hang Wu Suining He S.-H. Gary Chan
The Hong Kong University of Science and Technology
Department of Computer Science and Engineering, Hong Kong, China
{hwuav,sheaa,gchan}@cse.ust.hk

***Abstract*—Geomagnetism is promising for indoor localization due to its pervasive presence, high signal stability and unnecessity of extra infrastructure support. It can be fused with a pedometer (step counter) to further improve accuracy. Traditional fusions based on particle filter, however, are computationally complex. Their accuracy is barely satisfactory for large open indoor spaces (such as airports or malls), mainly because particles cannot converge properly in environments with high degree of freedom. They often assume knowledge or explicit user step length input, which is inconvenient or difficult to achieve in reality.**

We propose Mapel, which efficiently fuses geomagnetism with pedometer based on graphical model without manual user input. Mapel is applicable to any indoor environment. We first discretize indoor map into closely spaced lattice nodes (separated by, say, $0.5 \sim 1$ meter) in a graph, representing potential target locations. Using conditional random field (CRF), Mapel then conducts geomagnetism-pedometer fusion to estimate user location in the lattice. By jointly considering geomagnetism pattern and step counts, Mapel adaptively self-calibrates a step model for each user without explicit input. Extensive experiments conducted at our campus show that Mapel outperforms state-of-the-art schemes by a wide margin (cutting localization error by more than 40% and reducing computation time by more than 30%).

I. Introduction

The penetration of smart devices and advances in their sensing capabilities have enabled indoor localization. The signals studied include Wi-Fi RSSI, channel state information (CSI), visible light and ultra-sound. While impressive, these systems require costly or special infrastructure installation. Furthermore, due to multipath or interference, the RF signals are often not stable, which adversely affects localization accuracy. The survey processes of these systems are also rather time-consuming, where signal vectors have to be collected by standing at length at different fixed points in the area.

Using geomagnetic field for localization has recently emerged as a promising approach, since magnetic field is omnipresent and temporally stable. Its spatial variation with respect to the indoor locations (introduced by electrical appliances and building materials) can be easily used to differentiate them. Without the need for extra infrastructure support, it is also more cost-effective to deploy. Furthermore, its collection is efficient, because magnetic field data can be sampled at a high rate *while* walking. Meanwhile, indoor layout changes or metallic mobile objects only affect the field pattern within a limited scope [1]–[5]. Consequently, magnetic field exhibits much less fluctuation than RF signals.

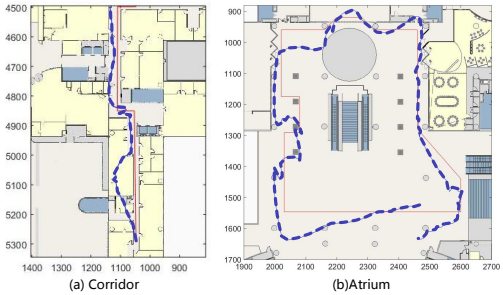

Fig. 1: Estimated trajectory of particle-filter-based magnetic localization in the corridor and campus atrium. The red solid line is the ground-truth trajectory and the blue dotted line is the estimated trajectory.

Due to advances of inertial sensing on step counts, smartphone orientation and user behavior, the pedometer (step counter) has been increasingly used for indoor localization [6], [7]. It can be fused with geomagnetism to improve localization accuracy. Prior works often employ recursive Bayesian filtering techniques (e.g., [4], [8]–[11]), which estimate user location by maximizing some joint probability or particle convergence given the magnetic field reading and walking distance of the target. Despite promising results, they are often computationally complex for a mobile device. They also assume the knowledge or explicit input of a user's initial position and step length, which is often inconvenient or difficult to achieve in practice.

Furthermore, these models work best in partitioned indoor spaces with narrow corridors, where the degree of freedom for the target (mobile client) is constrained. Figure 1 shows a typical example of a particle-filter-based magnetic localization performance in our university campus. Clearly, given corridor constraints in Figure 1(a), the estimated locations match the ground-truth well. However, as shown in Figure 1(b), for large spacious setting such as airports or shopping malls, their accuracy is barely satisfactory, mainly because the models (such as particle filter) cannot converge properly in environments with high degree of freedom.

To address the above drawbacks, we propose a novel graphical model fusing geomagnetism and pedometer for indoor localization. The scheme, termed Mapel (joint geo***ma***gnetism and ***pe***dometer ***l***ocalization), is applicable to any indoor environment without the need for infrastructure support, and is especially effective for open space settings. It self-calibrates the stride size (step length) of the target, and hence can adapt

2155-6814/17 $31.00 © 2017 IEEE
DOI 10.1109/MASS.2017.11

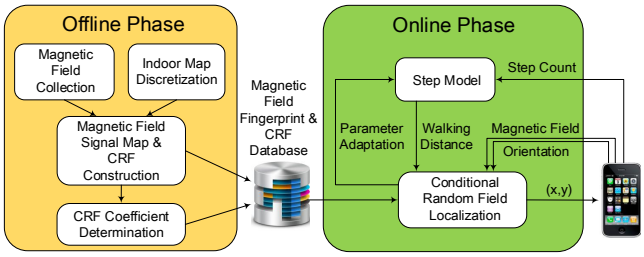

Fig. 2: System framework of Mapel.

to heterogeneous users without any explicit user input.

Mapel works as follows. We discretize the whole area into closely spaced nodes (separated by, say, 0.5 to 1 meter), which form a *lattice*. Each node of the lattice corresponds to a possible position of the target. Based on the joint information of step counter and the pattern of the collected magnetic field, target motion can be represented by neighboring transitions between nodes in the lattice. The transition probability and walking distance of the user can then be *jointly* estimated by applying conditional random field (CRF), with the target localized by maximizing the overall likelihood.

We show the system framework of Mapel in Figure 2. It basically consists of two phases, the offline phase (or the CRF calibration phase) and the online phase (or the localization phase). In the offline phase, a surveyor carries a smartphone and collects magnetic field data while walking on pre-defined paths. On the walking trajectory, the step counter is used to label the locations of geomagnetic field signals. Given the map information, we then perform indoor map discretization to construct the CRF structure (i.e., the lattice and feature functions). We then conduct offline training to determine the coefficients for the CRF feature functions. The signal map and the calibrated CRF (including structure, feature functions and their coefficients) are then stored in the magnetic field fingerprint and CRF database. In the online phase, a user measures the geo-magnetic field while his step counter returns the step count and walking orientation. The information is fed to the conditional random field localization module. Mapel works with a general step model, and then adaptively self-calibrates the physical walking parameters in the step model for a personalized model. It then returns the location based on the maximum likelihood.

The major contributions of this work are as follows:

- *A novel graphical model to fuse geomagnetic field and pedometer for joint calibration-free indoor localization*: We propose Mapel, a novel and efficient technique which fuses geomagnetic field and pedometer based on CRF. Mapel requires neither accurate or explicit input of user initial positions and step length, nor any offline device calibration, hence is more practical. Computationally, it is more efficient.
- *A self-calibrating algorithm to automatically estimate step length*: Mapel employs a novel self-calibrating algorithm that returns the walking distance and step length (stride size) of the user. It starts with a generic step model, and collects user data on the fly to train a personalized model. Specifically, it matches the magnetic field variation along the walking trajectory of the user with the stored magnetic fingerprint through derivative dynamic time warping [12] (DDTW) to estimate the walking distance of the user. As such, the distance has to be consistent with the step model parameters, which are obtained by regression.
- *Extensive experimental studies in various scenarios*: We have conducted extensive experiments on Mapel at our university campus, including an spacious atrium and constrained corridors. Our results validate its implementability, and show that it outperforms state-of-the-art algorithms by a large margin (cutting the location error by more than 40% and reducing the computation time by more than 30%).

This paper is organized as follows. We discuss related work in Section II. In Section III we show the offline phase of our system, including fingerprint database construction, map discretization, conditional random field (CRF) formulation, and CRF coefficient determination. We present location inference and step length self-calibration using CRF in Section IV. In Section V we discuss illustrative experimental results in our campus. We conclude in Section VI.

II. Related Work

Traditional fingerprint-based indoor localization has been extensively studied in the past decade. RADAR [13] and Horus [14] are among the earlier works for Wi-Fi fingerprinting, whose accuracy depends on the density of access points. The work in [11] fuses Wi-Fi RSSI signals with pedometers using particle filter. Other signals like channel state information (CSI), Bluetooth (or iBeacon) or FM rely on deploying specialized infrastructures, which may not be feasible for pervasive localization. Our scheme, Mapel, does not require infrastructure installation, and hence is more cost-effective to deploy with localization error less dependent on the infrastructure of the environment.

The fusion of magnetic field fingerprint signals and pedometers has been recently studied. MapCraft [15] uses the conditional random field (CRF) to localize users. In their scheme, sequential motion and sensor readings are fed to a graphical model for location estimation. However, MapCraft follows the corridors in a segment-like graph and has not considered the freedom of user movement in large open space indoor settings. Furthermore, its feature functions only consider fusing Wi-Fi RSSI signals with pedometers, burying the potential advantages of geomagnetism. Mapel addresses these issues by discretizing the indoor map into a lattice, with each lattice node representing a potential target location. Traditional CRF [16] is then adapted for the lattice instead of a segment-like graph, and dedicated feature functions are designed to leverage the geomagnetism. Infrastructure-based magnetic field localization has been studied before [3]. This early study focuses on using robots or special devices to collect magnetic fields. Different from this work, Mapel focuses on mobile localization based on smartphone sensors.

Fusion based on particle filter has been widely studied for magnetic field localization [17]. MaLoc [4] implements

a novel augmented particle filter to address motion estimation error. Magicol [10] considers a two-way particle filter to improve the fusion of Wi-Fi fingerprint and magnetic field. The work in [11] approximates the floor map by connected line segments. These works do not work well in open indoor environments. Mapel uses conditional random field which readily applies to such environments. It preprocesses the indoor map into lattice, and reduces the degree of freedom for target estimations. Besides, it does not need to know the initial user position. Though in this paper we focus on fusing pedometer and magnetism for mobile localization, Mapel is also amendable to other emerging signals like vision or visible light to improve their accuracy.

There have been other works on magnetic field localization. UnLoc in [18] proposes applying sparse magnetic field disturbance as the landmark for indoor localization. Different from their work, we measure the magnetic field to build an indoor signal map. Inspired by the observation that geomagnetism changes around the pillars or gates, LocateMe [19] maps the target location to these landmarks when a similar pattern of signal change is measured. FOLLOWME [1] leverages the walking patterns of earlier travelers to navigate the following users. These schemes work the best in narrow corridors where pedestrian walking patterns are constrained. Mapel is more general to apply in complex environments.

Step length adaptation has been recently studied. The work in [15] assumes different people may share similar step frequency coefficients. Different from it, Mapel dynamically adapts a personalized step model for different users, which helps improve the localization accuracy. The work in [20] assumes homogeneous users and has not considered the variation in step lengths among people. The work in [21] estimates or trains a personalized step model offline, which is not cost-effective or convenient in practice. As opposed to the above, Mapel can adapt to different users on the fly to achieve better accuracy. The works in [22] and [6] use particle filter to estimate the step model for each individual. Different from these works, Mapel uses magnetic field pattern in a trajectory and step count to self-calibrate the step length model.

III. Offline Phase

Table I summarizes the symbols used in our paper. In this section, we briefly discuss the offline phase of Mapel, including geomagnetic database construction and our CRF structure.

In Section III-A we describe how to construct magnetic signals fingerprint database. In Section III-B, we present how to discretize the map into lattice. Based on the discretized map, we discuss in Section III-C the formulation of conditional random field (CRF) for geomagnetism localization. We discuss the feature functions used in our formulation in Section III-D, followed by CRF coefficients determination in Section III-E.

A. Magnetic Field Signal Map Construction

We first present how we measure the magnetic field using a smartphone. The magnetic field vector $\mathbf{B}_p$ can be measured by

TABLE I: Major symbols used in Mapel.

Notations	Definitions
$\mathbf{s}_{i,t} = [a_i, b_i]$	Node of state i at time t
E	Set of edges formed in discretization
$\vec{\mathbf{x}} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$	N temporal inputs into CRF graphical model
$\vec{\mathbf{y}} = (\mathbf{y}_1, \ldots, \mathbf{y}_N)$	Possible N outputs from CRF graphical model
$\mathbf{B}_p$	Geomagnetism readings under phone coordinate system
$\mathbf{B}_o$	Geomagnetism observations under earth coordinate system
$\Psi_{j,t}(\vec{\mathbf{x}}, \vec{\mathbf{y}})$	Potential function of state j of output $\vec{\mathbf{y}}$ at t given inputs $\vec{\mathbf{x}}$
$\Psi_{ij,t}(\vec{\mathbf{x}}, \vec{\mathbf{y}})$	Potential function of states i and j of output $\vec{\mathbf{y}}$ at time t with inputs $\vec{\mathbf{x}}$
$f_1(\mathbf{x}^B_{t-1}, \mathbf{x}^B_t, \mathbf{y}_{i,t-1}, \mathbf{y}_{j,t})$	Feature function for magnetic field intensity
$f_2(\mathbf{x}^\theta_t, \mathbf{y}_{i,t-1}, \mathbf{y}_{j,t})$	Feature function for orientation
$f_3(\mathbf{x}^l_t, \mathbf{y}_{i,t-1}, \mathbf{y}_{j,t})$	Feature function for step length
$\mathbf{S}$	Set of states
N	Number of nodes in a single layer

smartphone's magnetometer [2], but the raw magnetic readings are under the smartphone's coordinate system. We transform the readings into the earth coordinate system by the yaw ψ, the pitch θ and the roll ϕ of the smartphone, i.e.,

$$\mathbf{B}_p = \mathbf{R}_x(\theta)\mathbf{R}_y(\phi)\mathbf{R}_z(\psi)\mathbf{B}_e, \tag{1}$$

where $\mathbf{B}_e$ is the magnetic field vector at the same location in terms of earth coordinate system, and $\mathbf{R}_x(\theta)$, $\mathbf{R}_y(\phi)$, $\mathbf{R}_z(\psi)$ are corresponding *rotation matrices* w.r.t the three axises of the smartphone [2]. Then we obtain

$$\mathbf{B}_e = \mathbf{R}_z^{-1}(\psi)\mathbf{R}_y^{-1}(\phi)\mathbf{R}_x^{-1}(\theta)\mathbf{B}_p \tag{2}$$

which is independent of the dynamic smartphone headings. However, we do not use $\mathbf{B}_e$ directly as the observation, since smartphone heading estimation is error-prone [4]. Instead we retrieve both the vertical and horizontal components (w.r.t. gravity), i.e., B_v and B_h, from $\mathbf{B}_p$. This is because the gravity sensor indicates the vertical direction and is stabler with location and time [2]. We combine them with the magnitude of $\mathbf{B}_p$ to generate the observation at location o, i.e.,

$$\mathbf{B}_o = (||\mathbf{B}_p||, B_v, B_h)\,. \tag{3}$$

Compared with Magicol [10] which proposes using the magnitude itself as the magnetic field fingerprint, our observation has three dimensions which contain more information regarding the magnetic field.

Based on the magnetometer readings, in the offline phase we simultaneously match the trace and collect the data as follows:

1) *Measuring magnetic fields and motions*: a surveyor walks in the area and the smartphone records all inertial data, including magnetic signals, gyroscope readings and accelerations during the walk. Note that all positions of interest should be walked through at a roughly constant speed for the better data-trajectory matching purpose introduced in the following. In our setting we walk through several survey paths which altogether cover all the positions of interest.

2) *Matching trajectory and data*: the ground-truth trace is mapped against the preplanned path based on walking distances, data timestamps and turns in order to determine the location of each step. We calculate the locations of

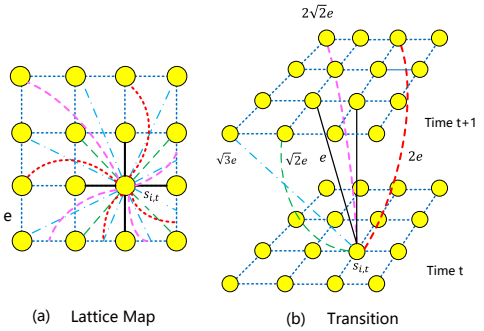

Fig. 3: Illustration of the lattice map and the transition map between different times for graphical model calculation. Each edge is labeled with its length.

intermediate steps proportionally according to estimated step lengths and overall distance between two turns. The location of each collected magnetic data is interpolated into the corresponding step proportional to the timestamp difference. Then the magnetic field fingerprint database can be built by extrapolating the magnetic field data of given positions using Gaussian interpolation [23].

It is worth noting that in practice, if it is detected that the layouts have been changed, we can recollect the magnetic field fingerprints near the changed layouts and replace them accordingly. However, for RF signals, a simple change of layout/access point (AP) locations or adding/removing an AP may lead to an invalidation of the whole original fingerprint and a whole fingerprint recollection is needed.

B. Indoor Map Discretization

The inner structures of a building impose hard constraints on available pedestrian positions. Hence, the main purpose of map preprocessing is to remove the degrees of freedom from the map where pedestrians are less likely to be. This step takes in a predefined floor plan as input, and generates a *lattice* consisting of discrete states (indoor locations), while preserving the physical hard constraints between states.

The indoor map discretization is as follows. First, we extract periphery edges from the floor plan by finding the contours of walls and doors [24]. Second, we divide the plan into small identical squares, or a *lattice*, with edge length e. Note that a large e degrades localization accuracy due to low map granularity, while a small e increases computational cost. In our localization system, we choose e as the average step length between $0.5\ m$ and $1\ m$ (say, $e = 0.8\ m$ in our system deployment). Third, we define $\mathbf{S} = \{\mathbf{s}_{i,t} | 1 \leq i \leq N\}$ as nodes of lattice for the moment of each step t, where N is the number of nodes in a layer. Figure 3(a) shows the lattice structure. We define the edges of lattice as all adjacent nodes between two consecutive layers

$$E = \{(\mathbf{s}_{i,t}, \mathbf{s}_{j,t+1}) \mid 1 \leq i, j \leq N, |a_i - a_j| \leq 2e, |b_i - b_j| \leq 2e\},$$

where a_i, a_j (b_i, b_j) represent the x-coordinate (y-coordinate) of nodes i and j in the lattice, respectively.

We illustrate an example in Figure 3(b). A state $\mathbf{s}_{i,t}$ at time t has the following edges to nodes besides $\mathbf{s}_{i,t+1}$: the ones along lattice edges (4 edges in black with length of e, and 4 edges in red with length of $2e$); the ones at the diagonal (4 diagonal edges in green with length of $\sqrt{2}e$, 8 edges in blue with length of $\sqrt{3}e$, and 4 edges in pink with length of $2\sqrt{2}e$) (here we refer to the spacial distance and ignore the one between the consecutive layers).

Finally, during map preprocessing, we discard nodes which are inaccessible, and remove their corresponding edges. This step is significant because there is typically a large number of locations that cannot be accessed from legal regions while removal of these nodes increases computational efficiency.

C. Conditional Random Field

We first briefly introduce the theory of conditional random field. Conditional random field (CRF) [16] is a probabilistic graphical model for computing conditional probabilities $p(\vec{\mathbf{y}}|\vec{\mathbf{x}})$'s among all possible outputs $\vec{\mathbf{y}} = (\mathbf{y}_1, \ldots, \mathbf{y}_n)$ given the inputs $\vec{\mathbf{x}} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$. It has been applied to many tasks in natural language processing and computer vision fields [16]. Based on the undirected graphical model, the conditional probability of states $\vec{\mathbf{y}}$ given $\vec{\mathbf{x}}$ can be written as

$$p(\vec{\mathbf{y}}|\vec{\mathbf{x}}) \propto \prod_{C \in \mathcal{C}} \Psi_C(\vec{\mathbf{x}}_C, \vec{\mathbf{y}}_C), \tag{4}$$

where $\Psi_C(\vec{\mathbf{x}}_C, \vec{\mathbf{y}}_C)$'s are the *feature functions*, or equivalently, *potential functions*, corresponding to the maximal cliques in the dependency graph [16]. A feature function defines the degree to which the observed signals (observations $\vec{\mathbf{x}}$'s) support the transition between two connected states (states $\mathbf{y}_{i,t-1}$ and $\mathbf{y}_{j,t}$ in our CRF).

Given the lattice map in Figure 3, we formulate the conditional random field in order to infer the final target locations. In our lattice map, we adapt the above CRF for target localization. We consider the magnetic field, user motion information (smartphone orientation, step length and step counts) as the sequence of inputs $\vec{\mathbf{x}}$. Then we implement a CRF to find the output sequence $\vec{\mathbf{y}}$, which is with the maximum conditional likelihood of all potential states $\mathbf{S}$ in the lattice. Specifically, the conditional likelihood by the unary and pairwise potential function is given as follows:

$$p(\vec{\mathbf{y}}|\vec{\mathbf{x}}) \propto \prod_{t=1}^{T} \left(\prod_{j=1}^{N} \Psi_{j,t}(\vec{\mathbf{x}}, \vec{\mathbf{y}}) \cdot \prod_{\substack{i,j \\ (i,j) \in E}} \Psi_{ij,t}(\vec{\mathbf{x}}, \vec{\mathbf{y}}) \right), \tag{5}$$

where the unary potential function (or the state factor at node j) is in the form of

$$\Psi_{j,t}(\vec{\mathbf{x}}, \vec{\mathbf{y}}) = \exp\left(\sum_{k=M^t+1}^{M} \lambda_k f_k(\vec{\mathbf{x}}, \mathbf{y}_{j,t}) \right), \tag{6}$$

where M is the total number of potential functions. The unary function represents the node-wise potential between the measured signals at the smartphone and the stored fingerprints.

Let M^t ($0 \leq M^t \leq M$) be the number of pairwise potential functions. The pairwise potential function (or transition factor

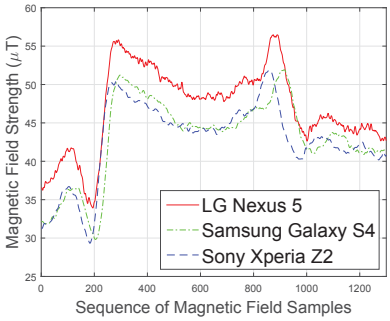

Fig. 4: Magnetic field strength of different devices along the same trajectory.

between $\mathbf{y}_{i,t-1}$ and $\mathbf{y}_{j,t}$) is defined as

$$\Psi_{ij,t}(\vec{\mathbf{x}}, \vec{\mathbf{y}}) = \exp\left(\sum_{k=1}^{M^t} \lambda_k f_k(\vec{\mathbf{x}}, \mathbf{y}_{i,t-1}, \mathbf{y}_{j,t})\right). \quad (7)$$

As potential target locations are within the lattice and can only travel between connected states, Mapel constrains potential locations and reduces indoor disperse mappings, thus avoiding high computation cost suffered by particle filter.

D. Feature Functions in CRF

Here we introduce potential functions (feature functions) in the CRF of Mapel. We specify all the features together with their corresponding observations and how they relate to state transitions. Note that in the CRF of Mapel, we match locations to different states, and consider user movements as transitions between states, and at each moment a user is considered to stay at one single state. Based on the magnetic field and motion sensor signals, in the followings we describe how to define the feature functions:

Magnetic Field: The first feature function, denoted as $f_1(\cdot)$, expresses the degree to which the magnetic field observations are consistent with the fingerprints generated from Section III-A. A unary feature function can be defined as

$$f(\mathbf{x}_t^B, \mathbf{y}_{i,t}) = -(\mathbf{x}_t^B - \mathbf{B}_i)^T (\Sigma_t)^{-1} (\mathbf{x}_t^B - \mathbf{B}_i), \quad (8)$$

where $\mathbf{B}_i$ is the magnetic field fingerprint data at position $\mathbf{y}_{i,t}$, and Σ_t is the covariance. However, magnetic field strength has been observed to differ by some constant offset for different devices as shown in Figure 4. Hence if the exact offset Δ between magnetometers is pre-calculated offline, we can replace $\mathbf{x}_t^B$ with $\mathbf{x}_t^B + \Delta$ in Equation (8) to obtain a usable feature function. Otherwise, measuring the offsets in the magnetic fields of two consecutive steps $t-1$ and t, instead of absolute values, is considered in our formulation to address the device heterogeneity problem.

Specifically, the unary function of Equation (8) is transformed into a pairwise function as in Equation (7): we measure the pairwise geomagnetism offset $\mathbf{B}_i - \mathbf{B}_j$ between states $\mathbf{y}_{i,t-1}$ and $\mathbf{y}_{j,t}$. Then we deduct this offset from the difference between two consecutive observations and have the following pairwise feature function

$$f_1(\mathbf{x}_{t-1}^B, \mathbf{x}_t^B, \mathbf{y}_{i,t-1}, \mathbf{y}_{j,t}) = -\vec{\mathbf{z}}^T (\Sigma_{t-1,t})^{-1} \vec{\mathbf{z}}, \quad (9)$$

where $\vec{\mathbf{z}} = (\mathbf{x}_{t-1}^B - \mathbf{x}_t^B) - (\mathbf{B}_i - \mathbf{B}_j)$. Here $\mathbf{B}_i$ and $\mathbf{B}_j$ are the magnetic field fingerprints at locations $\mathbf{y}_{i,t-1}$ and $\mathbf{y}_{j,t}$, respectively, while $\Sigma_{t-1,t}$ is the signal covariance between $t-1$ and t. With Equation (9), we only compare the difference between two adjacent states with the reading offsets of two consecutive steps. In this way, we avoid offline calibration between different smartphone magnetometers.

Walking Orientation and Step Length: The second and third feature functions consider the extent to which inertial measurements $(\mathbf{x}_t^\theta, \mathbf{x}_t^l)$ support transitions between two states:

$$f_2(\mathbf{x}_t^\theta, \mathbf{y}_{i,t-1}, \mathbf{y}_{j,t}) = I(\mathbf{y}_{i,t-1}, \mathbf{y}_{j,t}) \cdot \widetilde{f}_2\left(\mathbf{x}_t^\theta, \mathbf{y}_{i,t-1}, \mathbf{y}_{j,t}\right),$$
$$f_3(\mathbf{x}_t^l, \mathbf{y}_{i,t-1}, \mathbf{y}_{j,t}) = I(\mathbf{y}_{i,t-1}, \mathbf{y}_{j,t}) \cdot \widetilde{f}_3\left(\mathbf{x}_t^l, \mathbf{y}_{i,t-1}, \mathbf{y}_{j,t}\right),$$

where $I(\mathbf{y}_{i,t-1}, \mathbf{y}_{j,t})$ is an indicator function between states $\mathbf{y}_{i,t-1}$ and $\mathbf{y}_{j,t}$. We define $I(\mathbf{y}_{i,t-1}, \mathbf{y}_{j,t}) = 1$, if and only if the states (in lattice), $\mathbf{y}_{i,t-1}$ and $\mathbf{y}_{j,t}$, are connected. Otherwise, $I(\mathbf{y}_{i,t-1}, \mathbf{y}_{j,t}) = 0$. Two components of observations, namely the walking orientation $\mathbf{x}_t^\theta$ and step length $\mathbf{x}_t^l$ at time t between states $\mathbf{y}_{i,t-1}$ and $\mathbf{y}_{j,t}$, are considered to be independent. Two feature functions, $\widetilde{f}_2\left(\mathbf{x}_t^\theta, \mathbf{y}_{i,t-1}, \mathbf{y}_{j,t}\right)$ and $\widetilde{f}_3\left(\mathbf{x}_t^l, \mathbf{y}_{i,t-1}, \mathbf{y}_{j,t}\right)$, relate the walking orientation and step length with states, respectively. Specifically, $\widetilde{f}_2\left(\mathbf{x}_t^\theta, \mathbf{y}_{i,t-1}, \mathbf{y}_{j,t}\right)$ is defined as

$$\widetilde{f}_2\left(\mathbf{x}_t^\theta, \mathbf{y}_{i,t-1}, \mathbf{y}_{j,t}\right) = \ln\left(\frac{1}{\sqrt{2\pi}\sigma_t^\theta}\right) - \frac{\left(\mathbf{x}_t^\theta - \theta(\mathbf{y}_{i,t-1}, \mathbf{y}_{j,t})\right)^2}{2\left(\sigma_t^\theta\right)^2},$$

where $\theta(\mathbf{y}_{i,t-1}, \mathbf{y}_{j,t})$ is the orientation of the edge between two states, $\mathbf{y}_{i,t-1}$ and $\mathbf{y}_{j,t}$, and σ_t^θ is the orientation variance at time t. Edges from the current state to other neighboring ones are constructed in the lattice in order to represent sixteen possible headings in a discretized manner, as shown in Figure 3 in Section III-B. Another feature function of step length, $\widetilde{f}_3(\mathbf{x}_t^l, \mathbf{y}_{i,t-1}, \mathbf{y}_{j,t})$, is defined similarly, and we do not repeat it here for brevity.

In our system, we choose a typical linear frequency model [21] as our generic step model, i.e.,

$$step\ length = a \cdot f + b, \quad (10)$$

where f is frequency in Hz and a, b are constants which vary among people. Note that other models [25], [26] can easily be applied in our Mapel. As different users may yield different step lengths, we propose a novel algorithm which dynamically adapts the step model later in Section IV-B. Mapel considers conditional probabilities of several steps instead of only one step measurement within the particle, hence is more robust to sensor noise at one time stamp than a traditional particle filter.

It is worth noting that our CRF can be easily extended to other signals like Wi-Fi fingerprints. Through the implementation of a similar unary feature function for magnetic field like Equation (8), we can easily replace $\mathbf{x}_t^B$ with RF fingerprint data (such as Wi-Fi) to measure the degree of signal matching.

E. CRF Coefficients Determination

CRF model in Mapel combines these feature functions together into potential functions. We introduce as follows how coefficients λ_i's in Equations (6) and (7) are determined in offline CRF training. Training CRF is to find the λ_i such that the log-likelihood $\mathcal{L}$ of the training dataset T is maximized:

$$\arg\max_{\{\lambda_i\}} \mathcal{L}(T) = \arg\max_{\{\lambda_i\}} \sum_{(\vec{\mathbf{x}},\vec{\mathbf{y}})\in T} \log\ p(\vec{\mathbf{y}}|\vec{\mathbf{x}}). \tag{11}$$

where T contains magnetic field signal fingerprints, sensor data collected from walking trajectories, and corresponding ground truths of walking trajectories.

To compute the exact optimal λ_i in the CRF structure, we need to use junction tree [27] or loopy belief algorithm [23]. To improve the training efficiency, we first enumerate many possible values of λ_i, then use the location inference algorithm detailed in Section IV-A to obtain the estimations and choose the setting which maximizes the log-likelihood. Other choices of λ_i may have different influences on the localization performance, and detailed results are presented in the experimental evaluation in Section V.

IV. Online Localization

In this section, we present the online location inference using CRF in Section IV-A. In Section IV-B we show how Mapel can adaptively learn step models for different users. Then in Section IV-C, we discuss the computational complexity and some potential performance improvement methods for Mapel.

A. Infering Location

The final step of localization is to find the most likely sequence of $\vec{\mathbf{y}}$, i.e. the most likely trajectory, based on the observations. In practice, we prefer the model to generate estimated positions at every step. In fact, at each time the user can only stay in one position. Therefore, we adapt from Viterbi algorithm to compute the marginal probability of each step. More specifically, in each step, the algorithm finds the highest score (likelihood) along the path ending at state $\mathbf{s}$, and gradually fills in a lattice, i.e.,

$$\begin{aligned}\delta_t(\mathbf{s}|\vec{\mathbf{x}}) &= \max_{\mathbf{s}'\in\mathbf{S},(\mathbf{s}',\mathbf{s})\in E} \left(\delta_{t-1}(\mathbf{s}'|\vec{\mathbf{x}})\cdot\Psi_{\mathbf{s}',\mathbf{s},t}(\vec{\mathbf{x}},\mathbf{s}',\mathbf{s})\right),\\ \pi_t(\mathbf{s}|\vec{\mathbf{x}}) &= \arg\max_{\mathbf{s}'\in\mathbf{S},(\mathbf{s}',\mathbf{s})\in E} \left(\delta_{t-1}(\mathbf{s}'|\vec{\mathbf{x}})\cdot\Psi_{\mathbf{s}',\mathbf{s},t}(\vec{\mathbf{x}},\mathbf{s}',\mathbf{s})\right),\end{aligned} \tag{12}$$

However, in some cases, since some edges in E are longer than a typical step length, the feature functions described above are slightly modified to capture the motion information in two or more steps. For example, considering edge $e_0 \in E$ whose length is twice as long as a typical step length. In this case,

$$\begin{aligned}\delta_t(\mathbf{s}|\vec{\mathbf{x}}) = \max_{\mathbf{s}'\in\mathbf{S},e_0=(\mathbf{s}',\mathbf{s})\in E} &(\delta_{t-2}(\mathbf{s}'|\vec{\mathbf{x}})\cdot\Psi_{\mathbf{s}',\mathbf{s}'',t-1}(\vec{\mathbf{x}},\mathbf{s}',\mathbf{s}'')\\ &\cdot\Psi_{\mathbf{s}'',\mathbf{s},t}(\vec{\mathbf{x}},\mathbf{s}'',\mathbf{s}))\end{aligned} \tag{13}$$

where $\mathbf{s}''$ is the estimated location of user at time $t-1$ and can be estimated by using $\mathbf{s}'$ and walking direction and step length at time $t-1$. $\pi_t(\mathbf{s}|\vec{\mathbf{x}})$ can be defined accordingly.

In each step, the most recently filled column of the lattice $\delta_t(\cdot)$ is normalized. Let the total step of the pedestrian be N_s. The final sequence $\vec{\mathbf{y}}$ can be obtained by first finding

$$\mathbf{y}_{N_s} = \arg\max_{\mathbf{s}\in\mathbf{S}}(\delta_{N_s}(\mathbf{s}|\vec{\mathbf{x}})), \tag{14}$$

followed by the recursive search of

$$\mathbf{y}_i = \pi_{i+1}(\mathbf{y}_{i+1}), i \in \{N_s-1, N_s-2, ..., 1\}. \tag{15}$$

B. Self-calibrating a Personalized Step Model

As people's strides vary, simply using the general step model described above for all users may incur large localization error. Next we propose a novel algorithm which dynamically adapts the step model. Initialized with a generic step model, Mapel learns the step length models for different users to further maintain high localization accuracy.

We first present the basic idea. After each step, Mapel returns an estimated trajectory which matches the observations most. We can collect all the magnetic field fingerprint data on the estimated trajectory as the fingerprint data vector, each component of which is an observation $\mathbf{B}_o$. Similarly, we also have the ground truth of the magnetic field data vector. Recall that Figure 4 shows that along the same trajectory, the magnetic field sequence shapes are similar. Therefore, we adopt derivative dynamic time warping [12] (DDTW) to compare and match the most similar parts of two vectors. Then we estimate the walking distance using the real-world total distance of the matched part in the fingerprint data vector. Given the estimated walking distance, Mapel regresses the model parameters in Equation (10) and hence adapts to different users.

The details of our adaption algorithm are as follows. Observations in one step may not be applicable to estimating walking distance. In those areas where magnetic fields vary little, matching error with dynamic time warping [28] (DTW) can be large. Here we aggregate magnetic field readings from several consecutive steps together and perform DTW on the aggregated observations only when the range of magnetic field magnitude readings is larger than a threshold, say, 5 μT in our system. Since we prefer matching shape characteristics of series (such as peaks and slopes) rather than actual values, we perform DTW on the first-order derivatives on each dimension of magnetic field features (Derivative Dynamic Time Warping [12], or DDTW), and calculate the distance between the i-th observation $\mathbf{B}_t(i)$ at target and the j-th magnetic field fingerprint $\mathbf{B}_f(j)$ using $L2$-norm distance, i.e.,

$$D\left(\mathbf{B}_t(i),\mathbf{B}_f(j)\right) = \sqrt{\sum_{k=1}^{3}\left[der(\mathbf{B}_t(i,k)) - der(\mathbf{B}_f(j,k))\right]^2},$$

$$\text{where } der(\mathbf{B}_t(i,k)) = \frac{1}{2}(\mathbf{B}_t(i+1,k) - \mathbf{B}_t(i-1,k)),$$

$$der(\mathbf{B}_f(j,k)) = \frac{1}{2}(\mathbf{B}_f(j+1,k) - \mathbf{B}_f(j-1,k)),$$

$\mathbf{B}_t(i,k)$ and $\mathbf{B}_f(j,k)$ are the k-th dimension of $\mathbf{B}_t(i)$ and $\mathbf{B}_f(j)$, respectively, given the definition of magnetic field

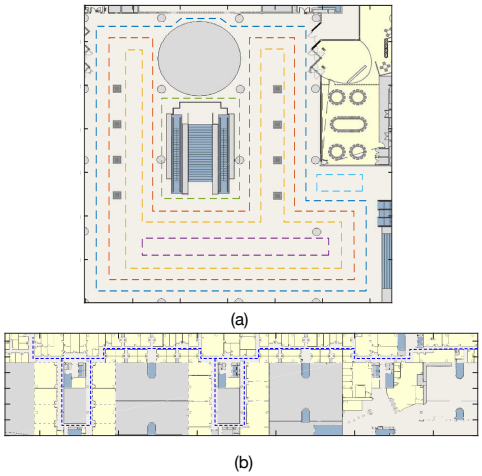

Fig. 5: Survey paths (dashed lines) in university (a) atrium & (b) corridors.

features in Equation (3).

After this we obtain a number of triplets: ⟨the total number of steps in one aggregation, sum of step frequencies, estimated walking distance obtained from DDTW matching⟩. Since $\sum_{i=1}^{S}(a+f_i\cdot b)=a\cdot S+b\cdot\sum_{i=1}^{S}f_i$ where S is the number of steps in one aggregation, the final personalized step model is obtained by linear regression on these triplets. It is further leveraged in Mapel to improve localization accuracy.

C. Complexity & Performance Improvement

The time complexity of estimating trajectory is $O(|\mathbf{S}|\cdot N_s)$, where $\mathbf{S}$ is the set of states. If the number of states is too large, the algorithm may suffer from high computational cost. In the case of real-time tracking, we also introduce some heuristics to improve the performance. First, Mapel conducts an exhaustive search over the whole site in the first few steps (initialization), until there are only a few states whose scores (likelihood values) are much higher than others. Then Mapel conducts localization over these states discovered. Second, if Mapel finds in some step that the sum of all states' scores (likelihoods) before normalization is notably small, then it stops and restarts the localization (when signal noise is large or under abnormal holding gestures by users).

The time complexity of step model learning is $O(KM_fM_o)$, where K is the number of triplets, M_f is the maximum length of fingerprint data vector in the triplets and M_o is the maximum length of magnetic field ground-truth observations in the triplets. Note that the step length model can be computed online, making the system adaptive to different users.

V. Experimental Evaluation

In this section, we evaluate the performance of Mapel in different environments. In Section V-A, we first introduce the system settings and comparison schemes. Then in Section V-B, we illustrate the experimental results.

A. System Settings & Comparison Schemes

Different types of smartphones are involved in experiments, including Samsung Galaxy S4, LG Nexus 5 and Sony Xperia Z2. We use Sony Xperia Z2 to perform site survey tasks. They all run the Android operating system no earlier than 2.3. We implement the server on a Dell PC with a 3.6GHz processor, 16G RAM and Windows 8.1. The mobile device performs inertial sensor sampling and walking state detection continuously in the background. When the user is detected to be walking, the step information (including step count, step frequency and walking direction) is logged and used for localization. Sampling frequency is set to be 25Hz for all sensors. From the experiments we observe that only a few steps of walking trace are sufficient to localize the user. Hence we only need to keep a small buffer for the walking trace.

In our experiment, we compare our Mapel with the following two state-of-the-art algorithms, and the detailed parameter settings follow and can be referred to their works.

- *Magicol* [10]: which measures not only the values but also the relative trend of magnetic field change, when the user is walking through an indoor corridor. Via dynamic time warping and particle filter, Magicol filters away incorrect locations and maps the target to location with the optimal trend matching.
- *MaLoc* [4]: which uses particle filter to yield potential locations. The particle filter first selects candidate locations with the best magnetic field matching, then reduces the weights of particles in incorrect positions. Specifically, we implement MaLoc by setting initial number of particles to be $2,000$.

We have conducted extensive experiments to validate the localization algorithms in two typical indoor environments: a large open campus atrium of $2,046$ m^2 (Figure 5(a)) and the long office corridors of $9,630$ m^2 (Figure 5(b)) , respectively. In floor maps, we also show the survey paths.

To obtain the ground truth of walking traces, we first set many landmarks (say, doors or corners) and measure their real positions in advance. When users are walking, they record the time when passing by those landmarks. Localization errors are calculated as all the Euclidean distances between the estimated locations in the historical trajectory and ground truths. Real-time localization errors are given by the errors at each step.

We have also conducted extensive experiments to further validate the accuracy of localization and step model personalization. In the experiment we invite eight volunteers to walk along the ground-truth trajectory presented in Figure 6(a). In summary, there are six males and two females whose heights and weights vary from 1.68 m to 1.87 m, and from 50 kg to 80 kg, respectively. For each user, we also compare the estimated walking distance obtained from the adapted step model with the ground truth by calculating the relative error ratio of the absolute value of difference between estimation and the ground truth of walking distance against the latter at each step.

B. Illustrative Experimental Results

Figure 6 shows the estimated walking traces of Mapel (blue dotted lines) in the campus atrium and corridors compared with the ground truth in red solid lines. We can observe the estimated trajectory with Mapel closely matches with the

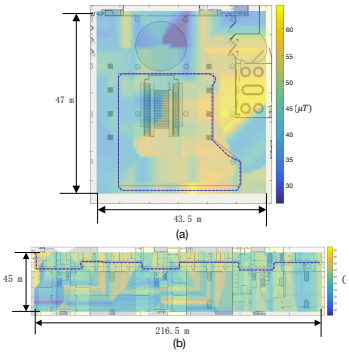

Fig. 6: Estimated walking trajectories of Mapel in (a) the atrium and (b) corridors. The red solid line is the ground truth and the blue dotted line is the estimated trajectory.

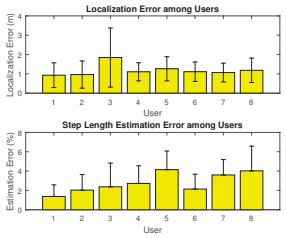

Fig. 7: Localization error and step length estimation error among users.

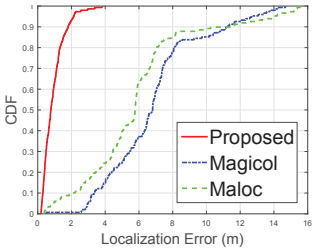

Fig. 8: CDF of indoor localization error in the campus atrium.

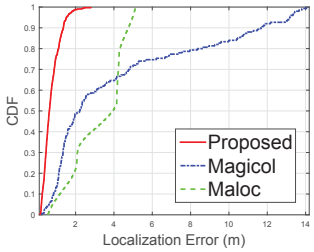

Fig. 9: CDF of indoor localization error in the corridors.

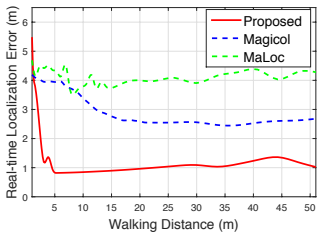

Fig. 10: Real-time localization error vs. user walking distance (campus atrium).

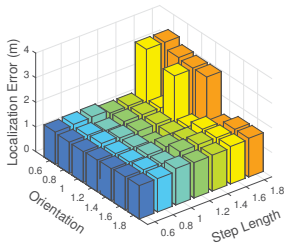

Fig. 11: Mean localization error vs. different ratios of λ between orientations and step lengths.

ground-truth. We compare our results with other localization systems such as Magicol and MaLoc. Figure 8 shows the CDF of localization errors in the university atrium. We can clearly see that our Mapel works well in large open space and outperforms the traditional particle-filter-based algorithms. In Figure 9, we also show the CDF of localization errors in the university corridors. In these figures, we also show that our scheme outperforms other state-of-the-arts with lower localization errors and smaller deviation (cutting localization error by more than 40%).

In this experiment we also invite another seven volunteers besides user No.1 to walk along the same trajectory in the atrium. Figure 7 shows mean localization error, step length estimation error and standard deviation for users. We can see that Mapel achieves high accuracy in location and step length estimation (errors are 1.18m and 2.80% on average, respectively) for different users. If we use generic step model [21], the average step length estimation error grows to 4.41%, showing that Mapel achieves better adaptivity with self-calibration.

Figure 10 shows the average real-time localization errors of different systems in the campus atrium. For the particle filter based systems, we repeat the followings ten times and show the average. In each experiment, we manually choose a random initial position near the accurate initial position and walk on the same trajectory afterwards. Recall that from Figure 1 we can see that the typical particle-filter-based system cannot converge at all given no initial position. In Figure 10, with a rough initial position, the particle filter can only provide the coarse location, which is of low localization accuracy. As the particle may not capture all possible step length and direction errors, Magicol and MaLoc cannot converge quickly in such large open spaces or complex environments. Since Mapel discretizes the states into finite transitions and constrains the estimation deviation, it can converge much faster (after the first 2 meters) and more accurately to the correct locations.

Since the final probabilistic score (likelihood) is normalized, we can divide all λ_i by λ_1 and check these ratios instead. Figure 11 shows the localization accuracy in the campus atrium under different settings of λ_i $(2 \leq i \leq 3)$, representing the weights of the step orientation feature and the step length feature, respectively. From the figure we can see that different weights only introduce little change in localization error (often less than 1.5m). Note that a too large feature weight should be avoided because it amplifies minor differences in the feature values of two tracking solutions into significant differences in the potential function, due to the exponential function in Equations (6) and (7). In the experiment we observe that all feature weights chosen from $[0.6, 1.8]$ already yield good performances applicable in various environments.

As the Mapel users may carry devices different from the one used for magnetic field fingerprint construction, we also evaluate the compatibility of Mapel among different smartphones. When no pre-calculated magnetometer offset is provided, we instead implement Equation (8) as the magnetic field feature function. Figures 12 and 13 show the CDFs of localization errors in the campus atrium with and without addressing the device heterogeneity problem in Nexus 5 and Samsung Galaxy S4, respectively. We can see that due to different offsets of these magnetometers, solely comparing the fingerprints in the database with collected ones leads to high localization error. If magnetic field differences between different positions are applied instead, Mapel successfully captures the device-independent features for better localization accuracy. Note that Mapel achieves consistent errors on Samsung Galaxy S4 because Figure 4 has shown minor differences in its readings from that of Sony Xperia Z2 in the same location.

Next we present the computation time of Mapel on Sony Xperia Z2. It is given by the average running time for a single

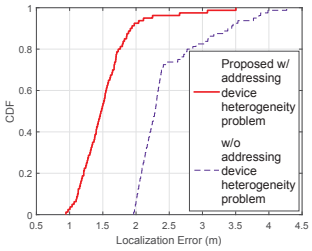

Fig. 12: CDF of Mapel localization error with and without addressing device heterogeneity problem (Nexus 5).

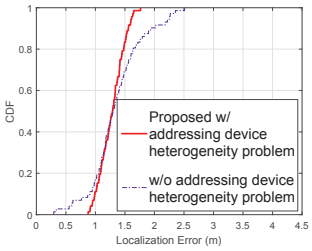

Fig. 13: CDF of Mapel localization error with and w/o addressing device heterogeneity problem (Samsung Galaxy S4).

step along the same trajectory. The execution time is 482.2 ms, 1808.8 ms and 743.8 ms for Mapel, Magicol and MaLoc, respectively. Mapel outperforms other state-of-the-art systems in online execution time (reducing computation time by more than 30%), maintains good user experience and achieves much lower localization error.

Finally, we show the average energy consumption of our system using Sony Xperia Z2. The average current is 472.32 mA, 251.14 mA and 781.17 mA for the Mapel application, screen energy consumption and continuous Wi-Fi scanning (with screen on), respectively. We can see that Mapel is much more energy efficient than continuous Wi-Fi scanning.

VI. Conclusion

In this paper, we propose Mapel, a novel magnetic field localization system based on the graphical model. We discretize the indoor map into the lattice structure. Each node in the lattice represents a state or the location of the target, and the connected edges from other nodes can constrain the target's potential locations into finite ones. Therefore, we reduce the degree of freedom for the target location estimation. Then we fuse geomagnetism and motion information from the step counter and implement conditional random field (CRF) to infer the target location. Furthermore, Mapel adaptively learns the step length model for each user to be pervasively deployed.

Compared with the traditional particle filter approaches, our Mapel converges faster and achieves much higher localization accuracy. Extensive studies in our university campus have shown that Mapel outperforms the state-of-the-art schemes by a large margin (cutting localization error by more than 40% and reducing computation time by more than 30%).

Acknowledgements

This work was supported, in part, by Guangdong Provincial Department of Science and Technology (GDST16EG04 under the contract 2016A050503024) and Natural Science Foundation of Guangdong Province (2014A030313154).

References

[1] Y. Shu, K. G. Shin, T. He, and J. Chen, "Last-mile navigation using smartphones," in *Proc. ACM MobiCom*, 2015, pp. 512–524.

[2] B. Li, T. Gallagher, A. Dempster, and C. Rizos, "How feasible is the use of magnetic field alone for indoor positioning?" in *Proc. IPIN*, Nov 2012, pp. 1–9.

[3] J. Chung, M. Donahoe, C. Schmandt, I.-J. Kim, P. Razavai, and M. Wiseman, "Indoor location sensing using geo-magnetism," in *Proc. ACM MobiSys*, New York, NY, USA, 2011, pp. 141–154.

[4] H. Xie, T. Gu, X. Tao, H. Ye, and J. Lv, "MaLoc: A practical magnetic fingerprinting approach to indoor localization using smartphones," in *Proc. ACM UbiComp*, 2014, pp. 243–253.

[5] H. Wu, S. He, and S.-H. G. Chan, "Efficient sequence matching and path construction for geomagnetic indoor localization," in *Proc. EWSN*, 2017, pp. 156–167.

[6] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: Zero-effort crowdsourcing for indoor localization," in *Proc. ACM MobiCom*, 2012, pp. 293–304.

[7] N. Roy, H. Wang, and R. R. Choudhury, "I am a smartphone and I can tell my user's walking direction," in *Proc. ACM MobiSys*, 2014, pp. 329–342.

[8] J. Ma, J. Qian, P. Li, R. Ying, and P. Liu, "Indoor localization based on magnetic anomalies and pedestrian dead reckoning‘," in *Proc. IEEE ION GNSS*, 2013, pp. 1033 – 1038.

[9] H. Hellmers, A. Norrdine, J. Blankenbach, and A. Eichhorn, "An IMU/magnetometer-based indoor positioning system using Kalman filtering," in *Proc. IPIN*, 2013, pp. 1–9.

[10] Y. Shu, C. Bo, G. Shen, C. Zhao, L. Li, and F. Zhao, "Magicol: Indoor Localization Using Pervasive Magnetic Field and Opportunistic WiFi Sensing," *IEEE JSAC*, vol. 33, no. 7, pp. 1443–1457, July 2015.

[11] S. Hilsenbeck, D. Bobkov, G. Schroth, R. Huitl, and E. Steinbach, "Graph-based data fusion of pedometer and WiFi measurements for mobile indoor positioning," in *Proc. ACM UbiComp*, 2014, pp. 147–158.

[12] E. J. Keogh and M. J. Pazzani, "Derivative dynamic time warping." in *Sdm*, vol. 1. SIAM, 2001, pp. 5–7.

[13] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proc. IEEE INFOCOM*, vol. 2, 2000, pp. 775–784.

[14] M. Youssef and A. Agrawala, "The Horus WLAN location determination system," in *Proc. ACM MobiSys*, 2005, pp. 205–218.

[15] Z. Xiao, H. Wen, A. Markham, and N. Trigoni, "Indoor tracking using undirected graphical models," *IEEE Trans. Mobile Computing*, 2015.

[16] R. Klinger and K. Tomanek, *Classical probabilistic models and conditional random fields*. TU, Algorithm Engineering, 2007.

[17] S. He, S.-H. G. Chan, L. Yu, and N. Liu, "Calibration-free fusion of step counter and wireless fingerprints for indoor localization," in *Proc. ACM UbiComp*, 2015, pp. 897–908.

[18] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury, "No need to war-drive: Unsupervised indoor localization," in *Proc. ACM MobiSys*, 2012, pp. 197–210.

[19] K. P. Subbu, B. Gozick, and R. Dantu, "LocateMe: Magnetic-fields-based indoor localization using smartphones," *ACM TIST*, vol. 4, no. 4, pp. 73:1–73:27, Oct. 2013.

[20] Z. Yang, C. Wu, and Y. Liu, "Locating in fingerprint space: Wireless indoor localization with little human intervention," in *Proc. ACM MobiCom*, 2012, pp. 269–280.

[21] W. Zijlstra, "Assessment of spatio-temporal parameters during unconstrained walking," *European Journal of Applied Physiology*, vol. 92, no. 1-2, pp. 39–44, 2004.

[22] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, and F. Zhao, "A reliable and accurate indoor localization method using phone inertial sensors," in *Proc. ACM UbiComp*, 2012, pp. 421–430.

[23] C. M. Bishop *et al.*, *Pattern recognition and machine learning*. springer New York, 2006, vol. 1.

[24] S. Suzuki *et al.*, "Topological structural analysis of digitized binary images by border following," *Computer Vision, Graphics, and Image Processing*, vol. 30, no. 1, pp. 32–46, 1985.

[25] J. Rose, J. G. Gamble, and J. M. Adams, *Human walking*. Lippincott Williams & Wilkins Philadelphia, 2006.

[26] S. H. Shin, C. G. Park, J. W. Kim, H. S. Hong, and J. M. Lee, "Adaptive step length estimation algorithm using low-cost mems inertial sensors," in *Proc. IEEE SAS*, Feb 2007, pp. 1–5.

[27] S. L. Lauritzen and D. J. Spiegelhalter, "Local computations with probabilities on graphical structures and their application to expert systems," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 157–224, 1988.

[28] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series." in *KDD workshop*, vol. 10, no. 16. Seattle, WA, 1994, pp. 359–370.