Efficient Sequence Matching and Path Construction for Geomagnetic Indoor Localization

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

Abstract

Geomagnetic field is highly applicable to indoor localization due to its pervasive spatial presence, high signal stability and infrastructureless support. Previous work in the area often fuses it with pedometer (step counter) via particles. These approaches are computationally intensive and require strong assumptions on user behavior.

To overcome the weaknesses, we propose Magil, a pedometer-free approach leveraging solely upon **mag**netic field for **i**ndoor **l**ocalization. To the best of our knowledge, this is the first piece of work using only geomagnetism for smartphone localization. Magil is applicable to any open or complex indoor environment. In the offline phase, it continuously collects and stores geomagnetic fingerprints while a surveyor is walking in pre-defined paths covering the indoor area. In the online phase, it identifies the indoor segments whose fingerprint variations highly match with the target observations. With a modified shortest path formulation, Magil selects and connects these matched segments and obtains the target locations and paths. We have implemented Magil, and conducted extensive experiments in our university campus. Our results show that Magil outperforms many state-of-the-art schemes by a wide margin (cutting localization error by more than 30%).

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless Communication

General Terms

Design, Experimentation, System, Performance

Keywords

Indoor localization; geo-magnetism; smartphone-based; shortest path problem formulation; sequence matching.

International Conference on Embedded Wireless
Systems and Networks (EWSN) 2017
20–22 February, Uppsala, Sweden

ISBN: 978-0-9949886-1-4

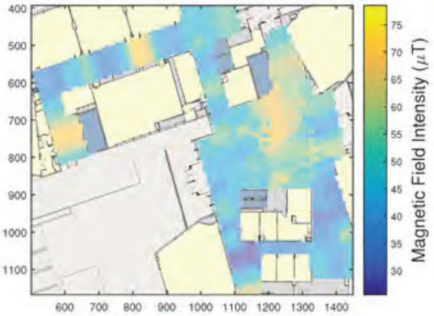


Figure 1: The magnetic fingerprint (intensity in μT) in a typical indoor environment (the x-axis and y-axis in all the maps in this paper are in pixels, and 40 pixels in the maps correspond to 3 meters in the physical world.).

1 Introduction

Recently there has been much interest in indoor localization, mainly due to the ubiquity of smart devices and advances in their sensing capabilities. The signals studied include Wi-Fi RSSI, channel state information (CSI), vision, visible light, ultra-sound, etc. While impressive, these systems require pervasive (and hence costly) or special infrastructure installations. Furthermore, the survey procedures of these systems are rather time-consuming, where signal vectors have to be collected by standing at length at different fixed points in the area.

Indoor localization based on geo-magnetic fields emerges as a promising technique because the fields exhibit local variations caused by electrical appliance and building materials [11]. Figure 1 shows the geo-magnetic field variations in a typical indoor environment, which can be used as fingerprints to identify target locations. Besides, geomagnetism possesses the following properties which are particularly applicable to indoor localization:

- *No need of extra infrastructure support:* Geomagnetic fields are omnipresent. Without the need of any extra infrastructure, it is more cost-effective to deploy and its performance does not sensitively depend on the infrastructure environment or the sensor density.
- *Robustness against indoor adjustment and mobile objects*: Geomagnetic fields are more stable as compared with other signals, leading to better performance. Fur-

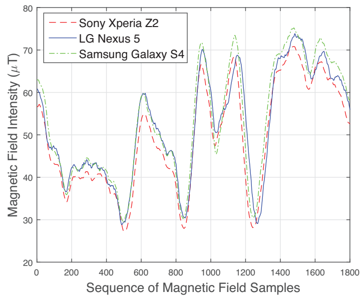


Figure 2: Magnetic field intensity of different devices along the same trajectory.

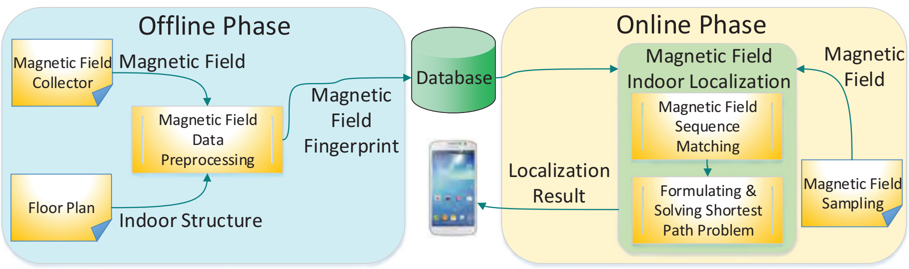

Offline Phase
Magnetic Field Collector
Magnetic Field
Magnetic Field Data Preprocessing
Floor Plan
Indoor Structure
Magnetic Field Fingerprint
Database
Localization Result
Online Phase
Magnetic Field Indoor Localization
Magnetic Field Sequence Matching
Formulating & Solving Shortest Path Problem
Magnetic Field
Magnetic Field Sampling

Figure 3: System framework of Magil.

thermore, an alteration in indoor layouts often has local impacts on the field patterns, with influence punctate only around the changed area while the whole signal map does not change significantly over so wide a scope as compared with other signals, and it has also been shown that the impact of mobile objects on the magnetic field is very limited [3, 11, 20, 28].

- *Deployability in heterogeneous mobile devices*: Geomagnetic field can be efficiently collected using mobile devices, because its data can be sampled at high rate *while* walking. Even though different devices at the same location may have different magnetic field readings, constant offsets are observed among these devices. Figure 2 illustrates magnetic field data collected using different devices along the same trajectory. We can see that, although the reading varies for different devices, the gaps between their readings are rather constant. This leads to simplicity in design, and universality in applying the algorithm across different device platforms.

Previous work on geomagnetic localization is predominantly on fusing pedometer (step counter) with magnetic field, using recursive Bayesian filters such as Hidden Markov Models (HMMs) [14] and particle filters [19, 28]. They estimate the location based on the maximum joint probability or particle convergence, given the magnetic readings and walking distance of the target. Despite promising, these approaches are often computationally complex for a mobile device. Furthermore, these models work the best for partitioned indoor environments characterized by narrow corridors, where the degree of freedom for the target is low. For large spacious settings, these models barely converge to the target location. Previous approaches also often have strong assumptions on user behaviors, relying heavily on a meticulously tuned pedometer which yields accurate step counts, walking distances and directions. In reality, the pedometer itself is not robust against movement, and error tends to accumulate over time [2]. They also require explicit and/or accurate initial target position and step length information, which is often not available or inconvenient to obtain in practical deployment.

To address the above, we propose **Magil**, a novel approach leveraging solely **mag**netic fields for **i**ndoor **l**ocalization. Without using any step counter or error-prone sensors such as gyroscope (which suffers from angular drift), Magil is more robust against different user behaviors. It is applicable to any indoor environment, including large open setting. Magil initially takes in the spatially-labeled geomagnetism signal map which is collected offline. During the online location estimation, Magil measures a sequence of magnetic signals along the path the user is walking. It then finds the trace of signals most matched with the user observations against the geomagnetism signal map, and returns the trace to the user.

In Magil, we first preplan survey paths to cover the area of interests. Then we collect magnetic field signals along the paths as *fingerprints*. Based on these fingerprints, in the online location estimation we use our proposed modified Smith-Waterman algorithm [21] to find the segments with fingerprint variation best matched with the target observations. In this way, target movements can be represented as a series of matched segments in fingerprints (and thus movements in real world locations). By modeling the localization problem as a modified shortest path problem, the matched segments can be ordered properly to yield the target locations over time.

In contrast to previous particle filter approaches, Magil requires neither meticulously tuned pedometers nor accurate or explicit input of initial user positions. It is also computationally efficient. To the best of our knowledge, this is the first work on using solely geomagnetism for smartphone-based indoor localization. We have implemented Magil in mobile phones, and conducted extensive experiments in our university campus. Our experimental results confirm the implementability of our algorithm, and show that Magil outperforms existing and state-of-the-art algorithms by a large margin (cutting the location error by more than 30% on average).

We show in Figure 3 the system framework of Magil, which consists of two phases, *offline* phase (or geomagnetic fingerprint collection phase) and *online* phase (or localization query phase). In the offline phase, a surveyor carries a smartphone walking in the area. While walking, *geo-magnetic field* signals and their corresponding timestamps are fed to the module *magnetic field data preprocessing*. Given the map information, the module generates the fingerprint signal map and stores it in the *database* for online use.

In the online phase, a user continuously measures the geomagnetic field, which is fed to the *magnetic field indoor localization* module. The module consists of two steps. First, in the module *magnetic field sequence matching*, we use our proposed modified Smith-Waterman algorithm [21] to find those segments with fingerprint variation best matched with the target observations. Second, in the module *formulating and solving shortest path problem*, we formulate a novel shortest path problem. By solving this problem, we reorder the matched segments properly and find the trajectories and locations of the user. The results are then returned to the user.

The remainder of the paper is organized as follows. In Section 2, we show how to construct the magnetic field signal fingerprint database. Localization with magnetic field is then presented in Section 3. In Section 4 we present our experimental results at different sites, followed by some discussions on Magil localization in Section 5. The related work is discussed in Section 6, before we conclude in Section 7.

2 Fingerprint Database Construction

In this section, we introduce how Magil constructs its geomagnetic fingerprint database. we first show how to collect magnetic fields using smartphones, and identify magnetic field features for fingerprints in Section 2.1. In Section 2.2 we describe how to construct the magnetic signal fingerprint database.

2.1 Measuring Geomagnetism Features

The magnetic field vector $\mathbf{B}_p$ can be measured by a smartphone's magnetometer [11]. However, the raw magnetic readings are under the smartphone's coordinate system. We hence need to transform the readings into the one under the earth coordinate system by the yaw ψ, pitch θ and 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 the 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 axes of the smartphone [11]. Then we obtain

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

In this way, we can measure the magnetic field $\mathbf{B}_e$ irrespective of the dynamic smartphone headings. We, however, do not use $\mathbf{B}_e$ directly as the observation, because smartphone heading estimation is error-prone [33] and these errors would be amplified on $\mathbf{B}_e$. Some works [11, 19] suggest using the magnitude of $\mathbf{B}_p$ as fingerprints because it is rotation invariant scalar quantity and quite stable. However there is only one fingerprint dimension, which reduces the uniqueness of fingerprints. On the other hand, most smartphones have been equipped with gravity sensors which sense the direction of gravity and is stable with location and time [11]. Therefore, we can retrieve both the vertical and horizontal components (w.r.t. gravity), B_v and B_h, of $\mathbf{B}_p$ and combine them with the magnitude of $\mathbf{B}_p$ to generate an observation at location o, i.e.,

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

2.2 Construction Steps

In the offline phase, we conduct magnetic field data collection and database construction as follows:

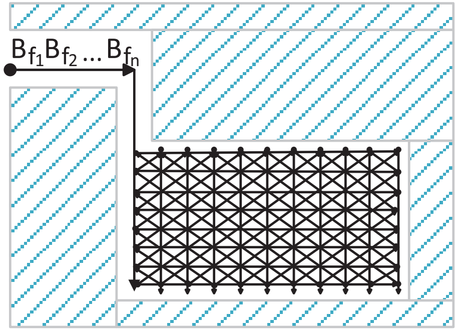


Figure 4: An example of survey paths and corresponding walking directions. One fingerprint magnetic field data vector $Seq_{fingerprint} = \langle \mathbf{B}_{f_1}\mathbf{B}_{f_2}...\mathbf{B}_{f_n} \rangle$ is also shown in the figure.

1. *Survey path planning*: we preplan several survey paths from the floor map we obtain in the map preprocessing step. Typically these paths should be along the corridors, across lobbies and at peripheries of obstacles and cover various walking directions. For an extremely large open space, we can add more paths to ensure the coverage on positions and walking directions of interests. An example is shown in Figure 4, where a large indoor area is covered by many survey paths in any potential directions. Note that we can adaptively align the survey paths according to the site shape. For example, in constrained corridors we align the trajectories along the accessible paths, while in open space we may align them in intersecting parallels. Note that for each survey path, we only need to walk along one direction to collect the fingerprint data.
2. *Magnetometer and motion sensor measurement*: a surveyor walks along paths and meanwhile his/her smartphone records magnetic field data and corresponding timestamps during the walk. Note that the surveyor should walk at a roughly constant speed for better data-trajectory matching purpose introduced in the following.
3. *Data-trajectory matching and data preprocessing*: actual trace is mapped against the preplanned path based on data timestamps and turns. We calculate the locations of intermediate magnetic field data proportionally according to their timestamps and the overall time interval between two turns. The location of each collected magnetic field data is interpolated into segments of preplanned paths proportional to their timestamp differences. After this, Piecewise Aggregate Approximation (PAA [10]) is performed separately on the magnetic field data of each survey path to reduce the data dimensionality. More specifically, denote the length of data as n. Then the data is partitioned into $\frac{n}{10}$ equally-sized frames, within each of which we calculate the mean value of the data. A vector of these mean values from all frames becomes the dimension-reduced representation of the data and is stored in the database.

It is worth noting that in practice, if we detect the layouts have been changed, we can recollect the fingerprints near the changed layouts and replace them accordingly; for RF signals, a simple change of layouts/access point (AP) locations or adding/removing an AP may lead to invalidations of the whole original fingerprint and a w-

Table 1: Major symbols used in Magil.

Notations	Definitions
$\mathbf{B}_p$	Geomagnetism reading under phone coordinate system
$\mathbf{B}_o$	Geomagnetism observation under earth coordinate system
$\langle \mathbf{B}_{o_1}\mathbf{B}_{o_2}...\mathbf{B}_{o_m}\rangle$	User observation sequence $Seq_{observe}$ whose length is m
$\langle \mathbf{B}_{f_1}\mathbf{B}_{f_2}...\mathbf{B}_{f_n}\rangle$	A fingerprint magnetic field data vector $Seq_{fingerprint}$ whose length is n
G	The corresponding graph generated by transforming all matched parts of fingerprints and user observations
N	The number of vertexes in G
M	The number of edges in G

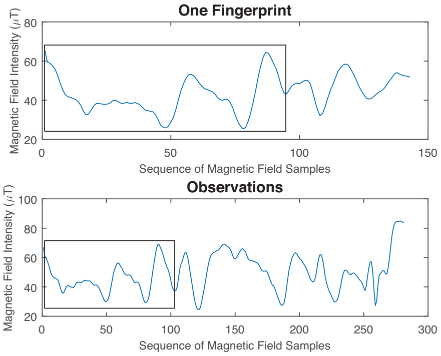


Figure 5: Magnetic field sample matching between fingerprint and observations. Note that only the parts in the boxes match each other well.

hole fingerprint recollection is needed.

3 Localization Based on Only Geomagnetism

In this section, we present the problem formulation and solution for Magil, an indoor localization technique based on geomagnetism only. We summarize the important symbols used in Table 1.

The user's smartphone first samples a sequence of magnetic field data, and Magil performs the similar PAA [10] on the geomagnetism data to obtain a shorter sequence of magnetic field observations. After that, we need to address the problem on how to determine the user location based on the signal sequence given the magnetic fingerprint database. In geomagnetism-based localization, two subproblems arise (in below) before the target can be effectively located.

From Figure 2 we note that along the same trajectory, the shapes of magnetic field sequences among devices are similar. As each sample is collected by a user somewhere in the survey path, the first subproblem, the so-called *sequence matching* problem, is: Given a vector with each component being an observation sampled by the user and a magnetic field fingerprint data vector obtained from the database, how can we determine whether and where they are similar?

Furthermore, user may traverse across any places, and thus different observations may match different magnetic field fingerprint data vectors. Therefore, the second subproblem, the so-called *path construction* problem, is: How can we perform "concatenation" on the matchings to obtain the whole estimated walking trajectory?

To address the above two subproblems, we propose in the following a novel geomagnetism-based localization algorithm. In order to match the geomagnetism sequence, we modify the Smith-Waterman algorithm [21] to determine the similarity between the signal observations from the user and the observations from survey paths (Section 3.1). Given the matching results, we convert them into vertices and a graph, formulate a shortest path problem (Section 3.2), and solve it efficiently via a modified Dijkstra algorithm (Section 3.3). We end by summarizing the computational efficiency (Section 3.4).

3.1 Sequence Matching for Magnetic Field

We first match the user magnetic observations against the fingerprint. Figure 5 shows a matching example between a observation sequence and the fingerprint. We can clearly see that the parts in the boxes match each other well. The problem here is how to identify the similar parts among sequences.

We devise a magnetic field sequence matching scheme by modifying the Smith-Waterman algorithm [21]. Smith-Waterman algorithm is a dynamic programming algorithm for performing local sequence alignment which has been widely used in bioscience (e.g. to determine similar regions between two strings of nucleotides or protein sequences).

We make modifications to the original Smith-Waterman algorithm to support the magnetic field sequence matching. We denote an observed sample (after performing PAA) as $\mathbf{B}_o$ and the user observation sequence as $Seq_{observe} = \langle \mathbf{B}_{o_1}\mathbf{B}_{o_2}...\mathbf{B}_{o_m}\rangle$ where m is the sequence length. Note that each magnetic field observation $\mathbf{B}_o$ can be viewed as a point in 3D space. We also denote a magnetic field fingerprint data vector as $Seq_{fingerprint} = \langle \mathbf{B}_{f_1}\mathbf{B}_{f_2}...\mathbf{B}_{f_n}\rangle$ where $\mathbf{B}_{f_i}$ is the i-th fingerprint magnetic field data (which is also an observation). However we cannot determine whether $\mathbf{B}_{o_i}$ matches $\mathbf{B}_{f_j}$ by simply calculating their Euclidean distance, since different devices may yield different magnetic field values even if samples are collected at the same place. Figure 2 also shows this characteristic. To address this, our key observation is that the differences between readings among different devices are almost the same despite locations. Let $dist(\cdot,\cdot)$ be the Euclidean distance between two samples. Based on the observation, we can make use of this and determine $\mathbf{B}_{o_i}$ and $\mathbf{B}_{f_j}$ match with each other if and only if $g(o_i,f_j)$, the average Euclidean distance between adjacent samples of $\mathbf{B}_{o_i}$ and $\mathbf{B}_{f_j}$ after the mean removal, is less than a certain threshold, i.e.,

$$g(o_i,f_j) < Threshold, \tag{4}$$

where

$$g(o_i,f_j) = C\cdot \sum_{t=-window}^{window} dist(\mathbf{B}_{o_{i+t}} - Mean(o_i), \mathbf{B}_{f_{j+t}} - Mean(f_j)), \tag{5}$$

$$\begin{aligned} Mean(o_i) &= C\cdot \sum_{t=-window}^{window} \mathbf{B}_{o_{i+t}}, \\ Mean(f_j) &= C\cdot \sum_{t=-window}^{window} \mathbf{B}_{f_{j+t}}, \\ C &= 1/(2\cdot window+1). \end{aligned} \tag{6}$$

In practice we empirically set $window = 5$ and $Threshold = 2\mu T$, which balances the robustness and reasonable computation time. We assign a score 1 for matches and the penalty cost for mismatches is set to an empirical value of -1 which yields a competitive performance in practice. We perform this modified Smith-Waterman sequence matching algorithm on each original magnetic field fingerprint data vector $\langle \mathbf{B}_{f_1}\mathbf{B}_{f_2}...\mathbf{B}_{f_n}\rangle$ and its corresponding reversed fingerprint data vector $\langle \mathbf{B}_{f_n}\mathbf{B}_{f_{n-1}}...\mathbf{B}_{f_1}\rangle$, since the user can walk in two different directions along the same path.

After running the sequence matching algorithm for a magnetic field fingerprint data vector and a user observation sequence, we can obtain several possible matched substrings with high matching scores. Note that a substring of the user observation sequence can match many substrings of the magnetic field fingerprint data vector due to the similarities in magnetic field data, and a substring of fingerprint magnetic field data vector can match many substrings of the user observation sequence because the user can walk around the same place many times. Another characteristic is that if substring $\vec{a}$ matches $\vec{b}$ with a high score, then another substring $\vec{c}$ whose position is near $\vec{a}$ can also match $\vec{b}$ with a relatively high score. We call this phenomenon *pattern repetition*. In order to select those substrings with high matching scores and reduce pattern repetition (thus reduce the time and space complexity), we introduce a new substring selection algorithm (Algorithm 1).

We use a parameter R in Algorithm 1 to determine the number of pattern repetitions. A large R will reduce pattern repetition, but it can also remove some correctly matched substrings. A small R can lead to both more pattern repetitions and more substrings. In our system we choose $R = 10$, an empirical value which balances the time complexity and the overall performance. Note that we also use several thresholds in the algorithm. The first one, $Threshold_1$, is to avoid the substrings whose scores are too low (thus not similar). The second one, $Threshold_2$, is to avoid the substrings too short to match to increase robustness. The last one, $Threshold_3$, is to avoid the substrings which have too many deletions and insertions during matching. This is because when a user walks along or near a survey path in the same walking direction and the same path as the surveyor, the observations are also order-preserving; we cannot skip too many magnetic field samples during matching since they walk through the same place. We will introduce some empirical values of these thresholds and analyze the system performance under different thresholds in Section 4.

Algorithm 1 Matching Substrings Selection Algorithm

Input: Fingerprint magnetic field data vector $Seq_{fingerprint}$, user observation sequence $Seq_{observe}$, matching score matrix $M_{m\times n}$ obtained from the previous magnetic field sequence matching algorithm

Output: Matched substrings

```
D ← $\mathbf{0}_{m\times n}$
while True do
    Select an entry $(a_1,b_1)$ with the highest score $M(a_1,b_1)$ satisfying $D(a_1,b_1) = 0$; if impossible, break the while loop
    if $M(a_1,b_1) < Threshold_1$ then
        Break
    end if
    Backtracking from $(a_1,b_1)$ to reconstruct the optimum alignment $\langle(\mathbf{B}_{f_{a_1}},\mathbf{B}_{o_{b_1}}),...(\mathbf{B}_{f_{a_t}},\mathbf{B}_{o_{b_t}})\rangle$ between two sequences by using $M_{m\times n}$
    if $t < Threshold_2$ then
        Break
    end if
    for all $1 \le p \le m, 1 \le q \le n$ do
        if $\sqrt{((a_1-p)^2+(b_1-q)^2} < R$ then
            $D(p,q) \leftarrow 1$
        end if
    end for
    Calculate the maximum length of platform of $\langle a_1a_2...a_t\rangle$ as $Plat_{\langle a_1a_2...a_t\rangle} = max\{|p-q|\,|a_p = a_{p+1} = ... = a_q\}$. Similar calculation is also performed for $\langle b_1b_2...b_t\rangle$.
    if $Plat_{\langle a_1a_2...a_t\rangle} \le Threshold_3 \wedge Plat_{\langle b_1b_2...b_t\rangle} \le Threshold_3$ then
        Yield these two substrings as matched substrings
    else
        Discard these two substrings
    end if
end while
```

3.2 Shortest Path Formulation for Path Construction

Since a user may traverse many places, matches between the user observation sequence and one magnetic field fingerprint data vector can cover only a fraction of the whole observation sequence. We need an algorithm to combine matches among different fingerprints of magnetic field data vectors to obtain a path which covers the whole observation sequence.

More specifically, we denote that for each magnetic field fingerprint data vector $Seq_{fingerprint_i}$ we find a set of substrings $S_i = \{s_{i,1}, s_{i,2}, ...s_{i,k}\}$ using Algorithm 1. Each $s_{i,j} = \langle \mathbf{B}_{f_{i,a_{j,1}}}\mathbf{B}_{f_{i,a_{j,2}}},...\mathbf{B}_{f_{i,a_{j,t}}}\rangle(1 \le j \le k)$ matches some part of user observations, say $\langle \mathbf{B}_{o_{b_1}}\mathbf{B}_{o_{b_2}},...\mathbf{B}_{o_{b_t}}\rangle$. We note that each substring $s_{i,j}$ can be also viewed as a corresponding physical route, since each sample can be mapped to a unique physical location. In the following we use the words substring and physical route interchangeably for convenience. Our target is to find an ordered sequence of substrings Seq which can together cover all user observations in order and *physically connected*. Here *physically connected* means that every two consecutive substrings in Seq are head-to-tail connected in physical locations. This constraint is necessary because user movement is continuous.

We approach this subproblem by transforming it into a shortest path problem. We consider all substrings as vertexes in a graph G. For any two vertexes, or two substrings s_i and s_j, there is a directed edge between them if there exists some physical location *LOC* such that both two substrings go through the nearby *LOC*. Say that the nearest sample in s_i to *LOC* is $\mathbf{B}_{f_{i,a_x}}$, and the nearest sample in s_j to *LOC* is $\mathbf{B}_{f_{j,a_y}}$. We assign two weights, b_x and b_y, to the edge, where $\mathbf{B}_{o_{b_x}}$ aligns to $\mathbf{B}_{f_{i,a_x}}$ in s_i and $\mathbf{B}_{o_{b_y}}$ aligns to $\mathbf{B}_{f_{j,a_y}}$ in s_j. The cost

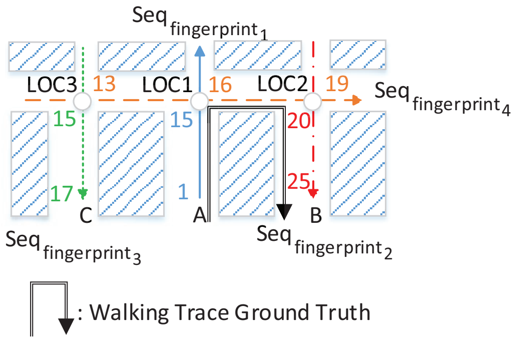
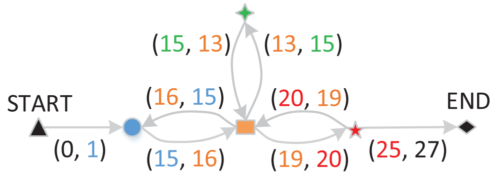


Figure 6: An example of location inference.



Vertex-Substring Correspondances

Vertex in Figure 7	Substring in Figure 6
▲	No matching
●	from A to LOC1
■	from LOC3 to LOC2
✦	from LOC3 to C
★	from LOC2 to B
◆	No matching

Figure 7: The corresponding graph G after transforming the example in Figure 6 into a shortest path problem.

of the edge is the absolute value of the difference between b_x and b_y. Note that b_x can be smaller or bigger than b_y due to the noise in magnetic field samples and alignment, and there may be many edges between two vertexes, as there can be infinity many potential LOCs if the user walks through the same place multiple times. To limit the number of edges between two vertexes, we only select those LOCs which are not so physically close, say at least $0.4m$ in Euclidean distance between any two LOCs.

Finally we need to align the beginning and the end of user observations. We add two new vertexes to G: $START$ connects to each of the substrings matching the beginning of user observations with two weights: 0 and the index of the first matched user observation (also the matched user observation with the smallest index), while END is connected by each of the substrings matching the end of user observations with two weights: the index of the last matched user observation (also the matched user observation with the largest index) and m, the length of the user observation sequence. Note that due to the noise in measurements, the beginning and the end of user observations may not be matched. Therefore, we propose that $START$ connects to the substrings with the smallest index of the matched user observation no greater than R_1, and END connects to the substrings with the largest index of the matched user observation no smaller than $m - R_1$. A large R_1 can tolerate more noise but introduce more edges which increases the time complexity, while a small R_1 requires a stricter matching of the sequences but reduces the number of edges. Here we set $R_1 = 6$ which balances the performance of our system. We require that a *valid path* is a simple path where two consecutive edges, e_1 with weights b_{1x} and b_{1y}, e_2 with weights b_{2x} and b_{2y}, satisfy that:

$$b_{1y} \le b_{2x} \tag{7}$$

This is because we can only align samples in the time order.

We illustrate an example in Figure 6, where each colored arrow represents a survey path with walking direction (and thus one corresponding fingerprint vector). Suppose the user walks along the trace A-$LOC1$-$LOC2$-B. Then after applying the magnetic field sequence matching and selection (Algorithm 1), we can obtain several matched substrings. In Figure 6, each colored number represents the index of user observations matching the fingerprint with the same color. For example, the first 15 samples of user observations $\langle \mathbf{B}_{o_1}\mathbf{B}_{o_2}...\mathbf{B}_{o_{15}}\rangle$ match a part of the blue trace from location A to $LOC1$ (so there are a blue number 1 and a blue number 15 near the blue magnetic field fingerprint data vector $Seq_{fingerprint_1}$, corresponding to the beginning and the end of the matched substring, respectively), and $\langle \mathbf{B}_{o_{13}}\mathbf{B}_{o_{14}}...\mathbf{B}_{o_{19}}\rangle$ matches a part of the fingerprint $Seq_{fingerprint_4}$ of the orange trace from location $LOC3$ to $LOC2$, etc. Note that the index of matched user observations around $LOC1$, $LOC2$ and $LOC3$ can be different due to the noise in measurement.

After transforming this example into a shortest path problem, we obtain a graph G shown in Figure 7 containing 4 vertexes, each of which represents a matched substring (a part of colored arrow), besides $START$ and END. Each vertex corresponds to the same-corresponding-color matched substring in Figure 6. We can see that the corresponding simple path from $START$ to END (which is also the only available simple path) (the term "path" means "simple path" in this paper) has edge weights $(0,1),(15,16),(19,20)$ and $(25,27)$, where 27 is the length of the user observation sequence. We can clearly see that the weights satisfy the condition mentioned before, thus the path is a valid path which is preferable.

Note that the corresponding path of trace A-$LOC1$-$LOC3$-C is not a valid path since $(15,16)$ and $(13,15)$ are weights belonging to two adjacent edges in the corresponding path, and they do not satisfy the constraint (7). Thus this path is not to be considered matching. If we walk along the trace A-$LOC1$-$LOC3$-C in reality, we will align the reversed fingerprint sequence corresponding to the orange arrow instead. Recall that the constraint (7) ensures that samples are aligned in the correct time order. Also, we note that there is not an edge connecting the green vertex (corresponding to a matched substring in the magnetic field fingerprint data vector $Seq_{fingerprint_3}$) to vertex END, since in the corresponding substring, the largest index of the matched user observation is only 17, which is smaller than $27 - R_1 = 21$ (thus not close to the end of the user observation vector). So the user does not walk in that direction the green arrow shows.

We denote the cost of a path as the sum of the weights of all edges in the path. For instance, in the above example, the cost of the valid path from *START* to *END* is $|0-1|+|15-16|+|19-20|+|25-27|=5$. Then all *valid paths* from *START* to *END* can be viewed as possible user trajectories, and the *minimum cost valid path* corresponds to the estimated user trajectory where substrings can match as many user observations in the correct order as possible. In this context the minimum cost path can be viewed as the shortest path from *START* to *END* interchangeably, if we view the cost of an edge as the distance.

3.3 Effective Solution to the Modified Shortest Path Problem

We denote N and M as the number of vertexes and edges in G, respectively. To solve the above shortest path problem, we need to convert the constraint (7) into a more usable one. If we add a new edge e_0 to G which only connects to *START* with both of weights 0, and add another new edge e_{end} to G which only connects with *END* with both of weights *INT_MAX* (a sufficiently large integer), and view edges in G as new “vertexes” in $\hat{G}$, and convert the weights of edges into weights of “vertexes” in $\hat{G}$, and add new “edges” in $\hat{G}$ between two adjacent “vertexes” e_1 and e_2 (here adjacent means that the corresponding edges of the “vertexes” are head-to-tail connected in G) if a *valid path* passes through those “vertexes” (or equivalently, $b_{1y} \leq b_{2x}$), then the original subproblem is equivalent to finding the shortest path (w.r.t. minimum sum of node weights instead of edge weights) from the “vertex” corresponding to e_0 to the “vertex” corresponding to e_{end} in $\hat{G}$. However a simple implementation of the above using traditional Dijkstra’s algorithm together with a min-priority queue can run in $O(N^3+N^2 logN)$ in the worst case, which is computationally expensive, not to mention there may be a large number of “edges” which consumes much memory to store. To efficiently reduce both time and space complexity, in reality we do not construct such $\hat{G}$. Instead we maintain the shortest path from e_0 to other edges separately. In each step we extract the unexplored edge with the minimum distance and update its neighbors’ distances if possible. The pseudocode is as follows (Algorithm 2), which is adapted from the standard Dijkstra’s algorithm.

In Algorithm 2 we use $Out(e)$ to represent the set of outgoing edges of e. We note a key observation that in Algorithm 2 we remove some edges from $Out(e_1)$ after relaxation to further reduce the time complexity (line 20). This is because for each edge e_2, if its *dist* value is updated from e_1 at the first time, then after that its *dist* value cannot be updated from other e_3 connecting to e_2. Dijkstra’s algorithm is a greedy algorithm which updates distances in an increasing manner, thus for any e_3 connecting to e_2 that updates *dist* later than e_1, if $b_{3y} \leq b_{2x}$, then Dijkstra’s algorithm guarantees that $dist[e_3] \geq dist[e_1]$; if $b_{3y} > b_{2x}$, then e_3 cannot further update $dist[e_2]$ due to the constraint in line 16 in Algorithm 2. In this way we can see that each edge in G can be updated at most once, and the overall time complexity can be reduced to $O(MlogM)$, or roughly $O(N^2 logN)$ after using our algorithm. Finally we can backtrack the shortest path from *END* to *START* (line 24-29) and obtain the estimated user trajectory by mapping the matched parts of the magnetic field fingerprint data vector back to the real world coordinates.

Algorithm 2 Finding Minimum Cost Path Algorithm

Input: Constructed graph G.
Output: Minimum cost path from *START* to *END*

```
 1: Add a new edge $e_0$ to $G$ which only connects to $START$ with both of weights 0
 2: Add another new edge $e_{end}$ to $G$ which only connects with $END$ with both of weights $INT\_MAX$, a sufficiently large integer
 3: $dist[e_0] \leftarrow 0$
 4: for edge $e \in G - \{e_0\}$ do
 5:     $dist[e] \leftarrow \infty$
 6: end for
 7: for edge $e \in G$ do
 8:     $back[e] \leftarrow NULL$
 9: end for
10: $Q \leftarrow$ all edges in G
11: while $Q \neq \emptyset$ do
12:     Extract $e_1 \in Q$ with the minimum distance $dist[e_1]$
13:     if $e_1 == e_{end}$ then
14:         Break
15:     end if
16:     for $e_2 \in Out(e_1), b_{1y} \leq b_{2x}$ do
17:         if $dist[e_2] > dist[e_1] + cost(e_2)$ then
18:             $dist[e_2] \leftarrow dist[e_1] + cost(e_2)$
19:             $back[e_2] = e_1$
20:             remove $e_2$ from $Out(e_1)$
21:         end if
22:     end for
23: end while
24: $Backtrack \leftarrow back[e_{end}]$
25: $MinCostPath \leftarrow \emptyset$
26: while $Backtrack \neq e_0$ do
27:     $MinCostPath \leftarrow MinCostPath \cup \{Backtrack\}$
28:     $Backtrack \leftarrow back[Backtrack]$
29: end while
```

In our implementation we aggregate 20 magnetic field observations (roughly corresponds to several steps’ observations) together after PAA and perform magnetic field sequence matching only on these aggregations instead of all the user observations, and concatenate these results to yield the final localization results in order to obtain robust results much faster. We also aggregate some first few samples after starting localization to obtain some initially estimated locations.

It is also worth noting that provided with Wi-Fi or iBeacon fingerprints, our algorithm can be modified to fuse with that signal easily. Say that at some time we collect both magnetic field and Wi-Fi signals. After comparing the RSSI values with the fingerprint, we know some potential locations of the user at that time. For the aggregation including that magnetic field observation, we can match it only with those magnetic field fingerprint data vectors which are close to those potential locations, instead of matching it with all fingerprint data vectors. This can help reduce the overall computation time and increase localization accuracy.

3.4 Time Complexity

The time complexity of the magnetic field sequence matching and selection (Algorithm 1) is

$$O\left(mn\left(1+\frac{1}{R^2}\right)\right), \tag{8}$$

where m is the length of user observation sequence, n is the maximum length of magnetic field fingerprint data vector and R is the parameter mentioned in Algorithm 1, since the time complexities for matching and substring selection are $O\left(mn\right)$ and $O\left(\frac{mn}{R^2}\right)$, respectively. Note that after substring selection algorithm we may select at most $O\left(\frac{mn}{R^2}\right)$ substrings.

After the above procedures, we convert the localization problem into a shortest path problem in $O\left(N^2\right)$, which can be solved in roughly $O(N^2 logN)$ by using Algorithm 2, where N is the total number of matched magnetic field substrings. So the overall time complexity of Magil is

$$O\left(\frac{m^2n^2}{R^4}\log\left(\frac{mn}{R^2}\right)\right). \tag{9}$$

4 Experimental Evaluation

In this section we evaluate the accuracy and efficiency of Magil through experiments. In Section 4.1, we introduce experimental settings and comparison schemes. In Section 4.2, we present illustrative experimental results.

4.1 Experimental Settings & Comparison Schemes

We implement our system on different mobile phones, including Sony Xperia X2, Samsung Galaxy S4 and LG Nexus 5. We use Sony Xperia X2 to perform site survey tasks. They all run Android operating systems later than 4.0 and are equipped with magnetometers. We implement the server on a Dell PC with a 3.6GHz processor and 16G RAM, running Windows 8.1. During localization, mobile client performs continuous background inertial sensor (magnetometer) sampling. Sampling frequency is set to be 25Hz. From the experiment we observe that only a few steps of walking traces are sufficient to localize the user. Therefore, in practice we only need to keep a small buffer for the walking trace, and discard the earlier part. Unless otherwise stated, we use the following thresholds: $Threshold_1 = 9$, $Threshold_2 = 15$ and $Threshold_3 = 10$. Later we will analyze the performance of our system under different thresholds.

In our experiments, we compare Magil with another two state-of-the-art magnetic field based localization algorithms, and the detailed algorithms and parameter settings are referred to their works.

- MaLoc [28]: which utilizes a particle filter together with INS to measure the user location. Candidate locations with the best magnetic field matching are selected, then particle filter further reduces the weights of incorrect locations and performs localization. Specifically, we implement MaLoc by setting the initial number of particles to be 2,000.
- Magicol [19]: which measures not only magnetic field values but also relative trends of magnetic field changes when a user is walking through indoor environment. By utilizing Dynamic Time Warping (DTW) and particle filter, Magicol maps target to locations with the best trend matching while filtering away incorrect locations. We implement Magicol by setting the initial number of particles to be 3,000 as suggested in their work.

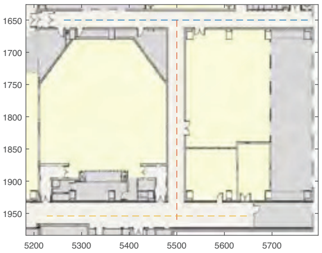


Figure 8: Survey paths (dashed lines) in the university corridors.

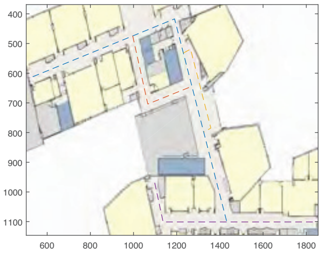


Figure 9: Survey paths (dashed lines) in a whole floor.

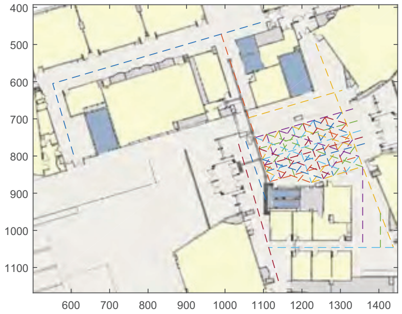


Figure 10: Survey paths (dashed lines) in a large indoor environment.

We have conducted extensive experiments to validate our localization algorithm in three typical indoor environments: a small indoor environment containing three short corridors ($1,091\ m^2$, see Figure 8), a larger indoor environment (a whole floor) containing more long corridors ($5,909\ m^2$, see Figure 9) and a large indoor environment containing open space and several corridors ($4,133\ m^2$, see Figure 10). In these figures we also show the survey paths. Note that in large open space, survey paths should cover various walking directions in order to achieve better localization accuracy.

The performance metrics are presented as belows. To obtain the ground truth of walking trajectories, we set many landmarks (say, the doors and pillars) and measure their locations in advance. During the experiment, users record the time when they pass by those landmarks to evaluate the real-time location estimation. Localization error is calculated as the Euclidean distance between ground truths and estimated locations in historical trajectories, while the real-time localization error is calculated as localization error at each step. The mean localization error is calculated as the mean of all the estimation errors along a trace.

4.2 Illustrative Results

Figures 11, 12 and 13 show the estimated walking traces compared with ground truths in corridors, a whole floor and

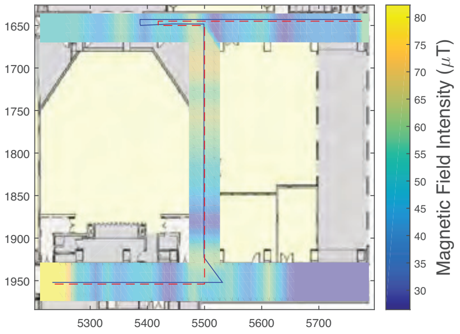


Figure 11: The estimated walking trajectory of Magil in the corridors. The red dot line is the ground truth and the blue line is the estimated trajectory.

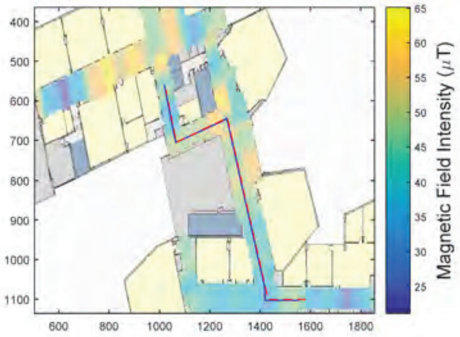


Figure 12: The estimated walking trajectory of Magil in the whole floor. The red dot line is the ground truth and the blue line is the estimated trajectory.

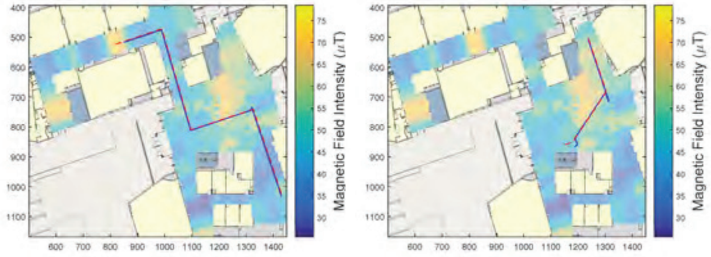


Figure 13: Two estimated walking trajectories of Magil in the large indoor environment. The red dot lines are the ground truths and the blue lines are the estimated trajectories.

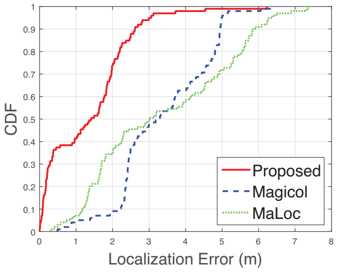


Figure 14: CDF of indoor localization error in the corridors.

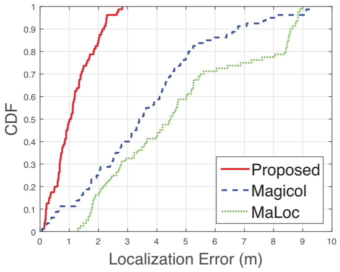


Figure 15: CDF of indoor localization error in the whole floor.

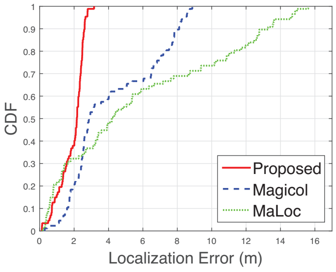


Figure 16: CDF of indoor localization error in the large indoor environment (the left trajectory in Figure 13).

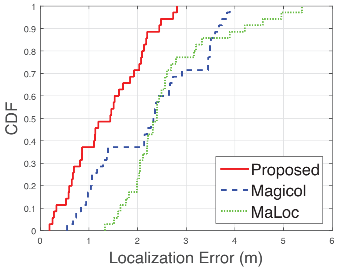


Figure 17: CDF of indoor localization error in the large indoor environment (the right trajectory in Figure 13).

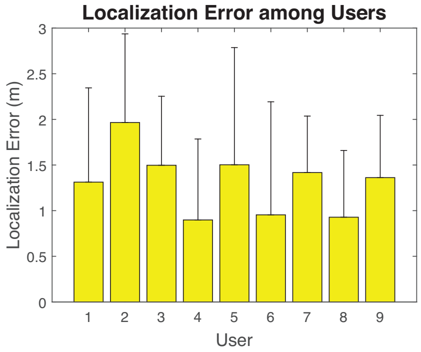


Figure 18: Mean localization error (with error bars) among users.

large indoor environment, respectively. We can see that estimations highly match ground truths.

We also compare Magil with other magnetic field based localization systems such as Magicol and MaLoc. From Figures 14, 15, 16 and 17 we can clearly see that our system works well in both narrow spaces (such as corridors) and large open spaces (as shown in Figure 13), and outperforms other state-of-the-arts with less error and smaller deviation.

In the experiment we also ask another eight subjects additional to user 1 to walk along the ground truth trajectory shown in Figure 11 to validate Magil's performance. In sum, there are seven males and two females, and their heights and weights vary from 1.68 *m* to 1.87 *m*, and from 45 *kg* to 80 *kg*, respectively. Figure 18 shows the mean error and standard deviation for each user. We can observe that Magil achieves high accuracy in localization among different users.

Figure 19 shows the average real-time localization error among different systems. 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 trajectories afterwards. We can see that our system can converge quickly and yield stable localization results, but other particle filter based systems cannot converge well. Note that traditional particle filter requires an explicit or accurate initial position of the target, and heavily relies on accurate estimation of user movement such as steps, walking distances and directions. Otherwise, given a coarse initial position, it can work with large localization error, as shown in Figure 20. S-

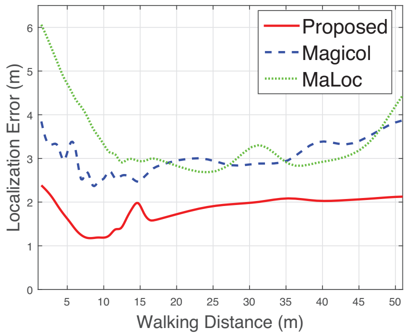


Figure 19: Real-time localization error versus user walking distances.

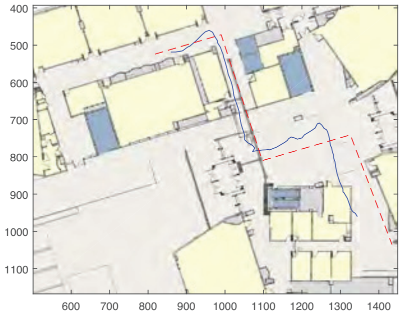


Figure 20: Particle filter based magnetic field localization results in the large indoor environment. The red dot line is the ground truth and the blue line is the estimation.

ince particle filter tries to capture all possible step length and direction error information by means of particles, it cannot converge quickly in large open spaces or complex environments [9], which still applies in magnetic field fusion. As the number of dimensions of magnetic field observations is less than Wi-Fi or Bluetooth, the dispersion of particles can be severe in large open spaces. Magil elegantly avoids this dilemma by matching magnetic field observations instead of estimating actual user movements which is error-prone [2]. Moreover, our system models user movements as transitions between several possible trajectories, which is computationally cheap.

We are also interested in the system performance under different user moving speeds. In this experiment a user walks along the trajectory shown in Figure 11 at three different speeds: walking in fast (roughly 1.8 m/s), normal (roughly 1.4 m/s) and slow (roughly 1.0 m/s) speeds. Figure 21 illustrates that the overall performance is not influenced much under different speeds. This is because our modified Smith-Waterman algorithm (Algorithm 1) can align the shapes of magnetic field observations, which are not altered under different walking speeds along the same trajectory. Different walking speeds can only affect the length but not the shape of the magnetic field observation sequence, which indicates the effectiveness of Magil.

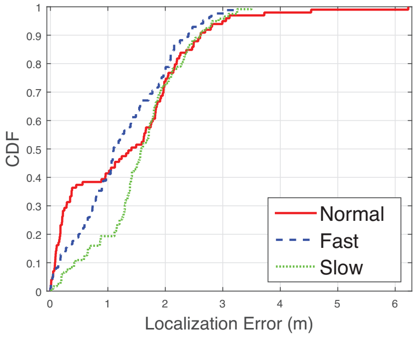


Figure 21: CDFs of the localization error versus user walking speeds.

Note that we use several empirical thresholds in our system setting. However, different thresholds may also influence the system performance.

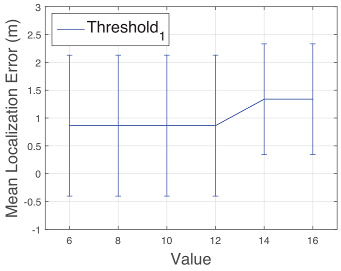


Figure 22: Mean localization error versus $Threshold_1$.

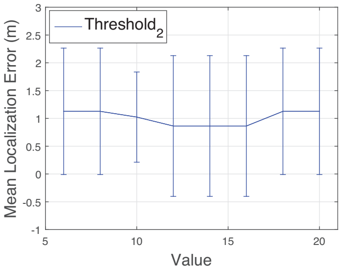


Figure 23: Mean localization error versus $Threshold_2$.

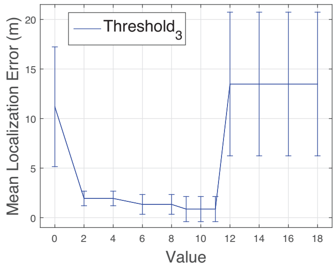


Figure 24: Mean localization error versus $Threshold_3$.

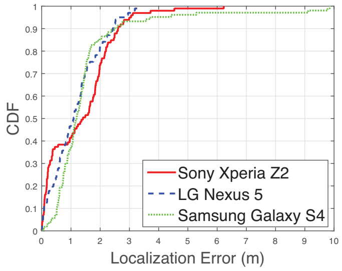


Figure 25: CDF of localization error among different smartphones.

Figures 22, 23 and 24 show the mean localization error and corresponding standard deviations under different thresholds. In this experiment the user walks along the trajectory shown in Figure 11. We can see that a too large $Threshold_1$ or $Threshold_2$ will incur large error, since it will skip more possible substrings, and small $Threshold_1$ or $Threshold_2$ can introduce more substrings which have too lower matching scores or are too short, respectively, under which the system performance degrades. A large $Threshold_3$ will allow more insertions and deletions during matching, which can lead to more wrong matchings, while a small $Threshold_3$ can also make the localization accuracy lower due to the stricter matching process. By setting those empirical thresholds mentioned above, we can maintain both good localization accuracy and robustness of Magil.

Since the user may use different devices other than the one used for magnetic field fingerprint collection, we also evaluate Magil's efficiency by using different smartphones. Figure 25 shows the CDFs of localization error among different smartphones. We can see that our system's performance is good and stable, since we use Equations (5) and (6) to calculate similarity between observations instead of using the magnetic field fingerprint itself directly to avoid calibrating different magnetometers.

Table 2: Computation time comparison between different magnetic field localization algorithms.

System	Proposed	Magicol	MaLoc
Time (ms)	498.1	1,781.4	700.7

Next we investigate the cost of Magil on a mobile phone (Sony Xperia Z2). For a responsive and user-friendly pedestrian tracking system, the typical processing time for each step during localization should be less than 600 *ms*. Any application with requirements exceeding this limit can result in a bad user experience, which is not preferable. Table 2 shows the execution time of various systems on the corridor's trajectory as shown in Figure 11. Time is calculated as the average time of processing a single step over the trajectory. We can see that Magil outperforms other systems in online execution time, whilst obtaining the lowest localization error.

Finally we measure the average energy consumption of our system using Sony Xperia Z2. The average current is 426.70 *mA*, 768.15 *mA* and 255.97 *mA* for Magil application, continuous Wi-Fi scanning (with screen display on) and screen energy consumption, respectively. We conclude that Magil is much more energy efficient than continuous Wi-Fi scanning.

5 Discussions & Future Work

For practical deployment of Magil, we discuss the following aspects for the completeness of our work.

- *Necessity of User Walking*: One requirement of our work is that a user needs to walk for some distance before Magil can obtain enough magnetic field data to analyze the pattern and localize the user. This is reasonable because if a user stands still, his/her magnetic field observation is almost fixed and stable, and we cannot achieve localization without offline device magnetometer calibration. We leave as future work investigating methods to infer location without asking the user to walk.

- *Survey Path Planing*: Note that we require survey paths in large open space to cover various directions, as shown in Figure 10. This is because Magil performs sequence matching on geomagnetic fingerprints, which also limits our localization results to several predefined survey paths. Since the magnetic field is omnipresent in the whole space, our ongoing work is to study effective methods to localize users based on the entire geomagnetic field in the site.

6 Related Work

We briefly review related works on mobile localization in this section.

6.1 RF-based Localization

Traditional fingerprint-based indoor localization, pioneered by Wi-Fi fingerprinting, has been extensively studied in the past decade [7]. RADAR [1], Horus [31] and WiGEM [6] have been proposed to achieve high accuracy for Wi-Fi localization. However, like many other RF signals, Wi-Fi signals suffer from multipath and interference effects. Other emerging signals such as channel state information (CSI) [25], Bluetooth (or iBeacon) [4] or FM [30] rely on deploying specialized infrastructures, which may not be feasible in pervasive computing. The popular signals explored for localization are Wi-Fi RSSI [1, 17], channel impulse response (CIR) [15], channel state information (CSI) [25], vision [29], visible light [24], sound [12, 13] and iBeacon [4, 32]. While the systems reported are impressive, they require costly or special infrastructure deployments. Furthermore, as some of the signals are not stable due to multipath or interference effect, this adversely affects the localization accuracy. The systems also involve time-consuming and laborious survey processes where signal vectors have to be collected at length by standing at different fixed points in the area. Our scheme, Magil, is more general to apply in different environments.

6.2 Geomagnetism-based Localization

Infrastructure-based magnetic localization has been extensively studied [3, 5, 16]. These early studies focus on using robots or special devices for magnetic field collection. Different from these works, Magil focuses on mobile localization based on smartphone sensors, which is more general to apply in mobile localization.

Some signal heuristics have also been proposed for geomagnetism-based localization. Unloc [23] proposes sparse magnetic disturbances as landmarks for indoor localization. Different from their work, we use the magnetic field to build an indoor magnetic field signal map. FOLLOWME [20] leverages walking patterns of earlier travelers to navigate the following users. Inspired by the observation that geo-magnetism changes around the pillars or gates, LocateMe [22] maps the target location to these landmarks when a similar trend of signal change is measured. The above works usually focus on narrow corridors where pedestrian walking patterns are constrained. Furthermore, the particles in spacious indoor areas may diverge significantly. Magil does not rely on the follower mode, and is more general to apply in complex environments.

6.3 Fusion-based Localization

Advanced fusion of fingerprint signals and pedometer has been recently studied in many works [8,9,26]. MapCraft [26] and the work in [18] both utilize the conditional random field (CRF) to localize the users. In their CRF, sequential motion and sensor readings are fed to the graphical model for location estimation. Magil, however, tries to model the problem into a signal matching problem which is different from their approaches, and achieves good localization accuracy without heavily relying on user walking direction estimations and error-prone step counters [2, 33].

Particle-filter-based fusion has been widely studied in recent years for magnetic field localization. Magicol [19] considers a two-way particle filter to improve the fusion of Wi-Fi fingerprint and magnetic fields. MaLoc [27, 28] implements a novel augmented particle filter to approach motion estimation errors. The work in [9] approximates the floor map by the connected line segments. All these works require a meticulously tuned pedometer to estimate actual user movements, including walking distances and directions. Unlike these works, Magil does not use the error-prone pedometers [2,33]. Instead, Magil leverages the fact that if a user passes the same place, the measured magnetic field signal pattern will be similar, and transforms the localization problem into a signal matching and path construction problem.

7 Conclusion

In this paper, we have proposed and studied Magil, a novel indoor localization scheme based on only geomagnetic field. Magil models the localization problem as a signal matching and path construction problem. After finding several fingerprint path segments with fingerprint variation best matched with the target observations, Magil employs a modified shortest path algorithm to select those matched parts and obtain the target locations.

Compared with the traditional particle filter approach, Magil is much more implementable (computationally efficient) and robust in accuracy when deployed on smartphones. It converges much faster, adapts to different users and achieves significantly better localization accuracy. Extensive experimental studies in our university campus have shown that Magil outperforms state-of-the-art schemes by a large margin (cutting localization error by more than 30% on average). Most importantly, our scheme does not require any pedometer information, and hence is more robust towards different user behaviors.

8 Acknowledgments

We would like to thank the shepherd and anonymous reviewers for their valuable comments and feedbacks. This work was supported, in part, by Guangdong Provincial Department of Science and Technology (GDST16EG04), Guangdong-Hong Kong joint innovation project (2016A050503024), and Natural Science Foundation of Guangdong Province (2014A030313154).

9 References

[1] P. Bahl and V. N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *Proc. IEEE INFOCOM*, volume 2, pages 775–784, 2000.

[2] A. Brajdic and R. Harle. Walk detection and step counting on unconstrained smartphones. In *Proc ACM UbiComp*, pages 225–234, 2013.

[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*, pages 141–154, New York, NY, USA, 2011.

[4] R. Faragher and R. Harle. Location fingerprinting with Bluetooth low energy beacons. *IEEE JSAC*, 33(11):2418–2428, Nov 2015.

[5] M. Frassl, M. Angermann, M. Lichtenstern, P. Robertson, B. Julian, and M. Doniec. Magnetic maps of indoor environments for precise localization of legged and non-legged locomotion. In *Proc. IEEE/RSJ IROS*, pages 913–920, 2013.

[6] A. Goswami, L. E. Ortiz, and S. R. Das. WiGEM: A learning-based approach for indoor localization. In *Proc. ACM CoNEXT*, pages 3:1–3:12, 2011.

[7] S. He and S.-H. G. Chan. Wi-Fi fingerprint-based indoor positioning: Recent advances and comparisons. *IEEE Communications Surveys & Tutorials*, 18(1):466–490, 2016.

[8] 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*, pages 897–908, 2015.

[9] 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*, pages 147–158, 2014.

[10] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra. Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and information Systems*, 3(3):263–286, 2001.

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

[12] H. Liu, Y. Gan, J. Yang, S. Sidhom, Y. Wang, Y. Chen, and F. Ye. Push the limit of WiFi based localization for smartphones. In *Proc. ACM MobiCom*, pages 305–316, Sept. 2012.

[13] H. Liu, J. Yang, S. Sidhom, Y. Wang, Y. Chen, and F. Ye. Accurate WiFi based localization for smartphones using peer assistance. *IEEE Trans. Mobile Computing*, 13(10), Oct 2014.

[14] J. Ma, J. Qian, P. Li, R. Ying, and P. Liu. Indoor localization based on magnetic anomalies and pedestrian dead reckoning'. In *Proc. ION GNSS+*, pages 1033 – 1038, 2013.

[15] A. T. Mariakakis, S. Sen, J. Lee, and K.-H. Kim. SAIL: Single access point-based indoor localization. In *Proc. ACM MobiSys*, pages 315–328, 2014.

[16] A. Markham and N. Trigoni. Magneto-inductive networked rescue system (MINERS): Taking sensor networks underground. In *Proc. ACM/IEEE IPSN*, pages 317–328, 2012.

[17] S. Nirjon, J. Liu, G. DeJean, B. Priyantha, Y. Jin, and T. Hart. COIN-GPS: Indoor localization from direct GPS receiving. In *Proc. ACM MobiSys*, pages 301–314, 2014.

[18] J.-g. Park and S. Teller. Motion compatibility for indoor localization. Technical report, MIT, 2014.

[19] 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*, 33(7):1443–1457, July 2015.

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

[21] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197, 1981.

[22] K. P. Subbu, B. Gozick, and R. Dantu. LocateMe: Magnetic-fields-based indoor localization using smartphones. *ACM Trans. Intell. Syst. Technol.*, 4(4):73:1–73:27, Oct. 2013.

[23] 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*, pages 197–210, 2012.

[24] Z. Wang, Z. Yang, J. Zhang, C. Huang, and Q. Zhang. Mobile devices can afford: Light-weight indoor positioning with visible light. In *Proc. ACM MobiSys*, May 2015.

[25] K. Wu, J. Xiao, Y. Yi, D. Chen, X. Luo, and L. M. Ni. CSI-based indoor localization. *IEEE Trans. Parallel & Distributed Systems*, 24(7):1300–1309, Jul. 2013.

[26] Z. Xiao, H. Wen, A. Markham, and N. Trigoni. Lightweight map matching for indoor localisation using conditional random fields. In *Proc. ACM/IEEE IPSN*, pages 131–142, 2014.

[27] H. Xie, T. Gu, X. Tao, H. Ye, and J. Lu. A reliability-augmented particle filter for magnetic fingerprinting based indoor localization on smartphone. *IEEE Trans. Mobile Computing*, 15(8):1877–1892, Aug 2016.

[28] 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*, pages 243–253, 2014.

[29] H. Xu, Z. Yang, Z. Zhou, L. Shangguan, K. Yi, and Y. Liu. Enhancing WiFi-based localization with visual clues. In *Proc. ACM UbiComp*, pages 963–974, 2015.

[30] S. Yoon, K. Lee, and I. Rhee. FM-based indoor localization via automatic fingerprint DB construction and matching. In *Proc. ACM MobiSys*, pages 207–220, 2013.

[31] M. Youssef and A. Agrawala. The Horus WLAN location determination system. In *Proc. ACM MobiSys*, pages 205–218, 2005.

[32] X. Zhao, Z. Xiao, A. Markham, N. Trigoni, and Y. Ren. Does BTLE measure up against WiFi? A comparison of indoor location performance. In *Proc. European Wireless Conference*, pages 1–6, May 2014.

[33] P. Zhou, M. Li, and G. Shen. Use it free: Instantly knowing your phone attitude. In *Proc. ACM MobiCom*, pages 605–616. ACM, 2014.