WaveLoc: Wavelet Signatures for Ubiquitous Localization

Swati Rallapalli*, Wei Dong†, Lili Qiu+†, Yin Zhang++†
IBM Research*
The University of Texas at Austin†

Abstract—Always-on localization is an important problem for a lot of context sensitive mobile computing applications. This paper proposes WaveLoc – that effectively uses measurements from a trajectory as its fingerprint for localization. Traditionally, signatures from single-points have been leveraged for localization although trajectory signatures offer a lot more information. One reason for this is that it is significantly more challenging to match measurements across trajectories than from single points. To tackle this challenge, WaveLoc divides the problem into the following two steps: (i) identify a user’s current trajectory by matching its measurements with those in the training traces (trajectory matching) and (ii) localize the user on the trajectory (localization). The core requirement of both steps is an accurate and robust algorithm to match two time-series that may contain significant noise and perturbation due to differences in speed, mobility, devices, and environment. WaveLoc addresses these by performing multi-level wavelet analysis of the measurements and applying an enhanced Dynamic Time Warping (DTW) alignment to the wavelet coefficients. Using both indoor and outdoor experiments, and a prototype implementation we demonstrate that WaveLoc is accurate and power efficient.

I. INTRODUCTION

Motivation: Localization has been extensively studied due to its important role in many applications from product guides in indoor retail stores to locating points of interest outdoors. Despite much research on localization, significant challenges remain. First, most existing works use Wi-Fi for localization. While the popularity of Wi-Fi has increased rapidly, it is still not always available. For example, in some developing countries/areas and certain stores, Wi-Fi is not deployed. Even when it is available, its density may not be high enough to allow accurate localization. Second, while it is natural to localize a mobile device based on measurements collected from an individual point, such point-based schemes yield limited accuracy due to significant aliasing. For example, existing Wi-Fi based schemes use received signal strength (RSS) or channel state information (CSI) from nearby access points (APs) for localization. CSI is more fine-grained than RSS and allows us to derive the signal-to-noise ratio (SNR) across OFDM subcarriers. However, for both RSS and CSI, it is common to have multiple locations that see similar values.

Magnetic field can also serve as a signature for localization [13, 39, 37]. The advantage over Wi-Fi is that this signature is ubiquitous. However, the magnetic field readings from different locations can be similar. Aliasing is a more serious issue with magnetic field and cannot be reduced by deploying more infrastructure.

Approach: Interestingly, measurements over a trajectory are distinguishable enough to be used as signatures. Therefore, WaveLoc explores the potential of using magnetic field or Wi-Fi measurements from trajectories as location fingerprints. However, this is very challenging as the trajectory signatures vary with speed, device configuration, variations in gait, etc.

WaveLoc addresses this and achieves high accuracy by breaking-down the localization problem as follows: (i) match a new measurement trace without location information with one of the training traces (i.e., trajectory matching – note that this involves only identifying the high level trajectory of a user e.g., which aisle is the user walking in? This step does not require identification of exact location, i.e., the x, y co-ordinate of the user), and (ii) narrow down to the exact location based on the matched point on the training trace (i.e., localization). This is because (i) measurements collected over a trajectory offer more information and trajectory matching is likely to be more accurate than point matching. Unlike the case of point-matching where aliasing can be very severe, i.e., many points can have similar fingerprint values, it is not very common to have multiple trajectories with similar fingerprint values. (ii) Once the trajectory in known, localizing a point within the trajectory is much easier than localizing a point in an entire area. Therefore, by leveraging more information and decomposing the problem into the two easier tasks, WaveLoc improves localization accuracy.

While it is evident that WaveLoc requires more information than point based localization, it is not necessarily more expensive to collect such data. Since people walk and drive in a few common trajectories (e.g., corridors in office building, aisles in grocery stores, and lanes on roads), we can collect measurements from users’ regular trajectories via crowd-sourcing [23] or compliant-walking proposed in [32].

Despite dividing the problem into two simpler steps, several significant challenges remain. First, it is hard to exactly match testing and training traces even if they are collected from the same trajectory, due to differences in speeds, trajectories, devices, environments, and noise. Second, localizing a point on the trajectory requires a highly accurate alignment between the testing and training traces, which is also challenging due to the differences in the testing and training data. Even a small amount of noise may lead to incorrect alignments. The core requirement of both steps is an accurate and robust algorithm to match two time-series that may be noisy and subject to various perturbations arising from different mobility, devices, and environmental factors. To find an alignment between the two time-series, WaveLoc applies well known Dynamic Time
Warping (DTW) [20], widely used in signal processing, speech recognition, and signature recognition. However, to improve robustness against different speeds and noise, WaveLoc: (i) performs multi-level wavelet decomposition on the measurement data and searches for the wavelet level that minimizes the distance between the two time-series, (ii) clusters the training traces from the same trajectory to capture different patterns that a trajectory may exhibit, (iii) enables continuous localization by combining trajectory matching and a novel procedure to detect the end point of a training segment, (iv) uses a number of techniques to address the singularity problem common in traditional DTW, and uses incremental DTW to reduce the computation cost.

Contributions: Our contributions are three-fold:

- Using real measurements, we show the potential of trajectory based localization (§ III).
- We propose WaveLoc that leverages multi-level wavelet approaches for trajectory matching and point localization to achieve robustness against noise and differences in mobility and speed. WaveLoc also combines them to achieve accurate localization over a full path with multiple segments (§ IV).
- Using walking and regular driving experiments with random lane changes, different speeds, we demonstrate that WaveLoc is promising. It significantly improves the localization accuracy indoors and yields considerable power saving over GPS outdoors. It is robust to the differences in devices, cars, speeds, and time-of-day (§ V).

II. RELATED WORK

Localization has been well studied over the past few years and in the interest of space we only aim to cover some of the most representative and related works here.

Point-based localization: Most works use point-based localization. They differ in the types of measurements and analyses. RADAR [2] uses signal strength of measurements for triangulation, Cricket [26] uses time difference of arrival between radio and ultra sound signals, VORBA [21] uses angle of arrival, Horus [44] uses maximum likelihood estimation using RSS measurements. [24], [45] use CSI for location distinction (i.e., determine if a node has moved from its position). PinLoc [30] and DeepFi [41] use CSI signatures from specific spots for localization. EZ [3] minimizes profiling effort by using measurements from unknown locations along with a few from known locations to extract propagation constraints and solves a non-linear optimization to obtain locations that best fit these constraints. CUPID [22] uses the direct path signal to estimate the distance and angle of arrival to mitigate multipath effect. LiFS [43] removes the need of profiling by creating a fingerprint space where the fingerprints are distributed based on their pairwise physical distance estimated from user mobility. Array-Track [42] uses small movements to suppress multipath effects in MIMO scenarios to enhance the accuracy, but requires modified APs. In general, aliasing limits the accuracy of point-based localization. Other interesting works like [28] (complementary to WaveLoc problem formulation) perform device-free localization, i.e., how to localize a user based on how she disrupts the WiFi signal map of a region.

Leveraging trajectories: There are some works that use trajectory information. Zee [27] uses a particle filter based approach. It eliminates improbable routes based on the map, so it can localize the user only after she walks a long enough trajectory and takes a few turns. Another class of works leverages Hidden Markov Models (HMMs). For example, WheelLoc [40] estimates the velocity and direction of a user using accelerometer and extracts cell tower IDs. Then it matches them to the most likely road segments on a map using Hidden Markov Model (HMM). [10], [12], [35], [36] are other works that also leverage HMMs in different ways to infer trajectories. While all these are very interesting approaches, they rely on a small number of paths sharing the same turning pattern or cell tower or relatively dense GPS locations. This may not always hold. Complementary to these works, we show that measurements along even a straight path without well known landmarks (e.g., cell tower IDs) already offer locally unique signatures for us to perform localization. This allows us to go beyond trajectory matching (i.e., figuring the high-level trajectory of a user, e.g., road segment that user is on, or aisle that user is walking in) and can perform fine-grained localization (i.e., determine the x, y co-ordinate). [15] presents work that leverages radial basis function fitting over RSSI trajectory data indoors. In comparison, we present a scheme that leverages multiple fingerprints (RSSI/magnetic field) indoors or outdoors, and conduct extensive evaluation under varying speeds etc. to demonstrate the effectiveness of our technique. Another class of approaches (e.g., CompAcc [6], UnLoc [39], SLAM [8], Walkie-Markie [51]) uses inertial sensors (accelerometer, compass) to determine the direction and distance of movement from a known location or landmark. WaveLoc complements these and can benefit from inertial sensors as discussed in § VI.

Magnetic field: [39] and [57] use magnetic field to identify landmarks. Since magnetic field is only leveraged from a few points, there can be significant aliases. [13] is a start-up that uses indoor magnetic field anomalies for indoor localization. Their technology is not disclosed to the public. [6] leverages magnetic field and compass distortions for indoor localization. It is a point based scheme, and also requires the device to be worn in a certain direction which may not always be feasible. [38] proposes another point-based localization scheme that
leverages magnetic field and faces aliasing issues.\textsuperscript{34} uses the magnetic field for indoor trajectory matching.\textsuperscript{33} uses the magnetic field for last-mile navigation.\textsuperscript{32} fuses Wi-Fi signatures with magnetic field to achieve accurate localization. WaveLoc advances state-of-the-art in the following ways: (i) performing point localization (beyond just trajectory matching), (ii) enhancing efficiency and accuracy using multi-level wavelet analysis, and (iii) applying to both Wi-Fi and magnetic field, and showing its benefit both indoors and outdoors.

**Summary:** WaveLoc leverages measurements at multiple points along a trajectory to reduce aliases and improve localization accuracy. It is general and can support different speeds, trajectories, and types of measurements. It complements the existing point-based localization by taking holistic view of measurements over a trajectory. Meanwhile, it can benefit from recent advances in point-based localization by incorporating any improvements in proximity metrics when computing DTW distance. Finally, magnetic field can be very useful for localization outdoors e.g., in downtown canyons buildings can block GPS signals, but they also interfere with earth’s magnetic field in a fashion very unique to various locations. But the potential of magnetic field in augmenting GPS outdoors, is so far not very well understood and we take a step towards bridging that gap.

### III. Motivation

We first introduce various types of location signatures. We then motivate the need for trajectory based localization by identifying the limitations of point based localization and showing the advantages of trajectory based localization.

#### A. Location Signatures

**Magnetic field:** The earth’s magnetic field is a ubiquitous signature that can also potentially be used for localization. The magnitude of the field from the earth alone varies between $0.3 - 0.6$ Gauss ($30 - 60 \mu T$).\textsuperscript{30} However, the earth’s magnetic field can be affected by many other factors, such as speakers, electric lines, appliances, etc. The impact of these factors differs across locations. This gives an opportunity to use magnetic field as a location signature.

**Wi-Fi signals:** A widely used location signature is Received Signal Strength (RSS), since RSS has a direct relationship with the distance from the transmitters. If the relationship between RSS and distance is known (e.g., in free-space), we can use RSS from three transmitters to uniquely determine the location. However, in practice, the relationship between RSS and distance is much more complex due to multipath effects, interference, and noise. This significantly limits the accuracy of RSS-based localization.

<table>
<thead>
<tr>
<th>Trace</th>
<th># Points</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total</td>
<td>Per Seg.</td>
</tr>
<tr>
<td>1</td>
<td>152</td>
<td>15</td>
</tr>
<tr>
<td>2</td>
<td>143</td>
<td>14</td>
</tr>
<tr>
<td>3</td>
<td>146</td>
<td>15</td>
</tr>
<tr>
<td>4</td>
<td>141</td>
<td>14</td>
</tr>
<tr>
<td>5</td>
<td>148</td>
<td>15</td>
</tr>
<tr>
<td>6</td>
<td>185</td>
<td>11</td>
</tr>
<tr>
<td>7</td>
<td>121</td>
<td>9</td>
</tr>
<tr>
<td>8</td>
<td>128</td>
<td>13</td>
</tr>
<tr>
<td>9</td>
<td>104</td>
<td>6</td>
</tr>
</tbody>
</table>

**TABLE I:** Accuracy of point based localization using magnetic-field outdoors.

Recognizing the limitation of RSS-based approaches, more recent works\textsuperscript{24}, \textsuperscript{45}, \textsuperscript{30} leverage the fine-grained Channel State Information (CSI) to improve the localization accuracy. From CSI, we can obtain RSS on every OFDM subcarrier (or subcarrier group). When $N$ transmitting antennas send to $M$ receiving antennas, the CSI consists of $M \times N$ matrices, where a matrix gives the amplitude and phase between a pair of transmitting and receiving antenna on a subcarrier. For example, the Intel Wi-Fi Link 5300 (iw51300) 802.11 a/b/g/n wireless network adapters that we use report the channel matrices for 30 subcarrier groups (around 2 subcarriers per group). Since CSI is not available on smartphones yet, we use desktops equipped with these Intel cards. We place them on wheeled carts for our measurements. This fine-grained signature improves accuracy, because different locations are less likely to have the same CSI.

#### B. Limitations of Point-based Localization

**Magnetic field:** Here we demonstrate that the accuracy of localization using magnetic field readings from a single point is poor. We evaluate point-based localization using the magnetic field outdoors. We use the driving traces that we collect as described in Section V. We refer to different drives/walks of the same route/path as a lap. A lap may contain more than one segment or street. We pick every 100-th point from lap 2 for testing and try to match these points with the points in lap 1 based on the closest magnetic-field value. We evaluate both with and without the knowledge of the specific segment. Table I shows the results. We can see that the accuracy is very poor. The error is hundreds of meters without the segment information, and tens of meters with the segment information. This is due to significant aliasing as points far apart may have similar magnetic field magnitude.

**Wi-Fi signals:** PinLoc\textsuperscript{30} uses CSI for localization. As the authors point out, CSI from locations even just 2 cm away can be very different. So PinLoc uses significant training data by having a Roomba robot move in a 1 m by 1 m grid for 4 minutes collecting the training data – 60 samples from each 2 cm by 2 cm grid in order to...
identify which grid a new measurement falls into. They further cluster the training data and localize a user to a 1m by 1m spot if multiple measurements are classified as falling into that spot.

To understand the performance of point based localization using CSI with sparse training data, we collect a CSI trace using the Intel 5300 cards (equipped with 3 antennas) based on the tool developed in [11]. We measure the CSI in a regular office building, which contains 44 grids, each spanning 2m by 2m. We use MCS 0 and 15 dBm Tx power for the measurement. We collect the CSI from 5 senders at 3 points in each grid. This gives us altogether 132 locations. Each location collects 30,000 packets. Each packet report CSI from 30 subcarrier groups on each of Rx antennas. The Intel 5300 cards use 1 Tx antenna and 3 Rx antennas for this configuration.

We quantify the performance based on the following information in the CSI: (i) phase, (ii) amplitude, and (iii) combination of phase and amplitude, to understand their impacts. (i) Phase signature: the initial phase of each transmission varies randomly. To get more reliable phase signature, we post-process the raw phase information to remove the random initial phase. Whenever $phase(s+1) < phase(s)$, where $s$ is the subcarrier index, we update $phase(s+1) = phase(s+1) + 2\pi$, since the phase of subcarriers in a frame should change monotonically. We then remove the random initial phase of the first subcarrier from the phases of all the subcarriers, i.e., $phase(s) = phase(s) - phase(1), \forall s$. We compute the mean CSI for each subcarrier group over all packets. It gives us a vector of $(3rx\_antennas \times 30 \text{ (subcarrier\_groups)} \times 5\text{ (senders)}) = 450$ mean phase values and use this as the signature of localization. (ii) Amplitude signature: it is simply a vector of 450 values, representing the average amplitude of each subcarrier group across all packets. (iii) Combined signature: for the CSI measurement on subcarrier $s$, once we get the corrected phase, $phase'(s)$, we compute the new complex CSI value: $a' + b'i$, with $a' = A(s) \times \cos(phase'(s))$ and $b' = A(s) \times \sin(phase'(s))$, where $A(s)$ is amplitude of the signal on subcarrier $s$. We obtain the combined signature as a vector of 450 complex values representing the mean over all packets.

We first evaluate the performance of localization when training data from the exact location is available. We divide the 30,000 packets from each location into a training trace of 20,000 packets and a testing trace of 10,000 packets. We then find the best match for each of the locations using these testing and training traces. We found that 132/132 locations matched correctly while using phase signature and amplitude signature and 131/132 locations matched correctly using the combined signature. On closer inspection of the one incorrect match from the combined signature, we find that the right match is still within top 5 closest matches with small differences in the signature. This shows that CSI from the exact same location is likely to be similar at least over a small time-scales, as in this experiment, where all traces were collected back-to-back.

Next we evaluate a more realistic case when there is no training data from the exact same points and check if we are able to match the signatures to the closest location. We take one location from each grid of size 2m by 2m as the training location, so altogether we have a total of 44 training locations, and 2 locations from each grid of the same size as the testing locations, which gives a total of 88 test locations. We deem correctly classified if we match to the closest location, to one of the top 2 closest locations or to one of the top 5 closest locations. As shown in Table II we match to closest location only 24% of the time using amplitude signature, 9% of the time using phase signature, and 11% of the time using combined signature. The limited accuracy even with non-trivial profiling (3 points per 2 m$^2$ grid) is because CSI at nearby locations can be very different. Specifically, the wavelength of the Wi-Fi signals is about 5.8cm for channel 36, 5.18 GHz, and phase is reversed every half wavelength, so even very nearby points can have very different phases. Amplitude information is robust against phase variations but still performs poorly due to measurement noise and poor correlation between geographic distance and signal strength.

<table>
<thead>
<tr>
<th>Error method</th>
<th>Amplitude</th>
<th>Phase</th>
<th>Combined</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dist. error</td>
<td>5m</td>
<td>7m</td>
<td>7.8m</td>
</tr>
<tr>
<td>Match to top 1</td>
<td>24%</td>
<td>9%</td>
<td>11%</td>
</tr>
<tr>
<td>Match to top 2</td>
<td>41%</td>
<td>13%</td>
<td>18%</td>
</tr>
<tr>
<td>Match to top 5</td>
<td>65%</td>
<td>35%</td>
<td>41%</td>
</tr>
</tbody>
</table>

TABLE II: Accuracy of matching to nearby locations.

Figures 1 (a), (b), and (c) further show the correlation between the real distances and the amplitude, phase and combined signature distances respectively. We see that as the real geographic distance increases, the difference between the signature distance may not always increase. This explains the poor accuracy of matching to the closest locations.

C. Potential of Trajectory-based Localization

Magnetic field: We measure the magnetic field along a few trajectories under different conditions. Figure 2 (a) and (b) compare the magnetic field between the two points on 1.5m long indoor trajectory measured using different devices (iPad vs. iPhone). While the curves have different shift and scale since the devices have different sensitivity and different speeds, their patterns look similar. Figure 3 (a) shows the magnetic field collected on a trajectory between Location 1 and Location 2 (that are 1.5m apart), using an iPhone on two different days, in...
To understand the impact of walking direction, we walk from location A to location B and then walk from location B to location A. As Figure 4(a) and (b) show, the measurements collected in the same direction look similar, while the measurements collected in the opposite directions look reversed on a 4m long indoor trajectory. This shows that it is sufficient to have the training traces in one direction and to have them reversed for the opposite direction.

Next we look at magnetic field pattern outdoors while driving. Several factors may affect the magnetic field patterns outdoors, such as measurement time, types of devices, and cars. Figure 5(a) shows the magnetic field pattern on a street in Redmond Town Center on two different days, and the pattern remains reasonably similar. Figure 5(b) shows the pattern across different devices is similar. Figure 5(c) further shows the pattern across different cars remains the same. In Section V we show that magnetic field can be leveraged to perform localization outdoors with different smartphone positions in a car (front-seat/back-seat). We also tried several other locations in the car like the dashboard. We found that magnetic field is stable unless the phone is placed within a foot from metallic objects like speakers in the car, which is not a natural position for a smartphone.

Wi-Fi Signals: Next we show that Wi-Fi signal over a trajectory can also be used for localization. In Figure 6 we plot RSS over the same 4m long indoor trajectory taken in two different walks and see that the pattern remains similar. Moreover, we see that pattern across different trajectories is different and omit the figures for brevity. In the following sections, we will use RSS and CSI information over a trajectory to achieve localization errors around 0.3 m.

IV. WAVELOC DESIGN

A. Overview

WaveLoc consists of a centralized server and mobile clients to perform the following tasks:

Training data collection: Traditionally, training data collection has been one of the major challenges hindering the adoption of trajectory based localization.
(b) Singularity problem

for the few areas it visits (e.g., Client occasionally downloads the training traces
communication cost and protect the users’ location pri-
Localization:
values can be annotated with interpolated co-ordinates.
known co-ordinates) and intermediate magnetic field
between two marked end points of a trajectory (with
roomba robots can be employed to run at constant speed
is more involved as GPS is not available. However,
indoors regular driving patterns to localize users later on using
annotated magnetic field traces from few volunteers’
e.g. from these regular trajectories. Outdoors, it is straight
forward since GPS is available and we just need the GPS
WaveLoc decomposes the problem into: fist, identify-
measurement. After finding the right trajectory, next step
is to localize the user on this trajectory.

C. Trajectory Matching & Localization

1) Distance Metrics: A key issue in trajectory match-
ing is to define an appropriate distance function to
quantify similarity between the location signatures and
is robust against different devices, speeds, time, environ-
mental factors while capturing the general trend. Some
simple distance functions include average and standard
deviation. However, such high-level aggregate statistics
lose valuable details about the time-series. Another natu-
ral distance function is the length of the longest common
subsequence (LCSS) [7]. It is well known and can be
efficiently solved by dynamic programming. However,
it only captures length of match and completely ignores
the unmatched parts which is also important. Moreover,
in unlike in string matching, in our scenario, even matching
points will not have the exact same measurements and
so we need to define a difference threshold to declare
measurements from two points as matching. This is hard.
WaveLoc uses dynamic time warping (DTW) [20].
DTW overcomes the limitations of LCSS by computing
the fine-grained distance over all points of the two
sequences. It stretches or compresses portions of the
input sequence to improve the alignment. This is useful
when we try to match traces collected at different speeds.

2) Classic DTW: DTW complete sequence: Given
sequences:  , , distance between any two points , DTW applies
dynamic programming to find the best alignment of
points on  to points on  . The technique is general
and can be applied to any additive distance function, and
extended to multiple dimensions. We use L1 norm (i.e.,
, ) in our evaluation. Other distance functions such
as L2 norm (i.e., ) yield similar results. Figure7

(a) DTW alignment (b) Singularity problem

Fig. 7: DTW examples.
Algorithm 1: Compute DTW distance.

(a) illustrates DTW alignment. DTW stretches required parts to find best alignment between the two curves. The DTW recursive relationship is on line 12 in Algorithm 1. DTW is ideal for matching two time-series collected at different speeds and sampling rates.

**DTW subsequence:** DTW subsequence finds the alignment and subsequence of $Y$ (training trace) that best matches the complete sequence of $X$ (test trace). Thus test sequence need not be complete. This is ideal for our scenario where we need to localize a user who is in the middle of her trajectory. It also outputs the alignment that gives the current location of the user. Interestingly DTW subsequence can be computed with the same complexity as DTW complete sequence by just changing the initialization: line 9 to $DTW(1, j) = d(x[1], y[j])$ to allow first point in test to align anywhere rather than forcing it to align with the first point in the training trace. Furthermore, the DTW distance is $min(DTW(n,:))$ (i.e., minimum value in the n-th column) instead of $DTW(n, m)$ so as to find a match for all elements in the test sequence to a portion of the training trace.

3) **Challenges of Applying DTW:** Singularity problem: A well-known issue in DTW is that multiple points on one curve may be incorrectly mapped to one point on the other curve [4]. DTW is effective when aligning two sequences that are similar in Y-axis values except acceleration or deceleration in time. The algorithm faces difficulties when the two sequences contain noise in the Y-axis value as well. Global differences, such as global shifts or scaling or other linear transformation, can be removed before applying DTW. Local differences (e.g., one series has a higher peak due to noise), which can be common, are much harder to deal with. In this case, DTW responds to the difference in Y-axis values by modifying time-axis, causing incorrect alignments. Figure 7 (b) shows an example of singularity problem, where the peak of trace 2 is stretched to match with multiple points on trace 1 since the peak of trace 2 is smaller than the peak of trace 1.

**Handling speed differences:** One of the major challenges we faced while implementing our solution, is to find the right alignment when test trace is collected at a higher speed than the training traces. DTW subsequence uses the complete test sequence and matches it to a smaller portion of the training sequence. If the test sequence is collected at a higher speed than the training sequence, ideally we want to match a point on the test sequence to multiple points in the training sequence. However, DTW fails to find such a match. Instead it prefers matching a point in the test sequence to one point in the training sequence since the distance tends to increase with the number of matching points.

**Handling diverse training traces:** Since we want to leverage training traces from many users training traces for one trajectory may come from diverse set of users (with varying device types, times of the day, positions of the devices, cars, as well as random events and noises). Using such diverse traces can improve our accuracy but how to effectively leverage different traces is an important design choice. Simply taking the average of DTW distance may mix up different patterns and lose important information. On the other hand, grouping traces based on how the measurement is collected may not be feasible due to lack of information. Further, factors affecting the patterns may vary across locations.

**Minimizing computation cost and power consumption:** Localization is run on the resource constrained mobile devices. DTW for continuous localization is an $O(MN)$ algorithm where $M$, $N$ are size of input sequences. To be a competitive alternative, the energy cost should be much lower than GPS.

**Full path localization:** Our signatures are collected continuously. To enable full-path localization we need to find the end point of each segment accurately which is challenging due to noise. Further, since it is impossible to find exact end-points, our algorithm should also be robust to imprecise cuts.

4) **WaveLoc:** Wavelet coefficients to handle different speeds and achieve accuracy: Using wavelet coefficients instead of the raw traces offers two major benefits. First, it allows us to handle different speeds. When the two time-series are collected at different speeds, we use wavelet coefficients at different wavelet levels (or scales). In particular, as mentioned in Section IV-C3 when finding $X$ in a subsequence of $Y$, the classic DTW subsequence fails when $X$ has higher speed than $Y$. Instead of matching $X$ and $Y$ wavelet coefficients at the same levels, we can find $X$ in a subsequence of $Y(s)$, where $Y(s)$ is the Y’s wavelet coefficient at level $s$. We pick the match by comparing DTW distance across all levels. The intuition for this is as follows: wavelet approximation coefficients (from low-pass filter) at every level decrease the number of coefficients by half. But the shape of the time-series remains similar to the original. Thus this provides a natural way to – in some sense sub-sample the signal, yet is much more accurate than simple sub-sampling. Given this background, imagine that test trace was collected at double the speed of training trace,
with same sensor sampling frequency, the number of samples in test trace will be half that of training trace. Now with noise, there is risk of mis-alignment between the two traces as test trace may not be stretched enough by simple DTW. But if we matched test trace with level 1 wavelet approximation coefficients (with some shifts introduced by coefficient computation removed) then there is a much better chance of correct alignment as test trace will be about same length as level 1 coefficients of training trace. To summarize, this gives WaveLoc the following benefit: even if it has training trace collected at only one particular speed, WaveLoc can still localize a test trace collected at any random speed chosen by the user and thus is a critical aspect of our algorithm design. In our experience, approximation coefficients (from low-pass filter) yielded higher accuracy than detail coefficients (from high-pass filter).

Another important benefit of using wavelet coefficients is that it improves efficiency. If we use the approximation coefficients at one level higher, as we mention above, it reduces the number of coefficients by half, and decreases the computation time of DTW to \( \frac{1}{4} \). In our evaluation level 2 coefficients give a good localization accuracy. Even if we need to match with wavelet coefficients from multiple levels, it is still efficient since the number of interesting levels is handful. The longest wavelet coefficients we consider is level 2 i.e., 1/4 of the length of the original time-series, which is 1/16 computation time. WaveLoc uses the Haar wavelets due to the simplicity and level 2 as the default. For multilevel, WaveLoc uses level 2 for test trace, and picks the best matching train level between level 2 and level 5. If going from level 2 to level 3 does not improve DTW distance of the alignment, WaveLoc stops trying higher levels. This serves two purposes: (i) efficiency, (ii) reducing the number of candidates for higher accuracy.

**Coping with singularity:** WaveLoc addresses the singularity problems by (i) pre-processing the data, (ii) using different local weights in DTW, and (iii) imposing a window constraint.

Specifically, to capture important features in the trace and reduce high frequency noise, we smooth the magnetic field traces using Savitzky–Golay filter \([22]\). Furthermore, we observe that outdoor magnetic field traces from the same segment may look alike but with a shift in the y-axis. A shift is present sometimes even when using the same device. Therefore we remove the mean of each trace before alignment.

To give different preference for horizontal, vertical, or diagonal, WaveLoc has local weighting as described in \([23]\). This is a modification to line 12 in Algorithm 1 to weigh \( d(x[i], y[j]) \) differently by \( w_h, w_v \) or \( w_d \) based on the step taken to reach \((i, j)\). This gives the recursion:

\[
DTW(i, j) = \min(\text{DTW}(i - 1, j) + d(x[i], y[j]) \times w_h, \\
\quad \text{DTW}(i - 1, j - 1) + d(x[i], y[j]) \times w_v, \\
\quad \text{DTW}(i - 1, j - 1) + d(x[i], y[j]) \times w_d),
\]

Algorithm 2: Final DTW subsequence.

\[
DTW(i, j - 1) + d(x[i], y[j]) \times w_h
\]

A lower weight on the diagonal helps avoid singularity problem, because by preferring diagonal paths, WaveLoc avoids stretching the row or column for matching. A lower weight on the horizontal helps match a short testing sequence (collected at a higher speed) in a longer train sequence, because it helps stretching the testing sequence and match it with multiple points on the training sequence. WaveLoc uses \( w_d = 0.6, w_v = 1, w_h = 0.6 \), which work reasonably well for all speeds.

Due to the singularity problem, some points in the beginning of the training trace may match with some point in the middle of the test trace. To prevent such an alignment from happening, WaveLoc imposes a window constraint as follows. Suppose the maximum ratio between the speeds of the two traces possible is \( \text{max_speed_ratio} \). Then the \( n \)-th point in one of traces can only align to any point between \( n \times \frac{1}{\text{max_speed_ratio}} \) and \( n \times \text{max_speed_ratio} \) of the other trace. To accommodate this, we change the loops in Algorithm 1 as follows: (i) line 6 and 8: for loop goes from 2 to \( \text{max_speed_ratio} \), and (ii) line 11: \( i \times \frac{1}{\text{max_speed_ratio}} \) to \( i \times \text{max_speed_ratio} \). WaveLoc uses \( \text{max_speed_ratio} \) of 2 as the default. The final DTW subsequence is presented in Algorithm 2.

**Clustering training traces:** Given that the time-series can exhibit different patterns for one trajectory, WaveLoc automatically partitions training traces into different clusters by using Meila-Shi spectral clustering algorithm \([19]\). It is the recommended algorithm in \([17]\) due to its excellent performance. Let \( W \) be the adjacency matrix of similarities between different pairs of training traces, with weight \( w_{ij} = 1/DTW(i, j) \), where \( i, j \) are indices of training traces. Let \( D \) be the degree matrix, which is a diagonal matrix with the node degree \( d_i = \sum j w_{ij} \) on the diagonal. We take the eigenvectors corresponding to the \( \sqrt{N} \) smallest eigenvalues of the normalized graph Laplacian matrix \( L_{rw} = I - D^{-1}W \) (where \( I \) is the identity matrix) and then call k-means clustering \([16]\) to cluster points by their respective \( \sqrt{N} \) components in these eigenvectors. WaveLoc finds the matching trajectory by finding the closest cluster that gives the
smallest average DTW distance to the testing trace. Then WaveLoc finds the closest training trace from this cluster and uses this trace for localization within the trajectory. For efficiency, the server performs clustering on the training traces offline in advance.

**Enhancing efficiency:** First, using wavelet coefficients already improves efficiency as discussed earlier. Second, for both accuracy as well as efficiency, for matching and identifying a new trajectory, at every intersection WaveLoc only considers the next possible 3 trajectories, *i.e.* go straight, turn left or turn right. Thus we refrain from matching against all possible segments in an area. We will describe how we identify which intersection using full path localization. Third, WaveLoc leverages the incremental nature of DTW. When DTW is called multiple times in one trajectory (*i.e.* go straight, turn left or turn right). Thus we refrain from matching against all possible segments in an area. We will describe how we identify which intersection using full path localization. Third, WaveLoc leverages the incremental nature of DTW. When DTW is called multiple times in one trajectory (*e.g.* for continuous localization), we only compute the newly added rows in the table. Every new reading corresponds to a new row, and the existing table entries can still be reused (dynamic programming property). This is called incremental DTW.

**Full path localization:** leverages trajectory matching and point localization as follows:

- **Identify the current trajectory:** DTW subsequence is called after every intersection to identify the right segment.
- **Continuous localization within a trajectory:** WaveLoc then periodically localizes the user on this segment using incremental DTW subsequence.
- **End point detection:** When WaveLoc finds that we are near the end of the training trajectory (via DTW subsequence), it first calls end point detection after waiting for a few seconds to ensure the trajectory is complete. Then WaveLoc end point detection finds the training segment as a subsequence in the current testing trace using DTW subsequence. Now, the point on the testing trace that aligns with the end point of the training segment is the intersection.
- **Identify the new trajectory:** In a new segment, WaveLoc first waits and accumulates enough information to allow correct matching. WaveLoc waits for \( \min(\min(L), \text{fraction} \times \max(L)) \) points, where \( L \) is the set of lengths of training traces that are candidates for the next segment. We use \( \text{fraction} = 0.5 \) outdoors and 0.75 indoors. A larger value is used indoors because segments are much shorter (*e.g.*, 4-6m vs. 100-200m). Then WaveLoc applies the DTW subsequence in step 1. To enhance robustness against end point misalignment error, WaveLoc matches the testing trace starting from the previously detected end point with all possible next segments. In addition, WaveLoc matches the testing trace involving the previous two segments with all possible two segments assuming the first segment is matched correctly but the cut point may not be detected precisely. If these two matching results are consistent, the result is used. Otherwise, WaveLoc uses the last three segments for matching, and takes a majority vote. In case there is no majority, it uses the result based on the last three segments since it contains more information. In all cases, incremental DTW is used to improve efficiency.

**D. Extensions**

Next we discuss a few extensions of our approach.

1) **Combining with other sensors:** Inertial sensors give valuable information for localization \([39], [27], [6], [56], [10], [55], [40]\). We can combine information from inertial sensors with the trajectory measurements to further enhance accuracy. Using inertial sensors, we can detect portions of traces where users do not move and remove those parts from the measurements before applying DTW. Moreover, we can also estimate user’s walking speed based on step counts using accelerometers. Accelerometer shows a periodic pattern when user is walking \([39], [27]\). Step-counting can be performed either using auto-correlation of accelerometer signal, counting peaks, or even by looking at the accelerometer signal in the frequency domain. Then we can estimate speed based on the number of steps per unit time and set the DTW window according to the estimated speed to enhance the matching accuracy. We evaluate the benefit of leveraging an accelerometer in \( V-E \).

2) **Using trajectory measurements as indoor landmarks:** Even when dense trajectory measurements may not be available, sparse trajectory measurements are still useful, because they allow us to identify landmarks. Such landmarks give us common reference points, and we can apply other localization approaches, such as dead-reckoning (*i.e.*, estimate distance based on accelerometer-based step counting and estimate direction using a compass), to localize between two adjacent landmarks. We evaluate the effectiveness of landmark identification indoors in \( V-E \). We focus on indoors, since we can rely on occasional GPS locks to get landmarks outdoors.

V. WAVELOC EVALUATION

A. Evaluation Methodology

**Magnetic field:** WaveLoc measures magnetic field at 32Hz using smartphones. It only uses the magnitude of the magnetic field since direction is sensitive to the phone orientation. The newer generation smartphones \([1], [5]\) have dedicated co-processors and significantly reduce the power consumption of collecting these measurements. Our indoor experiments are over 2 floors of an office building and outdoor experiments are from different cities, different cars, and different positions within the cars (front-/back-seat). Table \([II]\) summarizes our experiments.

**Wi-Fi:** We run the CSI and RSS experiments using Intel 5300 wireless cards \([II]\). We run the Wi-Fi experiments...
TABLE III: Outdoor magnetic field experiments.

<table>
<thead>
<tr>
<th>#</th>
<th>Area</th>
<th>Car (Seat)</th>
<th>Device</th>
<th>Laps/Segs/Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Shopping, Redmond</td>
<td>Accord (front)</td>
<td>Pantech</td>
<td>3/10/Evening</td>
</tr>
<tr>
<td>2</td>
<td>Shopping, Redmond</td>
<td>Accord (front)</td>
<td>Pantech</td>
<td>3/10/Night</td>
</tr>
<tr>
<td>3</td>
<td>Shopping, Redmond</td>
<td>Accord</td>
<td>Pantech</td>
<td>3/10/Night</td>
</tr>
<tr>
<td>4</td>
<td>Shopping, Redmond</td>
<td>Accord (front)</td>
<td>Prism</td>
<td>3/10/Evening</td>
</tr>
<tr>
<td>5</td>
<td>Shopping, Redmond</td>
<td>Accord (front)</td>
<td>Pantech</td>
<td>5/10/Night</td>
</tr>
<tr>
<td>6</td>
<td>Downtown, Redmond</td>
<td>Accord (front)</td>
<td>Prism</td>
<td>5/10/Evening</td>
</tr>
<tr>
<td>7</td>
<td>Residential, Austin</td>
<td>Accord (front)</td>
<td>Prism</td>
<td>5/10/Evening</td>
</tr>
<tr>
<td>8</td>
<td>Residential, Austin</td>
<td>Yaris</td>
<td>Prism</td>
<td>4/10/Night</td>
</tr>
<tr>
<td>9</td>
<td>Downtown, Austin</td>
<td>Yaris (front)</td>
<td>Prism</td>
<td>4/10/Evening</td>
</tr>
</tbody>
</table>

TABLE IV: Indoor magnetic field experiments.

in an office building within an area of 225 $m^2$. We experiment with all possible paths given the floor layout (which had 11 segments) and go over each path 3 times. Our measurement uses MCS 0 and transmission power of 15 dBm.

Fig. 8: Outdoors: schemes across different clustering.

(a) Trajectory Matching 
(b) Localization Median Error

B. Magnetic Field: Matching & Localization

1) Outdoors: We find that matching trajectory signatures in outdoor environments is a much harder problem, so we spend more space studying them.

Comparison of different schemes: Figure 8(a) compares the matching accuracy of DTW based approaches with longest common sequence (LCS) and average (AVG) using traces 1-5. We use traces 1-5 as they are traces obtained on the same route but under different scenarios. We quantify the matching accuracy based on the percentage of test traces that are matched correctly with the training traces. DTW refers to DTW without wavelet but only pre-processed y-value as mentioned in § IV-C4. LEVEL2 and LEVEL4 are DTW applied to the approximate wavelet coefficients at level 2 and level 4. We omit the curves of other levels for visual clarity. For LCS, we consider the points on the test and training traces match if they differ within 10% of the range of the Y-value in the training curve. Then the best matching training segment is the one that has the longest subsequence matching with the test trace. AVG uses the mean over an entire trajectory and finds the trajectory with closest mean. For all the schemes, we compare three clustering methods: random, natural – based on the collection setting (e.g., car type, device locations, etc.), and spectral – described in § IV-C4. We refer to different drives/walks of the same route/path as a lap. A lap may contain multiple segments. We pick every lap as test lap and in that turn all other laps become the training laps. To be exhaustive, we construct test cases using all possible approaches to an intersection.

First, we observe that DTW based approaches consistently achieve the highest matching accuracy. DTW accuracy is 10-30% higher than LCS and 25-117% higher than AVG. Second, using wavelet coefficients has comparable accuracy as original DTW. Third, spectral clustering gives better performance than natural clustering (by 10%), which outperforms random clustering (by 23%). Spectral clustering is better than natural clustering because the traces collected in the same way may not necessarily exhibit the same pattern and spectral clustering helps capture these patterns. So, we use the traces 1-5 with spectral clustering as the default outdoor traces in further evaluation. We refer to it as Combined trace. The 5 traces in the Combined trace were collected over multiple days within one week period. This is to demonstrate the stability of the signature over different days. For e.g., (i) the DTW distance between different measurements on same day on one sample segment was 0.2232 and (ii) between measurements from different days (1 week apart) it was 0.2514, which is a small 12% increase compared to (i).

Figure 8(b) compares localization accuracy. On each trajectory, we pick a point at 25%, 50%, and 75% of the total length and test for localization accuracy at each of these selected points. In LCS, we pick the last point returned on the training trace as the estimated current location. In CP (Closest Point), we match the current location in the testing trace to the closest matching signal value in the training trajectory. So CP is the only scheme that does not use the trajectory information. Figure 8(b) plots the median error. We compute error based on the GPS ground truth. The trend is similar to the matching results. DTW based approaches significantly outperform LCS and CP. For example, with spectral clustering, the median error of DTW (on the original time series) is around 13m, while the error of LCS is 62m and that of CP is 41m. The corresponding numbers for mean error are 16m, 63m, and 48m, respectively. LEVEL2 performs as well as DTW on original time-series with the median and mean errors close to 13m and 16m, respectively. Both the matching and localization results suggest using wavelet level 2 coefficients achieves similar accuracy as using the original time series. So we use LEVEL2 for evaluation below as default due to its higher efficiency – in other words this is the algorithm used by WaveLoc system. Since the mean and median errors show similar trends, we only report the median error from now on.

Impact of injecting additional speed differences: While speeds in the original traces are already different due to natural driving patterns, we further inject addi-
tional speed differences to our traces to better understand the impact. Figure 9 shows the result. Speed ratio 1 refers to the speed of the original non-modified traces although in reality they may not be collected at exactly same speeds. To simulate speed ratio of testing trace to training trace greater than 1 (i.e., the testing trace is faster), we linearly interpolate the training trace and for less than 1, we interpolate the testing trace. In this set of results, WaveLoc uses MULTI LEVEL algorithm as described in [8 IV-C4] with the base level of 2 and window size of 2. We compare with LEVEL2 with window set to \text{max\_speed\_ratio} = 2 (LEVEL2-W2) and with window set to the speed ratio (LEVEL2-W-INC). The latter has the benefit of knowing the speed ratio, which is not available in practice. Despite the benefit, our MULTI LEVEL performs better than both versions of LEVEL2: its matching accuracy is 73% and 54% for speed ratios of 2 and 6, respectively; and its localization error is 16m and 24m, respectively. In comparison, the accuracy of fixed wavelet level drops significantly: the matching accuracy of LEVEL2-W2 and LEVEL2-W-INC drops from 84% to 71% under a speed ratio of 2 and drops to 48% under a speed ratio of 6; their localization error increases from 13m to 23m under a speed ratio of 2, and increases to 61m and 48m, respectively, under a speed ratio of 6. This demonstrates the effectiveness of our multi-level wavelet scheme. Note that both LEVEL2 and MULTI LEVEL are algorithms adopted by WaveLoc. While the former is appropriate for regular conditions, the latter is powerful in handling significant speed differences.

In the interest of space, we only show the results for varying speed in Figure 10. The default speed ratio of 1 is also included for comparison. We consider traces 1—2 together and 3—4 together and use spectral clustering. First, we find that indoor accuracy is higher than outdoors due to more stable measurements and slower speeds. Second, matching accuracy does not deteriorate with higher speeds as (unlike outdoor traces) indoor traces have the mean magnetic field information, which does not vary with the speed. Third, the MULTI LEVEL is robust to such speed variation, and outperforms LEVEL2-W2 by up to 83% and LEVEL2-W-INC by 57% at a speed ratio of 6. In comparison, the schemes based on the fixed wavelet level degrade significantly with the speed: the error increases from 0.3m to 1.8m for LEVEL2-W2, and from 0.3m to 0.7m for LEVEL2-W-INC between speed ratios of 1 and 6. These accurate results indoors are encouraging as magnetic field is ubiquitous unlike Wi-Fi signatures.

![Fig. 9: Outdoors: Effect of varying speed.](image)

![Fig. 10: Indoors: Accuracy with varying speeds.](image)

### C. Magnetic Field: Full Path Localization

The accuracy here is different from the previous evaluations because the error is cumulative. Specifically if we make a wrong match on the current segment, all subsequent matches will likely be wrong since WaveLoc considers the end point of the current segment as the starting point of the new segment and uses that to determine the set of candidate training traces to match against. Similarly, the localization error also increases.

1) **Outdoors**: We show the benefit of WaveLoc in terms of (i) accuracy and (ii) power consumption.

**Accuracy**: We use the Combined trace, which has 3 intersections. At each intersection, there are 3 candidate next segments. We consider 2 routes: the route we drove on and the reverse route, and present the aggregate errors. Each route has 17 test laps, which gives 34 traces in total for testing.

In Figure 11, we show the performance at each segment in the path. When there is a matching error, all future localization will be erroneous due to searching on an incorrect trajectory. We assume the first segment is known (e.g., from GPS lock) so the matching accuracy is 100%. As expected, the initial segments have highest matching accuracy and lowest localization error. As the user travels more segments, the accuracy degrades due to cumulative errors. Matching accuracy goes down from 75% at the second segment to about 60% at the fourth segment. Localization error is 18m on the first segment, and about 46m on the fourth segment. So we recommend getting a new GPS reading at least once every 4 segments.

![Fig. 11: Outdoors: Accuracy across intersections.](image)

**Power consumption & running time:**
WaveLoc entirely runs on the phone to avoid dependence on connectivity. We use Monsoon power monitor [25] for the power measurements. We evaluate full path localization power consumption assuming 4 segment drive time of 2 minutes. So each segment takes 30 seconds. We compute matching 18 seconds after the start of every segment (to accumulate enough readings) and afterwards localization is run every 6 seconds till the end of the segment. On Samsung Galaxy S3, average running times for each localization and matching operation are 47 ms and 274 ms respectively. Thus it is very practical to run the computation on the phone. After 2 minutes, we turn on the GPS (automatically) and repeat the matching and localization as explained above for next set of 4 segments. Compared to using GPS all the time, power savings of WaveLoc can be as high as 55% on the Galaxy phone (and upto 45% on an older Huawei Prism phone) as shown in Figure 12. Assuming 2000 mAh battery capacity on the Galaxy phone, the power saving corresponds to 8 hours more battery life (6.4 vs. 14.4 hours) if the screen is off. Note that these measurements take into account the data collection and wavelet computation overheads.

2) Indoors: Figure 13 compares full path indoor localization. The trend is similar to outdoor case, where performance degrades as the user moves along. Interestingly, the absolute performance numbers are much better indoors. Even after 4 segments, the matching accuracy is still as high as 96% and the localization error is only 0.3m. This is likely due to a much slower moving speed, a smaller speed difference between different laps, and more stable indoor environment.

Next we use Wi-Fi signals for indoor localization. For matching, we consider the following baseline approaches: AVG RSS/AVG CSI, which use the mean RSS/CSI over an entire trajectory. These schemes still use trajectory information and are expected to be better than point based schemes. We use one sender, and vary the number of receive antennas from 1 to 3 on Intel 5300 card. For all schemes, we use magnitude of the CSI. As shown in Figure 14 (a), all schemes work reasonably well (e.g., over 80% accuracy) because trajectory is a fine-grained signature and allows unique path identification. Among them, we find CSI is generally outperforms RSS. For example, LEVEL2 (algorithm adopted by WaveLoc system) at 3 receive antennas achieves 100% matching accuracy with CSI and 96% with RSS. That is because CSI is a richer signature (30 values) as compared to RSS (1 value). Also, LEVEL2 is more robust than AVG.

Figure 14 (b) compares localization accuracy of our DTW based approaches with the following baseline schemes: CP RSS/CP CSI: given the current point, pick the point that has closest RSS/CSI signature in the deployment area. Unlike the matching results, DTW based approaches perform much better than others in localization. The localization error of CP is as high as 2.8m with CSI, and 4.2m with RSS with 3 receive antennas. Note that prior work reporting accurate localization with CSI [30] requires dense CSI measurement as described in § III. In comparison, LEVEL2 achieves 0.17m localization error with CSI and 0.25m with RSS.

E. Extensions

Combining with other sensors: To evaluate the case where users may pause in the middle of a trajectory, we collect indoor magnetic field data over 3 walks, 6 segments, around 5.5 m each, in an office building. During the first two walks, the user walks normally without stopping. During the third walk, the user randomly stops and waits anywhere in the segment for 30-60 seconds. We use the first walk as the training trace, and use the second walk as the testing trace. The corresponding localization error marked as “NoStop” in Figure 15 is within 0.2 m. We then use the third walk as the testing trace. The next two clusters of bars in the figure, marked “Stop” and “StopDetect” denote the errors when we either keep or remove the static portions, respectively. The error is around 1.7 m without detecting the stop, and decreases to 0.3 m after detecting and removing the stop, demonstrating the benefit of combining accelerometer and magnetic field information.

Using trajectory measurements as indoor landmarks: Next we evaluate the accuracy of using trajectory measurements as indoor landmarks. We use two areas: Areal from traces 1 and 2 in Table IV and Area2 from
traces 3 and 4 in the table. Area1 has 22 segments and spans 225 m², while Area2 has 20 segments and spans 1290 m². In our evaluation, we use one walk from the traces as the testing trace and the remaining walks as the training traces. As shown in Figure 15 both DTW and LEVEL2 (i.e., DTW applied to level 2 wavelet coefficients) yield high matching accuracy. Note that this result is different from the previous matching results as it is matched against many more candidates (21 and 19) rather than 3 possible next segments. This is useful for landmark detection because (i) if we have no idea of which intersection user is at, we can match against all possible segments in nearby area with high accuracy, and (ii) if only sparse trajectory measurements are available such that continuous localization is not possible, we can still use the sparse measurements in the nearby area and find the right match.

### VI. Discussion

Interestingly, while the worst places for GPS localization outdoors are urban canyons, under bridges etc. they are the best places to leverage magnetic field as these structures impose anomalies on earth’s magnetic field very unique to those locations. Thus we believe this should be a direction to pursue going forward. In this paper we show preliminary promise in showing that outdoor GPS can be augmented with intermediate localization using geo-magnetic field. Furthermore, urban canyons interestingly are also dense areas for Wi-Fi and cellular. Thus WaveLoc can be augmented to use multiple signals together for even higher accuracy.

### VII. Conclusion

We explore the promise of using trajectory signatures for localization. Matching noisy trajectory measurements from varying conditions is significantly more challenging than matching measurements from single points. To address this, we build WaveLoc that divides the problem into two steps: first, it identifies the right trajectory using our enhanced DTW subsequence, and second, localizes points on the trajectory based on the DTW alignment result. To enhance the accuracy, robustness, and efficiency, WaveLoc performs multi-level wavelet decomposition and clusters the training traces. WaveLoc achieves reasonable accuracy outdoors and saves 45-55% power over GPS. The accuracy is even higher indoors: with a median error of as low as 0.3m. Moving forward, we want to extend WaveLoc to leverage magnetic field and Wi-Fi together for a more robust and accurate localization scheme.

**Acknowledgements:** This work is supported in part by NSF Grants CCF-0916309, CCF-117009, and IIP-1546831.

**References**