LoSI: Large Scale Location Inference Through FM Signal Integration and Estimation

Tathagata Mukherjee*, Piyush Kumar, Debdeep Pati, Erik Blasch, Eduardo Pasiliao, and Liqin Xu

Abstract: In this paper we present a large scale, passive positioning system that can be used for approximate localization in Global Positioning System (GPS) denied/spoofed environments. This system can be used for detecting GPS spoofing as well as for initial position estimation for input to other GPS free positioning and navigation systems like Terrain Contour Matching (TERCOM). Our Location inference through Frequency Modulation (FM) Signal Integration and estimation (LoSI) system is based on broadcast FM radio signals and uses Received Signal Strength Indicator (RSSI) obtained using a Software Defined Radio (SDR). The RSSI thus obtained is used for indexing into an estimated model of expected FM spectrum for the entire United States. We show that with the hardware for data acquisition, a single point resolution of around 3 miles and associated algorithms, we are capable of positioning with errors as low as a single pixel (more precisely around 0.12 mile). The algorithm uses a large-scale model estimation phase that computes the expected FM spectrum in small rectangular cells (realized using geohashes) across the Contiguous United States (CONUS). We define and use Dominant Channel Descriptor (DCD) features, which can be used for positioning using time varying models. Finally we use an algorithm based on Euclidean nearest neighbors in the DCD feature space for position estimation. The system first runs a DCD feature detector on the observed spectrum and then solves a subset query formulation to find Inference Candidates (IC). Finally, it uses a simple Euclidean nearest neighbor search on the ICs to localize the observation. We report results on 1500 points across Florida using data and model estimates from 2015 and 2017. We also provide a Bayesian decision theoretic justification for the nearest neighbor search.

Key words: Global Positioning System (GPS)-free positioning; Frequency Modulation (FM) radio; signals of opportunity; nearest neighbor search

1 Introduction

Localization using ambient wireless signals (both indoor as well as outdoor) has generated considerable interest in recent times. Considerable work has been done both in the academia as well as in the industry with focus on civil and research use[1–6]. Indoor localization is important for ubiquitous computing[7, 8] and as Global Positioning System (GPS) is severely degraded in such environments[9], other modes of localization become necessary. The importance of outdoor localization in the absence of GPS cannot be overemphasized due to the heavy dependence on GPS in the society. The
problem of outdoor positioning has been studied for a long time with GPS being the most widely used large scale outdoor positioning system in use today.

With the coming of age of ubiquitous computing, ambient wireless signals have been extensively used for building indoor localization systems\(^\text{[2,10,11]}\) and the accuracy achievable with such systems continues to improve with time. Though there are no accepted standards for indoor localization, WiFi-based localization is the most common\(^\text{[11]}\) and other methods continue to be studied in detail. Examples of such systems are those based on Global System for Mobile communications (GSM)\(^\text{[12]}\) and Frequency Modulation (FM) signals\(^\text{[13]}\). For outdoor localization, the state of the art, in the last three decades, has been GPS\(^\text{[14]}\). The use of GPS has exploded in the last decade and today it has infiltrated almost every household in the world, courtesy of smartphones. Though it is hard to imagine a world where GPS is not available or unreliable, it is a real possibility. GPS can become unreliable due to spoofing by adversaries and it can become unavailable altogether due to jamming in contested environments, or disturbances caused due to natural causes like extensive cloud cover\(^\text{[15–22]}\). As a result, there is need for research into systems that can be used as a “fall back” when GPS becomes unreliable or is completely unavailable. Many research groups and organizations have been investing heavily in research into GPS-free positioning, navigation, and timing, which further demonstrate the importance of such technologies going forward\(^\text{[23–26]}\). It must be pointed out here that the development of GPS was pioneered by the US military and several reports in recent years have acknowledged the fallibility of GPS and the importance of alternative modalities for positioning\(^\text{[27]}\).

Assisted GPS\(^\text{[28]}\) and differential GPS\(^\text{[29]}\) are examples of systems that aim to augment and assist existing GPS systems either for accurate and reliable positioning or for obtaining faster “time-to-first-fix”. For example, Assisted GPS (A-GPS) uses cell phone towers to improve the quality of GPS positioning. It also downloads and stores information about the location of GPS satellites for improving the “time-to-first-fix”. Aside from this if GPS becomes unavailable, A-GPS systems can then use cell phone towers to localize the user. Differential GPS (D-GPS) uses two co-operating GPS receivers (one stationary and one moving) to localize the receiver in motion. If GPS becomes unavailable, then D-GPS ceases to work whereas A-GPS works only when the receiver has access to a cell phone network (GSM or Long-Term Evolution (LTE) networks). Thus if the receiver does not have access to a cellular network when GPS is lost, this method is of little use. Hence GPS-free positioning is required in these scenarios.

In this work we describe a large scale coarse localization system based on ambient FM signals and approximate FM map estimation. This Signals of Opportunity (SoOP)-based method can be used to approximately self-locate within a large area when GPS is degraded, unreliable, or unavailable. Our system cannot be used for navigation in the absence of GPS because it computes a coarse position estimate. However, given this coarse estimate, it is possible to use a secondary system like TERCOM\(^\text{[30]}\) or variations\(^\text{[31]}\) for accurate positioning and navigation. The system can also be used for detecting and overcoming GPS spoofing. Finally, our system can be used with existing GPS systems to decrease the “time-to-first-fix” without the user paying for additional cellular data as in the case of A-GPS.

The idea of using ambient Radio Frequency (RF) signals for localization is not new and has been studied before. Some of these techniques include anchor-based approaches\(^\text{[2,32–35]}\), those using Time Of Arrival (TOA)\(^\text{[36]}\), Time Difference of Arrival (TDoA)\(^\text{[37]}\), and Angle Of Arrival (AOA)\(^\text{[35]}\). Most of these techniques either require complex and expensive hardware, a complex time synchronization step, or multiple antennas. Some of them even require modeling the underlying multi-path in the environment in order to be useable. In addition to these techniques, there are some methods that use distance-dependent features of the RF signal, for example, the Received Signal Strength Indicator (RSSI)\(^\text{[38]}\), Signal-to-Noise Ratio (SNR), and the Stereo Channel Separation (SCS)\(^\text{[39]}\).

RSSI at the receiver is determined by the hardware of the receiver, the location and power of the transmitter, and the ambient medium. Given a transmitter \(t\), the RSSI at the receiver which is at a distance \(d\) from the transmitter is given by (see Ref. [40])

\[
r^T_d = r^T_{d_0} - \beta \log \left( \frac{d}{d_0} \right) + \eta_d \tag{1}
\]

where \(\eta_d\) is the noise at a location which is at distance \(d\) from the transmitter and \(d_0\) is a location which is at distance \(d_0\) from the transmitter where the RSSI is known beforehand and \(\beta\) is a weighing constant.
Under the model, \( \eta_d \) has a Gaussian distribution with zero mean and an unknown variance and models the uncertainty in the environment. Also the received signal strength given by Eq. (1) depends on the device used to sense the medium\(^{10} \). RSSI-based localization is usually easy to implement, since the RSSI can be read directly from the hardware, and the computation complexity is low. On the flip side, the accuracy of such systems is usually low, if the RSSI is used directly for the localization. This is primarily because the RSSI is affected by the state of the ambient medium, and the quality of the device used for measurements and the multi-path. Thus, for improving the accuracy, RSSI-based systems require careful feature engineering for use with a learning algorithm, so as to get acceptable accuracy\(^{2,10} \).

Another point worth mentioning, in the context of RSSI-based localization, is the fact that any learning algorithm used with the engineered features runs into the problem of model generalization in the presence of data obtained from heterogeneous devices. Any model for localization is usually created using training data which is obtained using a wireless device that is usually different from the device used to acquire the test data in real time. Moreover, data collected with the test device is usually not available beforehand and hence it is not possible to calibrate the model to account for the differences in the hardware of the two devices. As pointed out in Ref. [10], this has the potential to result in poor generalization performance which leads to a transfer learning problem, where the goal is to use a model learned with data obtained from a possibly unknown device and use the same for localizing an uncalibrated device in real time, using the sensed RSSI data. Our localization system is based on RSSI of FM signals and an estimated FM signals map and hence suffers from these disadvantages. As our goal was to build a fast and simple localization technique, we use careful feature engineering to alleviate these problems. Note that a Distributed Data Driven Application System (DDDAS) approach could be used as one can build a model of the RSSI response and calibrate the same for different environments (e.g., “FM model”). When the system is deployed, the real-time estimation can fine tune the general model in relation to the variations in the ambient environment, current sensor use, and platform measurement performance\(^{41} \). Thus calibration still plays a role and we will discussed this further later.

Inspired by the success of FM-based localization in indoor environments\(^{2} \), we use FM broadcast signals for building a large scale approximate outdoor localization system. In Ref. [2], the authors built an indoor FM-based localization system using a “fingerprint” based approach. However, for large scale outdoor localization, “fingerprint” based systems are hard to build because of scaling issues. As a result our system uses a large scale “FM model” estimated for the entire mainland United States. This system can be used for computing the approximate location of the user in the absence of GPS or when GPS is severely degraded or spoofed. Our goal is to get an approximate location fix, when GPS becomes ineffective. Once the approximate location is found, other methods can be used to improve the positioning accuracy of the affected GPS system and the whole localization system can be functional until GPS becomes available (reliable) again. A system for improving the approximate location of an Unmanned Aerial Vehicle (UAV) was built by Mukherjee et al.\(^{42} \) and several other methods\(^{8,30,31} \) can be used for improving this approximate location. The FM broadcast band falls between 88 MHz and 108 MHz in the US with a 200 kHz bandwidth per station. FM signals are Very High Frequency (VHF) and hence are less sensitive to weather conditions and terrain\(^{2,43} \), cover long distances, and can be used for both indoor\(^{2} \) and outdoor localization. Our method is based on RSSI, within the constrain of hardware costs, which is less than $25 for the entire system. This immediately rules out methods based on TOA, TDoA, and AOA which require more expensive of the shelf hardware.

Our method contains two distinct steps: the first step estimates the signal strength model and the final step tests the model for localization accuracy. Though we estimate the signal strength model in the first phase, unlike the training phase for traditional machine learning methods, we do not use observed RSSI values at different known locations for building this model. Instead we use data about FM transmitters across the US, consisting of their power of transmission and a constant power polygon (Figs. 1 and 2), as the input to this phase. The model computes an estimate of the RSSI at each point of our region of interest (the entire US mainland). In the testing phase, we use a software defined radio to read the FM power spectrum at an unknown location, and this is resolved into a location estimate using Bayesian estimation techniques.
Algorithm 1: Output a model of expected RSSI

1: function ESTIMATEMODEL($\tau$)  
2: $D \leftarrow$ GenGeoHash()  
3: for $x \in D$ do  
4: $HT[x] \leftarrow [-\infty]^{101}$  
5: end for  
6: for $(t, r, p) \in \tau$ do  
7: for $x \in GetPixels(t, r)$ do  
8: $v \leftarrow HT[x]$  
9: $y \leftarrow RayIntersect(p, Location(t), x)$  
10: $d_1 \leftarrow |y - Location(t)|$  
11: $d_2 \leftarrow |x - Location(t)|$  
12: $j \leftarrow FreqIndex(t)$  
13: $a = (\text{Power}(t), d_1, d_2, v_j)$  
14: $v_j \leftarrow \text{CALCULATEPOWER}(a)$  
15: $HT[x] \leftarrow v$  
16: end for  
17: end for  
18: return $HT$  
19: end function

and the estimated FM model. We would like to stress here that though our method can technically be called a "fingerprinting based method", it is very different from how "fingerprinting" works. In any "fingerprinting" based approach, one needs to collect the "fingerprints" whereas in our case we estimate them, which makes our approach different in two ways. First, the estimated values are not as precise as actual "fingerprints". Finally we use the data obtained from the FCC for our estimation and then use data obtained from a completely different device for the actual localization. This is not done for any "fingerprinting" based method.

The localization system that we describe has several interesting properties. The system reveals the DDDAS paradigm in which instrumentation data and executing application models become a dynamic feedback control loop, whereby measurement data are dynamically incorporated into a model of the system in order to improve the accuracy of the model. Again in the reverse process, the executing application model controls the instrumentation process to guide the measurement process. The DDDAS paradigm is used to build a large scale approximate positioning system that uses SoOP. We also demonstrate (1) the use of a model learned with data collected using unknown devices to approximately localize over a large area, (2) a simple feature engineering technique for the localization of an uncalibrated device using RSSI values only, and (3) that the resulting features can also be used to localize the uncalibrated device using models learned at a different point in time. We show empirically that there are negligible differences in the localization accuracy.
obtained using an old model that was learned a few years back against a new model that was learned more recently.

**Notation**: We denote locations in any region by lowercase letters as \( x, y \). \( \psi \) is used for the power spectrum and its subsets. Scalars including indexes are represented by lowercase letters like \( i, j, k \). Particular power values are accessed as \( \psi_i \). We note that a particular power value \( \psi_i \) is a scalar and can be assigned to scalars. Dictionaries are denoted by symbols like \( D, H, SS \). We note that dictionaries are same as functions and hence functions are also represented as such. Keys in a dictionary are always vectors. Values in a dictionary corresponding to a key \( x \) are accessed by \( D[x] \). Collections or sets are denoted by letters like \( S \). Random variables are denoted by letters like \( X, Y \). Expectation of a random variable is denoted by \( E \) and the probability distribution of a random variable is denoted by \( P \). When dealing with probabilities and probability distributions, the associated parameters are denoted by letters like \( \mu \) and \( \sigma \). This is true for all the scalars and constants that are associated with convergence assumptions of random variables and their distributions. \( \pi \) is reserved for representing prior distributions.

Next we describe other techniques that have been used for tackling similar or related problems.

## 2 Background and Related Work

In this work we are concerned with methods for absolute positioning, the problem of finding the absolute coordinates of a point in a fixed reference frame. Absolute positioning can be done using two approaches. The first approach relies on communications with the GPS whereas the second achieves its objective without any such communication. Traditional GPS-based localization\(^{[14]}\) uses GPS receivers to communicate with several GPS satellites. The received data is used to compute the distance of the object from at least four known GPS satellites using the idea of TOA\(^{[36]}\). The final position is found using trilateration. Current GPS systems, without modifications, suffer from several limitations, namely, lack of precision\(^{[44]}\), jamming\(^{[45]}\), and disruption and spoofing\(^{[46]}\).

To get around these problems, researchers have used the idea of assisted GPS\(^{[47]}\). In A-GPS, there is a dedicated A-GPS server that communicates with the GPS receiver. The data from the server is used to augment that the data obtained by the GPS receiver. Lee et al.\(^{[48]}\) used the idea of Radio-Frequency assisted GPS (RF-GPS). RF-GPS uses Differential GPS (DGPS)\(^{[49]}\) in order to correct the errors and improve the accuracy of GPS. More recently, work has been done in order to achieve centimeter level accuracy with GPS\(^{[29, 50-53]}\). Most of the centimeter accurate GPS systems use the idea of carrier phase differential.

Apart from GPS-based absolute positioning methods, there are absolute positioning techniques that do not depend on GPS. These methods are usually called GPS-free localization techniques. One of the most common forms of GPS free positioning is called Network Based Geolocation\(^{[47, 54, 55]}\). These methods are almost exclusively based on technologies that depend on wireless networks and use signal processing heavily. They use methods such as TOA, time difference TDoA, AOA, timing advance, and multipath fingerprinting\(^{[35, 36, 38, 56-58]}\).

McEllroy et al.\(^{[59]}\) developed a GPS free navigation system using the Amplitude Modulation (AM) radio transmission band with the TDoA method. The idea is to use an AM tower with known location and one reference receiver to estimate the second receiver’s location. In theory, this work can be extended to use FM stations. This method has an accuracy of 20m, but it relies on a known position of the reference receiver. Therefore, in a large area, it mandates the deployment of a large number of reference receivers in order to perform the localization. More recently, ground-based terrestrial transmitters have been used to create a terrestrial version of GPS, that does not depend on transmissions from GPS constellations. One such system is called Locata\(^{[60]}\) which has pioneered the use of ground-based transmitters that can be used as an alternative to GPS satellites. One of the major drawbacks of such a system is the need to invest in the creation of a large network of Locata transmitters across the world.

In order to implement a network-based geolocation system, without the need for special transmitters, one can use ambient radio signals. A straightforward approach to building such a system is through the use of a fingerprint database. Such systems are broadly categorized as fingerprint-based localization system and the fingerprints can be received signal strengths for WiFi or FM\(^{[61, 62]}\). They can also be readings obtained from inertial sensors, in case, such readings.
have unique characteristics at given locations\cite{63}. These methods have been extensively used for building indoor positioning systems\cite{13,34,62,63}. The receiver scans the fingerprint at a given location of interest and then compares it with a database, which contains the fingerprints for every possible location in the region of interest. The location of the query point is determined based on a match found in this database.

Laoudias et al.\cite{64} built such a system using smartphones. Smartphones are used to collect data on WiFi Access Points (APs) and create a database for the entire region of interest. This is done as a pre-processing step. Finally, in the query phase, the location of a point is determined by using an Euclidean nearest neighbor search, on the database, in the space of the RSSI values. A crowdsourced version of a similar system was implemented by Petrou et al.\cite{65} Adding another dimension to positioning, Konstantinidis et al.\cite{66} studied privacy preserving indoor localization using smartphones.

Fang et al.\cite{33} did an extensive study on FM fingerprint-based localization in a small area, for studying the possibility of meeting the FCC requirement\cite{67} (50 m error for 67 percent of calls for handset-based devices). They built a localization system using the RSSI values of the FM signals as fingerprints. They used the idea of correlation to compare the observed signal at a point with those stored in the database. They also compared the results obtained from FM and GSM signals as the sources. It is difficult, if not impossible, to extend these results to an entire country like the United States because of the large amount of data collection required.

Abdelnasser et al.\cite{63} implemented a system with fingerprints as data from different sensors on a mobile phone for indoor localization. They noticed that different indoor locations have unique signatures on one or more of the on-board sensors of the phone. For example, stairs have very different signatures in the accelerometer as compared to level floor. Such locations with unique signatures are called landmarks. Their system used these landmarks and combined them with dead-reckoning for indoor positioning. Azizyan et al.\cite{68} used a novel technique called SurroundSense to logically localize mobile phones using a method called ambience fingerprinting. The technique of ambience fingerprinting uses all possible sensor information available in the surrounding as a fingerprint. Recently, Aly and Youssef\cite{69} proposed the Dejavu system, which can achieve very accurate large scale localization using cell phones and crowdsourced data. An important variation of network-based geolocation is called Signals of Opportunity (SoO)-based localization\cite{70,71}. These systems all use the different types of available RF signals in the environment to create a fingerprint database. Different types of RF signals such as GSM\cite{12,34}, WiFi signal\cite{11,72}, FM\cite{13,62,73}, or TV signals\cite{74} can be used for the positioning. Unlike GPS, these systems can be used for indoor localization and are known to give errors of less than 3 m\cite{11,73}.

Indoor localization systems have also been built using strengths of sound waves\cite{75} as fingerprints. Tarzia et al.\cite{75} studied an acoustic fingerprint-based localization system. They used the Acoustic Background Spectrum (ABS) as a fingerprint for indoor localization. Another option is to use ultrasound for the fingerprints. For example, Hazas and Hopper\cite{76} studied the use of broadband ultrasound for indoor localization.

There are aspects of the problem of positioning using RSSI values that make the problem non-trivial both indoors as well as outdoors. One problem has to do with the fact that the device that is used for collecting the RSSI values for localization is oftentimes not fully characterized and is usually different from the device used for collecting the data for the training or fingerprinting step. This is called the “cold start” problem for heterogeneous device localization. Zheng et al.\cite{10} studied this problem in the context of indoor localization. In Ref. [10], the authors came up with robust feature representations for solving this problem. In another related work\cite{77}, the authors studied the problems that arise with the difference in the distributions of the training data and the test data. This problem is related to the problem of heterogeneous cold start device localization. However, in Ref. [77], the authors studied the adaptations that are required for using the model learned using data from one device to localize another device using the same modality in indoor environments. They treated the problem of localizing multiple devices as multiple learning tasks, and formulated it as a multi-task learning problem.

For any localization system to be robust, the positioning accuracy should not vary significantly over time. For example, in GPS, under normal operating conditions, the localization accuracy is usually fixed (around 20 m for commercial GPS systems). However, with any RSSI-based system, this guarantee is hard to establish as the RSSI values change over time.
Thus, the models developed at a given point in time might not be applicable at another time because the characteristics of the environment changed. In Ref. [78], the authors studied exactly this problem in the context of indoor localization. They used a novel semi-supervised algorithm based on hidden Markov models, which they called transferred hidden Markov model for solving this problem. As far as we know, these problems have not been studied in the context of large scale localization in outdoor environments.

Inspired by the success of FM-based indoor localization systems, we have implemented a large scale FM-based positioning system that uses a large scale signal map estimation phase. Our system is very similar to the system developed by Youssef et al. [79], which extended the ideas used in Smart Personal Object Technology (SPOT) [3]. Youssef et al. [79] studied the problem of FM-based localization using estimated ranking of FM stations in the city of Seattle. They used RadioSoft’s ComStudy [80] FM signal estimation package for predicting the FM map for the 28 FM transmitters in the Seattle area. They divided the region of interest into grids and for each grid they got a ranking of the 28 FM stations based on the estimated power values. In order to reduce the computational complexity, they grouped the 28 stations into 7 groups based on Pearson’s correlation coefficient between all the pairs of spatially co-located FM transmitters. Finally, at a location of interest, the FM spectrum from the 7 groups are measured and the location is determined by finding the ranking in the estimated database that best matches the observed ranking using Bayesian methods.

Though in Ref. [79] the method relies on determining the ranking based on the estimated power values as we do, the similarity ends there. In order to use the method of Youssef et al. [79], one would need to exactly know the location of the FM transmitters and would need to generate the estimation map for every region of interest differently, as their estimation used the terrain models for the region of interest. For a city other than Seattle, the number of FM stations would be different from 28 and the number of buckets different from 7 and hence this would completely change the map generation process for each region of interest. This would make it hard to scale this method to very large areas, like an entire country, because the map generation would be difficult due to the heterogeneity of the distribution of the FM transmitters and the difference in the terrain information. On the other hand, we use a large scale map estimation technique that does not depend on exact knowledge of the distribution of FM transmitters in a region of interest or the information about the terrain of the region. Thus every location, irrespective of the actual distribution of FM transmitters, is represented using a 101-dimensional vector. We use a feature selection mechanism coupled with a candidate selection algorithm to reduce the dimensionality and breadth of the search space, instead of creating groups of correlated stations, in order to make the system scalable.

In a more recent work, Margolies et al. [81] used 4G LTE data for localization. They used User Measurement Data (UMD) that consists of the Received Signal Strength (RSS), Reference Signal Received Power (RSRP), Propagation Distance (PD), and measurements of other metrics like throughput, latency, and information on dropped calls. The UMD is collected by the network and does not contain the user location information. The goal of the paper is to be able to predict the user location based on the observed UMD. To this end the authors [81] described the Network Based Localization (NBL) system. Their system operates in two phases: the offline phase uses GPS tagged UMD (GUMD) to build a coverage map that associates each location within the region of interest with an estimated 3-tuple of (RSRP, RSS, PD), which is finally used for localization. For making the problem computationally tractable for very large areas, the authors [81] divided the region of interest into a number of grids such that all positions within a grid is mapped to a representative. They used grid sizes of 16 to 21 which corresponds to cell widths of 522 m to 16 m, respectively. In the online phase they used a Maximum Likelihood (ML)-based approach and a weighted average-based approach for the location inference. We must point out that their method is similar to ours in that we also have two phases: one offline and the other online. Moreover we also overlay the region of interest with a grid of cells for reducing the computation involved. Finally the online phase of our algorithm uses a maximum likelihood-based approach (noting that Bayesian estimation is nothing but maximum likelihood with uniform prior). Our offline phase does not use any crowdsourced user data, and instead of using RSRP, RSS, and PD, our method simply uses the RSS for localization.

We choose FM instead of LTE for our work because it is harder to get data for LTE (GSM) transmitters for the entire United States. Our method should work
equally well with LTE (GSM) data. Moreover, our work can easily augment the Dejavu system\cite{69}, in the absence of crowdsourced data. Moreover, indoor localization systems\cite{62,75} can also use our system to improve accuracy and coverage area using a two-phase approach. In these settings, our system can be used as a coarse localization system and then once we have localized to a small area, the indoor/local positioning system can take over to give more fine grained results. It must be noted that for using our method in indoor environments, the offline estimation phase will need to change and will need to incorporate the floor plan of the building and location of nearby FM transmitters. Similar methods that use RF data for localization can benefit from our method, as long as transmitter data is available apriori, the receiver can acquire the power spectrum and has a small amount of processing capability. To our knowledge, we are the first to demonstrate FM-based localization without any GPS assist using large scale map estimation for an entire country. Moreover, the model estimation is done using data collected using unknown devices for which the calibration information is not available. Moreover, for the localization step, we collect data using an uncalibrated Realtek RTL2832U SDR\cite{82}. We also show through experiments that our method is agnostic to the temporal distribution of the RSSI values.

3 Localization Algorithm

Before discussing the different steps for localization, we outline the context for the application of our algorithms. First, we assume a priori knowledge of the transmitted power and location, with respect to a global coordinate system, of all FM transmitters in the region of interest. We denote the location of an FM transmitter by $r$ and the radius of its influence by $r$. Every FM transmitter is associated a power polygon. The $p$-dBu polygon, represented as a vector, corresponding to the transmitter is denoted by $p$ and is assumed to be available at the time of model estimation. Note that this polygon is obtained by physically measuring the power received from an FM transmitter and is assumed to be made available to us (in case of the US mainland by the FCC). However, we do not assume that we have any information about the device used for measuring this polygon.

As mentioned before our algorithm has two stages: model estimation and testing. The model estimation step is done off-line and the resulting model is used for the positioning in the testing phase. The first step of our algorithm estimates the expected value of the FM spectrum at each point across the entire region of interest. More precisely, given a location $x$, in geographical coordinates (we only consider the latitude and longitude as we want to self-localize on the surface of the earth), we determine the expected RSSI for the different channels of the FM spectrum at that location, and represent the same as a 101-dimensional vector (as there are 101 channels in the FM spectrum). The testing phase of our algorithm has three parts namely: feature detection, candidate selection, and localization, described in detail later. Given the estimates of the expected RSSI for each channel at every point in the region of interest, a major challenge of any localization algorithm, using the received power spectrum, is to determine a subset of the transmitters in the region that may correspond to the observed RSSI. We call the step that filters these transmitters from all the known FM transmitters in the region, as the candidate selection phase. This step helps us narrow down the search from the whole region of interest to within a few hundred square miles. The amount of reduction in the search area that we get, depends heavily on the distribution of the FM transmitters.

We denote the model representing the expected power at every point in a region of interest by a hash table $HT$. In this hash table $HT$, the keys are the location hashes (implemented using the python library geohash)\cite{83} for each location in the region of interest and the value for each hash is the estimate of the expected FM power spectrum, represented as a vector in $\mathbb{R}^{101}$. The received signal strength of each channel is denoted by $\psi_i, i \in [1, \ldots, 101]$. Note that at a location of interest, the $i$-th channel may or may not be available, $i \in [1, \ldots, 101]$ and hence the set of observed FM channels at a given location is a subset of the 101 possible FM channels.

One important aspect of our Location estimation using Signal Integration (LoSI) algorithm is the fact that it uses feature engineering to succinctly represent the received signal in a lower dimensional space. As noted above, the observed RSSI at any location of interest is a 101-dimensional vector. We use a simple feature engineering technique to embed the observed FM spectrum at a point of interest, in a low dimensional feature space with the aim of reducing the dimensionality and to get a robust representation of
the same. The goal is to find relevant features from the observed spectrum such that they are able to distill the identifying information, which can be used for positioning. We use a very simple feature descriptor that we describe in Section 3.2. Given the received power spectrum $\psi$, the extracted features are represented by $\rho$. Now we are ready to describe the LoSI system in detail.

We start with an algorithm for estimating the FM signal strength model at each point in the region of interest. Then we describe the testing phase where we describe the actual localization algorithm.

### 3.1 Model estimation

Our goal is to be able to learn a model that predicts the expected power across the FM channels at a point of interest $x$ based on the knowledge of nearby FM transmitters: the power at which they are transmitting and the description of a region around the transmitter that receives a fixed power from the transmitter (the power polygon). If the model is perfect, then the observed FM power spectrum at $x$ should agree exactly with the predicted spectrum. However, there are problems that prevent this straightforward comparison, which is done in any “fingerprinting” based approach, from being used for positioning in our setting. First, the model is not perfect and hence the predicted spectrum at a point $x$ may not be exactly the same as the observed spectrum at the point. Furthermore, the observed power spectra at a point is corrupted by noise and hence not all the observed data are accurate. The noise is a result of multi-path, hardware issues with the use of an uncalibrated receiver, interference with other Electro-Magnetic (EM) waves, as well as differences in ambient temperature and humidity. Finally, there are differences due to the fact that the learning phase uses data from actual fixed power polygons. These are collected with possibly several different devices, the nature and characteristics of which are unknown. Moreover, they are different from the device being used to sense the spectrum for localization during the testing phase. In short, the use of information from heterogeneous devices with unknown characteristics induces uncertainty.

For this paper, we estimated the expected FM power spectrum in the region of interest, the region is divided into regions (using geohashes$^{[84]}$ of a fixed precision). Going forward we denote this collection of geohashes by $D$. The data for estimating the expected FM spectrum consist of information about FM transmitters in the region of interest. More specifically, we assume that we know the geo-location of the transmitter $t$, the radius of its influence denoted by $r$, and the $p$-dBu contour plot for the transmitter denoted by $p$. We represent this contour plot as a $star$ polygon$^{[85]}$ with 360 vertices. Given this information, for every transmitter, we can estimate the expected power at all geohashes within its radius of influence. For points that are in the radius of influence of several transmitters, we get the total estimated power by adding the contributions from each transmitter. For a given transmitter $t$ and a point $x$ in the circle centered at $t$ of radius $r$ (the region of influence of the transmitter), we first compute the intersection of the line joining the points $x$ and $t$ with the $p$-dBu polygon $p$. This intersection can be computed using a line sweep algorithm$^{[86]}$. After this step, the problem reduces to that of interpolating or extrapolating the power at $x$, using the value of the power at the intersection, which is known to be $p$ dBu.

Now we formally describe this algorithm. It takes a list of tuples as input. Each tuple is of the form $(t, r, p)$. We use the following functions for Algorithms 1 and 2:

1. **Location** $(t)$ returns the coordinates of the transmitter $t$.
2. **RayIntersect** $(p, Location(t), x)$ calculates the intersection of the line (Location $(t), x$) from the tower location $t$ to a pixel location $x$, with polygon $p$ in local stereographic projection$^{[85]}$, which is centered at the tower location. The polygon is preprocessed to allow binary search$^{[85]}$. This allows us to implement this primitive in $O(n \log n)$ time$^{[87]}$. In our case $n = 360$ which is the number of sides of the polygon.
3. **GetPixels** $(t, r)$ gets all geohashes inside the coverage radius $r$ centered at the tower location $t$.
4. **GenGeoHash** $(t)$ generates all geohashes inside
that are invariant to the aforementioned conditions. Of observed RSSI directly. which is uncalibrated and finally the observed RSSI is corrupted by noise same device under different environmental conditions RSSI may be different from device to device and on the same frequency. Given a point of interest computing the power at a point

The goal of this step is to determine robust features from the observed FM spectrum for a given region of interest. (5) FreqIndex (t) returns the integer index in [1, 101] for a tower given its frequency. (6) Freq (t) returns the frequency of the tower given by t. (7) Power (t) returns the transmitted power from the tower t in kW. (8) Line 6 of Algorithm 2 uses a formula for computing the power at a point x. The formula being used can be found in Ref. [40] and is a standard for wireless networks. (9) Aggregate (d, a) takes the previous power reading a converted into linear scale and sums it with the current voltage reading d. Finally, it converts back to logarithmic scale, and returns the aggregated new power.

3.2 Dominant channel descriptors

Given a point of interest x that we want to localize, our algorithm starts by looking at the RSS values of the FM spectrum received at that point. Using the observed RSS values for the 101 channels directly might be problematic: the data is high dimensional, the observed RSSI may be different from device to device and on the same device under different environmental conditions and finally the observed RSSI is corrupted by noise which is uncalibrated[77]. All of this precludes the use of observed RSSI directly. Therefore, we need to find features that are invariant to the aforementioned conditions. Moreover, the features should be such that they distill enough locality information from the observed RSSI to be able to localize the point of interest. Thus the features should have enough discriminative power to differentiate between different locations with similar RSSI patterns. Intuitively, given the observed RSSI at a location, two characteristics of the received spectrum discriminate between different locations, namely, the received channels of the FM spectrum above the noise level at that location and the power received in those channels. It must be noted that there is a non-zero probability for two locations, in the region of interest, to have the same channels above the noise level at the same power. So there is a non-zero probability that the estimate of the location might be wrong using features based on these two characteristics. However, if the number of such features is large enough, then the probability of a collision resulting in mis-prediction is low. By ensuring that channels whose power is significantly above the noise level are selected, one can further reduce the uncertainty. Based on these considerations we decided to use features that use these characteristics for representing the received FM spectrum at a given region and carry out the location inference in this feature space. We call these features the Dominant Channel Descriptor (DCD) features.

Intuitively, to find DCD features, the extraction algorithm looks for channels that significantly “dominate” its local neighborhood in terms of received signal strength, thereby making sure that the power at these channels is significantly above the noise level. It also ensures that they have enough received power to be discriminative for location inference. Given the observed spectra denoted by ψ, the i-th channel is selected as a DCD feature if and only if it satisfies the condition:

\[
\min(\psi_i - \psi_{i-1}, \psi_i - \psi_{i+1}) > \nu,
\]

for \(i \in [2, \ldots, |\psi| - 1]\) and some constants \(\nu > 0\), that should ideally depend on the data and the device used for sensing the spectrum. Note that there are two boundary cases: the first one occurs when \(i = 0\) and the second one occurs when \(i = |\psi| - 1\). In case when \(i = 0\), the algorithm checks whether \(\psi_i - \psi_{i+1} > \nu\). Similarly for the case where \(i = |\psi| - 1\), it checks \(\psi_{i} - \psi_{i-1} > \nu\) to determine whether \(\psi_i\) is a DCD feature or not.

The feature extraction algorithm returns all the DCD features from the observed FM spectrum for a given
location, after sorting them by the value of \( \min(\psi_i - \psi_{i-1}, \psi_j - \psi_{j+1}) \) or the corresponding values for the boundary cases. We represent the extracted DCD features by \( p \). We also note that given the vector of selected DCD features, we can build a 101-dimensional bit vector corresponding to the observed FM spectrum. In this 101-dimensional bit vector, 1 is used for a channel that is selected as a DCD feature and 0 is used otherwise. We also use \( p \) to denote this DCD indicator vector, abusing the notation. But the particular vector being referred to should be clear from the context. The details of the algorithm are formally described in Algorithm 3.

We use the following in Algorithm 3:

1. Enumerate \( \psi \) accepts a list of powers \( \psi \) and returns a list of tuples \((i, \psi_j)\) for every element in \( \psi \).
2. \( \text{MinK} (p, k) \) returns the top \( k \) tuples based on the value of \( j \), given a list of tuples of the form \((p_w, j, i)\) where \( p_w \) is the power, \( j \) is a floating point number, and \( i \) is an integer.

The DCD feature extraction algorithm uses two parameters, namely \( k \) and \( v \) that are set to two cutoff values. These parameters control the number of DCD features that will be finally selected by the system. In our experiments we empirically found that the values \( k = 25 \) and \( v = 8 \) worked best and captured most of the relevant DCD features.

### 3.3 Inference candidates

The next step of our localization algorithm is candidate selection. Given the extracted DCD features from the observed RSSI at a location of interest, this step determines the candidates for location inference using the estimated model of expected FM spectra. More precisely, this step identifies locations across the entire region of interest that have the potential to generate the same DCD indicator as the current location. Thus this step eliminates a location from consideration, if the estimate of the expected spectrum at the location is such that it cannot generate the same DCD indicator as the observed one. Note that having the same DCD indicator at a known location only makes it a candidate for the location of interest. Further analysis is required to determine if it is indeed the location we seek. The selected locations for the final inference are called Inference Candidates (IC).

The problem of selecting ICs can be posed as a **Subset Query Problem**. The subset query problem is stated as follows: Given a set \( V \) of \( n \) vectors over \( \{0, 1\} \), build a data structure, which for a query vector \( q \) over \( \{0, 1\} \), detect if there is any vector \( p \in V \) such that \( q \) is a subset of \( p \) (in other words, \( p \land q = q \)). Due to its high importance, the subset query and partial match problems, as they are generally called, have been investigated for quite a while. It is believed that the problem inherently suffers from the “curse of dimensionality”, that is, there is no algorithm for this problem which achieves both “fast” query time and “small” space [88] in the general setting.

In order to formulate candidate selection as a **Subset Query Problem**, we start with the estimated model of the expected FM spectra in the region of interest, \( \mathcal{H} \mathcal{T} \). We recall that in \( \mathcal{H} \mathcal{T} \) each location is indexed by a geohash and corresponding to each geohash we have an expected FM spectrum recorded as a 101-dimensional vector. Using this we compute the DCD containment model denoted by a selection subset \( \mathcal{S} \mathcal{S} \), which is a dictionary where the keys are 101-dimensional bit vectors and the associated values are a list of geohash.

In order to compute \( \mathcal{S} \mathcal{S} \), we proceed as follows:

For the estimated power vector \( \xi \) corresponding to each geohash \( g \in \mathcal{H} \mathcal{T} \), we create the *Received Channel Indicator vector* (RCI) \( b \) such that the value at the \( i \)-th position is 1 if the estimated power corresponding to that channel is not \(-\infty\). The RCI \( b \) is used as the key for \( \mathcal{S} \mathcal{S} \). If \( b \) is already present in \( \mathcal{S} \mathcal{S} \), as a key, then as the current geohash generated this RCI vector, it is appended to the list of locations for this RCI. Otherwise a new entry is created for this RCI. The system stops when the entire region of interest has been processed.

Finally, \( \mathcal{S} \mathcal{S} \) is used for computing the ICs as follows: Given the DCD indicator vector for the

---

**Algorithm 3** Detect and output DCD features from the RSSI values

| Input: | \( \psi \): observed FM power spectrum |
|---|---|
| \( \mathbf{1:} \) | \( \text{function} \ 	ext{EXTRACTDCD}(\psi) \) |
| \( \mathbf{2:} \) | \( p \leftarrow [\] \) \> For DCD |
| \( \mathbf{3:} \) | \( k \leftarrow \text{cutoff1} \) |
| \( \mathbf{4:} \) | \( v \leftarrow \text{cutoff2} \) |
| \( \mathbf{5:} \) | \( \text{for} \ (i, p_w) \in \text{Enumerate}(\psi) \text{ do} \) |
| \( \mathbf{6:} \) | \( l_g \leftarrow \min\{p_w - \psi_{i-1}, p_w - \psi_{i+1}\} \) |
| \( \mathbf{7:} \) | \( \text{if} \ l_g > v \text{ then} \) |
| \( \mathbf{8:} \) | \( p_{-\text{add}}\{(p_w, l_g, i)\} \) |
| \( \mathbf{9:} \) | \( \text{end if} \) |
| \( \mathbf{10:} \) | \( \text{end for} \) |
| \( \mathbf{11:} \) | \( p \leftarrow \text{MinK}(p, k) \) |
| \( \mathbf{12:} \) | \( \text{return} \ p \) |
| \( \mathbf{13:} \) | \( \text{end function} \) |
observed RSSI $\psi$, the problem of locating the ICs reduces to that of finding the geohashes corresponding to the keys (RCI vectors) in $SS$, that contain the DCD indicator vector as a subset. This can be done using a simple AND operation in $\mathcal{O}(1)$ time. The geohashes corresponding to the matched RCI vectors are collected as the ICs. We skip the pseudo code for this because of its simplicity and space constraints. This step is assumed to be implemented using a subroutine InferCandidates ($\rho, SS$).

### 3.4 Location inference

The actual localization algorithm takes the estimated model $\mathcal{H}T$ and the DCD containment model $SS$ as input. It collects the observed FM power spectrum $\psi$ using an RTLSDR and then outputs an approximate location of the point at which the data was collected. The algorithm is given below. It uses the Euclidean distance between the observed power spectrum and the estimated power spectrum of the ICs restricted to the space determined by the DCD features from the observed spectrum.

In Algorithm 4, we use the following:

1. AcquireRTLPower () returns the power spectrum obtained from an RTLSDR.
2. EuclidDist ($\gamma, \psi, \rho$) returns the Euclidean distance between the spectrum $\gamma$ and $\psi$, restricted to the DCD feature space.
3. Power ($i$) returns the power spectrum of the geohash $i$.
4. Calibrate ($\psi$) calibrates the observed power spectrum using linear regression (see Section 4.2).

We must point out here that the complexity of the model estimation step of our method is linear in the total number of hashes in the region of interest. Finally, the worst case complexity of the location estimation step is also linear in the total number of hashes in the region of interest. We have provided a detailed complexity analysis of the algorithms involved in Section 3.6.

On important aspect of the localization algorithm is the choice of the Euclidean distance. This can be explained in a Bayesian Decision Theoretic framework. Bayesian methods for localization have been studied before[33]. In what follows we describe the Bayesian approach to localization and show how the use of the Euclidean metric can be justified in the context of this framework.

### 3.5 Bayesian decision theoretic analysis

The received FM signal for a single channel obtained from a measuring device is prone to error. Due to this error, the reading at a given location of interest is not exactly the same as the value estimated from the model $\mathcal{H}T$. In this section, we develop a Bayesian decision theoretic framework to justify our inferential procedure. As before let $D$ be the set of all possible locations. Let $\psi$ be the observed FM spectrum at a location of interest. We assume that $\psi \in O$, where $O$ is the set of all possible FM power spectra. Let $\mathbb{R}$ denote the set of all real numbers and $\mathbb{R}^{101}$ be the set of all possible FM power spectra. Our goal is to choose an estimator $T(\psi)$ of the location $x \in D$, which is a function of the observed spectrum $\psi \in \mathbb{R}^{101}$. To measure the discrepancy of the estimator from the actual location, we introduce a loss function $L$ which is a mapping from $\mathbb{R}^{101} \times D \rightarrow \mathbb{R}$. Clearly, $L$ takes in two arguments, the observed power spectrum $\psi$ and a tentative location $x \in D$ and then outputs a real number as a loss. Assume that the RSSI at the location of interest has a probability distribution given by $P$ and the location is assigned a prior $\pi$. Then the Bayes risk is defined as Ref. [89]:

$$\int E\{L(T(\psi), x)\} \pi(x)dx,$$

where $E$ is computed based on the data-generation mechanism assumed by the model under $P$. By an application of Fubini’s theorem[90], the Bayes risk can be re-written as Ref. [89]:

$$E \int \{L(T(\psi), x)\} \pi(x|\psi)dx,$$

where $\pi(x|\psi)$ is the posterior distribution of the location conditional on the observed power spectrum. In the Bayesian decision theoretic framework[89], the goal is to find the location $x \in D$, that minimizes the $\int \{L(T(\psi), x)\} \pi(x|\psi)dx$ for each $\psi$. Formally, we seek to find location $x \in D$ as $\hat{T}(\psi)$ such that for a given

---

Algorithm 4  Output approximate localization

**Input:** $\mathcal{H}T$: estimated model of expected power  
**Input:** $SS$: the DCD model

```python
1: function LOCALIZE($\mathcal{H}T, SS$)  
2:   $\psi \leftarrow$ AcquireRTLPower ()  
3:   $\rho \leftarrow$ ExtractDCD ($\psi$)  
4:   $M \leftarrow$ InferCandidates ($\rho, SS$)  
5:   $o \leftarrow \text{argmin}_{O} \text{EuclidDist}(\text{Power}(i), \text{Calibrate} (\psi), \rho)$  
6:   return Geohash.Decode ($o$)
7: end function
```
observed power spectrum $\psi$, $\hat{T}(\psi)$ satisfies:

$$
\hat{T}(\psi) = \arg\min_{T \in \mathcal{E}} \int \{L(T(\psi), x)\} \pi(x) \pi(\psi) \, dx,
$$

where $\mathcal{E}$ is the set of all estimators of $x \in D$. In the special case when $L$ is chosen to be a $0 - 1$ loss function\(^{\text{[89, 91]}}\), i.e.,

$$
L(T, x) = \begin{cases} 
1, & \text{if } T = x; \\
0, & \text{if } T \neq x,
\end{cases}
$$

for any estimator $T \in \mathcal{E}$, we have

$$
\arg\min_{T \in \mathcal{E}} \int \{L(T(\psi), x)\} \pi(x | \psi) \, dx = \arg\max_{x \in D} \pi(x | \psi).
$$

Now the problem reduces to the Maximum A Posteriori (MAP) decision rule. For the rest of the discussion we will use the MAP rule to explain the use of the Euclidean distance metric.

As before let the DCD features in the observed power spectrum at the location of interest $\psi$ be denoted by $\rho$. Note that $|\rho| \leq 101$, more specifically we assume that $|\rho| = l \leq 101$. Let us denote the channels represented in $\rho$ by $\epsilon$. We note that $\epsilon$ is a vector whose members are the channels that are represented in $\rho$. Thus we also have $|\epsilon| = l \leq 101$.

We also assume that we have access to the estimated model of the expected power spectrum, denoted by $\mathcal{H}T$, for the region of interest. Let us suppose that we have $n$ geohashes and at each geohash the estimated data is given by $\xi_i = \mathcal{H}T[i]$, $i \in [1, \ldots, n]$. We also assume that given the DCD features in the observed power spectrum at the location of interest, the ICs are denoted by $M$. Note that at each IC in $M$, the powers corresponding to the DCD features in the observed power spectrum is non zero. Let us denote each IC in $M$ by $x_i$, $i \in [1, \ldots, |M|]$. Then the problem of localization can be formulated as that of finding a mapping from $\rho$ to $x_i$ for some $i \in [1, \ldots, |M|]$. This in turn can be formulated as the following maximization problem:

$$
\hat{T} = \arg\min_{i \in [1, \ldots, |M|]} \pi(x_i | \rho) \tag{2}
$$

where $\hat{T}$ represents the predicted location. We observe that this is exactly the formulation that we arrived at using the Bayes decision theoretic framework.

Now using Bayes’ rule, we have

$$
\pi(x_i | \rho) = \frac{P(\rho | x_i) \pi(x_i)}{P(\rho)} \propto P(\rho | x_i) \tag{3}
$$

Observe that since $\pi(x_i)$ is uniformly distributed, maximizing the posterior $\pi(x_i | \rho)$ with respect to $x_i$ is equivalent to maximizing the likelihood $P(\rho | x_i)$. Hence Bayesian inference with posterior mode as the point estimator coincides with the maximum likelihood estimate in this case. Let us write the DCD features of the observed power spectrum as $\rho = \{\rho_j : j = 1, \ldots, l\}$. A single received power reading $\rho_j$ at channel $j$, observed at location $x$, can be modeled as a random variable $S_{ij}$ which is assumed to have a Gaussian distribution\(^{[92]}\) as follows.

The power $S_{ij}$ at location $x_i$ is assumed to be explained by the following linear model\(^{[93]}\):

$$
S_{ij} = \mu + V_{ij} + \epsilon_{ij},
$$

where $\mu$ is a generic factor that contributes to the observed power for each of the channels, $V_{ij}$ is the actual power contribution at the location $x_i$, and $\epsilon_{ij}$ is an idiosyncratic error term. A standard choice for the distribution of $\epsilon_{ij}$ is the Gaussian distribution with mean 0 and variance $\sigma^2$. In principle, one can allow flexible distributions, for example, a $t$ distribution which is more heavy tailed than a Gaussian distribution with heteroscedastic variance $\sigma_j$ specific to the $j$-th channel. Also, since the distribution of the DCD features is localized in a compact domain, one might consider truncated normal distribution as the error. However, it is a well known fact that the estimation of the mean of $S_{ij}$, i.e., $\mu + V_{ij}$, is robust to moderate changes to the error distribution. Hence, we stick to a Gaussian distribution with a common variance $\sigma^2$ for this analysis. Since $V_{ij}$ cannot be estimated based on just one replicate of the observation $S_{ij}$, we propose to have a regularization on $V_{ij}$ based on the understanding that neighboring locations will have similar predicted power for the channel $j$. Specifically, this regularization is supported by the estimated value of the channel $j$ at location $x_i$ and its neighborhood. Observe that the estimated value of the channel $j$ at location $x_i$ can be interpreted as an estimate of $\hat{V}_{ij}$ of $\mu + V_{ij}$.

Now the conditional distribution of $S_{ij}$, given $\hat{V}_{ij}$, can be written as

$$
f_{S_{ij}|\hat{V}_{ij}}(s) = \frac{1}{\sigma \sqrt{2\pi}} \cdot e^{-\frac{(s-\mu-\hat{V}_{ij})^2}{2\sigma^2}}, \quad -\infty < s < \infty \tag{4}
$$

This is a consequence of the distributional assumptions that we have made about the distributions of the random variables $S_{ij}$.

The conditional probability of the DCD features, given an IC, can be written as

$$
P(\rho | x_i) = \prod_{j=1}^{l} f_{S_{ij}|\hat{V}_{ij}}(S_{ij}) \tag{5}
$$
Note that we have made a simplifying assumption that the distributions across the channels are independent. This assumption holds because of the way the FM signals work. Otherwise, the signals from the different channels would cause interference and it would not be possible to listen to a channel without getting interference from the others.

Now substituting the density for $f_{S_{ij}|V_{ij}}$, we obtain:

$$P(\rho|x_i) = \prod_{j=1}^{l} \left\{ \frac{1}{\sigma \sqrt{2\pi}} \cdot e^{-\frac{(S_{ij}-\hat{V}_{ij})^2}{2\sigma^2}} \right\}$$

This in turn can be written as

$$P(\rho|x_i) = \left( \frac{1}{\sigma \sqrt{2\pi}} \right)^l \exp\left\{ -\sum_{j=1}^{l} \frac{(S_{ij}-\hat{V}_{ij})^2}{2\sigma^2} \right\}$$

Now taking logarithm on both sides we get the final predicted location as

$$\hat{\rho} = \arg\max_i \left\{ l \cdot \ln \frac{1}{\sigma \sqrt{2\pi}} - \sum_{j=1}^{l} \frac{(S_{ij}-\hat{V}_{ij})^2}{2\sigma^2} \right\}$$

But this is exactly equivalent to

$$\hat{\rho} = \arg\min_i \sum_{j=1}^{l} (S_{ij} - \hat{V}_{ij})^2$$

which is nothing but the minimum Euclidean distance between the DCD features in the observed power spectrum $\psi$ and the estimated power spectrum at the ICs. This mathematically establishes the optimality of using the Euclidean distance for the nearest neighbor search.

Consistency of the estimator $\hat{\rho}$. It is important to study the consistency of the estimator $\hat{\rho}$ as the number of selected DCD features, $l$, goes to $\infty$. We use the following heuristics to show that under reasonable assumptions on the true data-generation mechanism $P$, $\hat{\rho} \rightarrow x_0$ with high probability as $l$ goes to $\infty$ assuming that the true location is $x_0$.

Consider the situation when the true distribution of $\epsilon_{ij}$ is possibly different from Gaussian, but with mean 0 and sub-exponential tails. Then it is widely known that the least-squares estimate of the mean is robust to the distributional assumptions and one can safely assume a Gaussian distribution for $\epsilon_{ij}$. Then the distribution of $\sum_{j=1}^{l}(S_{ij} - \hat{V}_{ij})^2$ will concentrate around 0 provided the location $x_i$ is closest to the true location $x_0$. Hence $\hat{\rho} \rightarrow x_0$ with high-probability under $P$.

To better understand the robustness of our approach subject to misspecification of the error distribution, we consider the behavior of our estimator when the error is heavy-tailed. As an example, we consider the log-normal distribution on the absolute value of the error with location and scale parameters $m_\mu$ and $m_\sigma$, denoted by $LN(m_\mu, m_\sigma^2)$. Recall that $LN(m_\mu, m_\sigma^2)$ has a density for $x > 0$:

$$q(x|m_\mu, m_\sigma) = \frac{1}{\sqrt{2\pi x m_\sigma}} \exp\left\{ -(\log x - m_\mu)^2/(2m_\sigma^2) \right\}$$

(10)

It can be seen from Eq. (10) that $q$ has tails much heavier than a Gaussian distribution, being of the order $e^{-(\log x)^2/(2m_\sigma^2)}$, but has moments of all orders. If the true error is distributed as $LN(m_\mu, m_\sigma^2)$ and the practitioner is aware of that fact and uses a logarithmic loss, then for any estimator $\hat{\rho}$, the smallest possible variance is bounded below by the Crãmer-Rao lower bound[94] which is given by

$$\text{Var}(\hat{\rho}) \geq \frac{m_\sigma^2}{\Sigma}$$

(11)

Now, if the practitioner is not aware of the misspecified error, still uses a squared error loss function, and assumes that the error is distributed as $\mathcal{N}(V, \sigma^2)$, from Theorem 1 of Ref. [95], one easily obtains for a generic response $S$ with mean $V$ and an estimator $\hat{\rho}$ of $V$:

$$[\text{Var}(\hat{\rho})]^{-1} \leq l E_0 \left\{ \frac{f(S|V, m_\sigma)}{q(S|V, m_\sigma)} \right\}^2 \frac{(\log |S| - V)^2}{m_\sigma^4}$$

Now it is easy to see that

$$f(S|V, m_\sigma) \approx \frac{m_\sigma}{\sigma}, \text{ for moderate } S,$$

$$f(S|V, m_\sigma) \rightarrow 0, \text{ for large } S.$$

Hence

$$[\text{Var}(\hat{\rho})]^{-1} \leq \frac{l m_\sigma^2 m_\sigma^2}{\sigma^2} = \frac{l}{\sigma^2}$$

(12)

From Eq. (12), the variance of the estimator $\hat{\rho}$ when the error is misspecified is lower bounded by $\sigma^2/l$ where $\sigma^2$ is the variance of the error of the assumed model. As long as $\sigma \approx m_\sigma$, the estimator is robust in the sense that it continues to have the optimal variance bound Eq. (11) as if the error is known to be log-normal.

Robust extension of our method. In presence of outliers or when the distribution of $\epsilon_{ij}$ has heavier than exponential tails, the Gaussian distribution is not robust. In that case the distribution of $\sum_{j=1}^{l}(S_{ij} - \hat{V}_{ij})^2$ will not concentrate around 0 even if the location $x_i$ is closest to the true location $x_0$. A possible solution in that case will be to consider absolute-deviation loss:

$$\hat{\rho} = \arg\min_i \sum_{j=1}^{l} |S_{ij} - \hat{V}_{ij}|$$

(13)
or Huber’s loss\cite{huber1964robust}:

\[
\hat{T} = \arg\min_i \sum_j L_\delta(S_{ij} - \hat{V}_{ij})
\]

where

\[
L_\delta(x) = \begin{cases} 
\frac{1}{2\alpha^2}, & |x| \leq \delta; \\
\delta(|x| - 0.5\delta), & |x| > \delta
\end{cases}
\]

where \(\delta\) is a pre-specified small value. This function is quadratic for small values of \(x\) and linear for large values, with equal values and slopes of the different sections at the two points where \(|x| = \delta\). Huber’s loss ensures that the final estimate \(\hat{T}\) is robust to outliers in the data.

### 3.6 Complexity analysis

Our localization system operates in two phases: the offline phase estimates the FM map and the online phase calculates the location of an user from the sensed FM spectrum. As a result, we provide the complexity analysis of these two phases separately. We start with the offline phase.

**Complexity of offline phase.** Let us assume that there are a total of \(m\) geohashes in the region of interest. Referring to the pseudocode listing for the offline large scale FM map estimation algorithm (Algorithm 1), we note that line 2, that generates the geohashes, takes \(O(m)\) time. Similarly the for loop in line 3 also takes \(O(m)\) time. Further, let us assume that there are \(t\) FM transmitters in the region of interest. It must be noted that in general \(t\) is a constant and \(t < m\). Thus the for loop in line 6 executes \(O(t)\) times. However, the for loop in line 7 can execute \(O(m)\) times in the worst case and hence the total running time for the two for loops is \(O(tm) = O(m)\) as \(t\) is assumed to be a constant. If we assume that the distance computation primitives in lines 10 and 11 are constant time operations, then all the work inside the for loops can be done in constant time. It must be pointed out that for line 9, though the intersection can be computed in \(O(n \log n)\) time where \(n\) is the number of sides of the polygon \(p\), in our case \(n = 360\) and hence this operation is also constant time. Thus, the entire estimation algorithm works in \(O(m)\) time.

It must be pointed out that though the algorithm runs in linear time, in general \(m\) is very large, for example when the region of interest is an entire country like the United States. As a result the empirical execution time for this step is expected to be large in general.

**Complexity of online phase.** The online phase of our algorithm has four steps: acquiring the FM spectrum which is a constant time operation, computing the DCD features from the spectrum, finding the inference candidates, and finally computing the Euclidean nearest neighbor. For the computation of the DCD features, the algorithm makes a single sweep through the 101-dimensional vector of the FM power spectrum. For each channel, that is each allowed FM frequency, the algorithm checks whether it has enough power with respect to the adjacent channels. This is a constant time operation. The detected DCD features are stored such that the top \(k\) can be easily selected. This can be done by maintaining a sorted list of the identified DCD candidates. Since the list can have length at most 101, this operation is also constant time. Thus, given the sensed spectrum of 101 FM channels, the DCD features can be computed in \(O(1)\) time.

Selection of the inference candidates can be done in \(O(m)\) time, as before \(m\) is the total number of geohashes in the region of interest. To see this note that given the DCD feature vector, we first convert it to a binary vector that is \(O(1)\) operation. Then for each geohash, we compute a binary vector from the estimated FM spectrum at that location. Each such operation is \(O(1)\) and there are \(O(m)\) geohashes.

Finally the computation of the inference candidates involves another constant time operation, the AND operation with each of the \(O(m)\) estimated FM spectrum. Thus, the total time for all these operations together is \(O(m)\).

Finally the actual nearest neighbor search can be implemented in several different ways. The complexity of the implementation will depend on the algorithm used. However it must be noted that the computation of exact nearest neighbors in spaces having more than 8 dimensions is considered to be hard and there are few algorithms that can do better than the brute force search. As we carry out this search in the space of DCD features, the dimension of the search space is less than 101 and more than 5 (as we require at least 5 DCD features to be able to localize). Hence we use a brute force search that maintains a pointer to the minimum as new elements are inserted into the array. This will execute in \(O(m)\) time. Hence the total running time of the online phase is \(O(m)\). Note that this might seem to be a very pessimistic estimate as in general the number of inference candidates will be far less than \(m\). Thus in practice, if the expected number of inference candidates is \(n_c \ll m\), then the running time of the nearest neighbor search is \(O(n_c)\). However, the total
running time for the online phase is still $O(m)$ as it is dominated by the time for computing the inference candidates.

4 Experimental Setup and Results

For testing the localization algorithm, FM broadcasting signals were collected using different models of R820T2 SDR and DVB-T NESDR Mini 2 software defined radio tuner with an ESD-Safe Antenna input. This particular receiver and antenna were chosen because of reduced Size, Weight, Power, and Cost (SWaP-C) requirements and ease of use. A Globalsat BU-353-S4 GPS receiver was used to receive the GPS positioning information for cross validation purposes. All the experiments reported in this paper were done on systems running Ubuntu 15.04 and 16.04 with Intel i7 2.6 GHz CPU (released 2012 and later), 16 GB RAM, and a 256 & 512 GB SSD Drive. The data collection setup is shown in the appendix for interested readers.

We used Python extensively for building our software stack. We used rtl_power (https://github.com/keenerd/rtl-sdr.git) to gather the power spectrum using the SDR. We used the Computational Geometry Algorithms Library (CGAL) for processing geometry information. We used numpy, scipy, geopy, and many other Python libraries for implementing our algorithms.

4.1 Model estimation

Data obtained from FCC was used for our model estimation step (Algorithm 1). We used two models, the first was estimated using data obtained from FCC in 2015 and the second was computed using FCC data from April 2017. The first and second models contain a total of 18,339 and 22,491 FM stations in the CONUS, respectively. The range of these FM stations varies depending on their FCC classification. The total time taken by the estimation algorithm per FM station is <10 seconds on average and is done in parallel over four cores (and eight hyper threads). The current estimation algorithm is run on geohases with precision 5 (approximately equal to a distance of 2.36 miles between two consecutive rectangular cells). Our model generation takes less than a day to run on our current system. It is not optimized, written in Python, and can be sped up significantly. Visualization of a sample transmitter model is shown in Fig. 3. Once all the transmitter models have been estimated, we aggregate them into one large hash table which maps geohases to estimated power spectrum.

4.2 Testing data collection

We collected test data on multiple days in 2015 and 2017, spanning Tallahassee and Crestview (including Niceville and Shalimar, which are close together), FL, USA (Figs. 4 and 5). The data was collected by multiple co-authors using multiple cars and multiple RTL SDRs from different manufacturers. Each data sample consists of a GPS location and the FM power spectrum at that location. The data was collected on multiple days under different weather conditions with outside temperatures ranging from 60 degrees to 90 degrees Fahrenheit. Moreover, we deliberately collected data on days that were sunny or cloudy and also days where there was a drizzle as well as torrential rain. This was done to test the robustness of our features to compensate for these changes in environmental conditions. For people who are not acquainted with the region, Tallahassee is the capital of Florida and is located about 200 miles away from Crestview which is a small city in the Florida panhandle. Crestview is about 25 miles away from the city of Niceville which is about 11 miles away from the city of Shalimar. There are two routes between Tallahassee and these cities that we used. One of the routes is through the Interstate 10
which is the main highway that runs east-west across the southern United States. This is a high-speed highway with a speed limit of 70 miles an hour. We followed this route to Crestview and then followed highway 85 to Niceville and finally to Shalimar. We also collected data on the return journey using the same route. Figure 4 shows the route taken using Interstate 10 and the collected data points.

For the other route (which was used in 2015 as well during data collection in rain), we drove from Tallahassee to Niceville using Florida Highway 20, which has a speed limit of 60 miles an hour for the most part. From there we went through Shalimar to Crestview using highway 85. Finally, for the Tallahassee 2017 data, we drove around the Southwood community neighborhood of Tallahassee, which is both a residential and a commercial area. Figure 5 shows the route taken during this 2017 session and the associated data points.

The parameters used for rtl_power are integration time of 20 seconds, a tuner gain of 8 dB, crop percentage of 30% (discards data at the edges), and bin size of 1000. We use rtl_power in single shot mode, and do not keep it continuously running. These parameters were derived after numerous experiments and were kept constant throughout this study.

4.3 Calibration

Our localization method uses an estimated map of expected FM power spectrum, created using data obtained from FCC and uses it to localize FM spectra obtained using an uncalibrated RTL SDR. As we use data collected from different modalities together and due to the fact that our receivers are uncalibrated, there will be a mismatch between the power output from our estimation algorithm and the power obtained using the RTL SDR. A calibration step is required to bridge this divide and to account for factors like SDR reading error, receiver antenna gain error, and environmental conditions, which affect the observed RSSI but are beyond our control.

We used several different calibration methods, all of them were learned using randomly selected subsets of data collected using a randomly selected RTL SDR. For each calibration method, the learned parameters were used for localizing with all the other devices, without explicitly calibrating each one of them.

**Linear regression-based calibration.** For the first approach we used linear regression to map the observed
power to the estimated power. For a set of randomly selected locations, we considered the observed power at a given frequency (channel) against the power output by the estimation model at that frequency, ignoring the missing frequencies (if any) from the model. We used this to learn a regression line for calibration. We experimented with different sampling methods as follows: (1) First we randomly selected 97 data points which gave us a straight line $y = 0.91x - 45.66$. We call this the Fixed Size Calibration (FSC). The results of using this on one geohash are shown in Fig. 6. (2) For the second approach we randomly selected a fixed percent of the points and experimented with different splits of the data. However the accuracy was comparable to the first. (3) For the third, we randomly selected a subset of the data points and used a linear regression in the DCD feature space. Figure 7 shows the results. The resulting regression line is given by $y = 0.815817068492x - 48.2935730379$. We report results with both (1) and (3).

**Decision Tree regression.** One of the problems with the above calibration method is the fact that it is based on the assumption that a simple straight line can explain away the differences between the observed and the predicted data. However, as seen from Fig. 7, this is not true in general. In order to address this issue, we used a more generic regression technique, namely Decision Tree regression\[97\]. Decision trees are better at approximating functions that are non-linear and work by splitting the data into smaller subsets. For each subset it learns the regressor and hence the final regressor that is learned is a “piecewise” approximation to the unknown non-linear function. We experimented with Decision Tree regression using different test-train splits and depths. We show the results of using a Decision Tree regressor trained on a randomly selected set of 50% of the data with a depth of 5. We also used Support Vector Regression (SVR)\[98\] but the results from SVR were worse than that of the Decision Tree and hence we do not report them here. It must be noted that though we report the results using the decision tree-based calibration, it does not do as well as the linear regression-based calibration. Our guess is that the decision tree-based regressor over-fits the data and hence the generalization error is higher.

### 4.4 Temporal variation

There are two types of variations that might affect the positioning accuracy of our method. First, over time new FM transmitters are added and old ones are removed and hence the estimate of the expected FM signal strength at a given point in the continental US may change. On the other hand, the RSSI values obtained from the SDR varies with time because of reading errors from the SDR, moving objects along...
the transmission path, environmental conditions, and variable transmission power of the FM transmitters.

In order to study the variability in the received signal strength measurements, we fixed three different locations in Florida: Shalimar, Tallahassee, and Niceville, and recorded 1360 readings in Shalimar, 1792 readings in Tallahassee, and 1000 readings in Niceville under different conditions (indoors, outdoors, sunny, and rainy) over several months. Figure 8 shows the variation of received RSSI over time at the Tallahassee location. It must be pointed out that this data was collected over a period of time which is different from the times in which the test data was collected.

Figure 8 shows the plot of RSSI values from different locations and different channels. It shows 10 randomly selected channels across the different locations. For each channel we plot the time varying data and fit a Gaussian curve to the data. These curves were plotted after the DCD feature detector was run. The variance across the curves in Fig. 8 is only 0.617 dBm. This is important for the following reasons: (1) The plot shows that if we look at the data after the DCD extraction, the difference between the standard deviations across the Gaussians is very small. This shows that the DCD features are robust to temporal changes in the RSSI as well as changes due to location and environment. Hence it can be used for transfer learning tasks where either the model is old or the data is old. (2) It also justifies the assumption, in our theoretical analysis, that the variance can be assumed to be the same across channels and locations. We are now ready to describe the results of the location inference step, which we do next.

4.5 Location inference

We start by describing a method based on Friis equations\(^9\), which is a standard technique that uses a path loss model coupled with trilateration for localization. This is one of the most common location inference methods and hence we decided to test its efficacy and robustness in order to compare with our algorithm.

4.5.1 Friis equations

We started our experiments using Friis transmission equations directly for localization. The Friis equation\(^9\) was fitted using 1700 measurements in Tallahassee, FL, USA, and the average loss factor of \(X = -9.3218\) dB was chosen from the fitted equation and plugged into the Friis model. A dataset with 470 measurements was obtained at 30°23′47.0″N, 84°12′36.6″W and Friis model was used with the observed data for estimating the position of the receiver. Figure 9 shows the circles estimated using Friis equation. The center of each circle is the location of an FM transmitter and the receiver is estimated to be at a point on the circumference of each of these circles. The radii of the circles are the distances calculated from the Friis equation. Finally we estimate the position of the receiver using trilateration from these circles.

The average error for positioning with this method was around 7.40 miles, with a minimum error of 0.31 mile and a maximum error of 33.03 miles in Tallahassee, FL, USA. The loss factor \(X\), which is location dependent, can be affected by the terrain, weather condition, and objects between the transmitter and receiver. Since Friis equation results were not competitive with Algorithm 4 and because of the fact that it is hard to scale this method over very large areas,
we did not pursue this approach further.

4.5.2 Nearest neighbor positioning

As mentioned before, our algorithm for location inference is based on an Euclidean nearest neighbor search in the space of the DCD features. We used several other distance metrics for the nearest neighbor search and present the results here for comparison. More precisely, we used the Kendall-Tau distance, Cosine distance, City Block distance, and Correlation distance for comparison with our approach. We present the results of localization for Algorithm 4 using these distance metrics as well. The results of our experiments with different data sets and calibration methods are shown in Tables 1 – 4. As mentioned in Section 4.3, we also used DCD feature-based linear regression for calibration. However, as the results were not comparable with the linear regression or decision tree based methods, we have refrained from describing them here. They are available in the appendix for interested readers.

Table 1 Comparison of different metrics used for localization of two-way Tallahassee-Crestview 2015 data using estimation results from 2015 FCC data with linear regression-based calibration (Errors are in miles). Average error in ground truth is 0.98 mile using Vincenty distance[100].

| Metric       | Min err. | Med err. | Std. dev. | Avg. err. |
|--------------|----------|----------|-----------|-----------|
| Euclidean    | 0.33     | 2.82     | 2.92      | 3.4       |
| Kendall-Tau  | 18.96    | 725.43   | 654.22    | 933.96    |
| Cosine       | 15.53    | 734.46   | 557.78    | 851.45    |
| City-Block   | 0.33     | 2.75     | 2.78      | 3.44      |
| Correlation  | 1.1      | 673.85   | 495.2     | 795.8     |

Table 2 Comparison of different metrics used for localization of Tallahassee outskirts 2017 data using estimation results from 2017 FCC data and decision tree-based calibration (Errors are in miles). Average error in ground truth is 0.98 mile using Vincenty distance[100].

| Metric       | Min err. | Med err. | Std. dev. | Avg. err. |
|--------------|----------|----------|-----------|-----------|
| Euclidean    | 0.13     | 4.09     | 3.37      | 4.8       |
| Kendall-Tau  | 5.92     | 733.62   | 589.41    | 881.41    |
| Cosine       | 0.13     | 6.77     | 481.87    | 243.9     |
| City-Block   | 0.13     | 4.07     | 3.39      | 4.84      |
| Correlation  | 0.13     | 5.60     | 423.04    | 199.4     |

Table 3 Comparison of different metrics used for localization of two-way Tallahassee-Crestview 2015 data using estimation results from 2017 FCC data and decision tree-based calibration (Errors are in miles). Average error in ground truth is 0.98 mile using Vincenty distance[100].

| Metric       | Min err. | Med err. | Std. dev. | Avg. err. |
|--------------|----------|----------|-----------|-----------|
| Euclidean    | 0.13     | 4.09     | 3.37      | 4.8       |
| Kendall-Tau  | 5.92     | 733.62   | 589.41    | 881.41    |
| Cosine       | 0.13     | 6.77     | 481.87    | 243.9     |
| City-Block   | 0.13     | 4.07     | 3.39      | 4.84      |
| Correlation  | 0.13     | 5.60     | 423.04    | 199.4     |

Table 4 Comparison of different metrics used for localization of two-way Tallahassee-Crestview 2015 data using estimation results from 2015 FCC data and linear regression-based calibration y = 0.8064x – 51.42. Algorithm is top-k candidate based weighted average where weights are the distances in DCD space (k = 6) (Errors are in miles). Average error in ground truth is 0.98 mile using Vincenty distance[100].

| Metric       | Min err. | Med err. | Std. dev. | Avg. err. |
|--------------|----------|----------|-----------|-----------|
| Euclidean    | 0.25     | 5.28     | 59.14     | 9.88      |
| Kendall-Tau  | 18.03    | 670.67   | 331.76    | 722.37    |
| Cosine       | 0.24     | 10.16    | 448.38    | 250.80    |
| City-Block   | 0.25     | 5.31     | 87.31     | 12.35     |
| Correlation  | 0.11     | 6.89     | 476.04    | 226.63    |

As seen from the tables, none of the metrics used came close to that of the Euclidean metric (except the City Block metric which is an approximation to the Euclidean metric). Kendall-Tau has been shown to be effective in case of small areas[3] where important stations in the locality can be hand picked for positioning. However it suffers at large scales because it is not possible to select important stations over very large areas and also because of the availability of too many combinatorially similar spectra in the model. Another interesting thing to note is that the accuracy of the localization does not suffer too much when we use estimates of expected powers that are separated by a period of two years (2015 vs. 2017). The only
difference is that whereas using the 2015 estimates on 2015 data we can localize \( N = 916 \) points, using the 2017 estimates on the same data gives us localization for \( N = 691 \) data points. This is because of changes in the distribution of the FM transmitters from 2015 to 2017 which results in missing geohashes that cannot be localized. However, this serves our purpose, as here our goal was to show the efficacy of the DCD features for positioning with time varying models. Moreover, we could only localize 916 of the 924 data points from the 2015 two-way data with the 2015 estimates because at many locations there are FM frequencies that were received, but there are no nearby transmitters in our model for those frequencies. This is automatically detected by our candidate selection algorithm and no localization is performed for these points.

Though we provide results for the Kendall-Tau-based localization, first use in Ref. [3] for large scale positioning with FM within the confines of a city, we also want to compare the results of our algorithm with those obtained from other large scale positioning algorithms that are not based on FM. To that end we implemented the \textit{weighted average} method used for large scale localization using 4G-LTE and first reported in Ref. [81] and used it for FM-based positioning. For this method the final position is computed as the \textit{weighted average} of the top-\( k \) prediction candidates where the weights are the \textit{distances} in the DCD feature space. The results are shown in Table 4. It can be observed from Table 4 that for any distance metric, the \textit{weighted average} of the top-\( k \) candidates, for \( k = 6 \), does not improve the positioning accuracy but on the contrary makes it worse. However for the Tallahassee outskirts 2017 data, the Euclidean nearest neighbor and the top-\( k \) results are comparable.

The most time intensive part of our localization procedure is the acquisition of the FM power spectrum, which is currently about 20 seconds. There are several ways to speed this up, including a better SDR (HackRF One or Ettus B210 for example), or the use of custom hardware\textsuperscript{[13]}. Our current computation time for localization is less than 10 seconds in Python, which can be reduced considerably by changing the implementation to a low-level language like C or C++.

\textbf{Discussion.} It must be pointed out that the accuracy that can be achieved by our positioning system is limited by the resolution of the estimation algorithm. Our estimation algorithm used geohashes with precision of 5, which maps all the (latitude, longitude) pairs within a \( 3.04 \times 3.04 \) miles grid cell to the same hash value. Moreover the precision of the Globesat GPS receiver that we used is around 25 m and hence several locations (that are within the 25 m radius) are mapped to the same hash value. This limits the accuracy of our algorithm as it can only predict the (latitude, longitude) pair which is the “representative” of the predicted hash and not the exact one. Thus the error computed based on the ground truth GPS coordinates can vary from 0 to 2.14 miles for any hash. Figures 10 and 11 show the distribution of the errors introduced by approximating a set of locations with a single hash of precision 5. However, it can be seen based on the median, that using the Euclidean distance, majority of the position estimates are off by a single hash value as shown in Tables 3 and 4. Thus the uncertainty in the estimated position is at most a single pixel (hash), half of the time.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{fig10.png}
\caption{Distribution of errors due to the use of geohash with precision 5. Data: Tallahassee-Crestview 2015. Errors are in miles.}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{fig11.png}
\caption{Distribution of errors due to the use of geohash with precision 5. Data: Tallahassee 2017. Errors are in miles.}
\end{figure}
In order to represent the errors in terms of the hashes instead of miles we show a plot of the approximation factor in Figs. 12 and 13. Note that if the location inference algorithm predicts the correct hash for the point under consideration then it is a “perfect hit” in that it is predicting the optimal geohash. Otherwise there is an error and the positioning system approximates the optimal hash with a larger area (covering more than one geohash). Let the optimal geohash be defined with the length of its side $x$ and let the predicted cell be defined with a side of length $y$. Then we define the approximation factor as $1 + \varepsilon = \frac{y}{x}$. Note that the smaller the approximation factor, the better the approximation. We get a “perfect hit” when the approximation factor is 1.

It is observed from Figs. 12 and 13 that the approximation factor for the Euclidean distance is between 0 and 2 for the majority of the points, which shows that most of the time the predicted location is off by at most 1 geohash and sometimes its off by 2. We provide the plots of the approximation factors for the other metrics in the appendix. Except for the City Block distance (which is an approximation to the Euclidean distance), all others have very large variation for the approximation factors.

Finally we would like to point out that the number of DCD features extracted plays an important role in the accuracy achievable. As seen from Fig. 14, in general, the larger the number of DCD features, the lower the error.

5 Conclusion and Future Work

In this work, we have described algorithms for a passive approximate localization system using large scale FM map estimation using FM signal integration. Our system can be used for operating in environments where GPS is unreliable and can be used in conjunction with other positioning methods for approximate localization in GPS degraded environments. One of our contributions has been the creation of large scale FM map model using simple geometric primitives. The LoSI system can also be used standalone for detecting GPS spoofing. Going forward, our goal is to reduce the localization error by using multi-channel and multi-modal approaches. Also as noted before, the quality of the calibration plays an important role in the resulting accuracy that can be obtained from the system. Better calibration techniques can substantially reduce the errors and hence in the future we also plan to work toward the goal of finding a method for automatic calibration of the system.

Appendix

A Experimental Setup

For the experiments we used an RTLSDR with an FM antenna. The RTLSDR was plugged in using USB.

![Fig. 12 Approximation factor for Euclidean distance, where total points = 916. (Data: Tallahassee-Crestview 2015 with 2015 estimation and linear regression calibration)](image1)

![Fig. 13 Approximation factor for Euclidean distance, where total points = 111. (Data: Tallahassee 2017 with 2017 estimation and linear regression calibration)](image2)

![Fig. 14 Number of DCD features vs. average accuracy. Data: Tallahassee Outskirts 2017.](image3)
connector to a laptop computer running Ubuntu 14.04 LTS. The antenna was mounted on the collection vehicle as shown in Fig. 15. We used a GlobeSat GPS receiver shown in Fig. 15 for collecting the ground truth location information. The GPS receiver was also mounted outside the collection vehicle alongside the antenna (Fig. 15). The GPS receiver was plugged into the laptop computer using USB. The computer was mounted inside the collection vehicle.

B Experimental Results

B.1 Table of errors

In this section we show the results for the DCD feature-based linear regression calibration. Table 5 shows the results for the Tallahassee outskirts 2017 dataset. It must be noted that the average error in the ground truth for this data set is 1.20 miles. As seen from the table the median error is 5.47 miles. This is worse than the reported errors using the decision tree-based calibration.

In Table 6 we report the errors for the Tallahassee-Niceville one-way 2015 dataset using the FCC 2017 data for model estimation. Again we use a DCD feature-based linear regression for calibration.

B.2 Approximation factor: 2015 data

Here we present the distribution of the approximation factor of our nearest neighbor based location inference algorithm for the Tallahassee-Crestview two-way 2015 dataset with model estimated using 2015 FCC dataset and for metrics other than the Euclidean metric (which has been reported in the paper). We report the results for Kendall-Tau, Cosine, City Block, and Correlation metrics. It must be noted that the results have been obtained using a linear regression based calibration. Please refer to Figs. 16 – 19.

B.3 Approximation factor: 2017 data

Here we present the distribution of the approximation factor of our nearest neighbor based location inference algorithm for the Tallahassee-Crestview two-way 2015 dataset with model estimated using 2015 FCC dataset and for metrics other than the Euclidean metric (which has been reported in the paper). We report the results for Kendall-Tau, Cosine, City Block, and Correlation metrics. It must be noted that the results have been obtained using a linear regression based calibration. Please refer to Figs. 16 – 19.

Table 6 Comparison of different metrics used for localization of one-way Tallahassee-Niceville 2015 data using estimation results from 2017 FCC data and DCD linear regression-based calibration. (Errors are in miles.)

| Metric         | Min err. | Med err. | Std. dev. | Avg. err. |
|----------------|----------|----------|-----------|-----------|
| Euclidean      | 0.13     | 3.65     | 3.41      | 4.6       |
| Kendall-Tau    | 1.93     | 695.74   | 602.80    | 828.05    |
| Cosine         | 0.13     | 5.93     | 486.82    | 197.97    |
| City-Block     | 0.05     | 3.79     | 3.85      | 4.87      |
| Correlation    | 0.13     | 5.01     | 413.83    | 158.07    |

Table 5 Comparison of different metrics used for localization of Tallahassee outskirts 2017 data using estimation results from 2017 FCC data and DCD linear regression-based calibration. (Errors are in miles.)

| Metric         | Min err. | Med err. | Std. dev. | Avg. err. |
|----------------|----------|----------|-----------|-----------|
| Euclidean      | 0.74     | 5.47     | 4.4       | 6.82      |
| Kendall-Tau    | 4.68     | 749.98   | 455.66    | 903.5     |
| Cosine         | 2.3      | 608.45   | 543.16    | 676.16    |
| City-Block     | 1.21     | 5.72     | 4.41      | 6.87      |
| Correlation    | 2.29     | 646.57   | 505.05    | 656.85    |

Fig. 15 Equipment used for data collection: car setup, antenna, and GPS.

Fig. 16 Approximation ratio for Kendall-Tau distance, where total points = 916. Data: Tallahassee-Crestview 2015 with 2015 estimation.

Fig. 17 Approximation ratio for Cosine distance, where total points = 916. Data: Tallahassee-Crestview 2015 with 2015 estimation.
factor of our nearest neighbor based location inference algorithm for the Tallahassee outskirts 2017 dataset with estimated model using 2017 FCC data for metrics other than the Euclidean metric (which has been reported in the paper). We report the results for Kendall-Tau\cite{3}, Cosine, City Block, and correlation metrics. It must be noted that the results have been obtained using a decision tree regression based calibration. The results are presented in Figs. 20–23.

B.4 Results for top-k weighted average: Tallahassee-Crestview two-way 2015 dataset

Here we present the distribution of the approximation factor of the top-k ($k = 6$) weighted average based location inference algorithm for the Tallahassee-Crestview two-way 2015 dataset with the model estimated using the 2015 FCC data. We report the results for Euclidean, Kendall-Tau\cite{3}, Cosine, City Block, and correlation metrics. It must be noted that the results have been obtained using linear regression calibration. As pointed out in the paper, the top-k weighted average does not significantly improve the result in this case. On the contrary it leads to a significant deterioration of the results. Note that the detailed results for this method as obtained from this dataset have been presented in the paper. The results are presented in Figs. 24–28.
B.5 Results for top-$k$ weighted average: Tallahassee outskirts 2017 dataset

Table 7 shows the results of using the top-$k$ weighted average method with the Tallahassee outskirts 2017 dataset using the model estimated using 2017 FCC data and with decision tree regression-based calibration for positioning. As mentioned in the paper, the results from the top-$k$ weighted average, when the weights are the Euclidean distances, are comparable to those of the Euclidean nearest neighbor based method (and so are the results based
Table 7 Comparison of different metrics used for localization of Tallahassee outskirts 2017 data using estimation results from 2017 FCC data and decision tree calibration. Algorithm is top-$k$ candidate based weighted average where weights are the distances in DCD space ($k = 6$) (Errors are in miles). Average error in ground truth is 1.20 miles using Vincenty distance\cite{100}.

| Metric      | Min err. | Med err. | Std. dev. | Avg. err. |
|-------------|----------|----------|-----------|-----------|
| Euclidean   | 0.23     | 2.76     | 2.49      | 3.43      |
| Kendall-Tau | 19.16    | 726.38   | 506.04    | 837.55    |
| Cosine      | 60.29    | 670.68   | 469.74    | 786.60    |
| City-Block  | 0.23     | 2.68     | 2.21      | 3.27      |
| Correlation | 3.50     | 641.82   | 436.43    | 713.54    |

on City Block distance metric as it is an approximation of the Euclidean metric). However, the top-$k$ weighted average based method is computationally more expensive (specially when the value of $k$ is large) and the small gains in this case does not justify the increased computational cost.

We also present the distribution of the approximation factor of the top-$k$ ($k = 6$) weighted average-based location inference algorithm for the Tallahassee outskirts 2017 dataset using model estimated from 2017 FCC dataset and decision tree regression based calibration. We report the results for Euclidean, Kendall-Tau\cite{3}, Cosine, City Block, and correlation metrics. It must be noted that the results have been obtained using a decision tree based calibration. As pointed out in the paper, though the top-$k$ weighted average does not significantly improve the result in this case, it does not lead to a deterioration of the results and in the case of Euclidean metric (and hence the City Block metric) it gives results comparable to the Euclidean nearest neighbor based metric. However the computation costs are higher for this method. Please refer to Figs. 29 – 33.

Fig. 29 Approximation ratio for Euclidean distance, where points = 111. Algorithm Top-$k$ weighted average. Dataset: Tallahassee outskirts with decision tree regression calibration.

Fig. 30 Approximation ratio for Kendall-Tau distance, where points = 111. Algorithm Top-$k$ weighted average. Dataset: Tallahassee outskirts with decision tree regression calibration.

Fig. 31 Approximation ratio for Cosine distance, where points = 111. Algorithm Top-$k$ weighted average. Dataset: Tallahassee outskirts with decision tree regression calibration.

Fig. 32 Approximation ratio for City Block distance, where points = 111. Algorithm Top-$k$ weighted average. Dataset: Tallahassee outskirts with decision tree regression calibration.

References

[1] H. C. Chu and R. H. Jan, A GPS-less, outdoor, self-positioning method for wireless sensor networks, *Ad Hoc Networks*, vol. 5, no. 5, pp. 547–557, 2007.
Fig. 33 Approximation ratio for Correlation distance, where points = 111. Algorithm Top-k weighted average.

Dataset: Tallahassee outskirts with decision tree regression calibration.

[2] A. Popleteev, Indoor positioning using FM radio signals, PhD dissertation, University of Trento, Trento, Italy, 2011.

[3] J. Krumm, G. Cermak, and E. Horvitz, RightSPOT: A novel sense of location for a smart personal object, in UbiComp 2003: Ubiquitous Computing, A. K. Dey, A. Schmidt, and J. F. McCarthy, eds. Springer, 2003, pp. 36–43.

[4] F. J. Perez-Grau, R. Ragel, F. Caballero, A. Viguria, and A. Ollero, An architecture for robust UAV navigation in GPS-denied areas, *Journal of Field Robotics*, vol. 35, no. 1, pp. 121–145, 2018.

[5] Z. H. Wang, E. Blasch, G. S. Chen, D. Shen, X. P. Lin, and K. Pham, A low-cost, near-real-time two-UAS-based UWB emitter monitoring system, *IEEE Aerospace and Electronic Systems Magazine*, vol. 30, no. 11, pp. 4–11, 2015.

[6] Q. Miao, B. Q. Huang, and B. Jia, Estimating distances via received signal strength and connectivity in wireless sensor networks, arXiv:1801.09350, 2018.

[7] J. Hightower and G. Borriello, Location systems for ubiquitous computing, *Computer*, vol. 34, no. 8, pp. 57–66, 2001.

[8] K. Mohra, M. Watterson, Y. Mulgaonkar, S. K. Liu, C. Qu, A. Makineni, K. Saulnier, K. Sun, A. Zhu, J. Delmerico, et al., Fast, autonomous flight in GPS-denied and cluttered environments, *Journal of Field Robotics*, vol. 35, no. 1, pp. 101–120, 2018.

[9] M. Hazas, J. Scott, and J. Krumm, Location-aware computing comes of age, *Computer*, vol. 37, no. 2, pp. 95–97, 2004.

[10] V. W. Zheng, H. Cao, S. H. Gao, A. Adhikari, M. Lin, and K. C. C. Chang, Cold-start heterogeneous-device wireless localization, in *Proc. 30th IEEE Conf. Artificial Intelligence*, Phoenix, AZ, USA, 2016, pp. 1429–1435.

[11] E. Martin, O. Vinyals, G. Friedland, and R. Bajcsy, Precise indoor localization using smart phones, in *Proc. 18th ACM Int. Conf. Multimedia*, Firenze, Italy, 2010, pp. 787–790.

[12] A. Varshavsky, E. De Lara, J. Hightower, A. LaMarca, and V. Otason, GSM indoor localization, *Pervasive and Mobile Computing*, vol. 3, no. 6, pp. 698–720, 2007.

[13] Y. Chen, D. Lymberopoulos, J. Liu, and B. Priyantha, FM-based indoor localization, in *Proc. 10th Int. Conf. Mobile Systems, Applications, and Services*, Low Wood Bay, UK, 2012, pp. 169–182.

[14] P. Misra and P. Enge, Global Positioning System: Signals, Measurements and Performance, 2nd ed. Lincoln, MA, USA: Ganga-Jamuna Press, 2006.

[15] A. Cameron, GLONASS gone... Then back, http://gpsworld.com/glonass-gone-then-back/, 2016.

[16] GPS is a time bomb, http://www.locata.com/article/gps-is-a-time-bomb/, 2016.

[17] S. Cole, Backup PNT methods are essential for GPS-denied environments, http://mil-embedded.com/articles/back-up-pnt-methods-essential-gps-denied-environments/, 2016.

[18] M. Pomerleau, No GPS signal, no problem: Army works on ops in denied regions, https://www.c4isrnet.com/c2-commss/2017/06/05/no-gps-signal-no-problem-army-works-on-ops-in-denied-regions/, 2017.

[19] T. Hawks and B. McMahon, Time warfare: Threats to GPS aren’t just about navigation and positioning, https://www.defenseone.com/ideas/2017/05/time-warfare-anti-gps-arent-just-about-navigation-and-positioning/137724/, 2017.

[20] D. Hambling, Ships fooled in GPS spoofing attack suggest Russian cyberweapon, https://www.newscientist.com/article/2143499-ships-fooled-in-gps-spoofing-attack-suggest-russian-cyberweapon/, 2017.

[21] M. Pomerleau, Threat from Russian UAV jamming real, officials say, https://www.c4isrnet.com/unmanned/2016/12/20/threat-from-russian-uav-jamming-real-officials-say/, 2016.

[22] R. Beckhusen, Russia plans to turn civilian cell phone towers into cruise missile jammers, https://military.vice.com/en_us/article/xyg5c3/russia-plans-to-turn-civilian-cell-phone-towers-intocruise-missile-jammers, 2015.

[23] DARPA, Beyond GPS: 5 next-generation technologies for Positioning, Navigation and Timing (PNT), https://www.darpa.mil/news-events/2014-07-24, 2014.

[24] J. Lendino, DARPA to re-invent GPS navigation without the use of satellites, https://www.extremetech.com/extreme/202111-darpa-to-re-inventgps-navigation-without-satellites, 2015.

[25] SBIR, GPS-denied positioning using networked communications, https://www.sbir.gov/sbirsearch/detail/696781, 2015.

[26] R. Adams, DOD units, universities test air platforms in GPS-denied environment, http://www.executivegov.com/2017/09/dod-units-universities-test-air-platforms-in-gps-denied-environment/, 2017.

[27] F. S. Gady, Is the pentagon getting ready to dump GPS, https://thediplomat.com/2015/05/is-the-pentagon-getting-ready-to-dump-gps/, 2017.

[28] J. LaMance, J. DeSalas, and J. Jarvinen, Assisted GPS: A low-infrastructure approach, https://www.gpsworld.com/article/assisted-gps-a-low-infrastructure-approach/, 2002.
A. Savvides, C. C. Han, and M. B. Strivastava, Dynamic location determination by RF means, *IEEE Transactions on Mobile Computing*, vol. 17, no. 12, pp. 2396–2411, 2018.

A. L. Majdik, D. Verda, Y. Albers-Schoenberg, and J. P. Golden, Terrain contour matching (TERCOM): A precise low-altitude navigation aid, in *Proc. SPIE 0238, Image Processing for Missile Guidance*, San Diego, CA, USA, 1980.

S. Zhao, Y. M. Chen, H. Y. Zhang, and J. A. McEllroy, Use of FM signals for position determination, *IEEE Transactions on Vehicular Technology*, vol. 58, no. 3, pp. 1213–1223, 2009.

A. Popleteev, V. Osmani, O. Mayora, and A. Matic, A survey on secure localization in wireless sensor networks, https://cis.temple.edu/~jiewu/research/publications/Publication_files/BookChapter_SecureLocalizationInWirelessSensorNetworks.pdf, 2007.

S. H. Fang, J. C. Chen, H. R. Huang, and T. N. Lin, Is FM a RF-based positioning solution in a metropolitan-scale environment? A probabilistic approach with radio measurements analysis, *IEEE Transactions on Broadcasting*, vol. 55, no. 3, pp. 577–588, 2009.

V. Otsason, A. Varshavsky, A. LaMarca, and E. de Lara, Accurate GSM indoor localization, in *Proc. 7th Int. Conf. Ubiquitous Computing*, Berlin, Heidelberg, 2005, pp. 141–158.

S. H. Fang, J. C. Chen, H. R. Huang, and T. N. Lin, Is FM a RF-based positioning solution in a metropolitan-scale environment? A probabilistic approach with radio measurements analysis, *IEEE Transactions on Broadcasting*, vol. 55, no. 3, pp. 577–588, 2009.

V. Otsason, A. Varshavsky, A. LaMarca, and E. de Lara, Accurate GSM indoor localization, in *Proc. 7th Int. Conf. Ubiquitous Computing*, Berlin, Heidelberg, 2005, pp. 141–158.

S. H. Fang, J. C. Chen, H. R. Huang, and T. N. Lin, Is FM a RF-based positioning solution in a metropolitan-scale environment? A probabilistic approach with radio measurements analysis, *IEEE Transactions on Broadcasting*, vol. 55, no. 3, pp. 577–588, 2009.

V. Otsason, A. Varshavsky, A. LaMarca, and E. de Lara, Accurate GSM indoor localization, in *Proc. 7th Int. Conf. Ubiquitous Computing*, Berlin, Heidelberg, 2005, pp. 141–158.

S. H. Fang, J. C. Chen, H. R. Huang, and T. N. Lin, Is FM a RF-based positioning solution in a metropolitan-scale environment? A probabilistic approach with radio measurements analysis, *IEEE Transactions on Broadcasting*, vol. 55, no. 3, pp. 577–588, 2009.

V. Otsason, A. Varshavsky, A. LaMarca, and E. de Lara, Accurate GSM indoor localization, in *Proc. 7th Int. Conf. Ubiquitous Computing*, Berlin, Heidelberg, 2005, pp. 141–158.
of a software radio to evaluate signals of opportunity for navigation, in Proc. 19th Int. Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS 2006), 2006.

[60] J. Barnes, C. Rizos, J. L. Wang, D. Small, G. Voigt, and N. Gamble, Locata: A new positioning technology for high precision indoor and outdoor positioning, in Proc. 2003 Int. Symp. GPS/GNSS, Tokyo, Japan, 2003, pp. 9–18.

[61] M. M. Atia, A. Noureldin, and M. J. Korenberg, Dynamic online-calibrated radio maps for indoor positioning in wireless local area networks, IEEE Transactions on Mobile Computing, vol. 12, no. 9, pp. 1774–1787, 2013.

[62] Y. Chen, D. Lymberopoulos, J. Liu, and B. Priyantha, Indoor localization using FM signals, IEEE Transactions on Mobile Computing, vol. 12, no. 8, pp. 1502–1517, 2013.

[63] H. Abdelnasser, R. Mohamed, A. Elghory, M. Farid Alzantot, H. Wang, S. Sen, R. R. Choudhury, and M. Youssef, SemanticSLAM: Using environment landmarks for unsupervised indoor localization, IEEE Transactions on Mobile Computing, vol. 15, no. 7, pp. 1770–1782, 2016.

[64] C. Laoudias, G. Constantiniou, M. Constantiniades, S. Nicolou, D. Zeinalipour-Yazti, and C. G. Panayiotou, The airplace indoor positioning platform for android smartphones, in IEEE 13th Int. Conf. Mobile Data Management, Bengaluru, India, 2012, pp. 312–315.

[65] L. Petrou, G. Larkou, C. Laoudias, D. Zeinalipour-Yazti, and C. G. Panayiotou, Demonstration abstract: Crowdsourced indoor localization and navigation with anyplace, in Proc. 13th Int. Symp. Information Processing in Sensor Networks, Berlin, Germany, 2014, pp. 331–332.

[66] A. Konstantinidis, G. Chatzimiliouidis, D. Zeinalipour-Yazti, P. Mpeis, N. Pelekis, and Y. Theodoridis, Privacy-preserving indoor localization on smartphones, IEEE Transactions on Knowledge and Data Engineering, vol. 27, no. 11, pp. 3042–3055, 2015.

[67] FCC, FCC wireless 911 requirements, https://transition.fcc.gov/psrh/services/911-servicess/enanced911/archives/factsheet_requirements_012001.pdf, 2001.

[68] M. Aziyian, I. Constandache, and R. R. Choudhury, SurroundSense: Mobile phone localization via ambient fingerprinting, in Proc. 15th Annu. Int. Conf. Mobile Computing and Networking, Beijing, China, 2009, pp. 261–272.

[69] H. Aly and M. Youssef, Dejavu: An accurate energy-efficient outdoor localization system, in Proc. 21st ACM SIGSPATIAL Int. Conf. Advances in Geographic Information Systems, Orlando, Florida, 2013, pp. 154–163.

[70] C. C. Counselman III and T. D. Hall, Instantaneous radio-positioning using signals of opportunity, US Patent 6492945, December 10, 2002.

[71] C. Yang, T. Nguyen, and E. Blasch, Mobile positioning via fusion of mixed signals of opportunity, IEEE Aerospace and Electronic Systems Magazine, vol. 29, no. 4, pp. 34–46, 2014.

[72] M. Ocana, L. M. Bergasa, M. A. Sotelo, J. Nuevo, and R. Flores, Indoor robot localization system using WiFi signal measure and minimizing calibration effort, in Proc. IEEE Int. Symp. Industrial Electronics, Dubrovnik, Croatia, 2005, pp. 1545–1550.

[73] V. Moghtadeeae, A. G. Dempster, and S. Lim, Indoor localization using FM radio signals: A fingerprinting approach, in 2011 Int. Conf. Indoor Positioning and Indoor Navigation, Guimarães, Portugal, 2011, pp. 1–7.

[74] L. Engelbrecht and A. Weinberg, Location determination system and method using television broadcast signals, US Patent 5510801, April 23, 1996.

[75] S. P. Tarzia, P. A. Dinda, R. P. Dick, and G. Memik, Indoor localization without infrastructure using the acoustic background spectrum, in Proc. 9th Int. Conf. Mobile Systems, Applications, and Services, Bethesda, MD, USA, 2011, pp. 155–168.

[76] M. Hazas and A. Hopper, Broadband ultrasonic location systems for improved indoor positioning, IEEE Transactions on Mobile Computing, vol. 5, no. 5, pp. 536–547, 2006.

[77] V. W. Zheng, E. W. Xiang, Q. Yang, and D. Shen, Transferring localization models over time, in Proc. 23rd AAAI Conf. Artificial Intelligence, Chicago, IL, USA, 2008, pp. 1421–1426.

[78] V. W. Zheng, S. J. Pan, Q. Yang, and J. J. Pan, Transferring multi-device localization models using latent multi-task learning, in Proc. 23rd AAAI Conf. Artificial Intelligence, Chicago, IL, USA, 2008, pp. 1427–1432.

[79] A. Youssef, J. Krumm, E. Miller, G. Cermak, and E. Horvitz, Computing location from ambient FM radio signals [commercial radio station signals], in IEEE Wireless Communications and Networking Conf., New Orleans, LA, USA, 2005, pp. 824–829.

[80] ComStudy, http://www.radiosoft.com/index.php?id=983, 2016.

[81] R. Margolies, R. Becker, S. Byers, S. Deb, R. Jana, S. Urbanek, and C. Volinsky, Can you find me now? Evaluation of network-based localization in a 4G LTE network, in IEEE Conf. Computer Communications, Atlanta, GA, USA, 2017, pp. 1–7.

[82] RTL-SDR, http://www.rtl-sdr.com/, 2017.

[83] Gustavo Niemeyer, Geohash, https://web.archive.org/web/20080305223755/http://blog.libax.org/80/#post-85, 2019.

[84] A. Fox, C. Eichelberger, J. Hughes, and S. Lyon, Spatio-temporal indexing in non-relational distributed databases, in 2013 IEEE Int. Conf. Big Data, Silicon Valley, CA, USA, 2013, pp. 291–299.

[85] J. O’Rourke, Computational Geometry in C, 2nd ed. New York, NY, USA: Cambridge University Press, 1998.

[86] M. De Berg, M. Van Kreveld, M. Overmars, and O. C. Schwartzkopf, Computational geometry, in Introduction to Algorithms, 3rd ed. Cambridge, MA, USA: MIT Press, 2009.

[87] M. Charikar, P. Indyk, and R. Panigrahy, New algorithms for subset query, partial match, orthogonal range searching, and related problems, in Automata, Languages and Programming: 29th International Colloquium, Málaga, Spain, 2002, pp. 451–462.
Tathagata Mukherjee earned the MS and PhD degrees from Florida State University. Currently he is an assistant professor in Computer Science at the University of Alabama. His research interests include machine learning, data analytics, optimization, and applications of AI to cyber-security and malware analysis.

Piyush Kumar is the founder and president of CompGeom Inc. and an associate professor at the Department of Computer Science, Florida State University. He obtained the PhD degree from Stony Brook University in 2004 and the BS degree from IIT Kharagpur in 1999. His research interests include design and analysis of algorithms, computational geometry, and optimization and machine learning.

Erik Blasch is an IEEE fellow and principal scientist at the US Air Force Research Lab (AFRL). He is also a program manager for the AFOSR DDDAS program. He received the BS degree from Massachusetts Institute of Technology (MIT) in 1992, master’s degrees in mechanical (1994), health science (1995), and industrial engineering (human factors) (1995) from Georgia Tech. He served in active duty from 1996 in the United States Air Force. He has completed an MBA (1998), MSEE (1998), MS Econ (1999), and the PhD degree (1999) from Wright State University and he is also a graduate of Air War College (2008).

Debdeep Pati is an associate professor in the Department of Statistics at Texas A&M University. He obtained the BS and MS degrees from Indian Statistical Institute, Kolkata, India, and the PhD degree from Duke University at Durham, NC, USA. His research interests include Bayesian non-parametric methods, machine learning, optimization techniques, and graph theory.

Eduardo Pasiliao received the BS degree from Columbia University, New York, NY, USA, in 1992, and the MS and PhD degrees from the University of Florida, Gainesville, FL, USA, in 1995 and 2003, respectively. He is currently a senior research engineer with the Air Force Research Laboratory. He has co-authored over 100 peer-reviewed journal and conference publications. His research interests are in the areas of graph theory, combinatorial optimization, discrete mathematics, network science, and machine learning. He has been a member of the American Institute for Aeronautics and Astronautics and the Institute for Operations Research and Management Sciences.

Liqin Xu received the BS and MS degrees at Georgia Institute of Technology. At the time of this work, he was an intern at CompGeom Inc., Tallahassee, FL, USA.