APPLICATION OF SYNTHETIC APERTURE RADAR WITH WI-FI FOR INDOOR LOCALIZATION

by
Nafi Kawser Wazed

A thesis submitted to
the Faculty of Graduate and Postdoctoral Studies
in partial fulfillment of
the requirements for the degree of
Master of Computer Science

Ottawa-Carleton Institute for Computer Science
School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa
Ottawa, Ontario, Canada, K1N 6N5

Abstract

Indoor localization is the process of localizing people or objects inside a building in the same way GPS does in an outside environment. In recent years, researchers have successfully achieved improvement in indoor localization accuracy. Still there are many limitations to overcome in performing and achieving good accuracy in indoor localization. The interest in estimating the location of something inside a building with good accuracy is very strong.

In this thesis we first propose an indoor localization technique relative to Wi-Fi access points along with a novel heuristic search based algorithm, named MuSLoc. Through simulation and comparative studies, we have shown that MuSLoc outperforms other indoor localization models without the help of fingerprinting or crowdsourcing about the environment. MuSLoc provides almost the same accuracy in LOS (Line of Sight) and NLOS (Non-Line of Sight) environments with regular infrastructure that has recently been provided by smart phones. This model doesn't require any additional hardware support in order to perform well.

Further, we propose another indoor localization based Wi-Fi device tracker model, named MSTracker, which is able to track both moving and non-moving devices inside a building. This model is also free from specialized infrastructure and can perform well without any training data information. Through real time simulation and analysis we have shown that it performs more accurately than other available models.

Through extensive simulations in a real time environment and analysis of performance comparatives with other available models, we have shown that both MuSLoc

and MSTracker perform more accurately with COTS than any other method of indoor localization and tracking of objects inside a building. The complete package of MuSLoc and MSTracker can perform perfectly with recently available Wi-Fi modules and smartphones.

Acknowledgements

I would like to acknowledge my heartfelt gratitude to my honorable supervisor Prof. Dr. Amiya Nayak of Ottawa-Carleton Institute for Computer Science. Without his encouragement, enthusiasm and expert guidance throughout my work and during the writing of this thesis, I would not have been able to accomplish so much and complete it so successfully. I would like to thank Dr. Wei Gong for his valuable contributions, suggestions and discussions in preparing this thesis and associated papers. I would also like to thank my parents, as I am really grateful to them for their continuous support.

Contents

List of Acronyms

AoA Angle of Arrival
AP Access Point
COTS Commercial Off-The-Shelf
CSAR Circular Synthetic Aperture Radar
DoA Direction of Arrival
LOS Line of Sight
MIMO Multiple Input Multiple Output
MUSIC MUltiple SIgnal Classification
NLOS Non Line of Sight
RF Radio Frequency
RSSI Received Signal Strength Information
SAR Synthetic Aperture Radar
ToA Time of Arrival
ToF Time of Flight
WLAN Wireless Local Area Network

List of Figures

Chapter 1

Introduction

1.1 Background Information

Indoor localization is the process of locating devices or objects inside a building with the help of a building map or relative to some pre-localized points like Wi-Fi access points(APs), figures or some specific features. This process helps to track devices or objects inside a building. Using this process, people can even track themselves inside different corridors of a shopping mall or a large size floor of a building. In recent years, improvements in wireless communication and advent research in this field have transferred researchers' focus from outdoor environment localization to indoor environment localization. In outdoor units, GPS is successfully mitigating users need within tens of meters of accuracy. But in indoor environments, this accuracy scale should be a scale of tens of centimeters, and in recent years, researchers have had a lot of success in this area. Researchers have gained success in indoor localization processes through development of new specialized infrastructure like deployment of antenna arrays, acoustic beacon, APs, charge point specialization or a combination of these, which requires extra cost. In addition, a lot of research has also taken place on learning based systems like fingerprinting about the environment, crowdsourcing or evaluating images or features of environment, which requires a lot of site survey. None of these methods are successful in achieving centimeter scale accuracy. Or, they need

additional cost for specialized infrastructure or a lot of training for getting higher accuracy in indoor localization. Now a lot of research is ongoing to develop an indoor localization system that does not require additional fingerprinting or specialized hardware support.

One of the ways to do indoor localization is to find out the position of a device relative to the position of Wi-Fi APs (AP) already situated inside of a building. Usually a building has its own floor plan and Wi-Fi APs are installed by maintaining some specific regulations. So, according to the floor map, the location of Wi-Fi APs can easily be calculated. Now to localize a device relative to Wi-Fi APs requires doing beamforming or power profile creation of exchanged Wi-Fi signals between transmitter and receiver. Synthetic Aperture Radar (SAR), a technology for localizing a flying aircraft relative to its environment can be used for this process.

SAR is the technique of synthesizing the effects of large-aperture physical radar antennas and generating images for remote sensing objects [17]. This technique is usually used by air crafts for taking information about the surrounding and is presently being used by RFID localization techniques [44]. Each single antenna of SAR enabled devices, at the time of moving, tries to send and receive wireless signals from transceivers and emulates those signals for generating a multipath power profile with help of relative power. From this power profile receivers try to localize themselves relative to transceivers. In this thesis we are going to focus on the application of SAR in Wi-Fi topology for indoor localization. In real life, SAR can be applied in various ways for localizing or tracking purpose of an aircraft. One way is that the radar is fixed in place and the aircraft is moving. This is usually used to track the position of an aircraft. Another way is when the radar is moving to find out the position of a non-moving aircraft. This same process is considered when an aircraft moves in a circular way. We will first discuss already developed indoor localization models based on SAR and will observe localization performance by applying SAR on both the receiver side and the transmitter side. Receiver of Wi-Fi topology can be defined as mobile or portable Wi-Fi enabled devices and Wi-Fi Access Point as transmitter.

1.2 Problem Statement

1.2.1 SAR in Receiver Side

As we have stated in an earlier section, one efficient way to do fingerprinting free or special hardware free indoor localization is to mimic the main theme of SAR in wireless with the help of the relative position of Wi-Fi APs, assuming that the position of Wi-Fi APs are known based on an available building floor plan and emulate a large number of antenna array [3]. This model is not optimized since the relative power profile used has low resolution. More importantly, it requires either human interaction to decide on the largest peaks or a comprehensive search algorithm to find Direction of Arrival (DoA) candidates which is an extremely computationally intensive process.

Another vital problem is that SAR in its regular form is dependent on emulating a lot of snapshots from a large number of antennas which is usually installed in the body of an aircraft model. But it is not suitable to use directly in mobile devices as these devices have limitations regarding the number of antennas. It is really hard to imagine a smart phone carrying large size antennas in its body. Even creating a large scale antenna array from moving antenna, which is able to provide DoA of a signal, is not enough. The reason behind this is that SAR works perfectly with a large number of signal snapshots where receiver antennas are put in a fixed distance. This principle will not work with handheld mobile devices as moving trajectory of these devices in human hands are not fixed. Even SAR alone is not enough for providing a good resolution DoA estimation.

1.2.2 SAR in Transmitter Side

As we have already described, SAR in its regular form requires a large number of wireless signal snapshots from a set of antennas, which is larger than the number of antennas available in today's smartphones. Usually IEEE 802.11n standard APs (as it is available now and has frequently been installed inside buildings - thus maintaining

good density) come with at most 4 antennas installed in their bodies; usually they come with 2 antennas though. Another IEEE standard (802.15) carries 16 large antennas attached to its body yet it is less available inside a building than the IEEE 802.11n standard. So the unavailability of large numbers of antennas in Wi-Fi AP is a problem to solve when applying SAR.

Another relevant problem related to the SAR application in Wi-Fi AP is the absence of movement in AP devices. As we know, SAR is deployed onto an aircraft and performs its operation as it moves. Hence, SAR is able to track only moving objects, not static ones. But to be honest, most of the Wi-Fi devices inside a building are sitting on tables, shelves or carts. To track a static device, the SAR radar needs to move following a specific track. So, in its current form, SAR installed in a Wi-Fi AP will fail to localize stationary objects. In this thesis, the goal is to develop a model which will be successful enough to localize all types of devices, moving or static inside a building with the help of SAR and Wi-Fi AP.

1.3 Existing Solutions

Indoor localization is not a new concept. People have been conducting research on indoor localization for a long time. The present approaches for indoor localization can be divided into four categories: (1) *ultrasound-based*, (2) *light-density-based*, (3) *sensor-based* and *RF-based*.

Ultrasound-based indoor localization: Bat [34] and Cricket [35] are two -well-known examples of ultrasound-based indoor localization processes. In Bat, multiple RF-ultrasound receivers are attached to the ceiling and with the generation of matrix it tries to locate a device. Cricket tries to locate mobile devices which are already equipped with an ultrasound receiver with the help of fixed-position ultrasound beacon generators. These methods are successful in bringing indoor localization accuracy within a meter after a great deal of training and site surveys. Recently, Liu et al. [36] showed an approach of peer-to-peer, sound-exchange-based indoor localization where commercially available mobile devices exchange sound with their peers and find their own location based on the received signal strength with an accuracy of up to 2 m.

Light-density-based indoor localization: In Luxapose [37], researchers demonstrated how a mobile device equipped with a light sensor was able to localize itself at different positions inside a room by measuring the light intensity. Another recent work in this direction is "Surroundsense" [38], where mobile devices sense acoustic sound and light intensity to localize themselves. These works are really promising for indoor localization though their accuracy is not good enough compared to other methods available in the literature.

Sensor-based indoor localization: Recent smartphones' built-in motion sensors like accelerometer, compass and gyroscope have long been acknowledged as indoor localization vehicles [40,41], with reported accuracies between 2 and 6 m. These sensors are also coupled with GPS measurements for indoor localization purposes. The charging time of a wireless charge sensor (recently attached to smartphones) is also used as part of different localization techniques [41, 42], although they require additional infrastructure to perform accurately.

RF-based indoor localization: RF-based indoor localization is performed in various ways, such as leaning upon specialized infrastructure like custom APs [7] or RFIDs [4,44]. Another form is indoor localization with exhaustive fingerprinting [9] or crowdsourcing [12,46]. Some of the methods rely upon the received RF signal strength (RSS), which results in lower accuracy. One of the ways to work with RSS is by first building signal strength maps and then use them to find out the location [11, 47, 48]. Yet another way is to do either crowdsourcing [10, 49] about RSS or calibration [50] with RSS for map construction and localization of a mobile device.

Indoor localization with SAR: Related SAR implementation examples in Wi-Fi topology are Ubicarse [13] and ArrayTrack [5]. Ubicarse tried to use Circular SAR in smartphones to localize themselves relative to stationary APs. This model is not optimized since the relative power profile it uses has low resolution. More importantly, it requires either human interaction to decide on the largest peaks or a comprehensive search algorithm to find DoA candidates, which is an extremely computationally-intensive process. We will show in Chapter 3 how to respond to Ubicarse's failures and provide more dedicated solutions for smartphone localization.

On other hand, ArrayTrack used Linear SAR in Wi-Fi AP to track moving devices

inside a building but cannot localize stationary objects. It also fails to track devices keeping difference between levels. They also worked with lot of antennas, which are hardly available in Wi-Fi APs. We have succeeded in overcoming ArrayTrack's failures; an in-depth discussion on this subject can be found in Chapter 4.

SpotFi [39] is a recent example of indoor localization system where the researchers worked on sub-carrier information of the Wi-Fi signal, apply the MUSIC [27] algorithm to it and developed a model via the combination of Angle of Arrival (AoA) and estimation of Time of Flight (ToF) signal information. Both the AoA and ToF play good roles in suppressing the multipath effects of the Wi-Fi signal. For calculating ToF, extra hardware support is needed as this is not available with the present devices. They work with 32 subcarrier information of a single Wi-Fi signal, which is not enough to work with SAR when there are a lot of obstacles present along the way. Even this model can work efficiently if it is performed in an open-space office floor where people use small cabinets, but not in an office space split into cubicles.

1.4 Motivation and Objectives

As discussed above, SAR is one vehicle to conduct indoor localization without fingerprinting, specialized hardware or huge crowdsourcing about the environment. Yet SAR in its classical form is not directly applicable to Wi-Fi networking topology. Therefore, one must find a solution that will make SAR applicable to Wi-Fi networking topology with presently available Wi-Fi devices.

As stated above, already available SAR applications in indoor localization still suffer from serious drawbacks, which renders these models inapplicable to the real world. Additionally, their reported object localization accuracy is bound within 60cm to meter scale, which is not good enough for indoor localization purposes. These issues motivated us to develop a solution that would meet the existing indoor localization challenges.

1.5 Contribution

1.5.1 SAR in Receiver Side

We have developed MuSLoc, an indoor localization process for mobile devices where MUltiple SIgnal Classification (MUSIC) [27] is applied over SAR to find DoA in numerical form. We present the MUSIC algorithm in more detail in Chapter 2. This model can separate the signal space from the noise space and provide high resolution in DoA estimation by performing internal computations and without user interaction. We have configured MUSIC in a new way so that it works with regular Wi-Fi signals. First, we have generated a virtual antenna array from the received signal of two small MIMO [25] antennas usually available in current smartphones. We have considered this MIMO capability of antennas for differentiating between the received signal strengths from different APs. In addition, we have considered the distance between the two antennas as the distance between the elements of the matrix serving as MUSIC input.

MuSLoc has exhibited success in estimating DoA directly in number, with a median angle error of less than 3°. From a 2D localization experiment, we have observed that MuSLoc is capable of limiting the median error to 29cm along the X axis and to 34 cm along the Y-axis, even in presence of multipath signals. With the help of our developed Heuristic Location Estimation (HeLE) algorithm, this median error in X-axis is improved to within 12cm in the X axis and to 13 cm in the Y axis. When it comes to 3D global positioning, we are able to achieve less than 16cm accuracy, which is encouraging compared to other solutions.

1.5.2 SAR in Transmitter Side

We have proposed MSTracker, an indoor localization process that is also built along the marriage of SAR and the MUSIC algorithm. It can localize Wi-Fi-enabled moving and stationary devices inside a building and can be extended as a low-cost building security monitoring technique. This model is also able to calculate DoA by through

internal computations and without requiring any user interaction.

We have empirically confirmed that MSTracker can estimate DoA for a Wi-Fi-enabled target device with a median error in angle less than 2.8°. For location estimation, we have found that it caps the median error along the X axis to 22 cm, in the Y axis to 29 cm and in the Z axis to 25 cm. MSTracker can successfully distinguish between different floors of a building and can localize a moving device with a median error in the X axis of less than 25 cm, in the Y axis of less that 33 cm and with a DoA estimation error inferior to 3.4°.

1.6 Assumptions

We have assumed that all the electronic devices were installed with Wi-Fi transceiver and receiver. All the works and experiments are executed with a 5-GHz-band Wi-Fi signal because of firmware limitations [43]. The packet rate between transceiver and receiver will be 10 per second. All Wi-Fi devices we have worked on are considered having at least two antennas. All Wi-Fi APs' position are fixed and their locations are pre-calculated with the help of available floor plans of a building. For MuSLoc, we consider that each of the mobile devices can analyze Wi-Fi signals on their end and calculate AoA from them. On the other hand, the Wi-Fi APs in MSTracker are able to analyze the Wi-Fi signal. Since we conducted our experiments inside a university building, we also assumed that this environment contains enough multipath effects.

1.7 Thesis Organization

We have described the design and working procedures of our developed models for indoor localization in the following chapters. We have also discussed the working procedures of the developed algorithms with the help of illustrative figures and real-time simulation results. These proposed models are expected to outperform existing solutions in term of cost, hardware availability and localization accuracy.

The remainder of the thesis is organized as follows: Chapter 2 provides a literature review on different indoor localization techniques. Chapter 3 unveils the procedures related to the applicable model for SAR at the Receiver side. Chapter 4 provides a description of Wi-Fi devices' tracker model developed with SAR. Chapter 5 dissects real-time experimental results followed by conclusions and future directions in Chapter 6.

Chapter 2

Literature Review

Localization techniques are concerned with determining the location of an active cell phone or wireless transceiver with respect to a map or some specific features. The explosive growth of mobile devices, such as smartphones and tablets, is increasing the demand for more accurate location information. Using smartphones for accurate indoor localization opens a new frontier in mobile computing, thus offering enormous opportunities to enhance users' experiences in indoor environments. Despite significant efforts to advance indoor localization in both academia and industry in the past two decades, highly accurate and practical smartphone-based indoor localization remains an open problem. To enable indoor location-based services (ILBS), several stringent requirements for an indoor localization system must first be satisfied: (1) they must be accurate enough that they can differentiate massive users' locations (foot-level); (2) they should not depend on any additional hardware components or extensions on users' smartphones; (3) they are to be scalable to massive concurrent users. Current GPS, Radio RSS (e.g., Wi-Fi, Bluetooth, ZigBee), or Fingerprinting-based solutions can only achieve meter-level or room-level accuracy [3]. Some of these models also achieve accuracy of 1 meter, but they do not show the same performance every time or across all environments.

Many researchers are trying to develop indoor localization systems based on sound, light, Wi-Fi signal strength, angle of arrival of Wi-Fi signal, built-in device sensors and so on and so forth. Their main target is to develop a system which will accurately

show location information for a user. Each of the methods have some challenges to overcome for calculating the location of the user's handheld device. Some research studies have been successful in overcoming these challenges and have attained meter-scale accuracy results. Nevertheless, for indoor localization, centimeter-level accuracy is preferred and this is very difficult to obtain.

In this Chapter we are going to discuss existing solutions to the indoor localization problem that are available at present as well as their performance in location estimation. Before that, we are going to elaborate on the working procedures of MUSIC and SAR.

2.1 MUSIC algorithm

The MUSIC algorithm was first developed by Schmidt [28] in 1986 where he tried to estimate constant parameters of a received signal by designing a complete geometric solution without the help of noise present in signal. After that, he extended his geometric model to achieve a reasonable solution even when a lot of noise is present in the received signal. Among other existing models and algorithms for estimating the parameters of a signal, MUSIC is one of the most accepted high-resolution algorithms, although it requires a significant number of calculations. This algorithm is basically a good extension of the Pisarenko harmonic decomposition [29] algorithm.

MUSIC uses *eigenspace* method to estimate frequency content of a signal. It assumes that each signal consists of p complex exponential in presence of Gaussian noise. p consists of numerical values. It uses a *Correlation Matrix* R whose dimension is considered as $m \times m$, where m is the number of the signal source. And *eigenvalues* of this matrix are sorted in decreasing order. MUSIC creates signal subspace with p largest eigenvalues. The remainder of the eigenvalues, $m - p$ consist only of noise.

Let us check how MUSIC works with mathematical calculations. First of all, according to [30] Wi-Fi signal X can be written as:

$$X = Sa + N \tag{1}$$

where, S is a set of m numbers of *steeringvectors* and a is a set of signal phases or amplitudes. We are assuming that different signals are uncorrelated. Then the *Correlation Matrix* can be written as

$$\begin{aligned} R = E[XX^H] &= E[Saa^H S^H] + E[NN^H] \\ &= SAS^H + \sigma^2 I = R_s + \sigma^2 I \end{aligned} \tag{2}$$

where R_s is the signal covariance matrix or can be defined as signal space. This matrix is a $m \times m$ matrix. As we have already defined, this signal space consists of p peaks, it obviously has $m - p$ *eigenvectors* which corresponds to zero *eigenvalue*. Let, λ be one of those *eigenvectors*. Hence we can derive the following calculations:

$$\begin{aligned} R_s\lambda = SAS^H\lambda &= 0 \\ \Rightarrow \lambda^H SAS^H\lambda &= 0 \\ \Rightarrow S^H\lambda &= 0 \end{aligned} \tag{3}$$

Eqn. (3) states that all these $m-p$ *eigenvectors* are orthogonal to all m signal steering vectors. Now multiply the set of these *eigenvectors* with m *steering vectors*. Let this matrix of *eigenvectors* be defined as Q_n whose size is $m \times (m-p)$. Now, the frequency estimation function for MUSIC can be written as:

$$P_{MUSIC}(\phi) = \frac{1}{\sum_{i=1}^{m-p} |S^H(\phi)\lambda_i)|^2} \tag{4}$$

where λ_i are *eigenvectors* and $S(\phi) = [s(\phi_1)\ s(\phi_2)\ s(\phi_3)\ ....\ s(\phi_m)]$. As we have already said, Q_n is orthogonal to the signal steering vectors, and the denominator equal to zero when ϕ is in the direction of the signal. Therefore, the largest p peaks are the frequency of the signals.

Now, in practice, the signal covariance matrix R_s is not available. So we need to estimate R_s for our work. The most prominent way to do this estimation is to estimate *eigenvectors* matrix Q_n from the correlation matrix of Eqn.(2).

Let Q be an matrix of received Wi-Fi signals. Based on *eigendecomposition* we can partition this Q matrix into a signal matrix Q_s with m columns which corresponds to signal *eigenvalues* and into a noise matrix Q_n with $m - p$ columns. This Q_n matrix, consisting of noise *eigenvalues*, is exactly the same as the matrix of R_s

which corresponds to $zero - eigenvalue$. So we can say, Q_s represents the signal subspace and Q_n consists of the noise subspace. Usually, the smallest *eigenvalues* of R matrix are noise *eigenvalues* and Q_n is orthogonal to Q_s. The MUSIC algorithm heavily relies on this observation.

Root-MUSIC: The MUSIC algorithm has some limitations. One of them is accuracy which is limited by discretization. Another important problem is that it requires either human interaction to decide largest m peaks or needs an additional comprehensive search algorithm which is an extensively computationally intensive process. For practical implementation, we require a method which can result directly in numeric values for estimating direction. This is where Root-MUSIC becomes useful.

Root-MUSIC is a model based parameter estimation technique. We use *steeringvector* to estimate ϕ which is the direction of the signal. Let us consider the equation of *steeringvector* as:

$$z = e^{jkdcos\phi} \tag{5}$$

With help of this equation, if the values of z, k and d are known, we can easily find ϕ which is the direction of the signal in numerical form.

2.2 Synthetic Aperture Radar

SAR is a particular radar modality that is mainly used by an aircraft to take images about its surrounding, such as the landscape, rivers, hills, etc. These images are represented either in 2D or 3D space. SAR mainly uses the antenna motion over a target area and then performs conventional radar beam forming. The time required for the signal to travel from the antenna to the ground is taken as input for creating a large antenna aperture. The image resolution depends on the size of the constructed aperture.

At first, it is important to produce an array for SAR with the help of the received signals. The array can be two-dimensional or three-dimensional. Each element of the array represents the volume of the area of the target. It also consists of the reflection of an object on that surface. Initially, this reflection value is considered

as zero. After that, for each captured signal reflection, the entire volume calculation is iterated. The distance between the sender and reflection ground is calculated by means of the signal travel time [19]. This distance can also be estimated using the power of a received signal [17]. After all the reflected signals have been completely iterated, the basic SAR processing is finalized. The whole iterative process produces images that represent an approximate picture on the target object and surface.

SAR can be applied in several ways:

Stripmap SAR: In this process, SAR assumes a fixed pointing direction of the antenna attached to a tracking device like an aircraft [20]. The strip map is formed by using the SAR image reconstruction processes and the map length is equal to the length travelled by the tracker device itself. Fig. 1 shows the basics of Stripmap SAR.

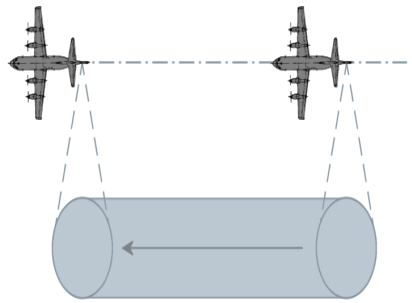


Figure 1: Example of Stripmap SAR [20]

Here, at traveling time, the aircraft constructs reflected images underneath its own position and combines them all to build the final image.

Spotlight SAR: Spotlight SAR [21] is another form of SAR where the radar beam is steered to keep a target under its beam coverage in order to achieve a higher resolution for as long as possible. By keeping that target under a beam from different angles, the SAR image is formed. It is mainly used to extend the spatial coverage of the target given the need to track it with high accuracy. Fig. 2 illustrates the beam steering process in spotlight SAR. Here, the aircraft is steering its beam towards a specific area during its flight so as to generate a high-resolution image of that area.

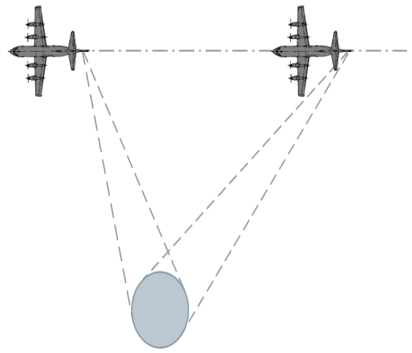


Figure 2: Example of Spotlight SAR [21]

Circular SAR: Circular SAR [22] works when an aircraft traverses a target area in a circular way while maintaining a fixed radius. As the aircraft flies in a circular fashion, it needs to aim its beam towards a specific point. Circular SAR helps an aircraft ensure that it is traveling in a circular way by maintaining a fixed distance from the target.

2.3 Sound-based Indoor Localization

Bat [34] used a specially configured mobile phone which is attached to a transmitter sound sensor. Numerous ultrasound RF-based receivers are attached to it, which generated a matrix by exchanging signals among senders and receivers. The sound sensor's location was calculated based on that matrix. Bat was successful in localizing that sound sensor attached to the mobile device with a median accuracy of 2 m in a regular room. An iOS application named "batphone" was developed, which needed a lot of training about surrounding sounds to discern between two available sound sources [6].

On other hand, Cricket's [35] main theme was to localize devices relative to ultrasound beacon packet generators whose position were assumed to be known. Here, the mobile device, whose location was going to be calculated, would be equipped with an RF-based sound receiver too. Actually, ultrasound-based indoor localization techniques require specialized hardware support for performing accurately. Cricket

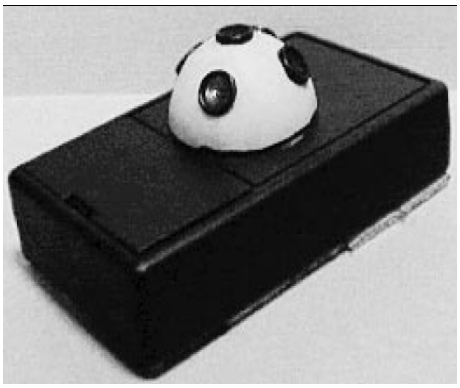


Figure 3: RF-based sound sensor attached to a mobile device for working with Bat [34]

was able to efficiently localize a device with a median error of 1.8 m. It exhibited a good performance in large areas; however, when the building layout consists of many small rooms, its performance degrades substantially.

Another example of sound-based indoor localization is one based on *acoustic background spectrum* [6], which relies on sound fingerprinting. A spectrum is designed based on the acoustic signal and is coupled with the "batphone" iOS app, which is already developed for sound-based signal responding. This system is able to distinguish between different sound frequencies, is robust to transient sounds and can easily discriminate among room level similarities.

Liu et. al [3] proposed an indoor localization method with the help of an anchor node and the acoustic signal. A coordination protocol for transmitting high-band acoustic signals localization is developed. In addition to that, they built an iOS app for getting acoustic signal information and used backend server indoor information and location-based services. They make acoustic beacons imperceptible to humans and improve its detection sensitivity for mobile phone sensors. To support multiple users, they replaced the regular beep signal with a newly designed wide-band modulation, a transmission waveform based on one-way synchronization. Furthermore, they tested their scheme in a smartphone without using mobile's own radio assistance. Moreover, for improving the low transmission rate of acoustic beacon signal, they employed symbol-interleaved beacon signals, e.g., Sa re ga ma pa... To enhance the accuracy, they resort to Time of Arrival (ToA)-based techniques.

In a mobile phone, they developed and implemented an app which would detect the signal and process it; they use information coming from ToA and apply NLOS mitigation for getting the final position inside the room. This model has some problems too. The first problem is the anchor node's coverage problems (i.e., it needs to cover the whole room). For this, either the transmission power or the detection sensitivity need to be increase, which may cause noise in the network. To counter this issue, they used advance signal processing systems in mobile as well as wide-band modulation techniques. The user movement needs to be continuously tracked with a very short time gap (sometimes just milliseconds difference). They tried to design better anchor network protocol for their model.

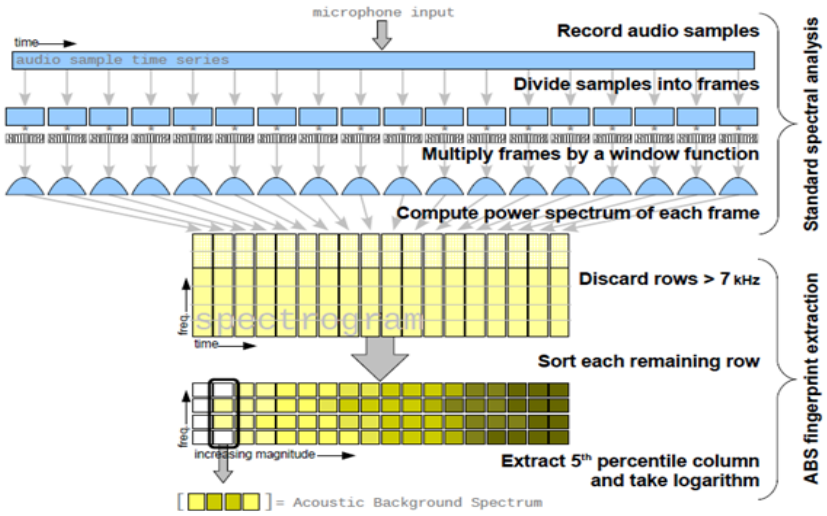


Figure 4: Acoustic Background Spectrum [6]

They introduced a new infrastructure which is compact and leans on sensible acoustic ambiance fingerprinting. It is able to distinguish between two rooms which are remarkably similar. For making a difference, they developed a linear-combination-distance algorithm. Their main application depends on tracking multiple objects. Their work is dependent on batphone, which increases the accuracy of batphone apps, which are already available in iOS. It is an improvement of a sound-based indoor localization system which is already developed for the iPhone that differentiates the

location between almost-same-size rooms. Recently, Liu et al. [36] showed a sound-fingerprinting-based localization approach that aims at finding the location of a set of already peered devices with some known position device. They exchanged sound signals among themselves and found the location based on the strength of the sound signals. The location estimation accuracy of the peer device is 2 m at the most.

2.4 Light-Density-based Indoor Localization

Luxapose [37] is a light-density-based indoor localization technique which tried to solve the indoor positioning problem using unmodified smartphone camera and slightly modified commercial LED luminaires. The luminaires are altered to allow rapid, on-off keying transmit their identifiers and/or locations encoded in human-imperceptible optical pulses. A camera-equipped smartphone, using just a single-image frame capture, can detect the presence of the luminaires in the image, decode their transmitted identifiers and/or locations, and determine the smartphone's location and orientation relative to the luminaires.

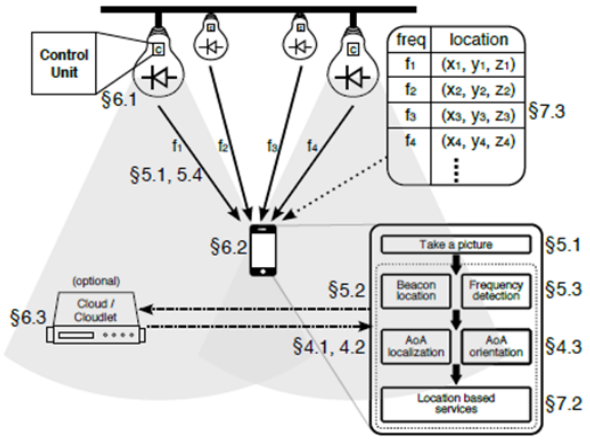


Figure 5: Luxapose: Location estimation with configured LED lights [37]

Continuous image capture and processing enables uninterrupted position updates. The driver circuits of emerging LED lighting systems can be easily modified to transmit data through on-off keying. A camera is intrinsically an angle-of-arrival sensor, so

the projection of multiple nearby light sources with known positions onto a camera's image plane can be framed as an instance of a sufficiently constrained angle-of-arrival localization problem. The whole process requires additional hardware. The location estimation accuracy can be bounded between 3 and 5 m depending on the resolution of the camera attached to the mobile phones.

Azizyan et al. [38] proposed a localization mechanism named "SurroundSense" that uses both ambient sound and light. Here, the phones sense their surroundings and use this ambient information to classify their location. Different surroundings have photo-acoustic fingerprints that can be sensed and used for localization purposes. For example, the ambient sound at Starbucks may include specific noise signatures from coffee machines and microwaves that are different from forks and spoons clicking in restaurants. Even lighting styles may be different in order to match with the type of service a place may provide bars with dim yellow lights versus Block Busters with bright white light. All the sensors together may produce a fingerprint. The acoustic fingerprint was generated from blocks of recorded clip at a specified rate. A spectral representation was computed by performing a Fourier Transform and the average power of the signal fell within a frequency range. An array of 48-dimensional fingerprints for a location was generated. Light and sound from each sensor were collected and matched with the previous trained data. The performance of the whole system was dependent on the number of training environmental features and on the data collection updates. The proposed approach hardly achieves a location accuracy within 3 m and requires a great deal of fingerprinting about light density to achieve this accuracy.

2.5 Sensor-based Indoor Localization

Recent smart phones' built-in motion sensors like accelerometer, compass and gyroscope have long been recognized as important sources for indoor localization [40–42]. They have shown an accuracy of about 2 to 6 m. These sensors are also coupled with GPS measurements for indoor localization purposes. In their research work, Constandache et. al. attempted to use electronic compasses and accelerometers present

in mobile phones. They devised Compacc [41] as a simple and scalable localization scheme. The idea is not fundamentally different from ship or air navigation systems, which have been around for decades. To overcome noisy phone sensors and complicated human movements, a person's walking patterns and possible path signature from a local electronic map has been used. Electronic maps enable greater coverage while eliminating the reliance on Wi-Fi infrastructure and expensive war driving. Results show a location accuracy of less than 12 m in certain regions. The main idea of their work was to leverage the mobile phones' accelerometer and electronic compass to measure the walking speed and orientation of the mobile user at a low energy cost. These readings can produce a directional trail that can be matched against walkable path segments within a local area map.

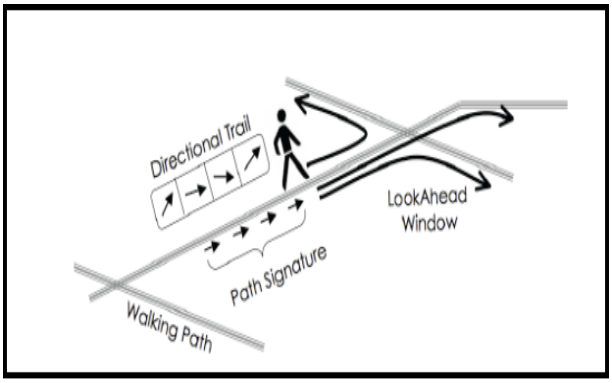


Figure 6: Compacc: Location estimation with the help of built-in accelerometer and gyroscope [41]

Shu et. al. proposed a wireless-charger-based localization system where the main idea is that "a mobile charger stops at different positions to charge sensors and utilizes the unique Time of Charge (TOC) sequences among wireless rechargeable sensors." [8]. Nearby sensors charge faster than far ones, which is their chief motivation. They divide the whole area in two different parts: inter-node division and inter-area division.

The authors tried to develop a relation between the time of charge and the distance based on time of charge. For a single sensor it is quite promising. For multiple sensors, they calculated the ratio of charge of each sensor and approximated their position.

Figure 7: TOC: Wireless Charger based localization technique [8]

This is almost ToA (Time of Arrival), with the only difference being that it is sensor-charge-based.

2.6 RF-based Indoor Localization

RF-based indoor localization is carried out in a variety of manners. One form of RF-based indoor localization leans upon specialized infrastructure like specialized APs [7], RFIDs [4, 44], Received signal strength [**?**, 47, 48, 52] or time of flight of signals [56–58]. Another form is exhaustive fingerprinting or crowdsourcing-based localization techniques.

Rai et. al. proposed a crowdsourcing-based approach [10] that leverages the inertial sensors (e.g., accelerometer, compass and gyroscope) present in the mobile devices to track them as they traverse an indoor environment while simultaneously performing Wi-Fi scans. This model is designed to run in the background on a device without requiring any explicit user participation. It depends on a map showing the pathways and barriers. A significant challenge that their model surmounts is tracking users without any priority, user-specific knowledge such as the user's initial location, stride length, or phone placement. Armed with just the map, they use Wi-Fi and inertial sensor measurements crowdsourced from the users' smartphones to automatically infer location over time and thereby construct a Wi-Fi training set.

Multi-dimensional particles are then created that also incorporate the uncertainty

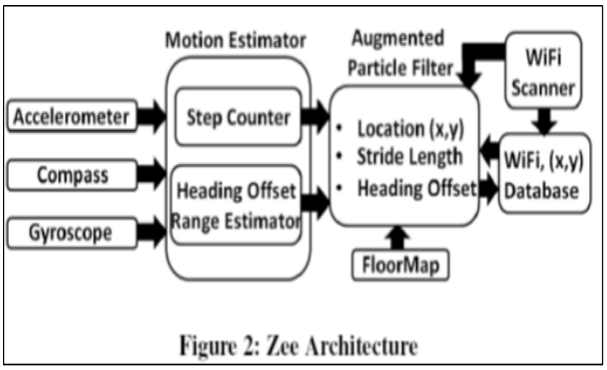


Figure 8: Zee: A crowdsourcing based approach [10]

in other aspects such as the user's stride length and their direction of walk. It employed placement-independent motion estimation to estimate the step count and the approximate orientation as well as Wi-Fi-based initialization to leverage partial Wi-Fi information to make an initial guess of the location(s) where a device might be. Backward belief propagation is applied to take advantage of the greater certainty in location at a later point in time and to trace back and reduce uncertainty in location at earlier times. Concurrently with estimating the location, the Zee scheme performs Wi-Fi scans and records the results indexed by time. As and when the location estimate for a particular time becomes available, the corresponding Wi-Fi measurement is annotated with the estimated location, thereby adding a record to the Wi-Fi training set.

Yang showed an approach of RF-signal-based indoor localization where he explored an alternative scheme that does not rely on the RF signature and leverages environmental physical features such as store logos or wall posters [49]. A user employs a smartphone to obtain relative position measurements of static reference points or physical features and the coordinates of these reference points are used to compute user locations. The advantage is that these are not affected by electromagnetic disturbances from microwaves, cordless phones or wireless cameras. The algorithm determines the suitable sensor and devises guidelines for the user to choose reference points. It also proposes a lightweight site survey method for service providers

to quickly estimate the coordinates of reference points and enhances image matching algorithms with a heuristic technique to automatically identify chosen reference points with high accuracy. Experiments have shown that the prototype achieves 4-5 m accuracy at 80% of the cases. Small malls and train stations require a one-time investment of only 2-3 man hours from service providers. Obtaining the signature map usually requires dedicated labor efforts to measure the signal parameters.

Gao et. al. worked with Zigbee technology to develop an indoor localization technique termed ZiFind [24]. ZiFind utilizes the low-power ZigBee interface to collect Wi-Fi interference signals and adopts digital signal processing techniques to extract unique signatures as fingerprints for localization. The design of ZiFind exploits the cross-technology interference in the unlicensed 2.4 GHz frequency spectrum. The mobile user utilizes a low-power ZigBee interface to detect the unique interference signatures induced by the Wi-Fi infrastructure and uses them as fingerprints to estimate the current location. ZiFind leads to significant power savings compared with existing approaches based on Wi-Fi interface, and yields satisfactory localization accuracy within realistic ranges. ZiFind clients employ low-power ZigBee radios instead of power-consuming Wi-Fi radios to collect fingerprints, which greatly reduces the energy consumption of localization. However, a challenge is that ZigBee radios cannot decode Wi-Fi frames and hence have no way of identifying the Wi-Fi APs they received the signal from. Moreover, the RSS samples of ZigBee radio may contain signals from various RF sources in the 2.4 GHz spectrum, such as Wi-Fi, ZigBee, Bluetooth, and even RF radiations from electrical appliances like microwaves. The extracted Wi-Fi fingerprints are used by a new learning algorithm called R-KNN to classify the location of user.

2.7 Received-Signal-Strength-Information-based Localization

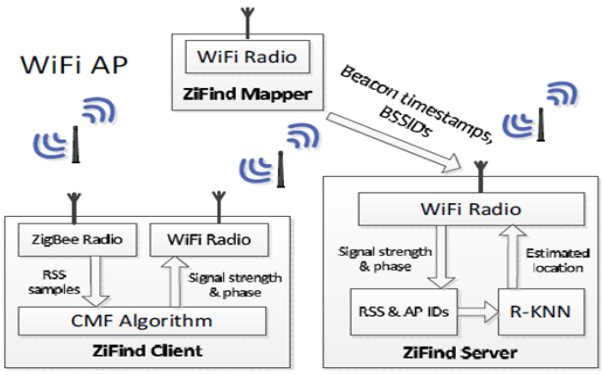


Figure 9: ZiFind technology [24]

In order to deal with the multipath effect, Wu et. al. proposed an indoor localization model named "Fine-grained Indoor Localization" (FILA) [52] to be used in WLANs. In order to attenuate the multipath effect, FILA used information about the channel state at the receiver side. To evaluate the performance of their model, the authors implemented FILA in commercial IEEE 802.11 NICs and then examined its performance in different indoor environments. Through the results of those experiments, the authors tried to show that FILA not only diminishes latency but also boosted accuracy in indoor localization compared to other existing RSSI-based methods.

Goswami et. al. developed WiGEM [54] using Gaussian Mixture Model (GMM) along with the Expectation Maximization (EM) algorithm. Here, they used a standard radio propagation model. Using this model they exploited the limitations in strengths of the received signal which can be found in the transmitter for various power levels. By removing any pre-deployment efforts, their proposed model enhances the methods based on infrastructure. By accessing RSS as well as the MAC ID of a device, WiGEM estimates the position of the target. By avoiding the training phase, WiGEM provides various crucial benefits.

Bahl et. al designed a software-based indoor localization system where they developed a point-based "environmental profiling scheme" [55]. Their method overcame

some of the difficulties induced by the changes in the surroundings, which in turn helps boost its efficiency. They also developed a Viterbi-like algorithm for "continuous user tracking". This algorithm employs real-time user motion patterns in order to detect his/her position.

Lim et. al. [53] proposed a robust localization system with zero-configuration cost in order to help the network services as well as management, which requires learning the location of a device. In order to estimate the position of the target, their proposed system uses the RSS measurements between the target device and its adjacent APs as well as between IEEE 802.11 APs in an on-line fashion. They used truncated singular value decomposition (SVD) methods to reduce the errors in measurements. Their proposed model needs zero configuration without using any supplementary exclusive hardware.

2.8 Signal's-Time-of-Fly-based Localization

Golden et. al. [57] proposed an indoor localization scheme with the help of WLAN communication between different devices. They developed a model based on the combination of RSSI and time of arrival of the signal. One of the important challenges of this model is to perform accurately even if no direct path between the devices is present. This model can localize a device with a median error of 4-5 m.

Youssef et. al. put forth a location detection algorithm named "PinPoint" [56]. This scheme works like a distributed system without using any preplanned implementation technique. Therefore the authors have claimed that their proposed system is appropriate for a situation which requires immediate deployment. The proposed system can also be considered as an asynchronous time-of-arrival method which works in two steps. The first step can be deemed as "ranging", where the distances between each node and all other adjacent nodes are computed. The second step can

be considered as "range amalgamation" in which at each node, the distances measured during the ranging step are used to estimate the topology of the network. This model requires a good deal of information to estimate the location within meter-level accuracy.

Mariakakis et. al. introduced a single-AP-based indoor localization method called "SAIL" [56]. In this system, the propagation delay between the target device and the AP is used to measure the distance between them. The proposed model attenuates the multipath effect by combining human mobility with the channel's impulsive responses. They used Kalman filtering in order to control the distance measurement errors. They also tried to reduce the dead-reckoning error by estimating the walking distance of a person without using any straightforward inputs.

2.9 SAR-based Indoor Localization

Kumar et. al. in Ubicarse [13] resorted to Circular SAR in smartphones to localize themselves relative to stationary APs. They used a correlation method over the received Wi-Fi signals and have created a relative power profile which helped them estimate DoA information. This model is not optimized since the relative power profile used has low resolution. More importantly, it requires either human interaction to decide on the largest peaks or a comprehensive search algorithm to find DoA candidates, which is an extremely computationally-intensive process. We will show in Chapter 3 how we circumvent Ubicarse's drawbacks and provide a more dedicated solution for smartphone localization.

On the other hand, ArrayTrack [5] used Linear SAR in Wi-Fi AP to track moving devices inside a building yet it failed to localize static devices. They used the MUSIC algorithm over a set of received signals for beamforming and required user interaction to define the peak points. This method also failed to track devices keeping difference between levels. They worked with a large number of antennas, which are hardly available with recent Wi-Fi APs installed inside a building.

Kotaru et al. proposed Spotfi [39] where they worked on sub-carrier information of the Wi-Fi signal and developed a model through the combination of angle of Arrival (AoA) and estimation of time of flight (ToF) information of signal. Both the AoA and ToF play good roles in suppressing the multipath effects in the Wi-Fi signal. For calculating ToF, they needed extra hardware support as this is not available with present devices. Again, they work with 32 subcarrier information of a single Wi-Fi signal, which is not enough to work with SAR when there are a lot of obstacles along the way. Even this model can work efficiently in open-space office floors, i.e., where people use small cabinets, but not in office spaces with cubicles.

Chapter 3

Indoor Localization with SAR in the Receiver End

As we have stated earlier in Section 1.3, SAR and MUSIC in their classical form are not directly applicable to mobile devices because of the limitation in the number of large shaped antennas in the mobile devices, which leads us to fail in configuring the antenna array [7]. An antenna array is a combination of multiple equidistant, physical antennas which are used to preserve information for transmitting and receiving wireless signals. There is a direct relationship between the signal's DoA and the antenna array, which is the main principle behind SAR. As classical SAR is dependent on emulating a large number of snapshots from different antennas, it is not suitable to use it directly in mobile devices given the limited number of antennas the latter carry. Even creating a large-scale antenna array from a moving antenna is not enough to provide DoA of a signal. The reason behind this is that SAR works perfectly with a large number of signal snapshots gathered from antennas separated by a fixed distance [13]. This principle will not work with handheld mobile devices as the moving trajectory of these devices is unpredictable and dynamically changing over time. Additionally, SAR alone is not enough for providing an accurate DoA estimation.

This is the reason that drives us to use the MUSIC algorithm together with SAR

for analyzing the signal's DoA. However, at the time of implementation of the MUSIC method as its own form over SAR in Wi-Fi, we have observed that MUSIC also requires a lot of antenna snapshots. To do this, we configured MUSIC in a newer way so that it can work with a regular WiFi signal. At first, we have generated a virtual antenna array from the received signal of two small MIMO antennas [25], which are usually available in modern smartphones. We have considered this MIMO capability of the antennas to differentiate between the received signals' strength from those in the APs and have taken the distance between the two antennas as the difference between the elements for configuring MUSIC's input matrix. With the help of the built-in motion sensors present in mobile devices, we tried to map the moving trajectory of the device both in linear and circular ways. We take into account both types as trajectories of mobile devices are totally unknown and undefined. When taking information from the receiving signals, we consider the signal received from both antennas and use the fixed distance between the antennas as a distance estimate between two snapshots. Thus, our new generated antenna array is able to provide a meaningful input to the MUSIC algorithm as it is needed for achieving a high resolution. Yet the MUSIC algorithm also needs user interaction to estimate the actual DoA in presence of the multipath environment, which is not acceptable for a pre-programmed mobile device. We tried to solve this problem by extending the MUSIC algorithm to a "Root-MUSIC" one. Root-MUSIC is capable of producing numeric values for DoA estimation [14]. Furthermore, Root-MUSIC provides a higher DoA resolution than that of MUSIC in a discrete environment, which is vital for handheld mobile devices as their moving trajectories are unpredictable.

Therefore, our procedures for mobile device localization with SAR application at the receiver end consists of Root-MUSIC with SAR, both linear and circular SAR [5, 15] for identifying the DoA of the incoming signal from APs to handheld mobile devices. These schemes try to estimate the relative position of the mobile devices even when these devices are moving. We named the whole system as **MuSLoc**. From here on, we will refer to MuSLoc as a model for the receiver-end SAR application. To make the localization performance more efficient, we introduced a new algorithm termed "HelE"', which will work coupled with MuSLoc to produce a more accurate

location of the target device. We will describe this algorithm later on as well as its implementation results.

For implementing MUSIC, we must first build a correlation matrix from the antenna array. This configuration requires a single Wi-Fi signal to be divided into amplitude, phase and steering vector. A major challenge for this pre-configuration from the Wi-Fi signal is the OFDM characteristics, which we have considered and have shown a way to deal with this in this Chapter. Emulating this correlation matrix, MUSIC separates the data space from the noise space and Root-MUSIC performs the DoA calculation.

3.1 MUSIC in SAR

As we discussed earlier, MuSLoc is the combined application of the MUSIC algorithm and the SAR system. The method performs real-time data collection and processing. First, it configures the antenna array by analyzing the received Wi-Fi signal and then tries to estimate the DoA with the help of the MUSIC algorithm. Like many other adaptive techniques, MUSIC works with a pre-configured correlation matrix of the received data. To configure the correlation matrix, the Wi-Fi received signal can be defined as [30]:

$$X = Sa + N \tag{6}$$

where S is a set of steering vectors $\{S(\phi_1), S(\phi_2), ..., S(\phi_n)\}$ of the n antenna elements, N is the noise, $\phi_1, \phi_2, ..., \phi_n$ are azimuth angles of between-antenna elements or snapshots and a is complex number representing the phase or magnitude; all the signals are uncorrelated.

Fig. 10 depicts an imaginary picture of a single antenna's circular way traversal with fixed radius r. Along its circular trajectory there are n points where this single antenna exchanges signals with its transmitter. Each time, the received signal phase angle will be varied for each position. The angle can be defined as ϕ_1, ϕ_2, ..., ϕ_n. As the antenna is maintaining a fixed-center point at the time of circular traveling, the transmitter will all the time maintain a fixed angle of arrival with the antenna,

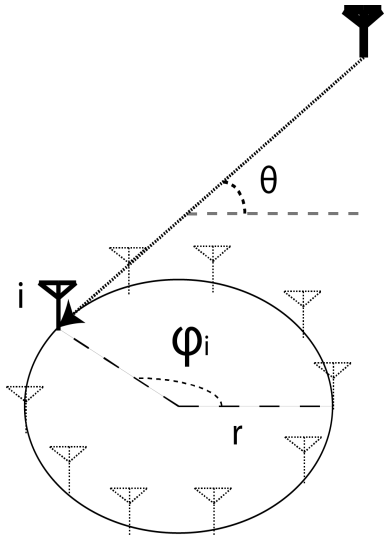


Figure 10: Circular path traversed by an antenna with n snapshots.

denoted by θ. After performing the Wi-Fi signal information X according to Eqn.(6), the correlation matrix will be

$$R = \frac{1}{K} \sum_{k=1}^{K} X_k X_k^T \tag{7}$$

where X_k is the input signal information at the k^{th} snapshot. $\{K = 1, 2, 3, ..., n\}$. The number of snapshots, K should be $K > 2N$, where N is the number of elements in the antenna array. This correlation matrix will be divided in two parts: the *signal covariance matrix,* R_s and the *noise covariance matrix,* R_n. The size of the matrix R_s is $N \times N$, which leads to $N - M$ *eigenvectors* to zero *eigen*values, where M is the number of incoming signals. These zero *eigen*values are orthogonal to M. The rest of the matrix, $N \times (N - M)$ of *eigenvector* makes up the *noise covariance matrix,* R_n, which is the basis of the MUSIC algorithm. In practice, R_s is not available. That's why we need to estimate R_n from the *eigenvectors* of the correlation matrix R. By doing *eigen*decomposition, we need to separate R_n from R. The *pseudo-spectrum* of MUSIC is given by the following equation:

$$P_{MUSIC}(\phi) = \frac{1}{\sum_{m=1}^{N-M} |s^T(\phi) q_m)|^2} = \frac{1}{s(\phi)^T R_n {R_n}^T s(\phi)} = \frac{1}{\|{R_n}^T s(\phi)\|^2} \tag{8}$$

Here, ϕ is the signal's direction of arrival. At the time of working with MUSIC in Wi-Fi scenarios, we have found that the received Wi-Fi signals are not directly applicable to the MUSIC algorithm. To address this problem, we have configured the steering vectors from the received Wi-Fi signal in a different way and have synthesized the received signal data. For steering vectors, we have to keep in mind the effects of perturbation problems, which may lead to a huge error in the final result. We will discuss our solution to this problem in upcoming Sections.

3.1.1 Acquiring Signal Space with an Antenna Array

For estimating DoA perfectly, all the elements of an antenna array need to be uniformly distributed (as shown in the Figure). In order to obtain the DoA, defined as theta in the figure, all the array elements should project the received signal toward that direction. This is done by multiplying the received signal by a complex number, called *steering vector*, after getting the signal information from all the snapshots. The configuration of the steering vectors is different for the circular way and the linear way [5, 15] . This thesis illustrates the working procedures in the circular way. The multiplication results linearly combined the *steering vectors* with the received signals, which will align their phases too [4]. We are now going to show the mathematical formulation behind this procedure.

Based on basic channel models, the information of the received wireless channel h_i at i^{th} snapshot can be written as the following complex number:

$$h_i = \frac{1}{d} e^{\frac{-j2\pi}{\lambda}(d + r\cos(\theta - \phi_i))} \tag{9}$$

where $i = 1, 2, 3, ..., n$ represent the snapshot positions, (r, ϕ_i) is the polar coordinate at the i^{th} snapshot, d is the distance from the source and λ denotes the signal's wavelength. At this point, considering Eqn (2), we have to multiply the *steering vectors* by the received signal. As we have discussed earlier, the *steering vectors* are different for each antenna in the array. Let us consider that each snapshot is an element of an antenna array. Now, the steering vector for a circular way can be written as:

$$S_i = e^{\frac{\pm j\pi}{\lambda} r cos(\theta - \phi_i)} \tag{10}$$

where θ is the direction of the source and $i = 1, 2, 3, ..., n$ represents the *steering vector* at the i^{th} element of the antenna array. This *steering vector* is derived by using Circular SAR [18]. MUSIC's pseudo-spectrum will be higher in θ, which is along the direction of the source. This is because the *steering vectors* are identical in phase. As this system is adopted from Circular SAR, its regular use responds to the translation of device positively, which will cause the indoor localization error estimation to fall in the meter scale, which is not acceptable. This problem would not appear in real radar, as it traverses a fixed way all the time and does not deviate much from the fixed track. As handheld mobile devices do not maintain a fixed track, it is hard to set up an antenna array with uniformly distributed *steering vectors* or SAR-derived antenna elements. In the following Sections, we will discuss this issue in more detail and unveil the approach we have chosen to solve this problem.

3.1.2 Translation-Resilient Steering Vector

In this Section we are going to discuss how to define the values for the *steering vector* in MuSLoc so that it performs perfectly even when the user moves his/her device along an unknown trajectory. Usually, in a regular mobile device at present time, this translation or deviation is calculated by the motion sensors attached to the device, which actually show the device's orientation. To calculate the textitsteering vector with this translation in mind, we have leveraged the MIMO capability of current wireless cards. We assume that each mobile device has two MIMO-enabled antennas. These two antennas receive wireless channels from the source. When a user twists the device, two antennas divide the wireless channels at each point of the twisted trajectory. By doing this, each point can produce an *steering vector* for its own position with the help of the motion sensors. As the distance between the two antennas is fixed in any device translation, each point in the trajectory space receives wireless channel information, thus keeping a fixed distance all the time. It then constructs a relative wireless channel from the received Wi-Fi signal, which

basically depends on the relative position between the two antennas rather than on the absolute position of the antennas. This process ensures that the element spacing in the *steering vector* is unique across all elements, which produces a large-size virtual antenna array. So, whether the user translates the device a lot, all the time, the so-constructed steering vector will be translation-resilient.

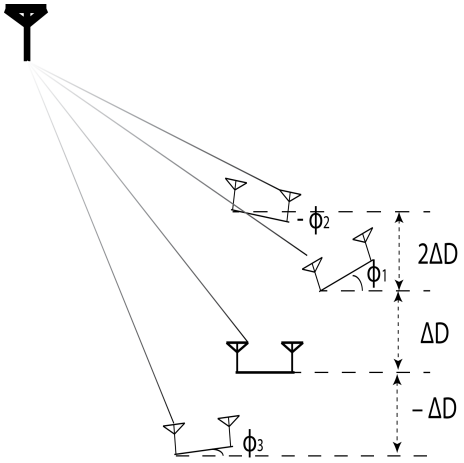


Figure 11: MIMO-enabled two receiver antenna movements in the user's hand.

Let us demonstrate the whole system as an example. We are assuming that the transmitter T is located at a far distance from the receiver. The receiver has two antennas; *antenna*1 and *antenna*2 as shown in Fig. 11 and the distance between them is d. The polar coordinates of the antennas' initial position are (D, θ), considering that the distance between the source and the receiver is D and the AoA at the receiver antenna is θ. Now, assume that the user has moved the device from its previous position. Let this position change be denoted by ΔD, which is much smaller than D; $\Delta D << D$ and that the orientation of the device is changed Φ considering X-axis as the base, (that is, Φ is the azimuth angle.) Hence, the new polar coordinates of the antennas are $(D \pm \Delta D, \theta)$. Considering D, the change of ΔD is negligible. As the deviation along the X-axis is zeros, the ΔD change in position is equivalent to considering the position change along the Y-axis, Δy whose value is positive all the time and relative to the position change of the two antennas. Now, according to the basic wireless channel modes, the wireless channel snapshots at the two antennas are given below:

$$h_{1,i} = \frac{1}{d} e^{\frac{-j2\pi}{\lambda}(d + \frac{\Delta y_i}{sin\theta})} \tag{11}$$

$$h_{2,i} = \frac{1}{d} e^{\frac{-j2\pi}{\lambda}(d + \frac{\Delta y_i}{sin\theta} + rcos(\theta - \phi_i))} \tag{12}$$

where λ denotes the Wi-Fi signal's wavelength and the change of Δx, Δy is much smaller than D; $(\Delta x, \Delta y << D)$ on the basis of the device's movement relative to the distance from the source.

From the above, the relative wireless channel will be derived by the following equation:

$$H_i = h_{2,i}.h_{1,i}^* = \frac{1}{d} e^{\frac{-j2\pi}{\lambda}(d + \frac{\Delta y_i}{sin\theta})} \frac{1}{d} e^{\frac{-j2\pi}{\lambda}(d + \frac{\Delta y_i}{sin\theta} + rcos(\theta - \phi_i))} = \frac{1}{d^2} e^{\frac{-j2\pi}{\lambda} rcos(\theta - \phi_i)} \tag{13}$$

If we substitute this relative wireless channel information in Eqn. (7), the signal information will be:

$$X = H \times S, \tag{14}$$

where H is the relative wireless signal and S is the set of steering vectors relative to the wireless elements.

3.1.3 Circular SAR Spatial Smoothing

As we have discussed at the beginning of this Section, the MUSIC algorithm depends on the assumption that all the input signals are uncorrelated. To ensure that the virtual antenna array we are configuring from the Wi-Fi signals is uncorrelated, we have done some spatial smoothing [26] before using it as an input for the MUSIC algorithm. It is really hard to derive the DoA with fully-correlated signals in MUSIC. When we are going to work with an antenna array formed via a circular-way traversal, spatial smoothing is especially needed.

Let us first look at Fig. 12, which shows the pseudo-random spectrum we have got from the application of the MUSIC algorithm to the set of spatially-smoothed

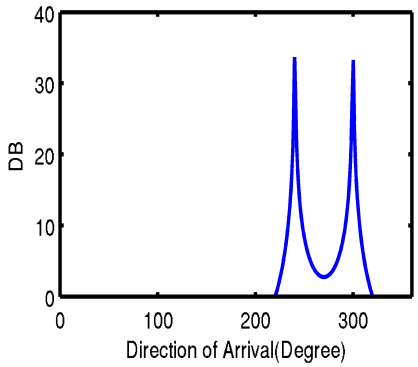


Figure 12: MUSIC's pseudo-random spectrum after applying spatial smoothing

Wi-Fi signals. As we already discussed before, the MUSIC algorithm distinguishes noise from signals, hence we can state that the pseudo-random spectrum of this figure is noise-free. The spectrum represents the DoA information of the received signals. We have reduced the spectrum to one lower than 0 dB as this power profile of a signal can not be inferior to zero. We are observing two peaks in the figure because of the presence of the multipath problem. Let us now take into account the received signal in Eqn.(15). As we are taking n snapshots in a circular way with radius d, we are considering that the number of elements in the virtual antenna array is n. Considering the multipath input, if the number of directions of the signal to impinge on antenna array elements is q $(\theta_1, \theta_2, ..., \theta_q)$, then the Equation can be written as:

$$X = \sum_{k=1}^{q} H_k S(\theta_k) \tag{15}$$

where S(θ_k) denotes any *steering vector* towards the direction θ. Note that here the problem lies in defining the value of q and the directions of the impinging signals $(\theta_1, \theta_2, ..., \theta_q)$ on the antenna array. To solve this, we have used a transformation matrix B to convert the steering vector of the configured antenna array into another virtual antenna array. The transformation matrix has been designed after considering special DFT and can be derived as:

$$B = \begin{pmatrix} 1 & \omega^{-h} & \omega^{-2h} & \dots & \omega^{-(n-1)h} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & \omega^{-1} & \omega^{-2} & \dots & \omega^{-(n-1)} \\ 1 & 1 & 1 & .... & 1 \\ 1 & \omega^{1} & \omega^{2} & \dots & \omega^{(n-1)} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & \omega^{h} & \omega^{2h} & \dots & \omega^{(n-1)h} \end{pmatrix} \tag{16}$$

where

$$\omega = e^{\frac{j2\pi}{n}} \tag{17}$$

We consider the size of this matrix to be $(2h + 1) <= n)$. The value of h, which determines the size of the matrix, should be chosen according to the following equation:

$$\max \left\{ h || h \leqslant \frac{p-1}{2}, \frac{|J_{h-p}(kr))|}{|J_h(kr))|} < \epsilon \right\} \tag{18}$$

After doing all this processing, the newly defined received signal is given below:

$$X = JBX \tag{19}$$

Here, J is a diagonal matrix. We proceed to formulate the MUSIC algorithm with a redefined, spatially-smoothed virtual antenna array which will be formed according to Eqn. (15).

3.2 Obtaining DoA in Numeric Values

For autonomous systems like mobile devices, it is very important to seek a solution with numeric values and make decisions automatically in the easiest way possible. With the classical MUSIC algorithm, a mobile device is unable to calculate the numeric value of the DoA without human intervention. To overcome this issue, we have introduced Root MUSIC [27] to mobile devices at this stage of our research work.

Root MUSIC is a model-based parameter estimation technique which is able to compute the DoA of the received signal with the help of the already-configured steering vector. The original Root-MUSIC algorithm does not work directly with circular-way antenna array. That is why we have formulated the circular antenna array in a different way, as already discussed in previous Sections.

Like the MUSIC algorithm discussed above, after constructing the correlation matrix, we have to separate the signal subspace, R_s from the noise subspace, R_n by performing its *eigendecomposition*. We then move forward with R_n. After calculating the noise subspace via MUSIC Eqn. (3), we will attempt to estimate the DoA from the following equation:

$$P_{MUSIC}^{-1}(\phi) = s(\phi)^T R_n {R_n}^T s(\phi) = s(\phi)^T C s(\phi) \tag{20}$$

where N is the number of received samples and n represents the number of elements in the antenna array. Now, according to the *steeringvector* in Eqn.(5), we can denote the set of *steeringvectors* as $S = [1, z, z^2, z^3, ..., z^{N-1}]^T$. Let $\mathbf{q}_m$ be the set of *eigenvectors* whose *eigenvalues* are zero. Hence, we can write the product of $\mathbf{q}_m$ and *steeringvector* as:

$$\mathbf{q}_m S = \sum_{n=0}^{N-1} q_{mn} z^n = q_m z \tag{21}$$

Now, we can extend Eqn.(25) as follows:

$$P_{MUSIC}^{-1}(\phi) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} z^n C_{mn} z^{-m} \tag{22}$$

From the above mathematical expression, our next step is to calculate the diagonal summation of C for all the samples and then estimate the number of incoming signals' DoA, M. As the Root-MUSIC algorithm also works with multipath signals, so estimating the number of incoming signals and their DoAs is an important step in achieving our goal. Root-MUSIC performs well even with low-SNR wireless signals. Finally, we will estimate DoA from the following equation:

$$\theta = \cos^{-1}\left[\frac{\Im \ln)(z_m)}{kd}\right], m = 1, ...M \tag{23}$$

where m represents the number of incoming signals, which is specially effective in presence of multipath signals; $\Im$ represents the imaginary part of the resulting information.

3.3 Extending to 3D Environment

It is hard to confine the aforementioned localization work only to the 2D space. MuSLoc can perform indoor localization in 3D environments as well. To this end, we have to perform steering vector calculation in the 3D space.

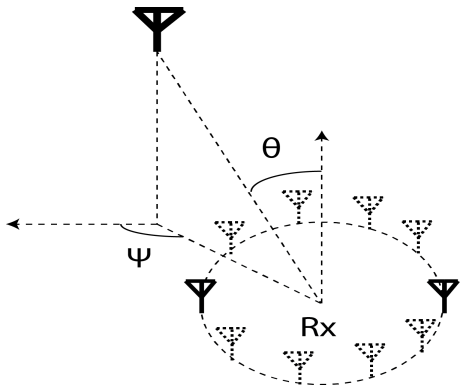


Figure 13: DoA estimation in a 3D environment.

From the basics of SAR, for both linear way and circular way, we have found that with the help of the polar angle and the azimuth angle of the incoming signal from the Wi-Fi transceiver, the DoA can be reliably estimated. It is possible to do so even when the user moves the device along the 3D space. Assume the azimuth angle and the polar angle of the incoming signal are denoted by θ and ψ, respectively. Let the position of the two antennas of the mobile device be represented by Φ as the polar angle and γ as the azimuth angle. Now, from Eqn. (5) and the basics of SAR in 3D environment, the steering vector will come in the form:

$$S_i = e^{\frac{+j\pi}{\lambda} rcos(\theta - \Phi_i) sin(\psi - \gamma_i)} \tag{24}$$

3.4 Multipath Suppression

Dealing with the multipath effect in indoor localization is one of the challenges for estimating a reliable DoA as a large number of wireless devices are present inside a building. Although the correlation matrix is considered as a solution for multipath suppression [32], working with MUSIC or Root-MUSIC sometimes provides more than one solution or root for DoA estimation [33]. To mitigate this multipath problem in indoor environments and estimate DoA more efficiently, we have proposed a Heuristic Location Estimation (HeLE) algorithm from the calculated DoAs. This algorithm allows reducing our localization error to 30 cm at the most.

At first we collect all the roots received from Root-MUSIC into an array. For each AP, we created an individual array. From our observations, we found that the length of these arrays is less than 4. After construction of all the arrays, we applied the triangulation method on three of the arrays, e.g., AP1, AP2 and AP3. At first, we took one element from AP1 and AP2 each and calculated an estimated position for the target object. Secondly, we select a new element from AP3 and again tried to infer a position by applying the triangulation method on AP2 and AP3. After computing this new position, comparing it with the previous one, if the distance between those two points is more than 10cm, we considered a virtual straight line between the points and the midpoint of that virtual line is regarded as the new position of the target device. We continue the same process with other elements of AP1, AP2 and AP3. Once completed, we leave the first one and start working with three new APs, e.g., AP2, AP3 and AP4. Once all the APs roots are traversed, the position we derived is the final estimated position of the target device.

Fig. 14 reports the working procedures of our novel algorithm. The red squares in this figure denote the location of the Wi-Fi APs. The location of these APs are calculated via the floor map. The blue cross represents the position of the target device in each of the pictures. Red crosses stand for the estimated positions with the help of MUSIC and the triangulation method. The red dashed line between the red cross points is considered a virtual straight line whose midpoint (depicted as a blue star) is the estimated location of the target. Every time the target's location

Algorithm 1 Heuristic Location Estimation

```
1: //Consider Traingulation(X,Y) as a procedure
2: // which returns intersection point of two DoAs
3: //Consider MidPoint(X,Y) as a procedure
4: // which returns middle point of a straight line with two points
5:
6:
7: m = 0, n = 0, LocEst = ∞, Temp = 0
8: M = Number of APs, N = Number of Roots
9: for i = 1 to M do
10:     for j = 1 to N do
11:         AP(m)(n) ← Numerical DoA estimated by Root-MUSIC
12:     end for
13: end for
14: LocEst= Triangulation(AP(m,n),AP(m+1,n))
15: m = m + 1,
16: while (n <= N) do
17:     while (m − 1 <= M) do
18:         Temp = Triangulation(AP(m,n),AP(m+1,n))
19:         if Distance(LocEst, Temp) > 10cm then
20:             LocEst = MidPoint(LocEst, Temp)
21:         end if
22:         m = m + 1
23:     end while
24:     m = 0, n = n + 1
25: end while
```

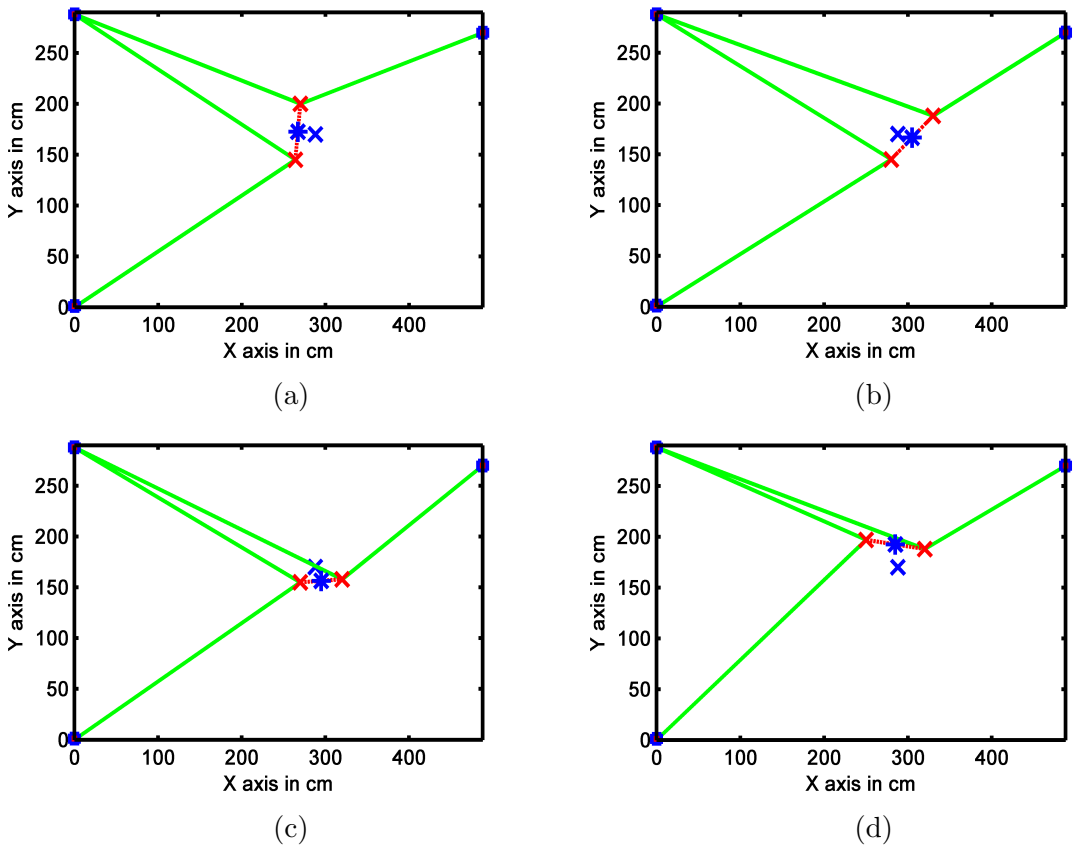


Figure 14: Convergence of position estimation with HeLE.

estimate is updated either in the global position of the device or along the X-axis or the Y-axis individually. According to Fig. 14, we can see that our HeLE novel algorithm is succeeds in estimating the location of a handheld device efficiently when outcomes of MUSIC or Root-MUSIC algorithm are highly distorted by obstacles or reflectors at the path, which causes multipath problems.

Our observations regarding the HelE algorithm are stated below:

- The complexity of the HeLE algorithm is $O(mn^2)$, where m is the number of APs and n is the number of solutions received from Root-MUSIC.
- Except in some cases, HeLE is capable of estimating the location with a median error of less than 13 cm along the X-axis and Y-axis estimation.

- A large number of APs can estimate the location of a device more accurately but with a higher computational overhead. For our experiment, we worked with up to 6 APs in one floor of a university building.

Chapter 4

Indoor Localization with SAR in the Transmitter End

As we have discussed earlier, mimicking the main theme of SAR in Wi-Fi is an indoor localization process involving fingerprinting or special infrastructure where the position of mobile devices are calculated relative to the fixed positions of Wi-Fi APs. For these methods, the APs positions are known in advance by consulting the building's floor map. Recently, Wi-Fi APs have been endowed with multiple antennas. We applied our work to IEEE 802.11n standard APs (as they are available now and have been frequently installed inside buildings, thus maintaining good density). These devices often come with at least 2 built-in antennas (although this number may go up to 4) [45]. APs with 16 antennas are commercially available as of 2010 [51]. Future Wi-Fi standards will be increasing the number of antennas in Wi-Fi APs, which will certainly benefit SAR-based localization techniques. We have named our study of applying SAR to a Wi-Fi AP scenario as **MSTracker**. In the rest of this Chapter, we will elaborate on the specifics of **MSTracker** with SAR in the transmitter end.

MSTracker can differentiate between the noise space and the signal space. This feature is exploited to provide high-resolution DoA estimation. This model can derive the DoA through internal computations and without direct user intervention. As we have shown in the previous Chapter, classical SAR cannot be implemented directly

into APs as it requires a large number of snapshots coming from several antennas, which are usually installed as part of an aircraft's fuselage. To solve this, we rotate the AP along a circular path at a fixed speed to create a large antenna array from the received signal. However, creating a large-scale antenna array from a moving antenna is not enough for providing an accurate signal DoA estimate. The reason behind this is that SAR works very well when multiple snapshots are taken from multiple, equidistant antennas. Another challenge to face at the time of working with SAR is that performing a relative power profile from Wi-Fi, the received signal or beamforming requires direct human interaction to estimate the DoA. To overcome this drawback, we have applied Root-MUSIC, an advanced version of the MUSIC algorithm, upon SAR to estimate an accurate DoA with reasonable computational effort. Root-MUSIC carries out a DoA estimation through internal calculation and returns a numerical value for it [14]. Root-MUSIC provides higher DoA resolution than MUSIC even in discrete environments. At implementation time, Root-MUSIC also requires a great number of antenna snapshots transmitted from a large number of antennas. To provide an adequate number of snapshots with two available antennas in Wi-Fi APs, we configured a virtual antenna array. We resorted to the MIMO capability of two antennas of IEEE 802.11n standard AP and attached a rotating stand to each AP so that it can move along a circular path while centered at its position. We rotated each AP in both ways, each time collecting the received signal information from both of the antennas and putting them together to configure the virtual antenna array. Thus, our newly generated virtual antenna array is able to assist the Root-MUSIC algorithm in its endeavour to produce a high-resolution DoA estimation.

We know from the previous Chapter that implementing MUSIC requires configuring a correlation matrix from a virtual antenna array. This configuration needs a single Wi-Fi signal to be split into amplitude, phase and steering vector. As the rotation of each AP is executed while maintaining a fixed track, each snapshot will preserve an equal distance between the matrix elements, thus giving rise to a virtual antenna.

4.1 Applying SAR to MUSIC

As we have already discussed above, MSTracker is developed with the help of the MUSIC algorithm as well as by using the SAR system. In MSTracker, the data collection and processing are both performed in real time. The MSTracker system works in two steps. First, the antenna array is configured and in order to do this, the output Wi-Fi signal from the receiver is used after being analyzed. In the next step, MSTracker is used in DoA estimation and to that end, it leans upon the MUSIC algorithm. Like many other methods applied in this area, a pre-configured correlation matrix of signals collected from the receiver are used by the MUSIC algorithm. In order to configure the correlation matrix, we can define the input signal as indicated in Eqn. (2). After computing and using X, we can represent the correlation matrix following Eqn. (3), (4). All these calculations will be performed at the Wi-Fi AP point. As Wi-Fi APs are not able to carry out these complex calculations, we have used a server with all the APs installed inside the building as shown in Fig. 15. The job of the server is to calculate the DoA by means of the MUSIC algorithm and store it in its memory according to the number of APs. MUSIC's spectrum procedures are already described in Section 3.1. We have applied the same procedures for applying MUSIC to SAR in the transmitter end.

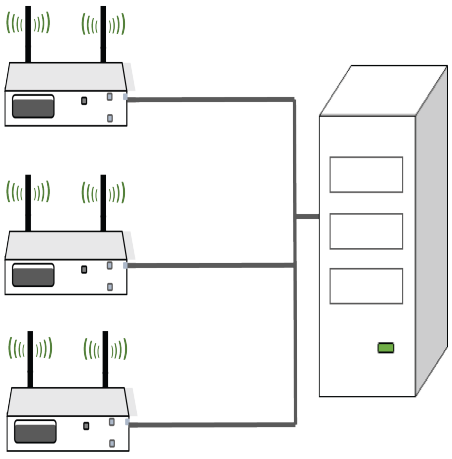


Figure 15: MSTracker working procedures

Fig. 15 portrays MSTracker's working procedures. We developed the entire MSTracker for the Wi-Fi AP side. All the Wi-Fi APs are connected to a server. For our experimental work, each of the APs installed in the building are located on a rolling rack whose height was almost equal to the height of the ceiling of that floor. We kept our AP on a rotational base in order to rotate it up to 360°. Every time, each of the APs exchanges signals with receiver devices and send them to the server, which calculates the DoA information from each of the APs' received signal and apply the triangulation [23] method to infer the device location. At the same time, our method utilizes some mechanisms to remove the multipath effect, which we will review in Section 4.5.

4.1.1 Configuring the Antenna Array

In order to correctly estimate the DoA, the elements of any antenna array should be uniformly distributed. In estimating the DoA, (represented by θ in Fig. 14), the Wi-Fi signal collected by the server through Wi-Fi AP should be projected in the same direction to that of the antenna array elements. After taking snapshots, each Wi-Fi AP rotates a ϕ angle from its previous position. This is known as the *phase angle*. We denote the distance between two antennas as d and r is the radius, which is equal to $d/2$. In order to obtain the signals' projection after receiving the information about the Wi-Fi signal at all snapshots, the Wi-Fi signal is multiplied by a complex number which is known as the *steering vector*. This multiplication results in linearly combining the steering vectors with the received Wi-Fi signals. As a result, the phases of the received Wi-Fi signal will also be aligned. Now, according to basic channel information, the received channel information will be:

$$h_i = \frac{1}{d} e^{\frac{-j2\pi}{\lambda}(d + r\cos(\theta - \phi_i))} \tag{25}$$

where $i = 1, 2, 3, ..., n$ denotes the snapshot position, (r, ϕ_i) are the polar coordinates at the i^{th} snapshot, d is the distance from the source and λ is the signal's wavelength. This equation represents the information of the received wireless channel h_i at the i^{th}

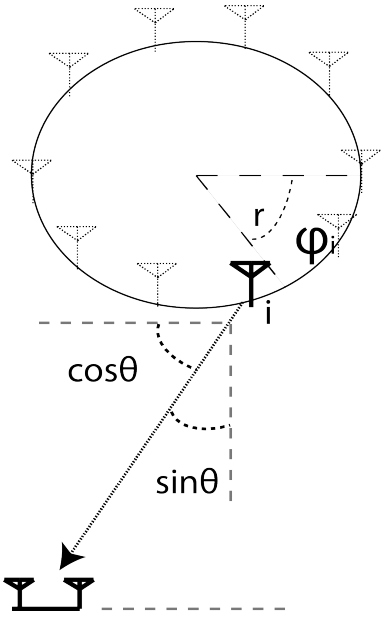


Figure 16: Wi-Fi AP in rotating condition.

snapshot. Now we need to multiply the received Wi-Fi signal by the *steering vectors* using Eqn. (20). As we have already illustrated, the steering vectors of an antenna array are different for every antenna in the array. For a stationary device, the steering vector at the i^{th} position can be written as

$$S_i = e^{\frac{\pm j\pi}{\lambda} r cos(\theta - \phi_i)} \tag{26}$$

where θ is the direction of the source and $i = 1, 2, 3, ..., n$ represents the *steering vector* at the i^{th} element of the antenna array.

For moving objects, this steering vector can be written as

$$S_i = e^{\frac{\pm j\pi}{\lambda} r sin(\theta - \phi_i)} \tag{27}$$

4.1.2 Steering Vector Configuration for a Wi-Fi AP

In this Section we are going to demonstrate how to configure the steering vectors for rotating MSTracker's Wi-Fi APs so that its DoA estimation does not hamper with this rotation. At first, we have defined the circular trajectory along which the

Wi-Fi APs will be rotated. We have assumed a full 2π circular way for experimental purposes. When defining the circular path, we have also considered that each Wi-Fi AP will take wireless signals' snapshot after traversing every "$\pi/12$" distance from its previous position. Yet this rotational movement becomes a challenge for configuring the steering vector. To correctly configure the steering vector in the presence of this rotation translation of Wi-Fi APs, we leveraged the current smartphones' wireless cards and two antennas' MIMO capability. MIMO-enabled smartphone antennas split the received wireless signal at each point of the trajectory space. Using this principle, each wireless signal snapshot can produce an individual steering vector for each antenna position. When moving in a circular fashion, the distance between two Wi-Fi AP antennas remains the same all the time. From this, we have tried to construct a relative wireless channel from the received signal of two antennas, which fully depends on the positional difference between the two antennas rather than on their absolute position. This procedure helps us form the steering vector with the same spacing between all the elements of a large-size virtual antenna array.

Let us concentrate on Fig. 16 to understand the whole mechanism. We are considering that a Wi-Fi AP is situated at a reasonable distance D from the receiver because Wi-Fi APs are attached to the ceiling of a floor or located at any corner of a floor space. The distance between two receiver antennas, $antenna1$ and $antenna2$ is denoted by d. The central point of Wi-Fi APs' circular path is O and r is the radius of that circular path. Now assume we are going to take snapshots N times along the circular path. So at each snapshot, the Wi-Fi AP will traverse π/N distance in a circular manner from its previous point. This circular distance traversal is defined by ϕ. Now the wireless channel snapshots at the two antennas, according to the basic wireless channel modes [31], are given below:

$$h_{1,i} = \frac{1}{d} e^{\frac{-j2\pi}{\lambda}(d)} \tag{28}$$

$$h_{2,i} = \frac{1}{d} e^{\frac{-j2\pi}{\lambda}(d + rcos(\theta - \phi_i))} \tag{29}$$

where λ denotes the Wi-Fi signal's wavelength. From the above expression, the relative wireless channel will be derived by the following equation:

$$H_i = h_{2,i}.h_{1,i}^* = \frac{1}{d}e^{\frac{-j2\pi}{\lambda}(d)}\frac{1}{d}e^{\frac{-j2\pi}{\lambda}(d+rcos(\theta-\phi_i))} = \frac{1}{d^2}e^{\frac{-j2\pi}{\lambda}rcos(\theta-\phi_i)} \tag{30}$$

where the value of d is set to $\frac{2\pi}{N}$, with N being the number of snapshots.

The subsequent procedures are the same as those discussed in Chapter 3.

4.1.3 Spatial Smoothing of the Wireless Signal

As we have stated earlier, the MUSIC algorithm requires uncorrelated input signals in order to correctly operate. To make sure the Wi-Fi signals carried by the newly configured virtual antenna array are uncorrelated, we have applied spatial smoothing [26] before using them as an input signal for MUSIC. With fully correlated signals, MUSIC hardly performs well to estimate the DoA. As we have configured our antenna array while moving the Wi-Fi AP antennas in a circular manner, it is crucial to apply spatial smoothing to the antenna array signals. The procedure we have followed to apply spatial smoothing is already discussed in Section 3.1.3.

4.2 Obtaining DoA in Numeric Values

For autonomous systems like mobile devices, Wi-Fi APs, etc., it is very important to arrive at a numerical solution in the easiest possible way. Here we mimick the same processs described in Section 3.2.

4.3 Multipath Suppression

One of the major challenges when working with wireless devices in indoor environments is mitigating the effects of multipath propagation of the wireless signals. Although the correlation matrix we configured for our work is considered a solution for multipath suppression, working with MUSIC or Root-MUSIC sometimes we experienced more than one solution or root for DoA estimation. So it can be stated that the correlation matrix is not enough to solve multipath problems for the Wi-Fi signal. To mitigate this multipath problem in indoor environments and estimate DoA more

efficiently, we have proposed an *iterative location estimation* (ITLE) algorithm, which allows us to reduce our localization error to 36 cm at the most.

First of all, we have buffered all the roots produced by Root-MUSIC for a wireless device from all the Wi-Fi APs active at that time. We have considered each AP, one by one, for creating the buffer array. From our experimental observations, we have decided to consider the three highest roots for an AP and configured the buffer array in the following way:

$$\begin{bmatrix} RootAP_{1,1} & RootAP_{1,2} & RootAP_{1,3} \\ RootAP_{2,1} & RootAP_{2,2} & RootAP_{2,3} \\ RootAP_{3,1} & RootAP_{3,2} & RootAP_{3,3} \\ \dots & \dots & \dots \\ RootAP_{n,1} & RootAP_{n,2} & RootAP_{n,3} \end{bmatrix} \tag{31}$$

After preparing this matrix, we applied the triangulation method on the first set of DoA values of the three APs e.g: $RootAP_{1,1}$, $RootAP_{2,1}$ and $RootAP_{3,1}$. We applied triangulation on $RootAP_{1,1}$ and $RootAP_{2,1}$ at first and estimated a position of that wireless device. Next, we run triangulation on $RootAP_{2,1}$ and $RootAP_{3,1}$ and derive the position for these two DoA values. After finding this new position, we calculated the distance between the two estimated positions. If the distance between those two points was more than 10 cm, we considered a virtual straight line between the points, the midpoint of which was taken as the new position of the target wireless device. We have established a threshold distance of 10 cm between two estimated positions as our experimental studies show that we are getting a median error of 8-10 cm in position estimation. Subsequently, we removed AP1's value and continued the same process over the first DoA values of AP2, AP3 and AP4; e.g., $RootAP_{2,1}$, $RootAP_{3,1}$ and $RootAP_{4,1}$. Once done with the first DoA values, we have taken the second values of active Wi-Fi APs and perform the same position estimation procedure. This process is repeated until all values of the matrix given in Eqn. 24 are traversed, where n represents the number of APs.

Algorithm 2 Iterative Location Estimation

```
//Consider Traingulation(X,Y) as a procedure
// which returns the intersection point of two DoAs
//Consider MidPoint(X,Y) as a procedure
// which returns the middle point of a straight line between two points


m = 0, n = 0, LocEst = ∞, Temp = 0
M = Number of APs, N = Number of Roots
for i = 1 to M do
    for j = 1 to N do
        AP(i)(j) ← Numerical DoA estimated by Root-MUSIC
    end for
end for
LocEst= Triangulation(AP(m,n),AP(m+1,n))
m = m + 1,
while (n <= N) do
    while (m − 1 <= M) do
        Temp = Triangulation(AP(m,n),AP(m+1,n))
        if Distance(LocEst, Temp) > 10cm then
            LocEst = MidPoint(LocEst, Temp)
        end if
        m = m + 1
    end while
    m = 0, n = n + 1
end while
```

4.4 Packet Collision and OFDM Carriers

It has already been observed that the possibility of preamble collision between two Wi-Fi signal transmissions is 0.6 % when the packet size is 1,000 bytes each [5]. Therefore, MSTRacker's performance is not hampered with the overlap between two packets. At each snapshot, this algorithm considers a group of transmitted and received signals and takes their average. So the effects of packet collision are mitigated.

At the same time, we know each of the Wi-Fi signal bears OFDM carriers. We used the "CSItool" [16] for extracting the signal information, including up to 30 OFDM sub-carriers. For our work, we again take an average of these and consider them as a single signal.

4.5 Distinction Between Building Levels

In reality, most of the Wi-Fi enabled mobile devices, like mobile phones, printers, scanners, etc., are kept on tables, moving carts or holding them in the hands, which typically means they are 1-1.5 m off from the ground. On the other hand, Wi-Fi APs are attached to the ceiling or placed at the top corner of a wall, which makes them be 2.5-3 m off from a building floor. Typically, each building floor carries individual APs attached to the ceiling. We have extended MSTracker to provide 3D localization following the basics of 3D Circular SAR [17]. As a result, we can calculate the height of a device from the floor level and also be able to track a device according to the level or floor number of a building.

MSTracker can perform indoor localization in 3D environments too. To accomplish this, we have to calculate the steering vector in the 3D space. From the basics of SAR (for both linear way and circular way), we have found that with the help of the polar and azimuth angles of the incoming signal from the Wi-Fi transceiver, the DoA can be accurately estimated, even when the user moves the device along the 3D environmental space. Assume the azimuth angle and the polar angle of the incoming signal are represented by θ and ψ, respectively. Let the position of the two antennas of the mobile device be Φ in polar angle and γ in azimuth angle. From Eqn. (20)

and the basics of SAR in 3D environment, the steering vector will be:

$$S_i = e^{\frac{+j\pi}{\lambda} r cos(\theta - \Phi_i) sin(\psi - \gamma_i)} \tag{32}$$

Chapter 5

Simulation Results

5.1 SAR in the Receiver Side

In this section we will evaluate the performance of SAR for indoor localization when applied at the receiver end. We have carried out experiments in various ways and will show the experimental results to get an overall sense of the performance of our proposed model, *MuSLoc*. We have evaluated MuSLoc's performance along three criteria: i) Characteristics , ii) DoA Estimation and iii) Device Localization

Characteristics: In this Section we have assessed the model performance by varying different parameters at the time of exchanging Wi-Fi signals.

DoA Estimation: We evaluated the model performance in terms of estimating the DoA of the Wi-Fi signal.

Device Localization: We will show how well our model performs when localizing a device relative to the Wi-Fi APs.

5.1.1 Implementation Details

For the experimental setup of our method, we used a DELL Inspiron 14R N4110 core i5 laptop (it comes with two receiving antennas) to which we added an Intel 5300 wireless card and ran it with Ubuntu Linux. For getting the movement (gyroscope, accelerometer readings) of the laptop in the user's hand, we attached a Samsung

Galaxy S4 mobile with it as shown in Fig. 17a (for working with a regular mobile gyroscope and real-time data, wireless antennas are positioned 3 cm away to work in the 5 GHz environment). We resorted to the "CSItool" [16] for obtaining information about the Wi-Fi signal. As the Wi-Fi signal carries OFDM information, each time we considered the average of the channels' OFDM information.

We developed our whole simulation environment using Matlab. At the time of collecting data, we asked our users to twist the laptop along its settings as much as they can with variation in speed and position. We also asked our users to walk and hold the device in their hand. We have employed 5-GHz-bandwidth-enabled APs and configured the laptop in such a way that it can exchange beacon packets with APs 10 times per second. We combined the gyroscope reading with the Wi-Fi signal information and simulated this to estimate the angle of arrival.

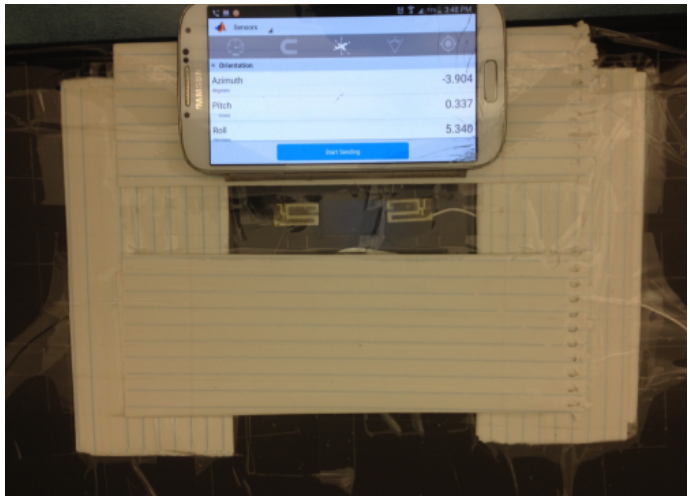


Figure 17: Device attached with mobile sensors for performing experiment

Fig. 17 illustrates the experimental setup for performing experiments with our model. We have attached mobile sensors to our laptop for getting their readings. We also bring out the Wi-Fi antenna of our laptop for improving the performance and analyzing the received signal. For real-time experiments, we have asked our users to carry this device with them and walk. We also asked them to twist this device as much as they can. We recorded our results and analyzed them later to see how well

our developed system performs.

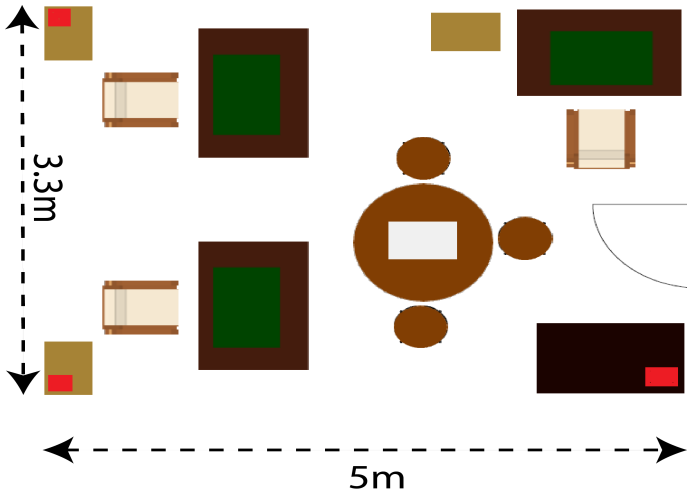
We have conducted our whole experiment in two environments. At first, we carried out an experiment inside a 150-square-feet room and then proceeded to repeat the same experiment at a university building's open group study space in order to to make sure we are in presence of a real-life complex multipath environment (as shown in Fig.18a, closed-room environment, and Fig.18b, open space in the first floor). Inside the closed room, we placed tables and chairs closely to create the multipath environment. We installed three APs in three corners of the room for localization purposes; they are marked as red squares in the figure. On the other hand, the university's open space was already equipped with shelves and group study tables.

We placed a few more shelves and tables to create a multipath environment and installed 6 APs (marked in the figure with red squares) inside the open space. We fixed the AP positions and tried to get information from direct measurements. In both of these environments, we were successful in localizing our device with a centimeter accuracy and estimating the DoA with a median error of less than 3°.

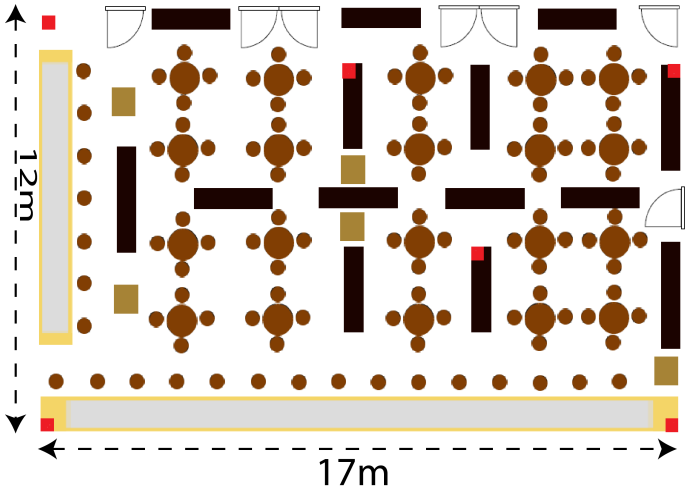
Baseline: We compared MuSLoc with another two-antenna-array-based localization system which also works on the DoA basics. We chose this method as it is more similar to MuSLoc. Like MuSLoc, this method also performs indoor localization without any additional infrastructure. Moreover, this technique works directly with the Wi-Fi signal, which is MuSLoc's chief objective. For OFDM information, we have taken the average of the total number of received channels (the CSI tool is limited to 30 OFDM channels).

5.1.2 Characteristics

Before evaluating the SAR performance in the receiver end on a real-time device localization problem, we are going to see how the whole system reacts to the received Wi-Fi signal, i.e., whether it takes a lot of time or lots of Wi-Fi signal information to estimate the DoA. The main purpose of this experiment is to check whether this method is feasible to work in a real-time environment at all.



(a) Floor plan for small closed room



(b) Floor plan for open-space study place in a university building

Figure 18: Custom-made environment for performing experimental analysis

Method: We performed our experiment in a small room first and then in a university group study open space, as already discussed above. We twisted the device from 0 to 360 degrees and we received this deviation reading from the mobile gyroscope. We tried to evaluate the results with a different number of Wi-Fi signal snapshots (created this variation with the device twisting speed). For configuring the *steering vector*, which is based on the number of antenna elements (in other words, the number of Wi-Fi signal snapshots), each time we registered the gyroscope reading (deviation in angle from the device's previous position) when a fixed deviation angle had crossed, e.g., 15°, 20° or 30°, and then add the channel information measured by the "CSItool" [16]. We tried to take readings from a distance of 2-16 m from an access point and finally evaluated the gathered data.

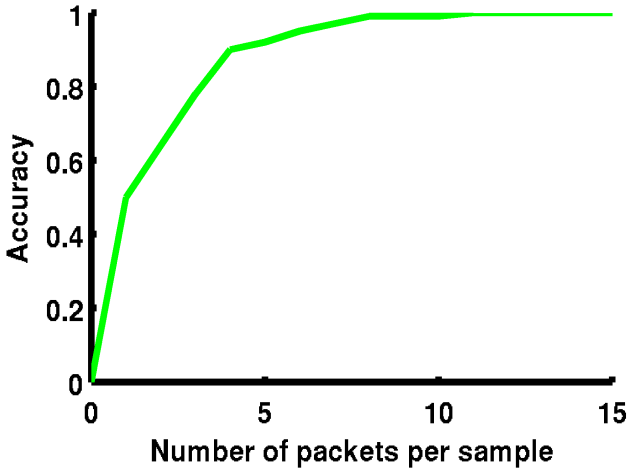


Figure 19: DoA accuracy based on packets per sample

Results: From the results in Fig.19 we can show how our system responds to the variation in the number of signal samples for accurately calculating the DoA. We have tried to twist our device up to 360° with a 15° deviation between each snapshot. From this figure, we can see that the accuracy in DoA estimation increased with a higher number of packets being exchanged. The DoA estimation is 95% accurate when we put 6 packets information in per sample. After that, the result becomes stable and the accuracy did not significantly improve. Hence, we fixed 6 packets per sample and used their average for performing each sample.

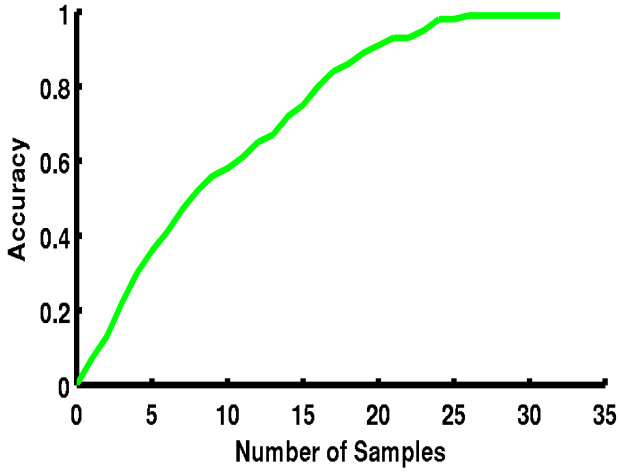


Figure 20: DoA accuracy based on the number of samples

After selecting the number of packets per sample, then we tried to determine how many samples were needed to achieve an accurate DoA estimation. We decided to use 6 packets per sample given our previous experimental results. We started our experiment with 4 samples per snapshots, just to realize that the DoA estimation accuracy was very low. We then increased the number of samples and checked the DoA estimation accuracy. When we reached 24 samples, we found out that we had achieved 95% accuracy in DoA estimation and after that the improvement in DoA estimation could be deemed negligible. Therefore, we decided to fix the number of samples to 24.

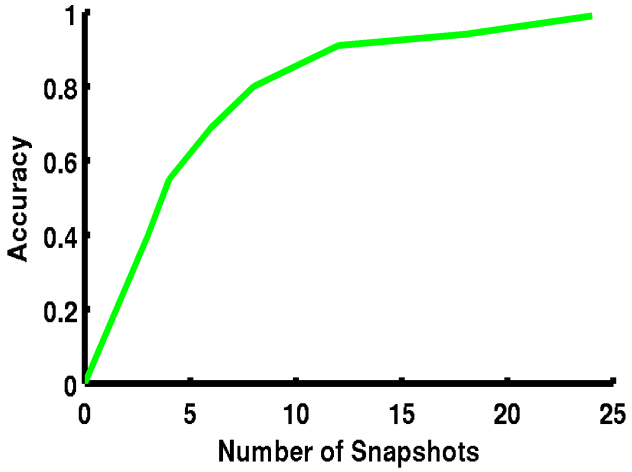


Figure 21: DoA accuracy based on the number of snapshots

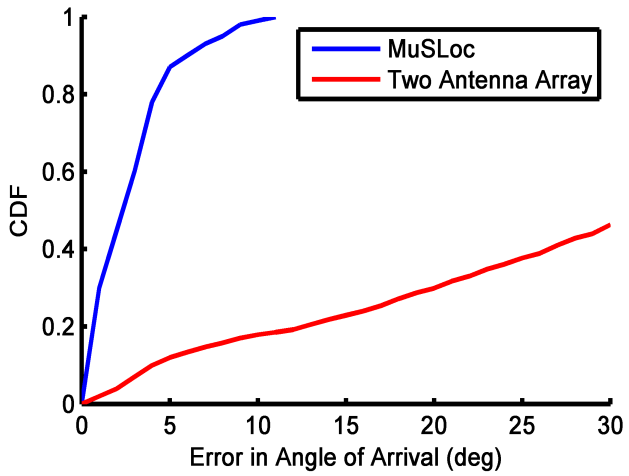
After fixing the number of samples to 24 and the number of beacon packets per sample to 6, we checked how many snapshots were required to get a good DoA estimate. We started experimenting with 3 snapshots at the beginning and checked the resulting DoA estimation accuracy. We noticed that the DoA estimation accuracy remained under 25%. We then started increasing the number of snapshots and observed that after reaching 12 snapshots in a circular way, the accuracy of DoA estimation exceeded 90%. We finally fixed "Diminishing Return" points for our experiment as: the number of samples was set to 24, the number of snapshots to 12 and the number of beacon packets to 6. With these configurations (almost 95 cases) we had observed that the median DoA estimation error remained smaller than 3°, which leveraged the device localization accuracy at the centimeter scale.

5.1.3 DoA Estimation

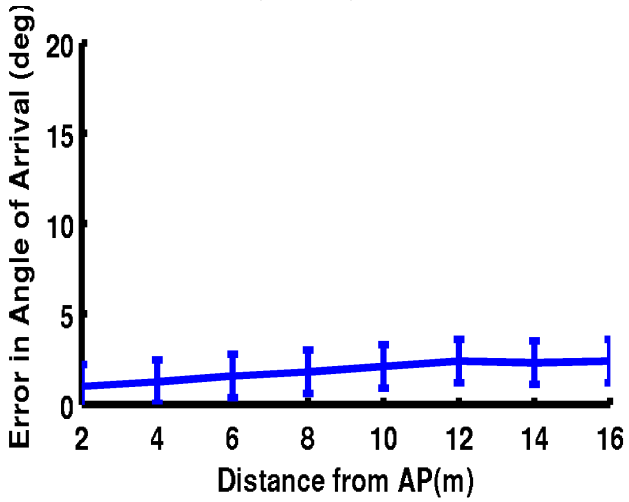
As we discussed earlier, we asked our users to twist the device at any point in the trajectory and we calculated the DoA estimation accuracy by collecting and evaluating the information from each movement of the device in the hands of the user.

Method: We performed the experiment of determining the DoA, which is easily understandable by an automated device, based on Equation (17). In both of the

environments, we first fixed the position of the APs and gathered their position information with respect to the room measurements and the floor map. Then we placed our device from 2-16 m away from different APs and evaluated our work. To simulate NLOS settings, in the closed room we laid a center table with a printing machine (marked in white color in Fig. 22b) and in the open-space study center of the university building we placed notice boards, standing book shelves, etc., which are good obstacles for the passing of a Wi-Fi signal.



(a) DoA error comparison for MuSLoc (in blue) and the two-antenna-array scheme (in red)



(b) AoA error as a function of distance from the APs

Figure 22: Performance in DoA estimation

Results: Fig. 22(a) and (b) reveal that this procedure achieves an accuracy with a median error of 2.8° approximately; that is, it stays within 4° even if the device is located far away from any of the APs. The two-antenna-array-based DoA system performs indoor localization with a median error of 45°, which is not feasible at all for an automated device. From Fig.20(b) we can also conclude that even without any knowledge in the device's twisting trajectory, MUSLoc's DoA estimation accuracy does not suffer much (the highest observed error is 7.8° from the actual DoA).

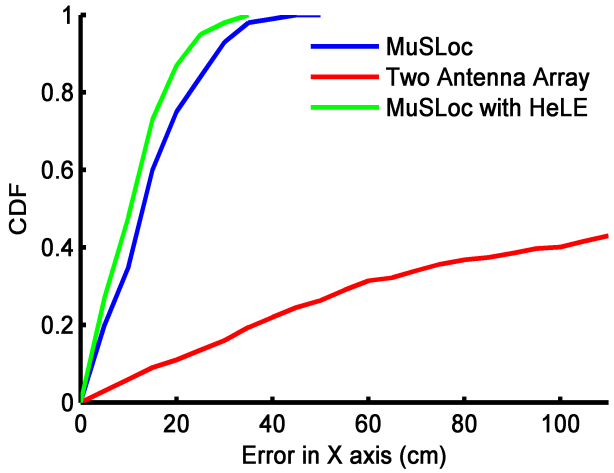
5.1.4 Device Localization

In this experiment, we are trying to localize our device with the help of the readings received from the different APs. We employed the triangulation method [23] to calculate the location of the device relative to the APs. After that, we tried to determine the accuracy in the localization with respect to the X axis and Y axis information used in the calculations.

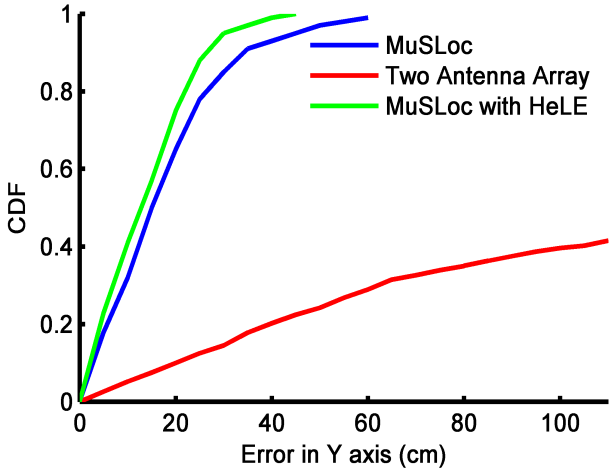
Method: We gathered data from 40 users and used them to evaluate the device indoor localization method. Each user was asked to rotate the device without keeping an eye on the rotating trajectory, which ensures random twisting of the device. After gathering DoA data, we determined the relative position of the device based on those of the APs, since their locations are already measured with respect to the environment, like room or open-space group discussion place we have identified earlier. For device localization purposes, we determined the position in the XY plane for each environment for both LOS and NLOS settings. We aggregate data from both settings to calculate the localization accuracy for the XY place, which will demonstrate our method's performance in real-life environments.

Results: Fig. 23(a,b) outlines the results for localizing our device along both the X axis and the Y axis. We used 6 APs and found out that our method localized the device with a median error of 20 cm along the X axis and 24 cm along the Y axis. Actually, these small errors in both dimensions happen not only because of the error in DoA estimation but also due to measurement inaccuracies in the relative APs position, in addition to limitations in the regular triangulation method for localization. To solve

h



(a) Error comparison for the X-axis position estimation. Blue represents MuSLoc and Red stands for the two-antenna array



(b) Error comparison for the Y-axis position estimation. Blue represents MuSLoc and Red stands for the two-antenna-array

Figure 23: Results representing localization performance in a real-time experiment

this, we develop HeLE and applied it together with MuSLoc. With the help of the HeLE algorithm, MuSLoc outperforms its location estimation performance. Along with HelE, MuSLoc succeeds in localizing that same device with a median error of 10 cm along the X axis and 12 cm along the Y axis. MuSLoc is clearly superior to the two-antenna-array-based technique, which yielded a the localization error confined to the centimeter scale in only 35-38 % of the cases.

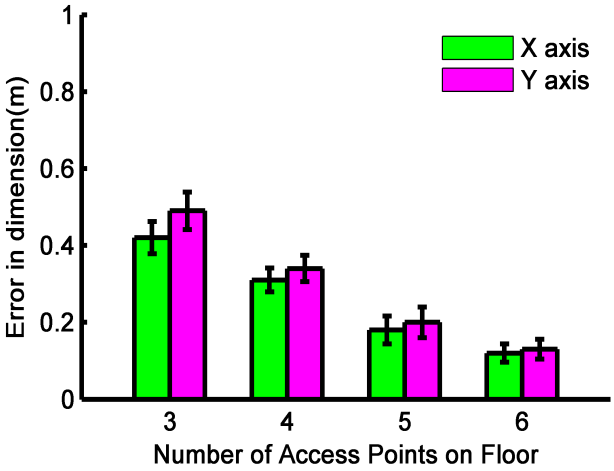


Figure 24: Error in 2D device location estimation

Fig.24 demonstrates the influence of a varying number of relative APs upon the localization accuracy in the X/Y dimensions. Using 3 APs, we discovered that we obtained an X/Y location accuracy with a median error of 50-60 cm. As we increase the number of Wi-Fi APs, this median error was lower than that one obtained with 3 APs. When 5 Wi-Fi APs are used, the X/Y location estimation error has dropped to a median error of less than 20 cm; for 6 APs, it comes under 15 cm. From these findings, we can conclude that if the relative position of the MuSLoc-installed device is measured with an appropriate number of APs, it efficiently mitigates the multipath effects present in NLOS settings.

5.2 SAR in Transmitter Side

In this Section we will discuss how SAR performs when applied to the Wi-Fi AP (transmitter) side. We already referred to this approach as "MSTracker"; hence, we will use the same name in the upcoming Sections. Its performance will again be discussed from 3 different angles: i) Characteristics, ii) DoA Estimation, and iii) Device Localization.

5.2.1 Experimental Setup

For experimental purposes with MSTracker, we have chosen the Electrical Engineering & Computer Science building of the University of Ottawa which already has plenty of office rooms, both big and small, and multiple corridors. We have our MSTracker installed on the Wi-Fi APs in different positions of the corridors and on the corners of a single building floor. We have used rolling racks for placing our newly configured Wi-Fi devices and have placed it on rotational bases, which helped rotate our devices up to 360°. Fig. 25 shows a snapshot of our floor plan in preparation for the experiment. The red squares indicate the positions of the APs, which we have situated there for performing the experiments.

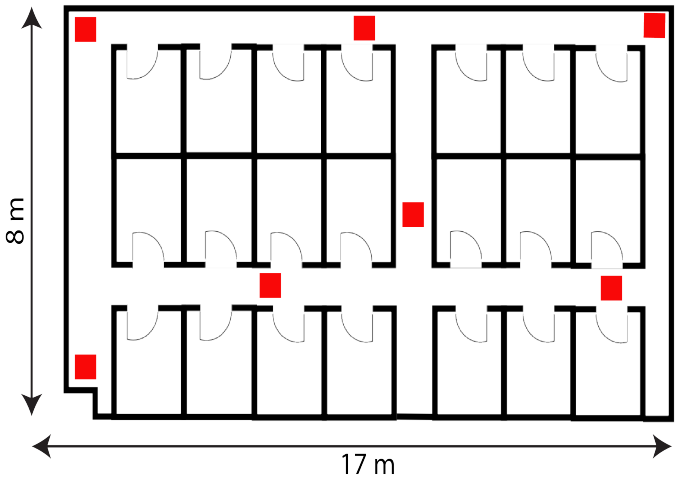


Figure 25: Floor plan for MSTracker

Fig. 26 demonstrates a sample experimental setup for MSTracker. The red circle in the figure denotes our configured AP. We have used a Dell Inspiron N4110 14R laptop as the server node, with Ubuntu Linux 10.04 server version as the operating system. The green circle represents the target Wi-Fi device whose location we are going to calculate. We have kept our AP on top of a roller rack which was almost at the ceiling height. We have attached a rotational base under the AP, which helped rotate the AP when evaluating the results of localizing a Wi-Fi device.

Figure 26: Floor plan for MSTracker

5.2.2 Characteristics

Method: We have performed our experiment in a university building corridor. The configured Wi-Fi device sat on a movable cart and we calculated its position with the help of the floor map. We twisted the AP from 0° to 360° along the predefined circular path. We also tracked the deviation reading received from the mobile gyroscope.

Results were evaluated with a different number of Wi-Fi signal snapshots (we created this variation with the device's twisting speed). For configuring the *steering vector*, which is based on the number of antenna elements (in other words, the number of Wi-Fi signal snapshots), each time we registered the gyroscope reading (angle deviation from the device's previous position) when a fixed deviation angle had crossed e.g., 15°, 20° or 30° and then added the channel information reported by the "CSItool" [16]. Readings were taken from a distance between 2 and 16 m from each AP and finally evaluated using the gathered data.

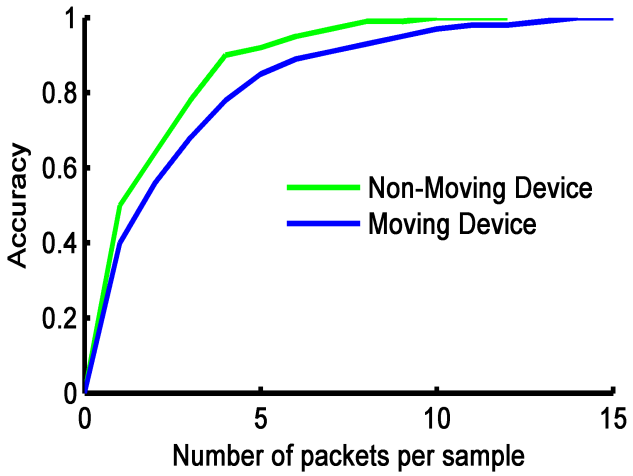


Figure 27: DoA accuracy based on packets per sample

Results: Fig.24 schematically portrays how our system responds to the variation in the number of signal samples for calculating the DoA accurately. Our device was twisted up to 360° with a 15° deviation between each snapshot. From this figure, we can see that the DoA estimation accuracy increased as the number of exchanged packets was raised. The DoA estimation is 95% accurate when we lean on 4 packets information per sample for a stationary device and 5+ packets for moving devices. After that, the result became stable and the accuracy did not significantly grow. Therefore, we set 5 packets per sample for both moving and stationary devices and use their average for processing each sample.

After selecting the number of packets per sample, we proceed to identify the

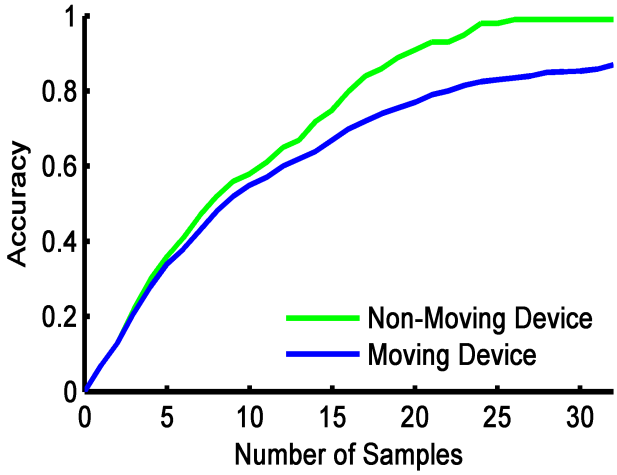


Figure 28: DoA sccuracy based on sample numbers

number of samples needed to arrive at an accurate DoA estimation. As we have already decided to use 5 packets per sample, we will stick to that value for this experiment. We started our experiment using 4 samples per snapshot but this lead to a very low DoA estimation accuracy for both stationary and moving devices. We then increased the number of samples and calculated the resulting DoA estimation accuracy. When we reached 20 samples, we learned that we had achieved 90% plus accuracy in DoA estimation for stationary devices and over 80% accuracy for moving devices. After that, the accuracy improvement rate becomes slow and almost stable. Hence, we decided to set the number of samples to 20.

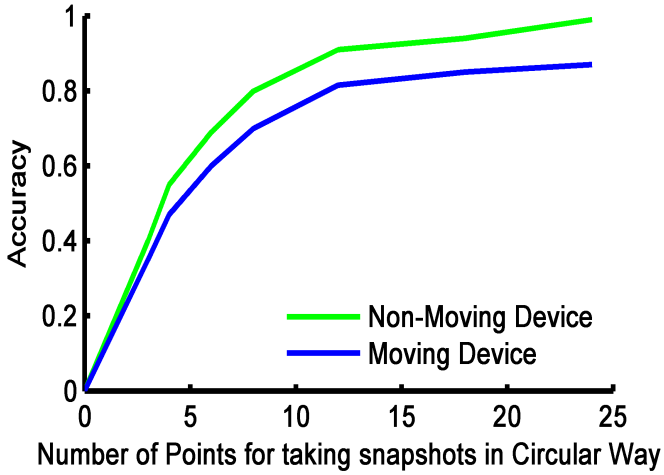


Figure 29: DoA accuracy based on the number of snapshots

After fixing the number of samples to 20 and the number of beacon packets per sample to 5, we investigated how many snapshots were required to estimate the DoA with a decent accuracy. We began experimenting with 3 snapshots and discovered that the DoA estimation accuracy remained under 25 %. Then we increased the number of snapshots and after reaching 12 snapshots in a circular way, the DoA estimation accuracy exceeded 90%. We have finally adopted "Diminishing Return" points for our experiment as follows: the number of samples was set to 20, the number of snapshots to 12 and the number of beacon packets to 5. With this configuration (almost 95 cases), we observed that the median DoA estimation error was confined to less than 2.8°, which leveraged the device localization accuracy at the centimeter scale.

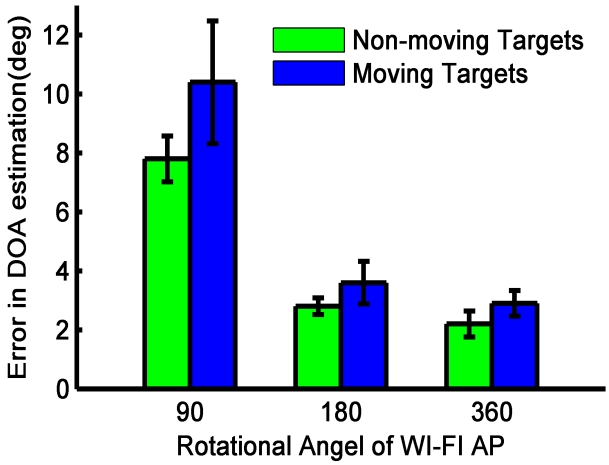


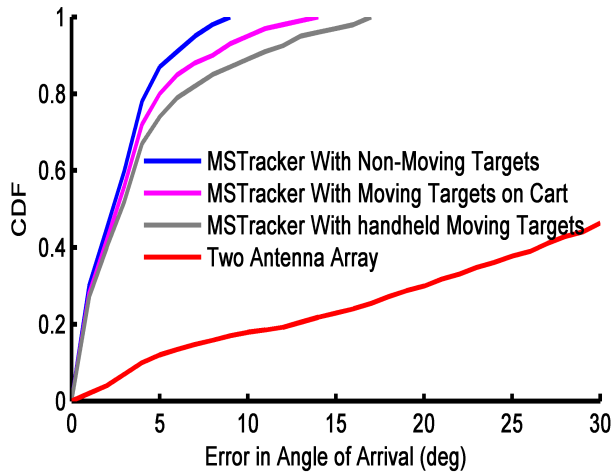
Figure 30: DoA accuracy based on degree of rotation

Finally, we wondered whether we needed to rotate the Wi-Fi AP to full 2π in order to get the same result, or whether a small portion of circular rotation could attain nearly the same result. From Fig. 20 we can infer that 90° rotation performs poorer than 180° and 360°. Our decision was to then rotate our device up to 180°. This is the value adopted in the subsequent experiments.

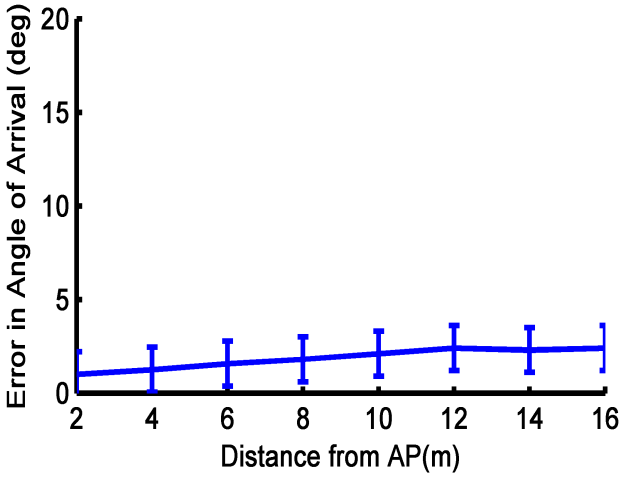
5.2.3 DoA Estimation

As we have discussed earlier, we rotated the Wi-Fi device and tracked the Wi-Fi enabled handheld devices. For stationary devices, we sat them on a table, or a shelf, even on the floor directly and tried to localize them. For moving devices, we used two processes. For mobile phones, we asked our users to walk to and fro at regular speed while keeping their mobile phone in their hand. For printers or scanners, we placed them on a moving cart and moved them a little slower than the phones.

Method: We performed the experiment of calculating the numerical DoA, which is easily understandable by an automated device based on Eqn. (24). We ran the entire experiment in different corridors and building floors and gathered their position information with respect to the AP positions according to the floor map. We placed



(a) Comparison of DoA estimation accuracy for various devices.



(b) AoA error for different distances from the APs

Figure 31: MSTracker performance in DoA estimation

our target devices from 2 to 16 m away from the Wi-Fi and calculated the DoA estimation accuracy for both stationary and moving devices. To simulate NLOS settings, we placed additional notice boards, standing bookshelves, signal reflectors etc., which are good obstacles for the passing of the Wi-Fi signal.

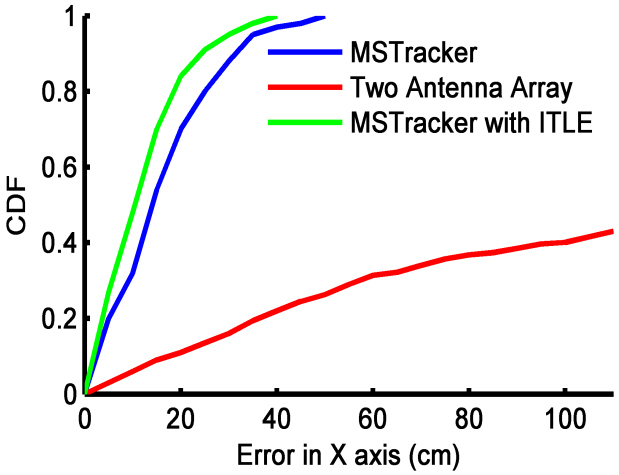
Results: Fig. 31(a) and (b) uncovered the fact that this procedure achieves an accuracy with a DoA median error of approximately 2.8° for stationary devices; this proves that the error remains within 4° even if the device is placed at a large distance from any of the APs. For moving devices, the DoA estimation accuracy varies with their speed but still performs much better than the two-antenna-array-based DoA scheme, which yields a a median DoA estimation error of 45°, hence not making this approach feasible for indoor localization purposes.

5.2.4 Device Localization

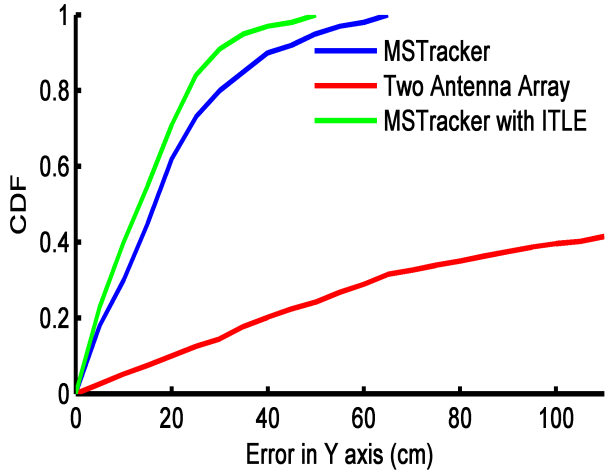
We attempted to localize our target devices with the help of the readings received from the different APs; we employed the triangulation method [23] to calculate the location of the device relative to the APs. After that, we determined the localization accuracy with respect to the X, Y and Z axes information resulting from the calculations.

Method: We recruited 40 users for data gathering and performance evaluation purposes. Each user was asked to walk at a regular speed while holding their target device in the hand. After gathering DoA data for both moving and stationary devices, we estimated the relative position of the devices based on the APs, whose locations were known beforehand. For device localization, we estimated the position in the X/Y plane for each environment with both LOS and NLOS settings. We aggregated data from both settings in order to compute the localization accuracy in the X-Y plane, which will demonstrate our method's performance in real-life environments.

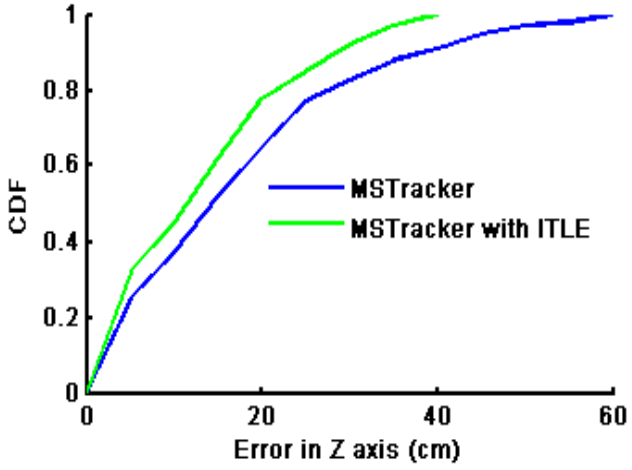
Results: Fig. 32(a,b,c) report the results of localizing our device in a 3D space. We used 6 APs and learned that our method localized the device with a median error of 23 cm along the X axis, 28 cm along the Y axis and 25 cm along the Z axis. Actually, these small errors along a particular dimension happen not only because of DoA estimation errors but also due to measurement inaccuracies in the relative



(a) Comparison of X-axis position estimation error.



(b) Comparison of Y-axis position estimation error.



(c) Comparison of Z-axis position estimation error.

Figure 32: Device localization performance results in a real-time experiment

APs positions as well as known limitations in the regular triangulation method. To counter these defficiencies, we developed the ITLE algorithm and applied it together with MSTracker. With the help of the ITLE algorithm, MSTracker outperforms its own location estimation performance and succeeds in localizing those same target devices with a median error of 8 cm along the X axis and 9 cm along the Y axis. We did not apply the algorithm for the Z axis. MSTracker is evidently superior to the two-antenna-array-based DoA technique, which was only capable of limiting the localization error to the centimeter scale in up to 35-38% of the cases.

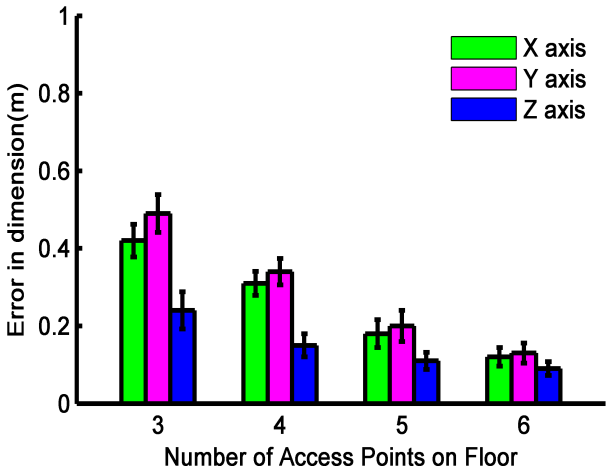


Figure 33: Error in 3D device location estimation

Fig. 33 confirms that MSTracker enabled Wi-Fi APs' performance to localize different devices in the X/Y plane. From the figure we can notice that, with 3 relative APs, the location accuracy in the X/Y axes oscillated between 45 and 60 cm. As we increased the number of relative APs, both the calculation complexity and the location estimation accuracy also soared. With 5 relative Wi-Fi APs, the location estimation accuracy is bounded by 20 cm and with 6 APs, this metric was further reduced to under 14 cm for both X and Y axes. Further increments in the number of APs should result in a more accurate location estimation for both moving and stationary devices.

Chapter 6

Conclusions

This thesis addressed existing problems of indoor localization with SAR in both the transmitter and receiver sides and emphasized on solutions for these. We first discussed various ways of solving indoor localization problems proposed by different researchers and highlighted their limitations and benefits. We then divided our work intwo different phases. First, we developed a novel technique to perform SAR-based indoor localization, called MuSLoc [1], where we applied the newly defined Root-MUSIC algorithm over Circular SAR as a feasible indoor localization solution. We also developed a novel algorithm to mitigate multipath problems inside a building. We tested our model inside a university building. The experimental results obtained after performing real-time evaluation have corroborated that our model outperforms other existing indoor localization models and methods in terms of both accuracy and cost. Our model is device-agnostic and has zero cost to implement as it requires no additional hardware support. We also solved some accuracy problems by combining Circular SAR with Root-MUSIC in order to obtain a centimeter-level-accurate indoor localization process.

In the next phase, we developed another indoor localization approach where we worked on the Wi-Fi AP side. We baptized this model as MSTracker [2]. We also developed a mathematical model for estimating the DoA with the help of the modified

MUSIC algorithm over SAR and devised an Iterative Location Estimation procedure for tracking devices inside a building more accurately. This model can track both stationary and moving Wi-Fi devices inside a building without any specially developed hardware support or environment. The proposed model can be extended as a security mechanism for a building. We have run MSTracker inside one of the buildings at the University of Ottawa campus. Our experimental results confirm that MSTracker outperforms the only available model named "ArrayTrack" [5] in terms of accuracy. MSTracker also estimates the location of different devices with a centimeter-level accuracy even in the worst cases. This model can distinguish between floors of a building, which helps track devices fairly smoothly at different building levels.

Although our two novel models exhibited superior accuracy in the indoor localization process and capped the location estimation error to a median error of 20 cm, there are still some limitations that deserve further thoughts. The developed algorithms work perfectly when estimating DoA and are fairly easy to implement; however they are still a tad complex in terms of time and space requirements. Both of the algorithms finished their execution in finite time. Again, for MSTracker we needed to rotate our device in order to solving certain SAR problems when tracking stationary devices.

In the future, we will focus on extending our work in the following ways:

- We will continue our experiments with the developed algorithms to reduce their time and space complexity.
- We will keep searching for more accurate and straightforward ways to solve issues in SAR-based indoor localization and will aim to develop a model that further refines our current techniques by making them more accurate, efficient and easy to work with.
- We will continue our work on MSTracker so that we can develop an updated version of this model which can overcome its existing challenges.

Bibliography

[1] K. W. Nafi, W. Gong, A. Nayak, "MuSLoc: A centimeter level indoor localization with COTS", Submitted to *IEEE Transaction on Instrumentation and Measurement.*

[2] K. W. Nafi, W. Gong, A. Nayak, "MSTracker: Wi-Fi device tracker with centimeter accuracy", in preparation.

[3] K. Liu, X. Liu, and X. Li, "Guoguo: Enabling Fine-grained Indoor Localization via Smartphone" *In proceedings of MobiSys13*, Taipei, Taiwan, June 25-28, 2013, pages 235-248.

[4] J. Wang, D. Katabi "Dude, where's my card?: RFID positioning that works with multipath and non line of sight" *In proceedings of SIGCOMM*, Hong Kong, China, August 12-16, 2013, pg: 51-62

[5] J. Xiong, K. Jamieson, "Arraytrack: A fine grained indoor localization system" *In proceedings of 10th USENIX Symposium on Networked Systems Design and Implementation* , 2013

[6] S. P. Tarzia, P. A. Dinda, R. P. Dick, G. Memik, "Indoor Localization without Infrastructure using the Acoustic Background Spectrum" *In proceedings of MobiSys*, Bethesda, Maryland, USA, June 28-July 1, 2011, pages: 155-168

[7] K. Joshi, S. Hong, S. Katti "PinPoint: Localizing Interfering Radios", *In proceedings of 10th USENIX Symposium on Networked Systems Design and Implementation*, 2013, pages 241-254,

[8] Y. Shu, P. Cheng, Y. Gu, J. Chen, T. He, "TOC: Localizing Wireless Rechargeable Sensors with Time of Charge", *In proceedings of IEEE INFOCOM*, Toronto, Ontario, Canada, 2014, pp.388-396.

[9] H. Liu, Y. Gan, J. Yang, S. Sidhom, Y. Wang, Y. Chen, F. Ye., "Push the limit of WiFi based localization for smartphones.", *In proceedings of the 18th annual international conference on Mobile computing and networking*,pp: 305-316, 2012.

[10] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, R. Sen., "Zee: Zero-effort crowdsourcing for indoor localization.", *In proceedings of the 18th annual international conference on Mobile computing and networking*, pp: 293-304, 2012.

[11] J. Wang, D. Fang, X. Chen, Z. Yang, T. Xing, Lin Cai,"LCS: Compressive Sensing based Device-Free Localization for Multiple Targets in Sensor Networks." *In proceedings of INFOCOM*, 2013, Turin, pages: 145 - 149.

[12] Y. Chen, N. Crespi, L. Lv, M. Li, A. M. Ortiz, and L. Shu,"Locating using Prior Information: Wireless Indoor Localization Algorithm." *In proceedings of ACM SIGCOMM*, Hong Kong, China, August 12-16, 2013, Pages 463-464.

[13] S. Kumar, S. Gil, D. Katabi, D. Rus, "Accurate Indoor Localization With Zero Start-up Cost.", *In Proceedings of the 20th annual international conference on Mobile computing and networking*, September 7-11, Maui, Hawaii, USA, Pages 483-494

[14] Z. W. Geem ,"Music-Inspired Harmony Search Algorithm: Theory and Applications" Springer, December 8, 2010

[15] R. Fenby ,"Limitations on directional patterns of phase-compensated circular arrays", *In Radio and Electronic Engineer*,vol.30, issue.4, 1965, pages: 206 - 222.

[16] D. Halperin, W. Hu, A. Sheth, D. Wetherall ,"Tool release: Gathering 802.11n traces with channel state information." *ACM SIGCOMM CCR*, 2011.

[17] P. J. Fitch,"Synthetic Aperture Radar", Springer, 1988.

[18] M. Soumekh,"Synthetic Aperture Radar Signal Processing", Wiley, 1999.

[19] L. J. Cutrona,"Synthetic Aperture Radar", McGraw Hill, 1970.

[20] D. E. Wahl; C.V.Jakowatz Jr.; P. A. Thompson; D. C. Ghiglia, "New approach to strip-map SAR autofocus." *In Digital Signal Processing Workshop*, 2-5 Oct 1994, pp.53-56,

[21] J. Mittermayer; A. Moreira; O. Loffeld, "Spotlight SAR data processing using the frequency scaling algorithm" *IEEE Transactions on Geoscience and Remote Sensing*, vol.37, no.5, pp.2198-2214, Sep 1999.

[22] M. Soumekh, "Reconnaissance with slant plane circular SAR imaging *IEEE Transactions on Image Processing*, vol.5, no.8, pp.1252-1265, Aug 1999

[23] M. Li, W. Trappe, Y. Zhang, B. Nath, "Robust statistical methods for wireless localization in sensor networks.", *In proceedings of Information Processing in Sensor Networks*, IPSN 2005, pages 91-98.

[24] Y. Gao, J. Niu, R. Zhou, G. Xing, "ZiFind: Exploiting Cross-Technology Interference Signatures for Energy-Efficient Indoor Localization", *In proceedings of INFOCOM*, pp.2940-2948, Turin, Italy, 2013.

[25] 4G Americas,"MIMO and Smart Antennas for Mobile Systems. July, 2013.

[26] M. Waz, J. Sheinvald, "Direction finding of coherent signals via spatial smoothing for uniform circular arrays.", *IEEE transactions on antenna and propagation*, Vol.42, No.5, pages: 613 - 620, May 1994.

[27] W. Y. Yang et. al. "Matlab Simulink for Digital Signal Processing" Hongrung Publishing Company, 2012.

[28] R. O. Schmidt, "Multiple Emitter Location and Signal Parameter Estimation," *IEEE Trans. Antennas Propagation*, vol. AP-34, March 1986, pp.276-280

[29] V. F. Pisarenko, "The retrieval of harmonics from a covariance function Geophysics" Journal of Roy. Astron. Soc., vol. 33, pp. 347-366, 1973

[30] D. Tse, P. Viswanath, "Fundamentals of Wireless Communication." Cambridge Press University, May 2005.

[31] A. Goldsmith, "Wireless Communication." Cambridge Press University, 2005.

[32] R. V. Nee, "Spread-spectrum code and carrier synchronization errors caused by multipath and interference." *IEEE Transactions on Aerospace and Electronic Systems*, vol.29, pages: 13591365, 1993.

[33] A. Vesa, "Direction of Arrival Estimation using MUSIC and RootMUSIC Algorithm." *18th Telecommunications forum TELFOR 2010*, Serbia, Belgrade, November 23-25, 2010.

[34] A. Ward, A. Jones, and A. Hopper "A new location technique for the active office. *IEEE Personal Communications*, vol.4, issue.5, pages: 4247, Oct. 1997.

[35] N. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket location-support system." *In proceedings of the ACM MobiCom Conference*, Aug. 2000, pages 3243.

[36] H. Liu, Y. Gan, J. Yang, S. Sidhom, Y. Wang, Y. Chen, F. Ye, "Push the Limit of WiFi based Localization for Smartphones", *In proceedings of MobiCom*, Istanbul, Turkey, August 2226, 2012, pages 305-316.

[37] Y. Kuo, P. Pannuto, K. Hsiao, and P. Dutta, "Luxapose: Indoor Positioning with Mobile Phones and Visible Light" *In proceedings of SIGCOMM*, Chicago, Illinois, August 17-22, 2014, pages 447-458.

[38] M. Azizyan, R. R. Choudhury "SurroundSense: Mobile Phone Localization Using Ambient Sound and Light" *Mobile Computing and Communications Review*, Volume 13, Number 1.

[39] M. Kotaru, K. Joshi, D. Bharadia, S. Katti, "SpotFi: Decimeter Level Localization Using WiFi" *In proceedings of SIGCOMM*, August 17 - 21, 2015, London, United Kingdom, pages 269-282.

[40] M. Niessner, A. Dai, and M. Fisher, "Combining Inertial Navigation and ICP for Real-time 3D Surface Reconstruction.", Eurographics, 2014.

[41] I. Constandache, R. R. Choudhury, I. Rhee, "CompAcc: Using Mobile Phone Compasses and Accelerometers for Localization", *In proceedings of SIGCOMM*, Helsinki, Finland, August 13-17, 2012.

[42] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury "No need to war-drive: Unsupervised indoor localization." *In proceedings of MobiSys*, Lake District, United Kingdom, 2012, pages 197-210.

[43] J. Gjengset, J. Xiong, G. McPhillips, and K. Jamieson, "Phaser: Enabling phased array signal processing on commodity wifi access points", *In proceedings of MobiCom*, Sept 7-11, Maui, Hawaii, pages 153-164.

[44] J. Wang, F. Adib, R. Knepper, D. Katabi, and D. Rus. "RF-compass: Robot object manipulation using RFIDs." *In proceedings of MobiCom*, Sept 30-Oct 4, Miami, Florida, 2013, Pages 3-14.

[45] Intel Product description: Intel Ultimate N WiFi Link, "Link: https://web.archive.org/web/20090126154959/http://download.intel.com/network/connectivit

[46] Y. Tian, R. Gao, K. Bian, F. Ye, T. Wang, Y. Wang, X. Li, "Towards Ubiquitous Indoor Localization Service Leveraging Environmental Physical Features." *In proceedings of INFOCOM*, April 27 - May 2, 2014, pp: 55-63.

[47] T. Roos, P. Myllymaki, and H. Tirri, "A probabilistic approach to WLAN user location estimation." *International Journal of Wireless Information Networks*, vol.9, issue.3, 2002.

[48] A. Smailagic, D. Siewiorek, J. Anhalt, D. Kogan, and Y. Wang. "Location sensing and privacy in a context aware computing environment." *In Pervasive Computing*, 2001.

[49] Z. Yang, C. Wu, and Y. Liu, "Locating in fingerprint space: Wireless indoor localization with little human intervention." *In proceedings of Mobicom*, August 22-26, Istanbul, Turkey, 2012, pages 269-280.

[50] M. Youssef and A. Agrawala, "The Horus WLAN location determination system." *In Proceedings of ACM MobiSys*, pages: 205-218, 2005.

[51] xirrus. Xirrus Corp. (http://www.xirrus.com).

[52] K. Wu, J. Xiao, Y. Yi, M. Gao, and L. Ni, "Fila: Fine-grained indoor localization," *In proceedings of INFOCOM*, 25-30 March, Orlando, FL, pages: 2210 - 2218.

[53] H. Lim, L.-C. Kung, J. C. Hou, and H. Luo, "Zero-configuration, robust indoor localization: Theory and experimentation" *In proceedings of INFOCOM*, Pages 1-12, Barcelona, Spain, April 2006.

[54] A. Goswami, L. E. Ortiz, and S. R. Das, "Wigem: A learning-based approach for indoor localization", *ACM CoNEXT*, December 69 2011, Tokyo, Japan .

[55] P. Bahl, V. N. Padmanabhan, and A. Balachandran, "Enhancements to the radar user location and tracking system", *Microsoft technical report*, 2000.

[56] M. Youssef, A. Youssef, C. Rieger, U. Shankar, and A. Agrawala, "Pinpoint: An asynchronous time-based location determination system", *In proceedings of MobiCom*, June 19-22, 2006, Uppsala, Sweden, pages 165-176.

[57] S. A. Golden and S. S. Bateman, "Sensor measurements for Wi-Fi location with emphasis on time-of-arrival ranging", *IEEE Transaction on Mobile Computing*, vol.6, issue.10, October 2007, pages 1185-1198 .

[58] A. T. Mariakakis, S. Sen, J. Lee, and K.-H. Kim, "Sail: Single access point-based indoor localization", *In proceedings of MobiSys*, Bretton Woods, NH, USA, 2014, pages 315-328.