INDOOR LOCALIZATION OF SMARTPHONES

A THESIS

Submitted by

NARAYAN SRINIVASAN

for the award of the degree

of

DUAL DEGREE

(BACHELOR OF TECHNOLOGY + MASTER OF TECHNOLOGY)

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY MADRAS

CHENNAI-600036

JUNE 2020

THESIS CERTIFICATE

This is to certify that the thesis entitled **"INDOOR LOCALIZATION OF SMARTPHONES"** submitted by **NARAYAN SRINIVASAN** to the Indian Institute of Technology, Madras for the award of the degree of **Bachelor of Technology + Master of Technology,** is a bonafide record of research work carried out by him under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Prof. Devendra Jalihal
Professor
Department of Electrical Engineering
Indian Institute of Technology Madras
Chennai – 600 036.

Place: Chennai

Date: June 2020

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my project guide Prof. Devendra Jalihal for his valuable guidance, support, constructive comments and encouragement during the course of this project. I would also like to thank Karthik and Swaathi from Skcript, Chennai, who offered me the chance to work on this problem to be presented for use at DWTC, Dubai. I am highly grateful to my faculty advisor Prof. Aravind R, for his support throughout my institute life.

I am indebted to all the professors and staff of the Department of Electrical Engineering for making my time at IIT-Madras so valuable. I would like to thank the members of Centre for Innovation for developing my interest in technology and their valuable suggestions. Finally, I would like to thank my parents, my friends particularly Foram Trivedi and Ajeyaa Gk for their unconditional support throughout my college life which has been priceless to me.

ABSTRACT

The need for an Indoor Localization system has grown in the past few years. Meanwhile the wireless sensors have been getting better and cheaper at the same time. This opens up the avenue to build a robust Indoor Localization system that can track a user's location in real time. Different technologies like WiFi, Bluetooth Low Energy (BLE), RFID and UWB have been used to build such services. Incase of smartphones most approaches rely on Received Signal Strength Indicator (RSSI) to estimate the distance, but they suffer from poor accuracy in fine localization as the method is prone to multipath fading and noisy fluctuations in indoor environment. In this thesis, we build a cloud-based Indoor Localization module that employs a BLE based localization module and a IMU based dead reckoning module which are combined using a innovative particle filter. This system is cost and power efficient, scalable all while being able to provide a localization accuracy under 2m consistently which is enough for most practical purposes.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

RSSI – Received Signal Strength Indicator

BLE – Bluetooth Low Energy

KF – Kalman Filter

PF – Particle Filter

DR – Dead Reckoning

DL – Deep Learning

SVM – Support Vector Machine

IoT – Internet of Things

PoA – Phase of Arrival

ToF – Time of Flight

TDoA – Time Difference of Arrival

IPS – Indoor Positioning System

RFID – Radio Frequency Identification Device

WKNN – Weighted K-Nearest Neighbours

FFT – Fast Fourier Transform

INTRODUCTION

With the advent of Internet of Things (IoT), a lot of services have erupted to improve the quality of life for users. Proximity based services have recently witnessed great interest and are put to use in various indoor settings for providing information, targeted advertisements and promotional offers. We see that localization in an outdoor setting has been one of the most researched topics of the past decades and has spawned the creation and quality of services like ride-hailing, food delivery, navigation amongst many others.

Indoor localization modules could help in improving the quality of life and the business prospects at the same time. It could help people navigate complex yet unfamiliar spaces like large malls, airports, trade center exhibitions etc. Among recent literature about indoor localization, wireless based indoor localization methods take up most proportion of them.

The widely accepted system in the outdoor setting is called the Global Positioning System. The construction materials used in modern buildings severely impact the signal quality of GNSS indoors. Indoor environments are also filled with a lot of fixed and moving obstacles while give rise to a plethora of undesirable effects causing reflection, scattering and refraction of signals. These reasons among others result in a very low GPS accuracy indoors. Another reason why we'd need a reliable IPS is because GPS is accurate up to around 5m under open sky for smartphones. This worsens when near bridges, buildings or trees. While a 5m accuracy would suffice for a turn by turn navigation app, it won't be the case for a library or a mall.

Fig.1 Indoor Localization System (Pictured)

The market for BLE based positioning is up and growing [1], we also see it already being adopted in airports, museums and malls [1,2,3]. BLE beacons are usually small, advertisement-emitting, battery-powered devices designed for easy deployment in a variety of indoor environments. Currently, there is richness in the BLE beacons availability regarding vendor, price range, supported protocols, transmission powers, advertisement frequency capabilities and battery lifespan [4]. WiFi is another option, but BLE provides higher achievable accuracies, lower battery drain, lower network traffic over WiFi [5,6].

Different localization techniques have been discussed in the literature for IPS, each have their own advantages but come with some constraints. Time of Arrival (ToA), Angle of Arrival (AoA), Phase of Arrival (PoA) are some of the methods used in the past. These methods however accurate require sensors that aren't available in a common modern-day smartphone. WiFi-RTT is a new protocol in WiFi 802.11mc, which allows for a computation of Return Tie of Flight (RToF) based on which the distance to the smartphone can be accurately estimated to <1m error. However, there are barely any devices (Routers and Smartphones) that are compatible with this protocol and even those that are compatible are extremely expensive and not scalable. Received Signal Strength Indicator (RSSI) based methods have been very popular in recent years, primarily due to the ease of implementation and lack of need for special hardware on the receiver. This method uses the correlation between the RSSI value and the distance of the receiver from the transmitter. While off the top implementation of RSSI based methods suffer from poor localization accuracy, in recent years there has been lot of research done in this region to improve it.

Aside from wireless technologies based localization, a lot of research has gone into using the IMU sensors namely the gyroscope, accelerometer and magnetometer in the smartphones to estimate the relative position. Recent development of microelectromechanical systems has made the sensors cheaper, accurate and widely available in most smartphones. While Dead Reckoning methods usually suffer from the problem of drift while taking measurements over longer time durations, we find that in shorter time periods these methods prove quite accurate.

BACKGROUND AND RELATED WORK

In this section we look at recent literature in the field of Indoor Localization. We discuss various localization techniques used, different types of wireless technologies, evaluation metrics for any IPS model.

1.2.1 Localization Techniques

Time of Flight (ToF) [22] : Here we use the signal propagation time between the Transmitter and the Receiver to compute the distance between them. ToF value is just multiplied by the speed of light 3 x 10^8 m/s, this means that we require strict synchronization of the clocks in transmitter and receiver. ToF estimation accuracy depends on two things, signal bandwidth and sampling rate. A high sampling rate is necessary to ensure that the signal doesn't arrive in between sampling periods. Frequency domain super-resolution is commonly used to obtain the ToF with high resolution from the channel frequency response. In most cases without a direct Line of Sight (LoS) path, the accuracy of ToF based systems would be poor. The signal usually comes with a timestamp corresponding to the time of transmission. Let T_t be the time of transmission and T_r be the time of receiving, then the distance d between the two is calculated as,

$$d = (T_r - T_t) * (3 * 10^8) \text{ m} \quad \text{.....(1)}$$

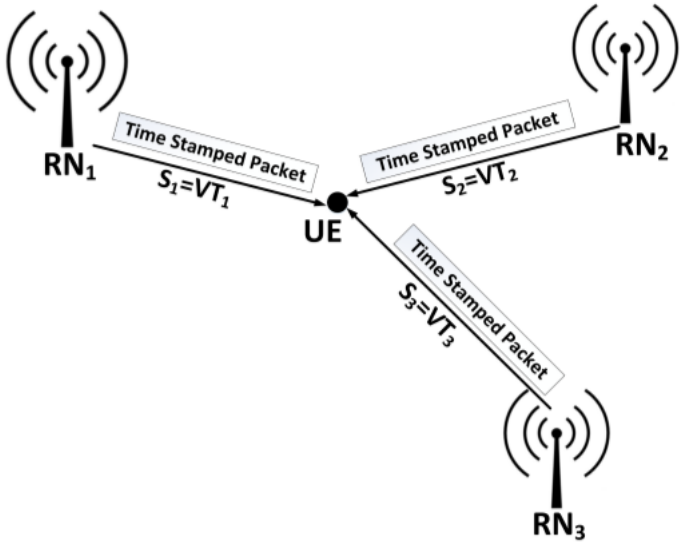


Fig 2. Time of Flight (ToF) localization

Angle of Arrival (AoA) : This method relies upon an array of antennas to calculate the angle at which a signal is incident on the array by using the different in time of arrival on the individual antennas. With just 3 transmitters we will be able to estimate the users location in 3D environment. This method is understandably more accurate when the distance between the transmitter and the receiver is small compared to long distance ranging where a small difference in angle would result in a large localization error. It also involves too much calibration and handcrafting compared to other methods. [20,21]

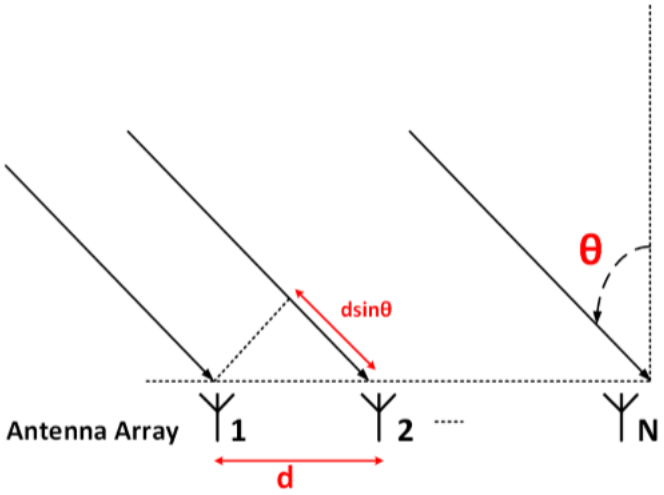


Fig.3 Angle of Arrival

Time Difference of Arrival (TDoA) : While the previous two methods discussed above require strict synchronization between the receiver and the transmitter, it is often hard to achieve that. In this method, we attempt to localize a device based on the difference in Time of Arrivals of signals coming from different transmitters. This method only requires synchronization between all the transmitters. The TDoA from at least three transmitters is needed to calculate the exact location of the receiver as the intersection of the three (or more) hyperboloids. The accuracy in TDoA method also depends on signal bandwidth, sampling rate and the presence of Line of Sight. [18,19]

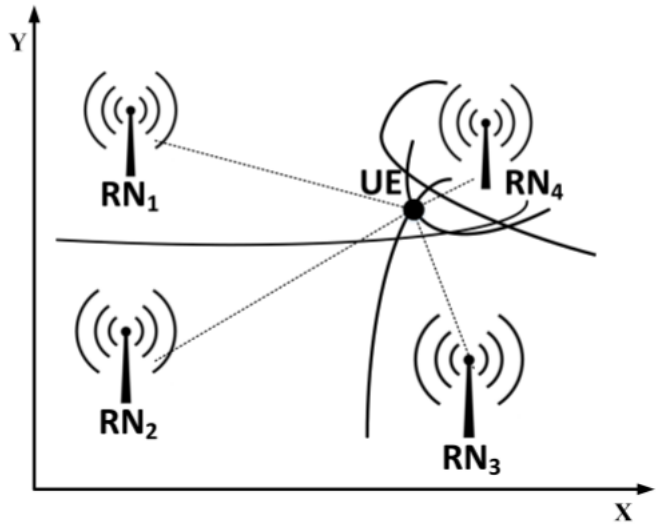


Fig.4 Time Difference of Arrival

Phase of Arrival (PoA) : PoA based approaches use the phase or phase difference of carrier signal to estimate the distance between the transmitter and the receiver. There are a number of techniques available to estimate the range or distance between the Tx and Rx using PoA. If the phase difference between two signals transmitted from different anchor points is used to estimate the distance, TDoA based algorithms can be used for localization. This method again requires an antenna arrays, which is not available in a regular smartphone. [16,17]

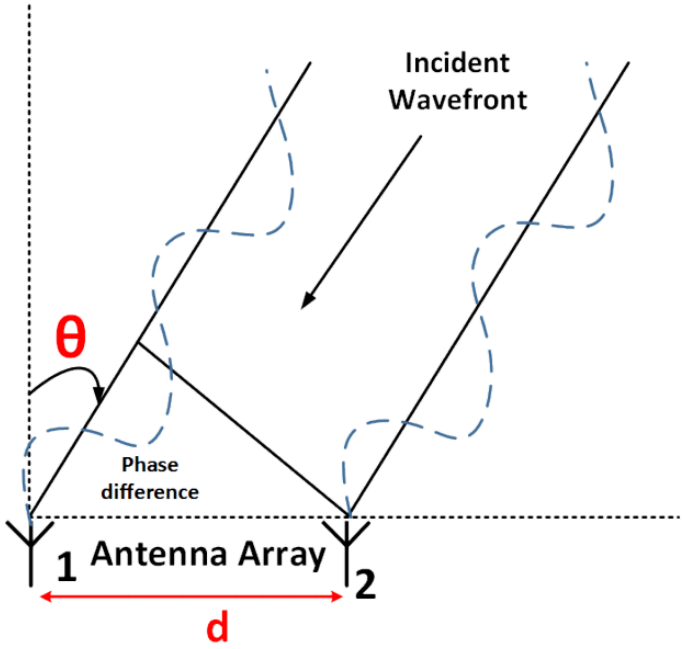


Fig.5 Phase of Arrival

RSSI Based Methods (RSSI) : Received Signal Strength is a measure of the strength of the signal at the receiver end and RSSI is an indicator of the RSS. There is correlation between RSSI value and the distance between the receiver and the transmitter. This approach is most widely used due to the ease of implementation and lack of requirement of additional sensors. Mostly implemented using either a multilateration or a fingerprinting approach [13,14,15] this method is prone to lower accuracy than others, but using sophisticated algorithms and probabilistic filtering we can get good accuracy with this too.

1.2.2 Wireless Technologies

WiFi (IEEE 802.11) : WiFi is one of the most popular methods used for RSSI based localization. Almost all the current smartphones are WiFi enabled and hence availability is not an issue. However, existing WiFi networks are normally deployed for communication rather than localization purposes and therefore novel and efficient algorithms are required to improve their localization accuracy. Moreover, the uncontrolled interference in the ISM band has been shown to affect the localization accuracy. However, WiFi based localization systems in recent times have been used to produce extremely precise localization systems. [9,10,11,12]. WiFi access points are hard to deploy in large numbers due to the price and power consumption.

Bluetooth Low Energy : Bluetooth (or IEEE 802.15.1) consists of the physical and MAC layers specifications for connecting different fixed or moving wireless devices within a certain personal space. The latest version of Bluetooth, i.e., Bluetooth Low Energy (BLE), also known as Bluetooth Smart, can provide an improved data rate of 24Mbps and coverage range of 70-100 meters with higher energy efficiency, as compared to older versions[8]. Also iBeacons are very cheap and widely available. The power consumption of the BLE Beacons are lower, albeit lower localization accuracy is achieved using this system.

Radio Frequency Identification Device (RFID) : RFID is primarily intended for transferring and storing data using electromagnetic transmission from a transmitter to any radio controlled receiver device [7]. It has been used to build many localization systems in the recent past. They are of two types, Active and Passive RFID. Active RFID systems are connected to a power source and can operate at large distances from the reader unlike passive RFID systems which have a limited range. However, the active RFID technology cannot achieve submeter accuracy and it is not readily available on most portable user devices

OUR MODEL

3.1 System Overview

Our model employs a Client-Server architecture in order to be scalable and step away from computational constraints imposed by different smartphone models users might possess. The user installs our client-app that collects data from BLE Beacons and transmits it to a server. The client-app also periodically transmits the data from smartphone IMU i.e gyroscope, magnetometer and accelerometer to the server. All the computation happens at the server end, and the device periodically requests the server to send its updated location coordinates.

The server is implemented using Flask-Python, and contains two POST endpoints where the client can send in BLE and IMU data respectively. The server also has one GET endpoint through which the client can request its location. All data is transmitted and received as HTTP requests. Client is an Android app running on a Oneplus 6T smartphone, while the server is a GPU-enabled Ubuntu machine rented on Google Cloud Platform. There are plenty of BLE Beacons available in the market, but for the purpose of this study we have used Estimote Beacons. Like most BLE beacons, these feature an internal battery and an app that lets us control transmission power, advertising interval as well as the protocol (Eddystone/iBeacon).

Fig.6 Available BLE Beacons

The system is mainly divided into 3 modules,

1. BLE Module
2. Dead Reckoning Module
3. Particle Filter

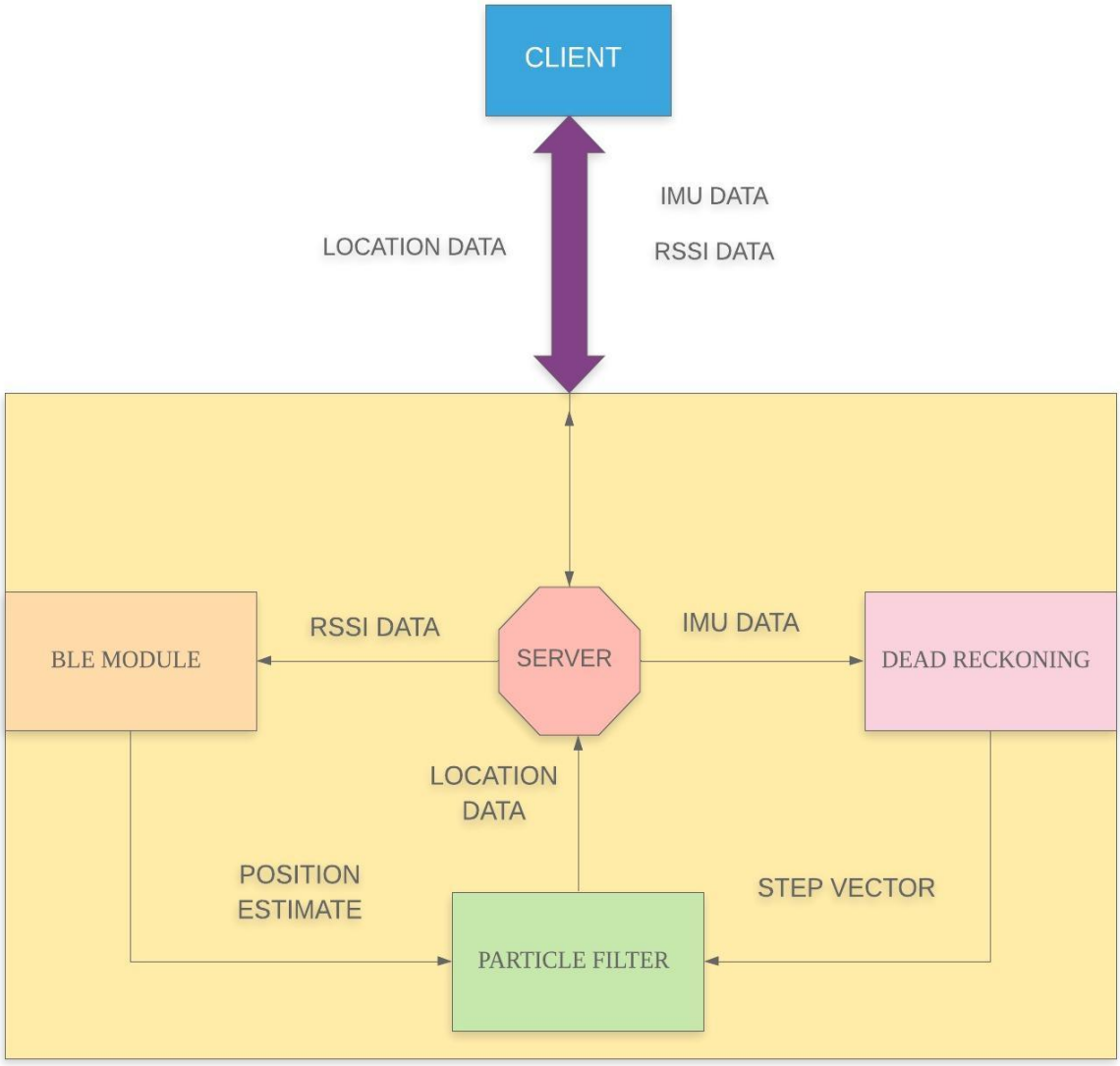


Fig.7 System Design

Before going into the details of the implementation, it is important to understand how the system views the map. The map is seen as a grid of 0.25 x 0.25m cells. Each cell is either an occupiable cell or a blocked cell. Blocked cells involve walls, furniture and other non-navigable areas. The first step in setting up this IPS module, is to define a map that corresponds to real life orientation of the indoor environment under consideration. The user's location is just computed as one of the grid cells on the map, and all errors are calculated as the Euclidean distance between the actual and estimated coordinates.

In order to make the system scalable, Particle Filter is implemented in a parallel manner using the PyCUDA[23] library. This allows us to do the resampling and computation of the particle weights in parallel for all particles, letting us use a larger number of particle with lower latency. This is a good reason to use particle filter[24] to fuse the sensor values aside from the fact that we can easily add other sensor measurements by just adjusting the weighting function without changing the structure of the filter.

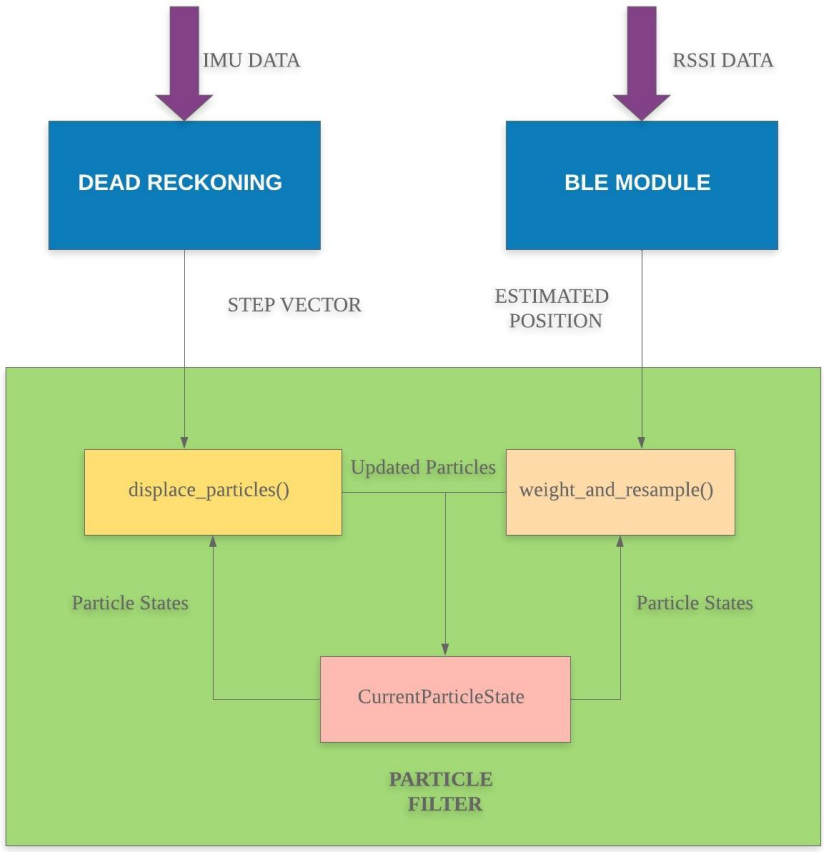


Fig. 8 Particle Filter Working

Particle filter initializes itself by randomly sampling particles all across the occupiable regions of the map. Now there are three functions a particle filter has to perform during it's run time,

1. **Predict State** – This is run by the function displace_particles( ), which is invoked whenever the Dead Reckoning module computes a step vector using the IMU data, the step vector is a 2D vector that tells the filter about the displacement of the user in the last time window. This function basically displaces all the particle in the direction of the step vector with a probability p = 0.85 and keeps them stable the other times.

2. **Compute Weight** – This is run by the function weight_and_resample( ) which invokes weight_particles( ). This is triggered whenever the BLE module computes a new position estimate based on RSSI data it receives. The particles are weighted in a parallel manner using PyCUDA[23] implementation.

3. **Resample Particles** – After computing the weights of the particles, this function resample_particles ( ) is invoked. This function basically resamples particles according to the new weight computed in the previous step of the process. The resampled particles update the CurrentParticleState vector.

Everytime an operation borrows the CurrentParticleState argument it places a lock on the vector so that if at all another operation wants to use it, it must wait for the previous operation to update the CurrentParticleState vector. This ensures order in computational graph. Another function a particle filter performs is providing location data to the Client whenever requested. The particle filter basically computes a weighted average of the top-k particles in the filter, to get the user location.

3.2 BLE Model

This model uses only the RSSI data coming from the BLE beacons to predict the position of the user. This can be extended to methods using WiFi RSSI and even other ranging methods discussed before, we would only need to modify the final location estimate function and nothing else in the whole system. For the transmission, BLE operates at an industrial, scientific, and medical (ISM) frequency band of 2.4 GHz. The frequency band is divided into 40 channels spaced at 2 MHz apart. Among the 40 channels, three channels are used for an advertisement. A BLE device uses these advertisement channels to broadcast its advertisement packets continuously.

The BLE Beacons transmit data in a periodic manner based on the advertisement frequency set , we can also adjust the transmission power of the signal which of course comes at the cost of battery life. The underlying concept we use is that the Received Signal Strength at the client end correlates with the distance from the beacon. This relationship is usually described using a path loss model [25] ,

$$RSSI(d) = RSSI(d = 1m) - 10n * \log(d)$$

Where,

d = Distance of the receiver from the transmitter

n = Path loss coefficient

The way the RSSI values received from the beacon are in the form of packets that contain the following data,

[RSSI Value , Timestamp , Beacon ID]

The model maintains a modified Kalman Filter [26] for each beacon, so that the RSSI values can be filtered through them as done in [27,28]. This is because there is a lot of undesired effect on the signal that's received like fast fading, blocking by obstacles or the person themselves, signals getting dropped etc. So it's important to estimate the actual value of the RSSI value and not just take the value that is received at this sensor.

Let’s do a preliminary analysis of RSSI values coming from a beacon, to see what kind of noise exists and how we can work on fighting the noise. We can see that there does exist correlation between RSSI values and the distance and employ a Kalman filter to solve the problem. The reason why we apply Kalman filter[27] is because of the ease of implementation and it’s ability to estimate the maximum aposteriori estimate of the state under Gaussian assumptions. Now RSSI isn’t exactly gaussian, but we still find that Kalman Filter helps us clear the unwanted noise in the data. Also to be noted that we can overcome the gaussian assumption constraint by employing a UKF (Unscented Kalman Filter).

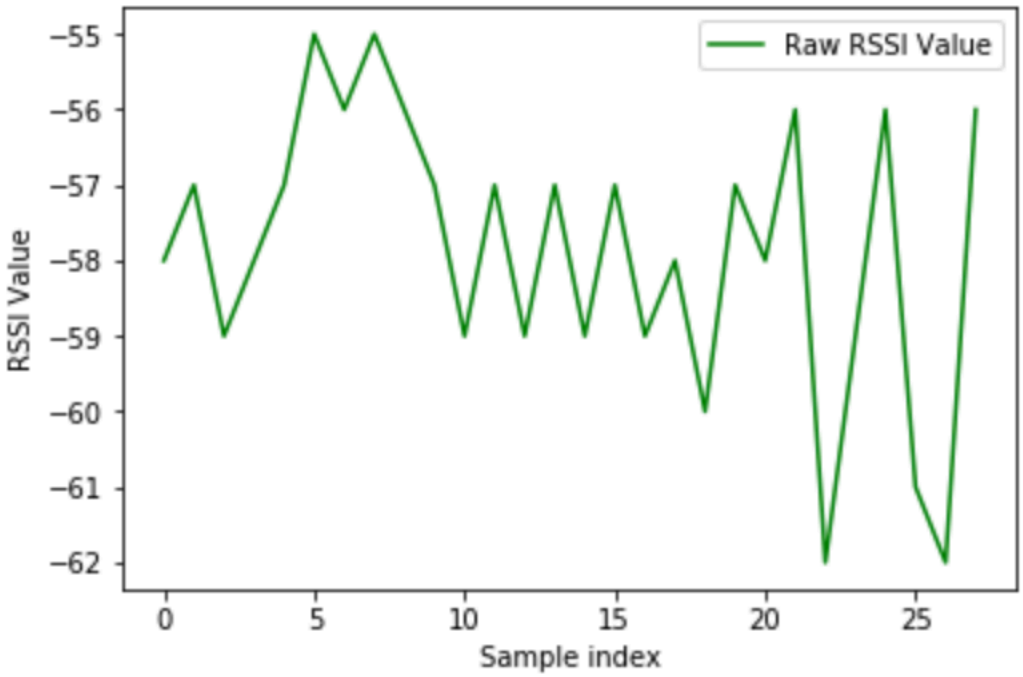


Fig.9 RSSI Value from Beacon at 1m

We can see that the value generally oscillates around a mean of - 58.5 dBm but there is significant noise in the system. The noise as we can observe isn’t gaussian, but it’s fairly symmetrical about the mean. Despite this we would like to use probabilistic filters to estimate the true value of RSSI before using it for further localization problems.

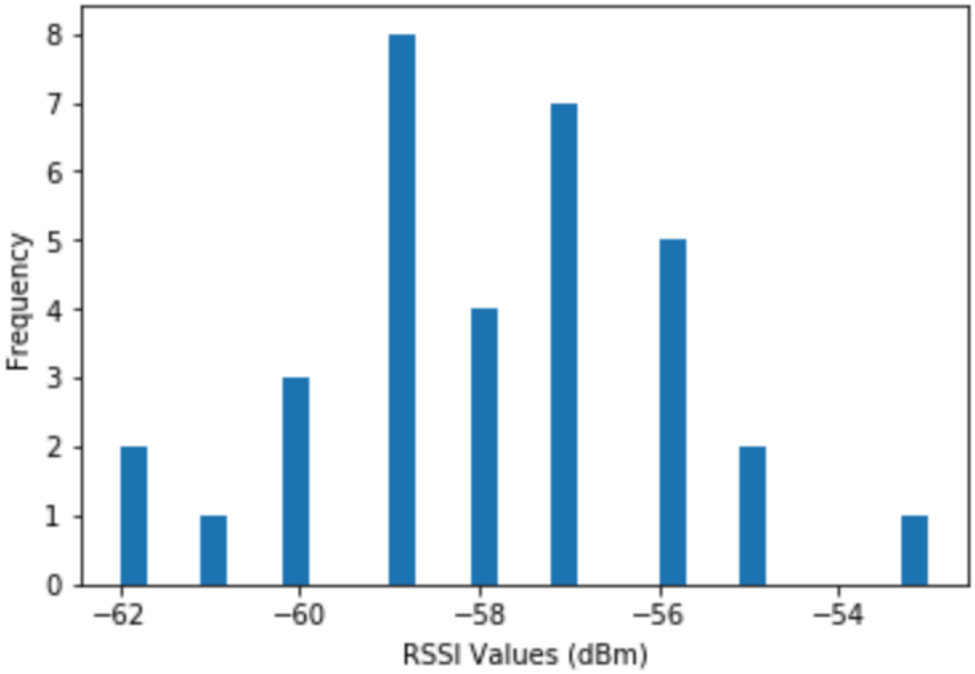


Fig 10. Symmetrical Distribution of Noise

The main feature of the Kalman filter is that we calculate the Kalman Gain, which signifies how useful is this current measurement. If it's deemed too noisy, it is discarded unlike the mean and median filters where every measurement is given equal importance. We estimate the parameters of the Kalman filters using a em algorithm [29] (with the first 5 measurements coming from the beacons), before initializing the filter.

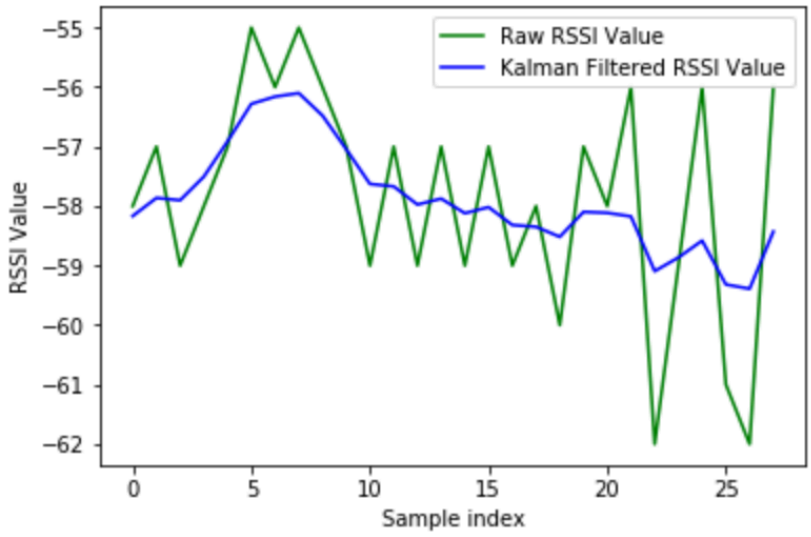


Fig 11. Kalman Filtered RSSI Value

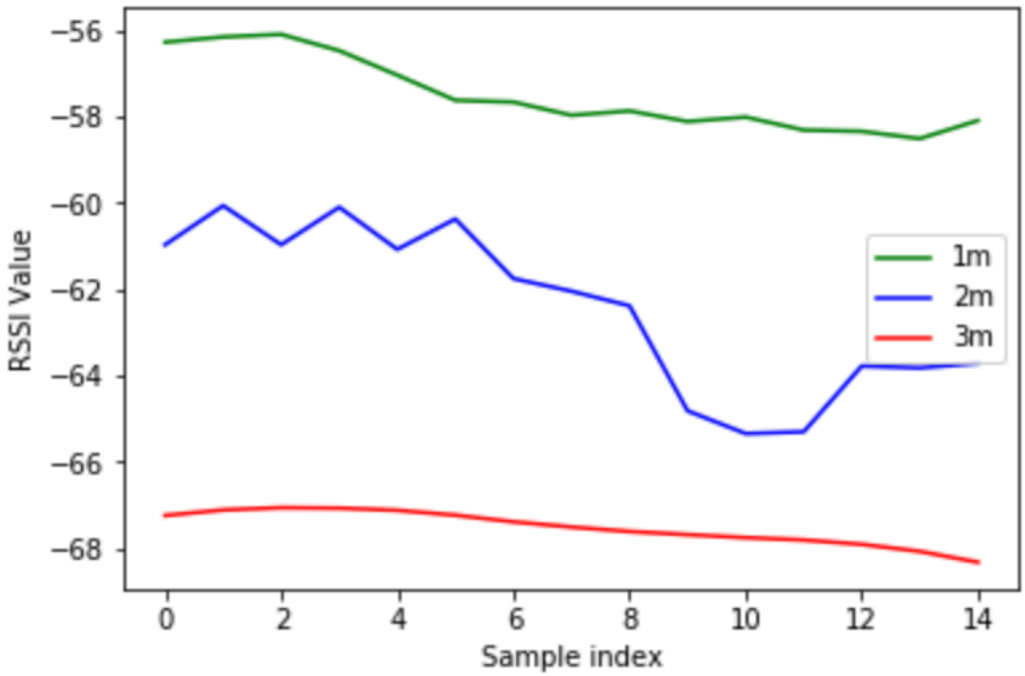


Fig 12. Kalman Filtered Values at different distances

We can see that post Kalman filtering we are able to get a reasonable accuracy on the RSSI data. The way the system handles the data is that in order to prevent RSSI data getting too old, the system maintains a map of BeaconID -> [Timestamp, Filtered RSSI]. Periodically the system runs a check and any RSSI value more than 5 seconds old is just cleared from the system. This is because as we walk away from beacons, we stop receiving signals from them and it's important to note that we are far away from said beacon. We don't usually a see a beacon signal get dropped for 5mins straight.

Now there are two main approaches to BLE based localization, Multilateration and Fingerprinting. Multilateration [30] based approach work by applying the path-loss model to the RSSI value and estimating the distance to a beacon. Once we have distances to multiple beacons we find a coordinate that minimizes the mean squared error in the distances we computed to beacons. This method suffers from the fact that Line of Sight is utterly essential for this method to have any good accuracy. This method also doesn't take into account the correlation between RSSI values of different beacons at a given location.

The other popular method is finger-printing [31,32,33], where we collect RSSI value from beacons at predefined reference points in the map. This data is stored in a database and is used in online localization. This method doesn't exactly require line of sight and only expects that the RSSI value at any particular location when filtered remains reasonably stable. Even if one beacon fluctuates we'd have the data from other beacons to be able to localize the person using this.

The offline phase of fingerprinting is designed for learning the RSSI at each reference point. At this stage, we collect RSSIs from all beacons in four directions at each measurement location. This is done to ensure that we account for blocking by the human body. This is a one-time process and needn't be updated unless there's a significant change in the surroundings. In the online phase, when the user's device starts transmitting the RSSI data to the server, the server periodically attempts to estimate the position of the user's device.

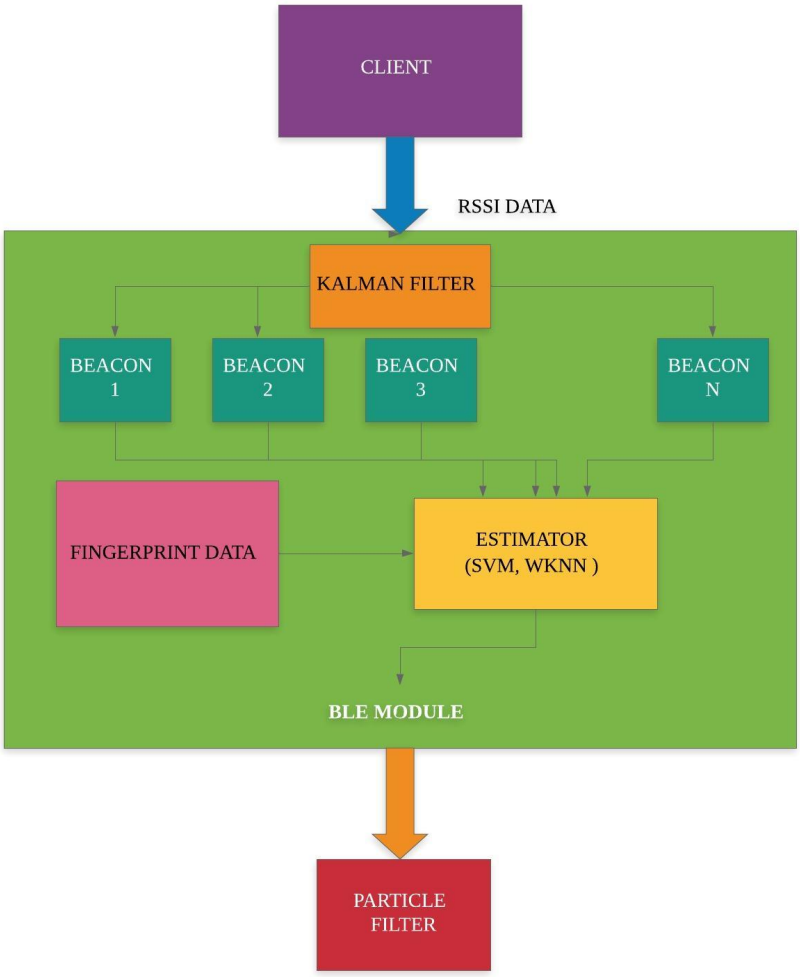


Fig.13 BLE Module

Once we have the required fingerprints collected from the Reference Points (RP) we store them in a database. There are many methods studied in literature to estimate positions based on fingerprinting in our model, we have adopted a modified version of Weighted K- Nearest Neighbours and an SVM based regressor. The methods are described below in detail.

Weighted K-Nearest Neighbours : As the name suggests this method involves simply picking the K closest reference points to the current RSSI fingerprint we have through online data. The first question we must answer is how do we define similarity. We use the following metric in our model. [31]

$$Similarity = cosine\text{-}sim(d_i , d_j) * (euclidean\_distance (d_i , d_j))^{-1}$$

This metric is used because we take into account both the magnitude as well as the direction of the vector into account as opposed to using only one of them. Secondly while considering similarity we would ideally want to use the beacons that are more accurate, which can be loosely translated to say that we want to use the beacons that are closer to us to localize than the ones that are far. The way we achieve this is my using the top-p beacons which have higher RSSI values. This makes sure that erroneous beacons or unavailable beacons donot hamper our localization accuracy. Finally once we have ranked all the RPs in order of similarity, we pick the top k RPs by similarity and find a weighted average of the coordinates of the RPs weighted by the similarity metric. The values p and k are determined experimentally.

SVM Regressor : In the past few years machine learning research has progressed a lot and hence it is imperative that we try to leverage the power of machine learning in our problem statement. However in this problem large datasets are pretty hard to find, as it's an extremely time consuming process to be able to collect RSSI fingerprints in a large area. Thus complex approaches like Deep Neural Networks, often tend to overfit the data and don't generalize well enough. In this model we have used a SVM to predict the location of the user given a online fingerprint. The model was trained by splitting the fingerprints into training, evaluation and testing data and fitting the training data using an SVM. The hyperparameters are tuned experimentally. [34]

In this model we don't combine SVM and the W-KNN approach but for future work we can aim to build a hybrid approach that leverages the strength of both approaches by adding a probabilistic filter on top of this. We take the previous BLE estimated values and take a moving average to get the current estimate. This is to ensure that we don't have a sudden erroneous values affecting our particle filter weights.

Evaluation is one in two steps, we first check the room detection accuracy where we take the predicted location by the BLE module and check if it lies in the same room as the actual user. This is to establish the coarse localization capabilities of the BLE module. Down below we have shown the map we have in use for testing purposes.

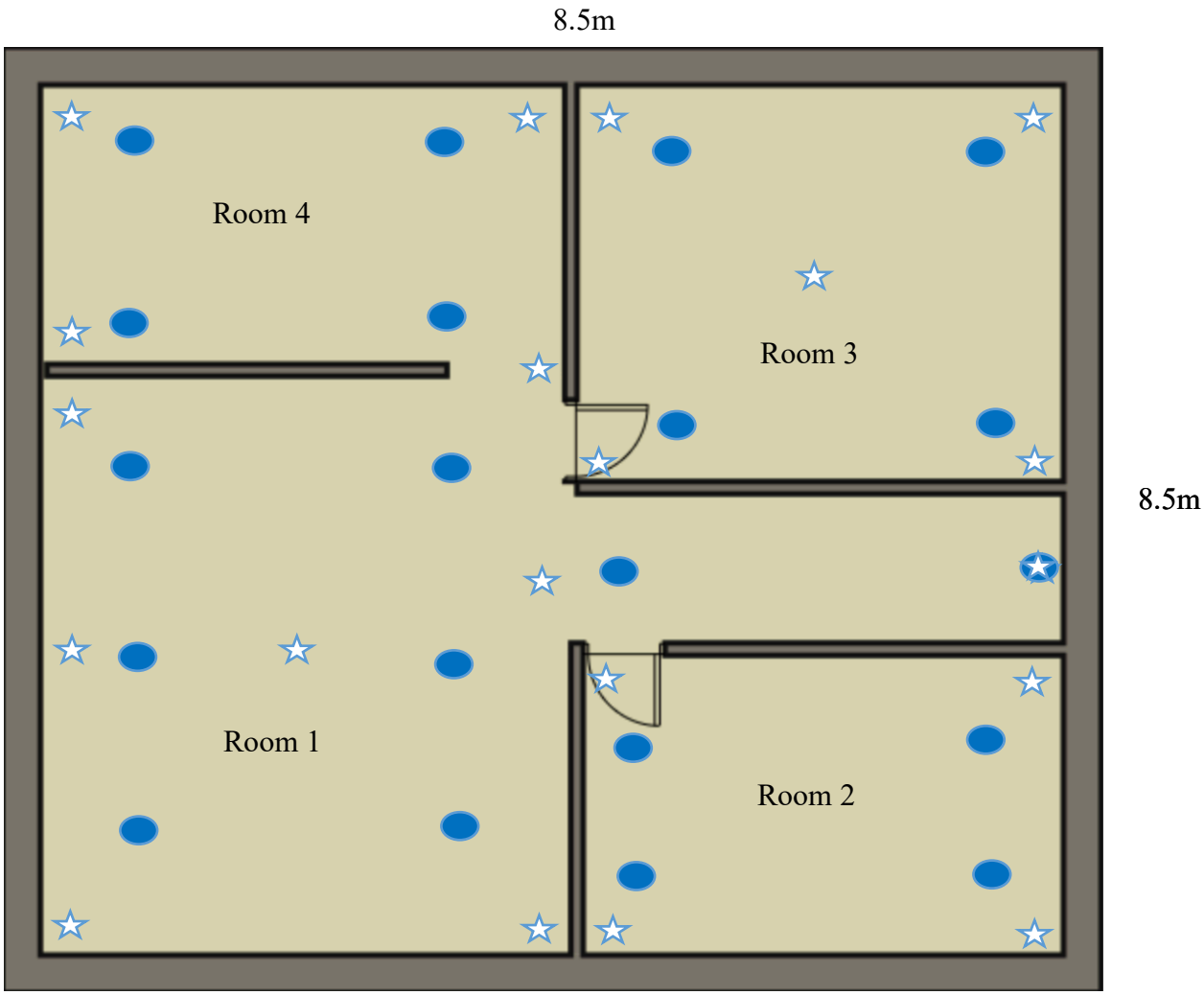


Fig.14 Beacon Placement Map

The ☆ shows the position of the Estimote Beacons with respect to our map. For simplicity sake we don't show the positions of furniture and obstacles in this illustration. ● Denotes the points where fingerprint data is collected. We chose 8 random points in room 1, 4 each in room 2,3,4 and walked around these points to test online localization.

Room Detection Results

Method	Accuracy
Simple Multi-lateration	58.5 %
W-KNN (P = 5, K = 4)	90.1 %
W-KNN (All Beacons, K = 4)	81.4 %
SVM Regressor	83.8 %

Table 1. Room Detection Results

We also observe the benefits of using only the 5 beacons with highest RSSI to compute the similarity metric, this is because the beacons that are further have a larger chance of being erroneous and noisy, both during online and offline phase of measurement.

Euclidean Distance Accuracy Results

While our methods do provide reasonable accuracy for usage in a mall, expo setting. We proceed to tap the measurements from IMU in the smartphone to further better the accuracy of the model.

Method	Error (50^{th} Percentile)
Simple Multi-lateration	4.7m
W-KNN (P = 5, K = 4)	2.8m
W-KNN (All Beacons, K = 4)	3.5m
SVM Regressor	3.3m

Table 2. Euclidean Distance Accuracy Results

3.3 Dead Reckoning

Recently, with the development of artificial intelligence technologies and the popularity of mobile devices, walking detection and step counting have gained much attention since they play an important role in the fields of equipment positioning, saving energy, behaviour recognition, etc. In this module we aim to use the sensor data from smartphones to predict the relative position of the smartphone with respect to the starting point. While using it for long-term localization would lead to inaccuracies due to drift, we can leverage accuracy for short time windows.

The following sensors are used commonly for the purpose of step vector computation.

Accelerometer : Measures acceleration which is the rate of change of velocity of a body in its own instantaneous frame of rest. This means that we can determine the orientation relative to the direction of gravity in relation to the frame of the earth, at least when moving slowly. The main weakness of the accelerometer is that it can be very noisy.

Gyroscope : Delivers the angular velocity and is the most dynamic sensor of the three because it provides real time information about rotation changes immediately.

Orientation Sensor : The sensor provides the 3 axis orientation of the smartphone in Earth frame of reference. It's a software sensor. The orientation sensor derives its data by processing the raw sensor data from the accelerometer and the geomagnetic field sensor. Because of the heavy processing that is involved, the accuracy and precision of the orientation sensor is diminished. To overcome this we use the getRotationMatrix() method in conjunction with the getOrientation() method to compute orientation values.

In literature people have tried to broadly use two types of methods for the purpose of Dead Reckoning based localization, it's integration and step counting. Because of the erroneous accelerometer readings and errors in the Rotation Matrix, it is usually hard to get a good accuracy by double integration methods without using highly complex algorithms. We would hence rely on step counting methods and fuse that data with the orientation to get the step vector.

If we look at the accelerometer, gyroscope data we observe that there is a clear distinction in the area where the person is walking and the area where the person isn't. Let's observe this data from this hand-held smartphone.

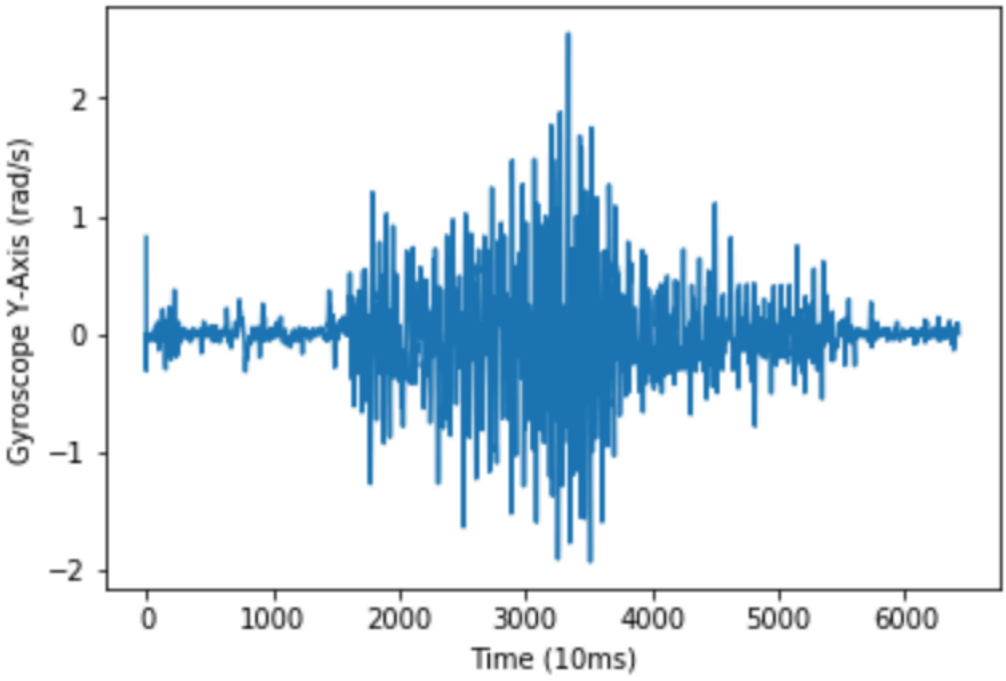


Fig 15. Gyroscope Y axis from smartphone

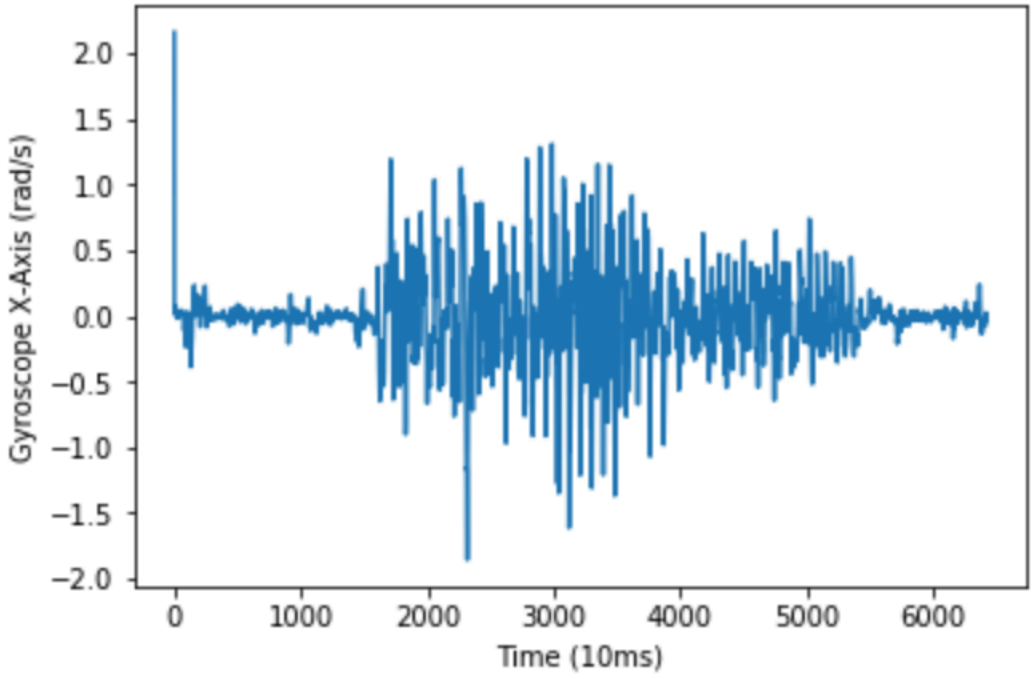


Fig 16. Gyroscope X axis from smartphone

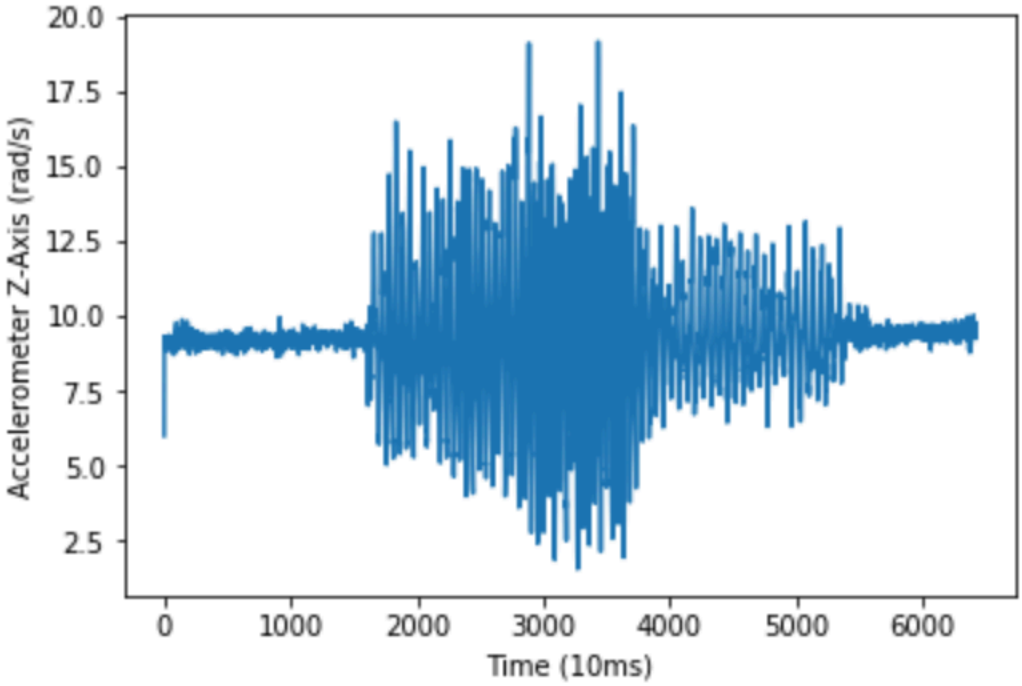


Fig 17. Accelerometer Z axis from smartphone

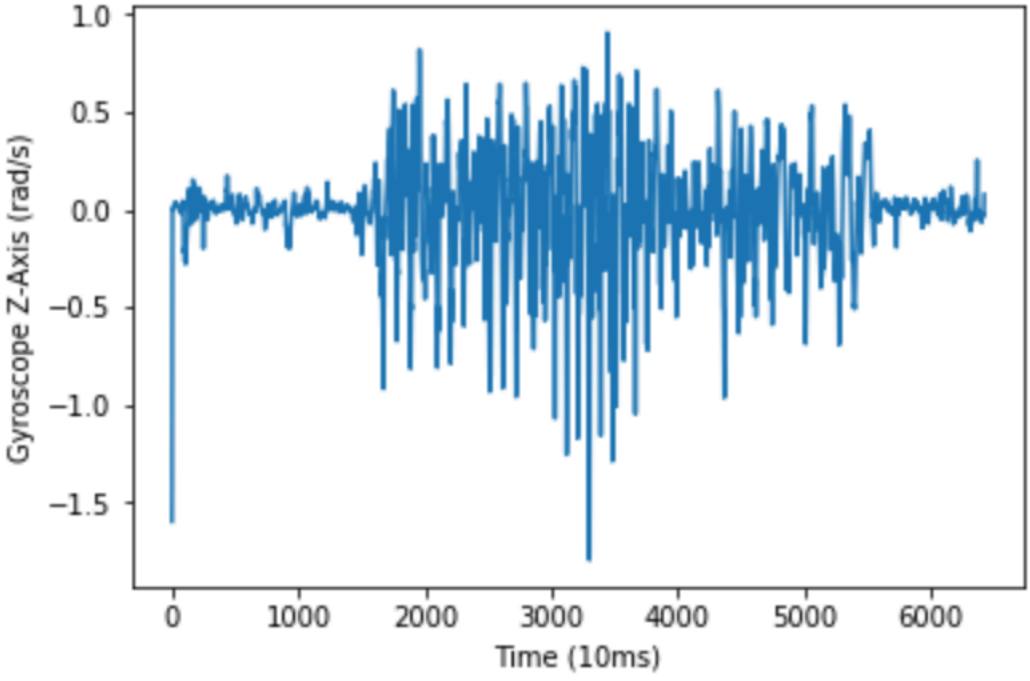


Fig 18. Gyroscope Z axis from smartphone

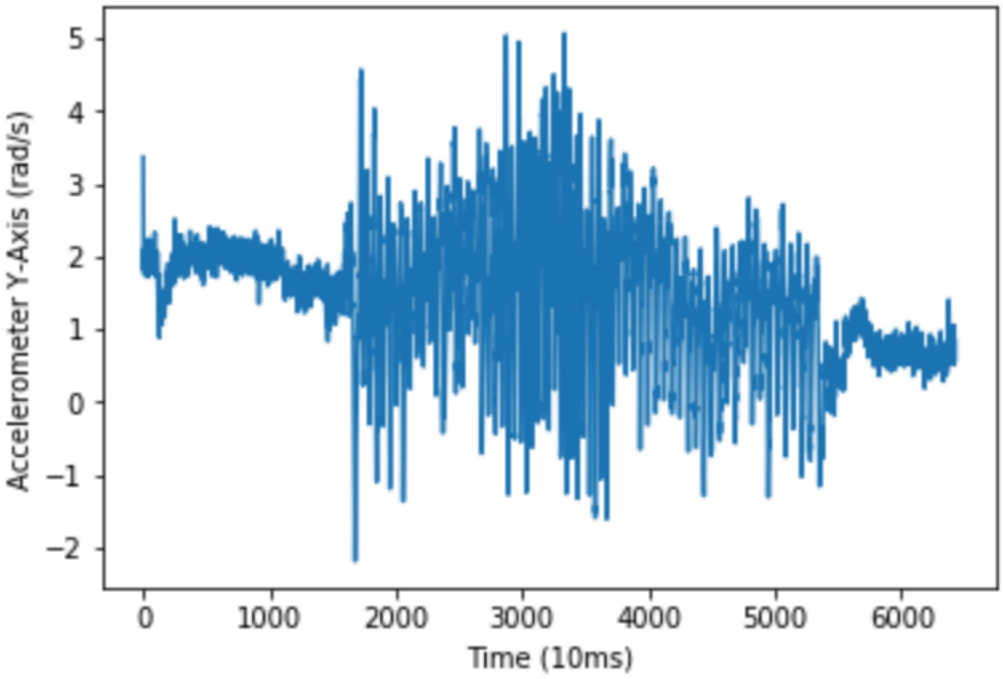


Fig 19. Accelerometer Y axis from smartphone

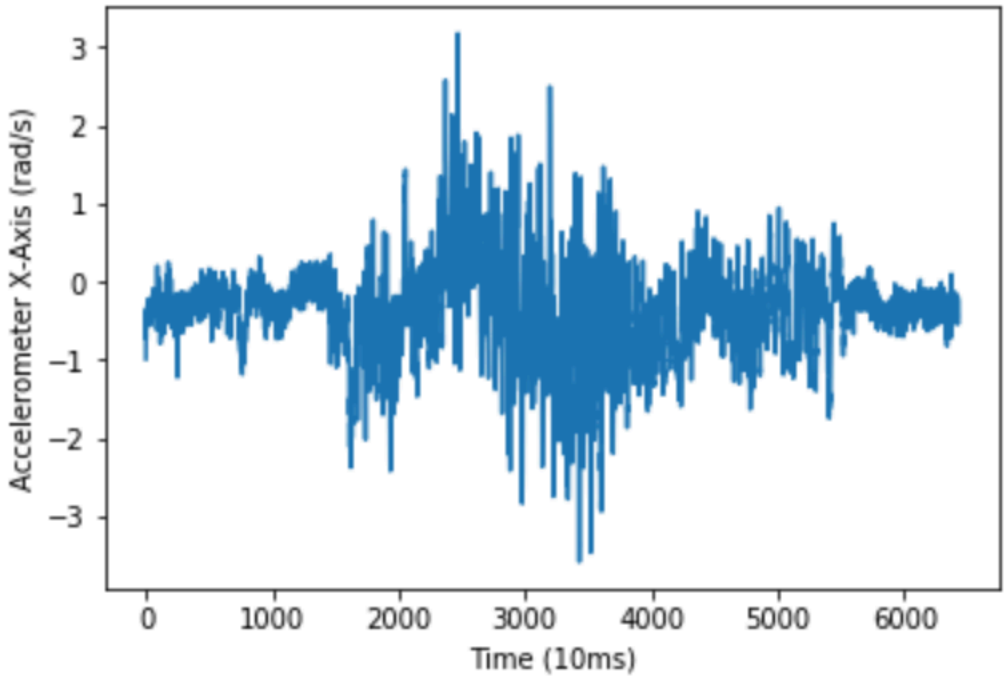


Fig 20. Accelerometer X axis from smartphone

Now that we have established that we definitely have a clear signal we should be able to extract from the IMU data to detect steps the person is taking, the actual method to use is still under question. Broadly the following methods are used in the past,

Thresholding: These The thresholding approach counts steps by judging whether sensory data satisfy some predefined thresholds that differ according to various devices held by different users at different positions. It is very hard to find a single threshold that'd work well across different smartphone placements. [35,36]

Peak-Detection : Peak detection works based on the principle that the peaks in the IMU data refer to steps taken, while this correlation is strong and it doesn't rely on predefined thresholds. [37,38,39,40]. It does suffer from the issue of interference from noise, and false peaks. Usually this problem is overcome using a low-pass filter and heuristics. Zero crossing approaches suffer from similar difficulties [44]

Autocorrelation Based : This approaches aims to use the periodicity in the graph to detect the steps, usually we only require one threshold for this approach namely above what autocorrelation value are we willing to accept it as a step. While we do get good accuracy using this approach there's a very high computational cost, hence we can't use it for real-time systems. [43]

Frequency Domain Approaches : Again we attempt to use the periodicity in the value of the walking frequency. When a person is walking, and we take a windowed fourier transform of a time period where there's movement we'd get a peak at the walking frequency. This approach usually gives good accuracy when used on gyroscope and in large parts doesn't depend on the orientation of the smartphone (handheld, pocket etc). However this approach does suffer from resolution defects and computational overheads. [45,46,47,48,49]

To sum up, even though great efforts have been invested, it is still challenging to detect and count steps with unconstrained smartphones in an accurate and efficient manner. We attempt to break the process of detecting steps into 2, and use a neural network based approach to build a system that can localize a smartphone irrespective of its placement and orientation.

Data collection is done by using our android client app, and walking around with different smartphone placements (Handheld, Swinging hand, Front Pocket, Trouser Pocket). We then take pieces of this data in a time window manner of 1 sec each. We then label this data as whether the user is walking or if the user is not walking. A neural network is trained using the 3 axis Accelerometer and Gyroscope data to predict if the user is walking. If the user is walking then we employ another neural network designed to predict the number of steps (1/2) taken by the user in the given interval.

The reason we use this approach is because it avoid the handcrafting of heuristics and thresholds and lets the model learn to adapt to all kinds of smartphone placements. Also there is no case of drift in this method, because the data is used only in 1 sec time windows in an independent manner so the errors don't carry forward. Once we have the number of steps taken by the user, we sample a gaussian with the average step length of a person as the mean. We can also take the users approximate height as an input and use average step length as 0.41* height of the person. This lets us compute the total displacement of the person. We now use the orientation data and multiply it with the magnitude displacement to get the final value of step vector.

Dead Reckoning Results

We just use our smartphone and walk around our map, and compute the error as follows.

$$Error = Abs(Actual\ Steps - Estimated\ Steps)/Actual\ Steps$$

Method	Error
Butterworth Filter + Peak Detection	0.76
FFT on Gyroscope	0.89
FFT on Accelerometer	0.85
Deep Learning Walk Detection + Steps	0.91

Table 3. Step Detection Results

3.4 Particle Filter

The particle filtering method refers to the process of obtaining the state minimum variance distribution by finding a set of random samples propagating in the state space to approximate the probability density function and replacing the integral operation with the sample mean. Although the probability distribution in the algorithm is only an approximation of the real distribution, due to the non-parametric characteristics, it can get rid of the constraint that the random quantity must satisfy the Gaussian distribution when solving the nonlinear filtering problem, and can express a wider distribution than the Gaussian model.

In our model the particle filter begins by randomly initializing particles throughout the map uniformly, and the particle filter as discussed before performs 3 main functions, displace particles, reweight and resample. We also discussed how a lock is placed on the CurrentParticleState vector in order to avoid race conditions.

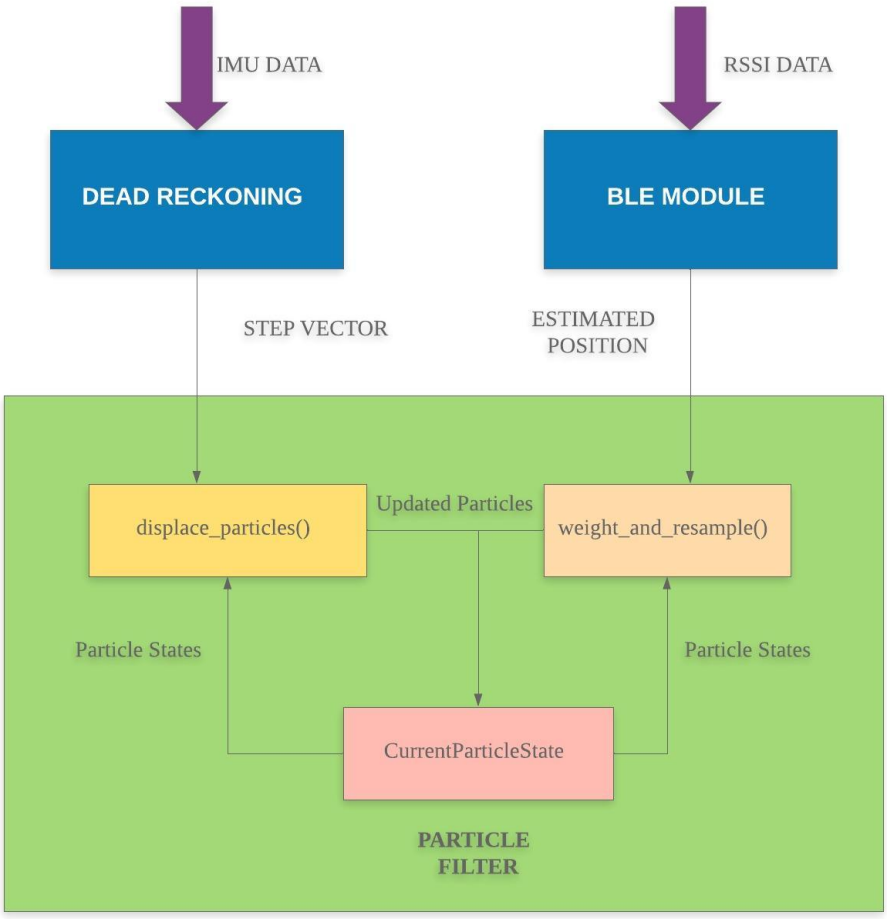


Fig.21 Revisit Particle Filter

In this section we will discuss our improvements to the particle filter, and how it will help us achieve better accuracies. The main areas of discussion would be in regards to how we would weight the particles when the BLE module comes up with a position estimate, how we would resample the particles and how do we use the fact that we already know the underlying map.

Weighting Particles : Whenever the BLE Module gives us a position estimate, we assign a weight to each particle as an inverse of its Euclidean distance to the given position estimate. It means that closer the particle to the BLE estimate the higher weight it will get. This means that eventually the particles will start to converge around the area of the BLE estimates.

$$Weight(i) = (Euclidean\_Distance(P_i , Pos\_Estimate))^{-1}$$

Map Based Penalty : The underlying map is known to us, and it is imperative that we put the information to use while improving the accuracy [50,51]. Our model achieves it by adding a penalty to the particle weight for any particle that moves into a wall or into an occupied zone. For a particle close to the wall, we might end up displacing it through the wall due to an erroneous step vector computation. We should hence not discard the particle but merely keep its original position and discard the step vector while also reducing the weight of the particle. This makes sure we take both the step vector and the map into account while weighing the particles. The way it is penalised is that, let d be the distance between the particle's supposed location and the closest free cell. In case of wall we compute d as the distance between it's supposed position and the closest free-cell beside the wall. The penalty is multiplied with the original particle weight

$$Penalty = e^{-p*d}$$

Post this we normalize the weights of all particles and resample.

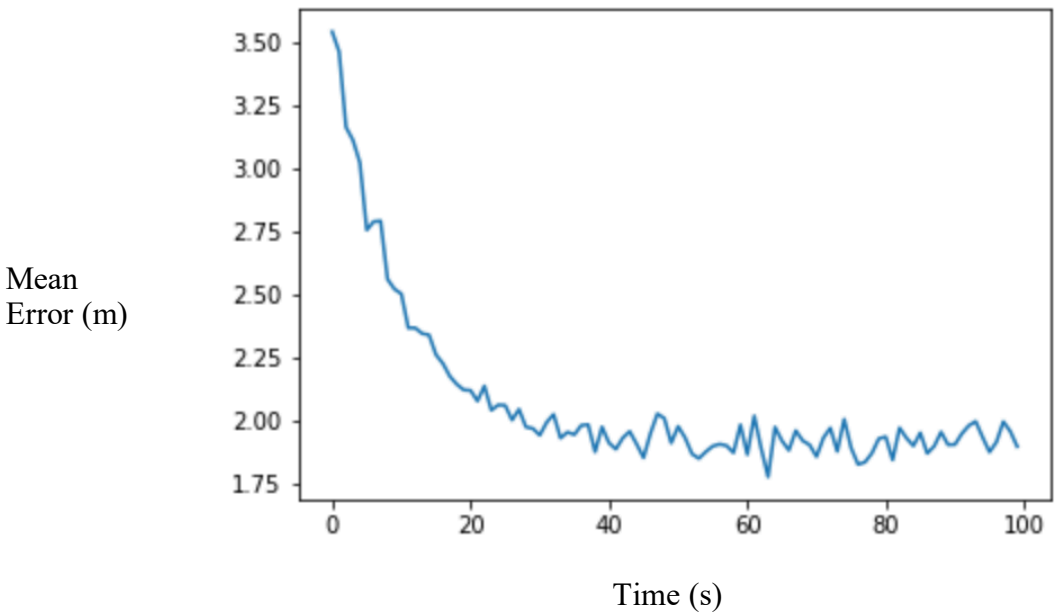
Resampling Particles : The resampling works by choosing an existing particle based on the probability distribution defined by the weight of the particles. We then pick a point around this point such that it lies in a "free" cell and the distance from the original particle will be a random gaussian with a mean of 0.5m. We also find the closest 4 beacons for the particle and sample 5% of total particles in the quadrilateral formed by these beacons, this is a small heuristic designed to avoid degeneracy of the particles. We notice that the actual user lies inside the quadrilateral formed by the closest beacons ~91% of the time.

Results

We now compute the mean-localization error using the particle filter approach, we can observe that this approach yields better results compared to using only the BLE module. Another caveat is that as the user moves around the map, our map based penalty makes our particles converge much faster, because if a user takes a right turn and moves ahead based on the DR module and there's no turn in that place then those particles will lose weight even if not validated by the BLE Module. Moreover every 1 sec the BLE module releases its position estimate which is used to make the particles converge towards true location. We have run the system with N = 10000 particles as it was experimentally proved to be most optimal.

Method	**Error (50th Percentile)**
Particle Filter (BLE + IMU) Converged	1.9m
W-KNN (P = 5, K = 4)	2.8m
W-KNN (All Beacons, K = 4)	3.5m
SVM Regressor	3.3m

Table 4. Final Results



The graph is taken from the time we got the 5th BLE measurement in the particle filter while the user was still. Post that as the user moves from room to room walking, we find how the map penalty helps the particle converge, producing an accuracy that couldn't be achieved by BLE Module alone.

SUMMARY AND CONCLUSION

To summarize we built a low-cost energy efficient scalable BLE and IMU based localization module which was able to produce usable accuracy of 1.9m in our test environment. This accuracy is stable enough to be used in a museum, mall or expo setting. We handled the challenges of a fluctuating RSSI values, lack of LoS signals, extracting step counting data irrespective of smartphone placement while handling erroneous sensors and finally leveraging map occupancy data to improve the accuracy of the overall model to a usable level. We also implemented the system in a Server-Client model thus offloading all the heavy computational load outside of the user device, thus we can provide smooth performance even in low end smartphones as long as they have good internet connectivity. Parallelizing every aspect of the particle filter also helps us operate with a large number of particles (N = 10000).

There is however a lot of scope for future research and development in this field. We could look at using Recurrent Neural Networks for step detection, as RNNs in recent years have shown lot of promise in the field of sequence understanding. We also couldn't run a test in a large environment for our project and conducting one could open our eyes towards errors that were possibly missed in our current limited test setting. Another approach worth looking at is trying to approximate the BLE radio map, so that we can estimate the probability of receiving a particular signal value at a given location. Finally, we would also suggest looking at better ways to resample in the particle filter, this has also been an area of active research. We can also extend the approach to WiFi and other wireless signals.

REFERENCES

1. Bluetooth SIG, Inc. Bluetooth Market Update. Available online: https://www.bluetooth.com/wp-content/uploads/2020/03/2020_Market_Update-EN.pdf
2. Use of Bluetooth Beacon Technology in Smart Airport. Available online: https://www.amarinfotech.com/use-of-bluetooth-beacon-technology-in-smart-airport.html
3. Wang, C.S. An AR mobile navigation system integrating indoor positioning and content recommendation services. *World Wide Web* 2018. [Google Scholar] [CrossRef]
4. The Hitchhikers Guide to iBeacon Hardware: A Comprehensive Report by Aislelabs. Available online: https://www.aislelabs.com/reports/beacon-guide/ (accessed on 21 December 2018).
5. Faragher, R.; Harle, R. Location fingerprinting with bluetooth low energy beacons. *IEEE J. Sel. Areas Commun.* 2015, *33*, 2418–2428. [Google Scholar] [CrossRef]
6. Davidson, P.; Piché, R. A Survey of Selected Indoor Positioning Methods for Smartphones. *IEEE Commun. Surv. Tutor.* 2017, *19*, 1347–1370. [Google Scholar] [CrossRef]
7. S. Holm, "Hybrid ultrasound-RFID indoor positioning: Combining the best of both worlds," in 2009 IEEE International Conference on RFID, pp. 155–162, IEEE, 2009.
8. F. Zafari, I. Papapanagiotou, and K. Christidis, "Microlocation for Internet-of-Things-Equipped Smart Buildings," IEEE Internet of Things Journal, vol. 3, no. 1, pp. 96–112, 2016.
9. D. Vasisht, S. Kumar, and D. Katabi, "Decimeter-level localization with a single Wifi access point," in 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16), pp. 165–178, 2016.
10. S. Kumar, S. Gil, D. Katabi, and D. Rus, "Accurate indoor localization with zero start-up cost," in Proceedings of the 20th annual international conference on Mobile computing and networking, pp. 483–494, ACM, 2014.
11. J. Xiong and K. Jamieson, "ArrayTrack: a fine-grained indoor location system," in Presented as part of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13), pp. 71–84, 2013.

12. M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, “Spotfi: Decimeter level localization using WiFi,” in ACM SIGCOMM Computer Communication Review, vol. 45, pp. 269–282, ACM, 2015.

13. Z. Yang, Z. Zhou, and Y. Liu, “From RSSI to CSI: Indoor localization via channel response,” ACM Computing Surveys (CSUR), vol. 46, no. 2, p. 25, 2013.

14. P. Castro, P. Chiu, T. Kremenek, and R. Muntz, “A probabilistic room location service for wireless networked environments,” in International Conference on Ubiquitous Computing, pp. 18–34, Springer, 2001.

15. A. Haeberlen, E. Flannery, A. M. Ladd, A. Rudys, D. S. Wallach, and L. E. Kavraki, “Practical robust localization over large-scale 802.11 wireless networks,” in Proceedings of the 10th annual international conference on Mobile computing and networking, pp. 70–84, ACM, 2004.

16. A. Povalac and J. ˇ Sebesta, “Phase of arrival ranging method for UHF ˇ RFID tags using instantaneous frequency measurement,” in ICECom, 2010 Conference Proceedings, pp. 1–4, IEEE, 2010.

17. M. Scherhaufl, M. Pichler, D. Muller, A. Ziroff, and A. Stelzer, “Phase of-arrival-based localization of passive UHF RFID tags,” in Microwave Symposium Digest (IMS), 2013 IEEE MTT-S International, pp. 1–3, IEEE, 2013.

18. S.-Y. Jung, S. Hann, and C.-S. Park, “TDOA-based optical wireless indoor localization using LED ceiling lamps,” IEEE Transactions on Consumer Electronics, vol. 57, no. 4, pp. 1592–1597, 2011.

19. J. Xu, M. Ma, and C. L. Law, “Position estimation using UWB TDOA measurements,” in 2006 IEEE International Conference on UltraWideband, pp. 605–610, IEEE, 2006.

20. S. Kumar, S. Gil, D. Katabi, and D. Rus, “Accurate indoor localization with zero start-up cost,” in Proceedings of the 20th annual international conference on Mobile computing and networking, pp. 483–494, ACM, 2014.

21. J. Xiong and K. Jamieson, “ArrayTrack: a fine-grained indoor location system,” in Presented as part of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13), pp. 71–84, 2013.

22. W. Dargie and C. Poellabauer, Fundamentals of wireless sensor networks: theory and practice. John Wiley & Sons, 2010.

23. https://developer.nvidia.com/pycuda

24. *S. Godsill, "*Particle Filtering: the First 25 Years and beyond*,"* ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, United Kingdom, 2019, pp. 7760-7764, doi: 10.1109/ICASSP.2019.8683411.*

25. Nowak, Thorsten & Hartmann, Markus & Zech, Tobias & Thielecke, Jorn. (2016). A path loss and fading model for RSSI-based localization in forested areas. 110-113. 10.1109/APWC.2016.7738133.

26. Rudolph Emil Kalman. 1960. A New Approach to Linear Filtering and Prediction Problems. Transactions of the ASME–Journal of Basic Engineering 82, Series D (1960), 35–45

27. Röbesaat, Jenny et al. "An Improved BLE Indoor Localization with Kalman-Based Fusion: An Experimental Study." *Sensors (Basel, Switzerland)* vol. 17,5 951. 26 Apr. 2017, doi:10.3390/s17050951

28. Alam, S.A., Gustafsson, O. Improved Particle Filter Resampling Architectures. *J Sign Process Syst* **92,** 555–568 (2020). https://doi.org/10.1007/s11265-019-01489-y

29. Comparison of Expectation-Maximization based parameter estimation using Particle Filter, Unscented and Extended Kalman Filtering technique S.B.Chitralekha[*]J.Prakash[**]H.Raghavan[*]R.B.Gopaluni[***]S.L.Shah[*]

30. Method for Improving Indoor Positioning Accuracy Using Extended Kalman FilterSeoung-Hyeon Lee,[1] Il-Kwan Lim,[2] and Jae-Kwang Lee. Emerging Trends in Mobile Collaborative Systems 2016

31. P. Kriz, F. Maly, and T. Kozel, "Improving indoor localization using Bluetooth low energy beacons," *Mobile Information Systems*, vol. 2016, Article ID 2083094, 11 pages, 2016.

32. Z. Jianyong, L. Haiyong, C. Zili, and L. Zhaohui, "RSSI based Bluetooth low energy indoor positioning," in *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 526–533, Busan, South Korea, October 2014.

33. F. Subhan, H. Hasbullah, A. Rozyyev, and S. Bakhsh, "Indoor positioning in Bluetooth networks using fingerprinting and lateration approach," in *2011*

International Conference on Information Science and Applications, pp. 1–9, Jeju Island, South Korea, April 2011.

34. Evgeniou, Theodoros & Pontil, Massimiliano. (2001). Support Vector Machines: Theory and Applications. 2049. 249-257. 10.1007/3-540-44673-7_12.

35. Alzantot M., Youssef M. UPTIME: Ubiquitous pedestrian tracking using mobile phones; Proceedings of the Wireless Communications and Networking Conference; Shanghai, China. 1–4 April 2012; pp. 3204–3209. [Google Scholar]

36. Hu W.Y., Lu J.L., Jiang S., Shu W. WiBEST: A hybrid personal indoor positioning system; Proceedings of the IEEE Wireless Communications and Networking Conference; Shanghai, China. 7–10 April 2013; pp. 2149–2154. [Google Scholar]

37. Ying H., Silex C., Schnitzer A., Leonhardt S., Schiek M. Proceedings of the 4th International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2007), Aachen Germany, 26–28 March 2007. Volume 13. Springer; Berlin/Heidelberg, Germany: 2007. Automatic Step Detection in the Accelerometer Signal; pp. 80–85. [Google Scholar]

38. Chen G.L., Fei L.I., Zhang Y.Z. Pedometer method based on adaptive peak detection algorithm. J. Chin. Inert. Technol. 2015;23:315–321. [Google Scholar]

39. Lan K.C., Shih W.Y. Using smart-phones and floor plans for indoor location tracking. IEEE Trans. Hum. Mach. Syst. 2014;44:211–221. [Google Scholar]

40. Yang X., Huang B. An accurate step detection algorithm using unconstrained smartphones; Proceedings of the 27th Chinese Control and Decision Conference; Qingdao, China. 23–25 May 2015; pp. 5682–5687. [Google Scholar]

41. Kappi J., Syrjarinne J., Saarinen J. MEMS-IMU based pedestrian navigator for handheld devices; Proceedings of the 14th International Technical Meeting of the Satellite Division of the Institute of Navigation ION GPS; Salt Lake City, UT, USA. 11–14 September 2001. [Google Scholar]

42. Ailisto H.J., Makela S.M. Proceedings of SPIE—The International Society for Optical Engineering, Orlando, FL, USA. Volume 5779. SPIE; San Jose, CA, USA: 2005. Identifying people from gait pattern with accelerometers; pp. 7–14. [Google Scholar]

43. Rai A., Chintalapudi K.K., Padmanabhan V.N., Sen R. Zee: Zero-effort crowdsourcing for indoor localization; Proceedings of the 18th Annual International Conference on

Mobile Computing and Networking; Istanbul, Turkey. 22–26 August 2012; pp. 293–304. [Google Scholar]

44. Jayalath S., Abhayasinghe N. A gyroscopic data based pedometer algorithm; Proceedings of the International Conference on Computer Science & Education; Colombo, Sri Lanka. 26–28 April 2013; pp. 551–555. [Google Scholar]

45. Barralon P., Vuillerme N., Noury N. Walk detection with a kinematic sensor: Frequency and wavelet comparison; Proceedings of the IEEE International Conference of Engineering in Medicine and Biology Society; New York, NY, USA. 30 August–3 September 2006; pp. 1711–1714. [PubMed] [Google Scholar]

46. Dirican A.C., Aksoy S. Step Counting Using Smartphone Accelerometer and Fast Fourier Trransform. Sigma. 2017;8:175–182. [Google Scholar]

47. Stephane . Wavelet Tour of Signal Processing. Academic Press; 1999. pp. 83–85. [Google Scholar]

48. Nyan M.N., Tay F.E., Seah K.H., Sitoh Y.Y. Classification of gait patterns in the time-frequency domain. J. Biomech. 2006;39:2647–2656. doi: 10.1016/j.jbiomech.2005.08.014. [PubMed] [CrossRef] [Google Scholar]

49. Wang J.H., Ding J.J., Chen Y., Chen H.H. Real time accelerometer-based gait recognition using adaptive windowed wavelet transforms; Proceedings of the 2012 IEEE Asia Pacific Conference on Circuits and Systems; Kaohsiung, Taiwan. 2–5 December 2012; pp. 591–594. [Google Scholar]

50. D. Bernstein and A. Kornhauser, “An introduction to map matching for personal navigation assistants,” 1998.

51. C. White, D. Bernstein, and A. Kornhauser, “Some map matching algorithms for personal navigation assistants,” Transportation Research Part C: Emerging Technologies, vol. 8, no. 1-6, pp. 91–108, 2000.