InLoc: An End-to-End Robust Indoor Localization and Routing Solution using Mobile Phones and BLE Beacons

Vivek Chandel
Innovation Labs
Tata Consultancy Services Limited
Kolkata, India (700156)
Email: vivek.chandel@tcs.com

Nasimuddin Ahmed
Innovation Labs
Tata Consultancy Services Limited
Kolkata, India (700156)
Email: nasim.ahmed@tcs.com

Shalini Arora
Mobility Center of Excellence, Hi-Tech
Tata Consultancy Services Limited
Noida, India (201309)
Email: shalini.arora@tcs.com

Avik Ghose
Innovation Labs
Tata Consultancy Services Limited
Kolkata, India (700156)
Email: avik.ghose@tcs.com

***Abstract*—This paper discusses 'InLoc', an accurate and a robust positioning and tracking system using commercial mobile devices, with an integrated feature of route finding for the user from a source to the desired destination. The system exploits easily available building floor maps in raster form, with an easy conversion to vector model, eliminating the need of specially designed vector maps, thereby making the system scalable for large indoor maps and more implementation friendly. This also enables the vector map to be used in both Particle Filter based IMU tracking, and routing. Additionally, an efficient method for independent fusion of location information from phone IMU sensors and Bluetooth Low Energy (BLE) beacons is demonstrated. The method caters both dynamic and static properties of the system state. Furthermore, the paper proposes a novel approach for estimating the distance from BLE beacons using RSSI (Received Signal Strength Indication) measurement. InLoc can be readily used for any size of building floors for applications like tracking, routing and guiding system, emergency evacuation, meeting planners etc. requiring no separate effort to rebuild the vector map from scratch. The system yields a mean tracking error of less than 0.4m in location, and yields 0.9m as an average positioning error using fusion.**

I. Introduction

Indoor Positioning System (IPS) is one of the most actively researched areas in the domain of mobile sensing with the advent of integrated small and power efficient mobile sensors. This may also be backed by the fact that an accurate positioning system is almost always required in large indoor areas like conference venues, complex hospital buildings, big indoor shopping areas, where people tend to lose their path, and a step-by-step tracking and routing system becomes a requisite. Global Positioning System (GPS) has solved this problem for the outdoor domain by providing a positioning accuracy of less than $3.5m$ [1]. An application which provides a location alone is not much useful to a user if one is not able to course a path from his/her present location to the desired destination. A number of gadgets are available in the market [2] that leverage GPS technology for routing, creating a complete guidance system for outdoors. InLoc is such an application, which provides positioning, tracking, and routing to the user, but for indoor premises where GPS loses its accuracy badly.

978-1-5090-2425-4/16/$31.00 ©2016 IEEE

Among various technologies, WiFi is widely accepted due to its ubiquitous nature and adaptivity to the existing infrastructure. However, WiFi infrastructure is meant for network communication. Thus, placement and density of access points are not suitably optimized for localization. Moreover, WiFi is a power hungry protocol. As an alternative technology, Bluetooth Low Energy (BLE) is becoming more and more common in location aware applications with the introduction of BLE beacons, which provide attractive form factor and long-lasting battery life [3]. They are being extensively researched in context-aware applications, as those for shopping malls for feeding the customers with relevant promotions from a nearby store. Existing location-aware systems relying on BLE can be broadly categorized as: *trilateration with propagation based* and *fingerprinting based*. Trilateration is a method to estimate the location based on the distances measured from three or more reference points at known locations. The key technique used in this approach is converting the BLE Received Signal Strength Indication (RSSI) into distance by using the free-space propagation model. We leverage our proposed system with BLE technology in order to derive an initial BLE location fix of the user in the building much like a GPS fix, and then periodically assisting the IMU tracking system based on the well-researched Particle Filter, whenever needed.

The *novel contributions of this work* lie in our vector representation and modeling of the indoor raster floor map, which enables us to incorporate a particle filter based tracking, along with routing in a single application. For achieving

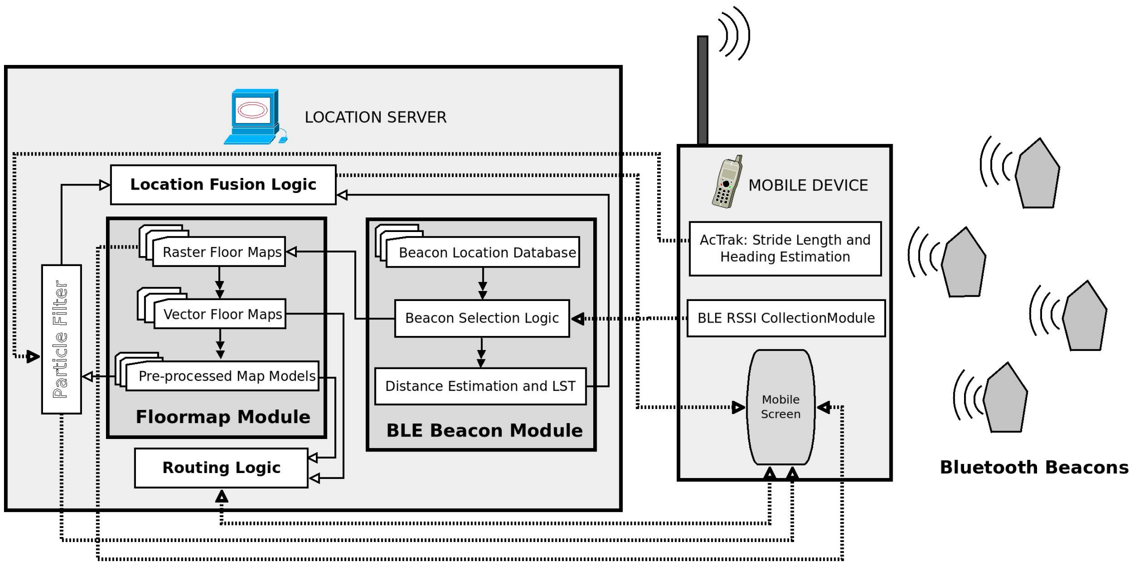


Fig. 1. InLoc Architecture

particle filter based tracking using IMU, a floor map is required to be in a representation which can clearly classify walkable and non-walkable areas for the particle filter. But routing requires the floor map to be represented in the form of a graph with defined nodes and arcs. InLoc is the first application to the best of our knowledge incorporating both tracking and routing. Another contribution made by this work is a novel method to estimate distance from BLE beacons using RSSI measure, and localizing using those distances. Instead of a well-researched RF fingerprinting approach, we map the RSSI features to one of the distance bins, followed by application of Least Squares Trilateration (LST) to estimate the location [4]. Also this work discusses a completely novel and effective yet simple technique to perform an independent fusion of locations from particle filter and BLE beacons catering both static and dynamic states of the system, improving the overall accuracy.

II. Related Works

As already discussed, localization using mobile phone sensors has been researched actively and it continues to be so with the advent of technologically more advanced sensors. Phone embedded IMUs have been actively researched for localization using pedestrain dead reckoning [5][6][7][8]. The cited works use a small area using a custom built map in order to test the respective systems, opening up a practical challenge of implementing the systems on very large floor maps, where manual creating of vector maps might be tedious and erroneous. Particle filter is a famous choice for tracking a user indoors by exploiting the floormap geometry [5][9][10][11].

In order to implement graph based algorithms like shortest path finding etc., efforts have been made to vectorize the floor map information. Link et. al. [12] has tried to do a map-matching (also Shih et. al. [13]) for tracking using the data received from the IMU sensors and an additional altimeter. This incremental learning of floor map may not be required if already existing blueprint maps of the buildings can be leveraged, which is the goal of InLoc. Link et. al. [12] uses OpenStreetMap for manually creating the vector maps, requiring efforts to input the map data to the OpenStreetMap system, which is impractical in case of fast deployment of the system on multi-floor, large, and dense buildings. Jensen et. al. [14] uses a separate hand-made connectivity and accessibility graphs for object tracking, which again is impractical for large areas. Stroila et. al. [15] performs route visualization by creating an open area map requiring user to manually mark walkable and non-walkable areas. Han et. al. [16] uses ArcGIS for representing the vector floormap, and performing routing.

It should be noted that none of the works above, requiring substantial user intervention, detail out the architecture where both routing and a particle filter based tracking can be performed on already available indoor raster floor maps, which InLoc tries to achieve. Also Subbu et. al. [17] states the need for a practical floor map system which can provide indoor tracking alongwith efficient routing.

As discussed in last section, BLE location-aware applications can utilize trilateration with propagation or fingerprinting. An efficient approach exploiting the propagation model is proposed by Zhao et. al. [18], demonstrating empirical models for BLE in different conditions such as line-of-sight(LOS), Non-line-of-sight(NLOS) or outdoor and indoor. Another approach by Ashwin et. al. [19] provides an attractive method to infer the position of mobile robot using bluetooth, and uses trilateration.

Location fingerprinting is a very familiar technique as it is robust and moderately accurate in complex real-life deployments. Traditional fingerprinting-based approach relies on constructing a radio map of RSSI measurements attributed to reference locations in training phase. Later in the positioning phase, the user location is estimated by comparing signal strength measurements with the training model. An important work in fingerprinting is by Faragher et. al. [20] which provides a detailed study in BLE based fingerprinting, establishing a quantitative comparison with WiFi fingerprinting. Another interesting fingerprinting application by Chen et. al. [21] proposes a Bayesian fusion method which combines the statistical features of RSSI and the prior information from a motion model. In its training phase, this approach generates the radio map using the RSSI, and its statistical mean and variance. Later in its positioning phase, Bayesian estimation method combined with motion model is utilized to facilitate the user location. The work presented by Zhang et. al. [22] evaluates three fingerprinting algorithms: k-nearest neighbors (k-NN), neural networks, and support vector machine (SVM). The result demonstrates that k-NN is the best candidate for localization in real-life applications. Although traditional fingerprinting method posseses moderate accuracy but it is quite sensitive to the environment. Gradually, it becomes ineffective due to the multipath effect and dynamic changes in environment.

III. System Overview

We present a modular architecture of InLoc in Figure 1 with all the functional connections.

A typical workflow of the system starts when the user is at some unknown location in the building and starts InLoc. An initial location fix (section IV-B) is estimated using BLE beacons present in the area; the relevant floor map is downloaded from the server and the same is displayed on the user's screen. The user then starts walking and the inertial module along with the particle filter estimates (section IV-C) and displays the user's location in real-time. The location derived by particle filter is intermittently corrected by fusing it with the location obtained from BLE beacons. At any time, the user can select a destination they desire to reach, and a shortest path to the destination from the user's current location is displayed (section IV-D, Figure 5). The workflow is depicted in Figure 2. We now detail out the algorithms used by InLoc system.

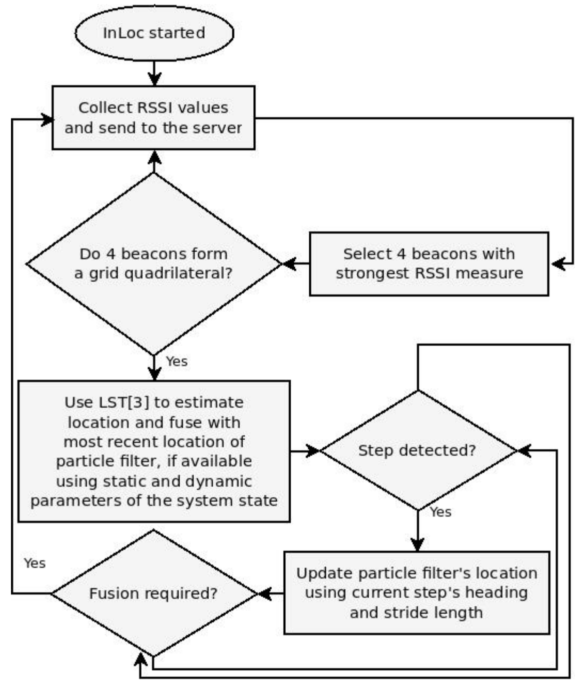


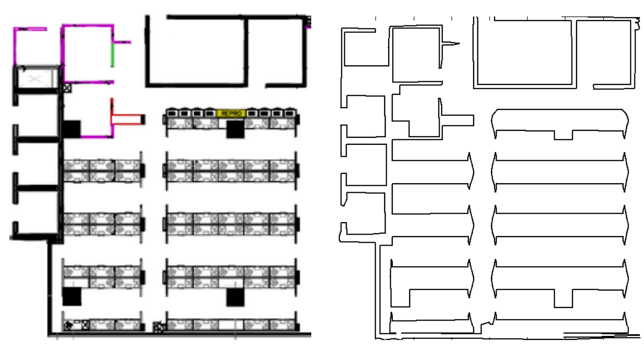
Fig. 2. InLoc Operational Flowchart

IV. Detailed Methodology

We first discuss the indoor map model which we use both for particle filter in IMU based tracking, and also for routing purpose.

A. Map Processing

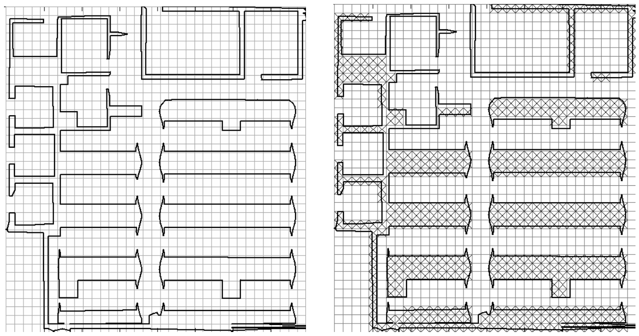
1) Map Vectorization: For implementing a tracking/routing system for purposes of guiding, emergency evacuation etc. in a large building, a vector floor plan is required. Normally, building administrations can provide raster blueprints. In order to make the system readily implementable without special efforts to re-create the map geometry (section II) by studying the topology from scratch, we use such raster plans (including fire exit plans) and build our model using them. The typical approach of manually creating the topology may require substantial effort when indoor floor plans are of the order of thousands of sq. m. A sample of a typical floor map is shown in Figure 3(a), which is a part of our workplace. We convert this raster floor plan to a set of closed polygons using the standard edge detection techniques [23] based on adaptive brightness cutoff. The resultant floor map is depicted in Figure 3(b) showing the polygonal forms of all the obstacles present in the map (*obstacle polygons*). Let us denote this set of obstacle polygons as $\mathbf{P}$. It should be noted that this kind of representation can be directly used with particle filter, since a clear representation of walkable and non-walkable (valid/invalid) areas has been made, which otherwise is done manually in traditional approaches. But for the purpose of routing, this representation should be re-interpreted in the form of nodes and arcs so that the state of the art routing algorithms can be readily applied.



(a) Raster Floor Map (b) Vector Floor Map

Fig. 3. Raster Blueprint Map and its Conversion to a Set of Obstacle Polygons

2) Vector Map Preprocessing: In this stage, we divide the map into a grid of uniform cells, as shown in Figure 4(a). In order to cater human stride length within two cell-hops (assuming it to be well below $100cm$), and also to balance with the intensive pre-processing, we assign the grid square a size C_l of $50cm$, which we have seen to behave well in the areas of more than $5000m^2$.



(a) Vector Map Divided into Cells (b) Valid/Invalid Cells

Fig. 4. Determination of valid/invalid cells during pre-processing

In order to create a meaningful geometrical model, we define two properties for a cell C:

Validity: C is valid if it represents a walkable area, implying the particle filter can propagate in C, and that while making routing decisions, C may be included in the shortest path. Let the set of all the valid cells on the map be $\mathbf{C_v}$. This decision is trivial if C covers an entirely walkable or an entirely non-walkable region. But for the third category of cells, we can have following approaches:

- We can render C invalid if it touches/intersects an obstacle polygon. The flaw with this approach is that this can declare narrow gateways, even a little less than $2 \times C_l$ un-walkable.
- In order to prevent the above problem, we follow the rule that $C \in \mathbf{C_v}$ iff $Area(C \cap P_i) < A_o$ where A_o is % overlap of C's area with $P_i \in \mathbf{P}$. We select A_o as 50%.

The valid/invalid cells are depicted in Figure 4(b).

Connectivity: Let us denote the set of i^{th} hop neighbor for C as $\mathbf{C_h^i}$. So, all the cells within i^{th} hop may be written as $\mathbf{C_{ha}^i} = \bigcup C_h^n$. We define $\mathbf{C_h^i} = \{\phi\} \ \forall \ C \notin \mathbf{C_v}$. Also denote the center coordinate of C as C_m, and for another cell $D \in \mathbf{C_v}$, let the line joining C_m and D_m be L_c^d. So we define connectivity, of C with another cell D as:

$$\mathbf{CO_C(D)} = \begin{cases} 1 & L_c^d \bigcap p = \{\phi\} \ \forall \ p \in \mathbf{P} \\ 0 & \exists p \in \mathbf{P} \text{ such that } L_c^d \bigcap p \neq \phi \end{cases}$$

For C, we define its i^{th} order adjacency list A_C^i as:

$$A_C^i = \{X : \ X \in \{D : \ \mathbf{CO_C(D)} = 1 \ \forall \ D \in \mathbf{C_v}\} \bigcap \ \mathbf{C_{ha}^i}\}$$

The map pre-processing stage computes $A_C^2 \ \forall \ C \in \mathbf{C_v}$. For routing only A_C^1 is used ($A_C^i \subset A_C^{i+1}$). For this purpose, we use GEOS C API [24] which is an open source geometry engine. This preprocessing is performed for all the floor maps of a building and is a one time processing. The preprocessed data is stored in the location server database, and data of relevant floor map is loaded as decided by the current location of the user. We emphasise here that we perform no complex image processing for deriving the nodes and arcs-based model, but just use easily available raster floor plan to create a simple vector model of a polygon set automatically, which is used both in particle filter and routing, detailed in sections IV-C and IV-D.

B. BLE Beacon-based Location Estimation

InLoc tries to estimate the first fix of the user's location by leveraging the BLE beacons spread in the building in a defined fashion. We now discuss the topology in which we arrange the beacons in the indoor area, and the details on how InLoc estimates the location of the user using beacons.

1) BLE Grid System: It has been observed that the BLE RSSI characteristics are more stable than the WiFi. But as a downside, the fading of RF signals from BLE devices is much more [18]. This reduces the range of the usable RSSI, and hence the beacons need to be placed closer to each other. As already discussed in section I, instead of using a normal fingerprinting approach to directly estimate the location, we use the same for estimating accurate distances from the beacons. LST is then used for estimating the user's location. For LST to work on the floor map, data from at least 3 anchor beacons is required. For a better accuracy, we lay the beacons in a set of continuous quadrilaterals of any dimensions on a floor. Denoting the set of all the beacon grids on a floor as G with i^{th} grid as $G(i)$ and the set of beacons in $G(i)$ as $\mathbf{B(i)}$, we place following restriction on the grid quadrilateral $G(i)$:

$$Distance(a, b) < 8 \ \forall \ a, b \in \mathbf{B(i)}$$

This condition makes sure that atleast 4 beacons are present in a usable RSSI range at every location on the floor, which corresponds to about $8 - 10m$ as we observed.

We store the beacon database in the system with an entry for each beacon b as $(HID, X, Y, \{G(i) : b \in B(i) \ \forall \ i\})$, where HID is the beacon hardware address, (X, Y) is the physical location of the beacon on the floor map and i denotes the grid numbers to which this beacon belongs to.

2) BLE Location Estimation: For estimating initial location of the user, RSSI values are collected from all the visible beacons and 4 beacons with strongest RSSI values are chosen and checked against the beacon database whether they belong to a grid quadrilateral. In case they don't, the process is repeated. We devise a novel approach for location estimation with BLE beacons. We explore the supervised machine learning based approach to derive the beacon specific model which accurately maps the RSSI measurements to distance. We then perform iterative LST to determine the absolute position. The novel contributions of this approach against usual fingerprinting are threefold:

- Various statistical features of RSSI measurements are derived and attributed to the beacon itself rather than to its absolute location.
- Beacon-specific model includes both beacon-to-beacon variations and also the environment in which it is placed. This is advantageous as all beacon models equally contribute to location estimation. As we have implemented

LST, the accuracy of positioning is maximized even if any erroneous distance from any beacon affects the same.

RSSI to Distance Mapping. Signal propagation model introduces large errors even after rigorous regression of equation parameters. Hence, we use a machine learning approach, which is an extended version of the work presented by Van et. al. [25]. It is a statistical approach that exploits the temporal and spatial variations of RSSI. We investigated the RSSI characteristics at various distances from beacons and observed that different statistical parameters (Max, Min, STD, Mean etc.) vary significantly at different distances. Considering these parameters, Van et. al. [25] has established a linear relationship between RSSI and distance:

$$D_i = -5 * S_i - 68$$

where, D_i is the distance between the tag and sensor, and S_i is the RSSI measure. However, this relationship is quite susceptible to a complex environment. We devise a per-beacon-model based technique where statistical parameters are used as features attributed to a specific distance. A supervised training process is performed to generate the training model for each BLE beacon. Later in positioning phase, a k-NN classifier is used to infer the distance from the training model.

Per-Beacon Training Phase. This involves computing various statistical features, assigning these features to distance bins, and subsequently combining them to generate the training model. This phase includes the following:

- *Data Collection:* For every beacon in the premises, a set of RSSI measurements are collected from several reference points at distances of $1m - 8m$ with $1m$ interval inside the grid. While gathering the data at a distance d, the user moves over a circular path with radius d covering most LOS/NLOS reference points. The true locations of the reference points are determined using the digital floor map's scaling factor. To accomplish the experiment, we deploy Estimote BLE beacons and choose the broadcast interval to be $900ms$.
- *Normalization:* Each data fingerprint gathered at a different distance is normalized using the following:

  $$NormRSSI_i = (RSSI_i - MIN)/(MAX - MIN)$$

  where MAX and MIN are the are global maximum and minimum RSSI values for the distance in the range of $1 - 8m$.
- *Feature Extraction:* As the user moves through the experiment, a sliding window with 80% overlap in time domain is used to compute the typical statistical features from RSSI for each distance. The total samples captured for each distance are 60, and using 80% overlap, the windows with 15 samples each are created. One typical feature point can be represented as:

  $$P_i = \big(Mean, STD, Mode, Median, Max, Min : Ds\big)$$

  where i is the window number and Ds denotes the actual distance. The training model is generated with all fingerprints for all the windows over every distance.

Testing Phase: Distance Estimation. In testing phase, first the grid of top 4 beacons is identified, as already discussed, and RSSI measurements are gathered from the beacons situated in the grid. As described in the training phase, sliding window with 80% overlap is applied and subsequently features are calculated from RSSI data. These features and training models are fed to a k-NN based classifier to estimate the distance from each beacon. k-NN is a simple instance-based non-parametric learning algorithm that stores all training instances and classifies new instances based on Euclidean distance. For each window, one distance is estimated. The mode of all derived distances is selected as the final distance.

Trilateration: Location Estimation. Trilateration module is deployed to determine the final position of the user. Static beacon locations and associated grid details are fed as input to this module. This module receives all distances from beacons inside the grid and performs LST [4] to obtain the absolute user location.

C. Inertial Tracking

After the user has been located in a building, request is made to the location server to download and show the location's relevant floor map on the device's screen. At this point, the user can opt for a route if they want to go to a destination by selecting the location of the same on the screen. Routing is discussed in section IV-D.

1) The Particle Filter: For a real-time location tracking, we use IMU sensors embedded in the phone to send the per-step updates to the location server. The location server runs an instance of a particle filter (PF) for each user, where the step updates are exploited to estimate the most probable location of the user, and the same is reflected back to the user.

PF estimates user's location at each step by propagating its state through the step using step length and the heading estimated during that step. A large no. of particles ensure that the errors in sensor observations are neutralized properly. It uses map information of valid/invalid areas in order to propagate the particles. For n^{th} step, the propagation equations in 2-D are given by:

$$x[n] = x[n-1] + (SL[n] + \delta SL[n])(\cos(\theta[n] + \delta\theta[n]))$$
$$y[n] = y[n-1] + (SL[n] + \delta SL[n])(\sin(\theta[n] + \delta\theta[n]))$$

where $SL[n]$ is the stride length estimated for n^{th} step, and $\theta[n]$ is the heading of the user with respect to the map coordinate system. $\delta SL[n]$ and $\delta\theta[n]$ are the observation errors, which are modeled as zero mean Gaussian noise. We have observed that the variance measure can be experimentally tweaked as per the nature of the floor map. It controls the freedom with which the particles propagate.

Let the maximum no. of particles allowed be N. For every step update, the PF creates new particles for every existing one by applying the propagation equations. Let the set of particles at step s be $P(s)$, and the set of child particles generated for i^{th} parent particle be $C(s,i)$, where every particle $p \in C(s,i)$ is such that p lies in one of the cells in A^2_C where C is the cell where parent particle lies. We go for re-sampling if $|C(s,i)| \forall i$

falls below a certain percentage, in which case a new particle is generated around a live particle for every particle died by landing up in a location in cell $C \notin A_C^2$. We deem the user's location at step s after propagating to step $s+1$ as the location of particle $p \in P(s)$ such that $|C(s,p)| = max\{|C(s,i)| : i \in P(s)\}$.

2) Heading and Stride Length: Accuracy of PF depends directly on the accuracy of stride length and heading parameters of a step. Li et. al. [9] has used a stride length method with their PF which requires to be trained for every user for a good accuracy. We use an improved model from our work in Chandel et. al. [26] to avoid this personalisation, where we integrate height factor into the model, and derive activity-particular parameters (walking, brisk walking, and running).

Getting accurate heading is more challenging than stride length, since there are multitude of error sources, and the results might depend on the surroundings, since magnetometer is the prime sensor used for this purpose. In order to prevent the undesired deflections in magnetometer, we fuse its data with the angular information from gyroscope on a per-step basis. This minimizes the drift effect introduced by gyroscope, and also serves the purpose of neutralizing magnetic disturbances in the area. The key idea behind this fusion is that the magnetometer and gyroscope show similar variance characteristics only when there is no external magnetic disturbance.

D. Routing

Routing is a very important feature of a localization application, which improves the usability of the positioning system by complementing the user's current location with the information about the path to any destination they might want to reach to. As an example, there may be a scenario when a delegate needs to attend to a meeting. The destination might already have been added to their planning calender. User may then select the meeting and the best path to the meeting venue would be displayed after determining their current location. As already discussed in section I, InLoc provides both positioning and routing features.

For routing, we use the grid representation to derive the graph $G = (E, V)$ which is exploited to perform routing using well-known routing algorithms. We define:

$$E = \{L_c^d : C, D \in \mathbf{C_v} \ \& \ D \in A_C^1\},\ V = \{C_m : C \in C_v\}$$

This definition of G can be used readily with well-known routing algorithms like Dijkstra's, A* etc. for estimating a shortest path from current location of the user. We emphasize here that the metric for defining 'shortest' may either be distance or time. Although we present our algorithm taking in consideration the former in this work, where the edge weight is simply the edge length, but our model is well-capable of calculating the time metric, which can be achieved by defining a congestion parameter for all elements of E at regular intervals. This is possible from the crowdsourced proximity information from BLE beacons alone, traced from the users using InLoc in the area.

E. Location Fusion

While tracking the user, particle filter, though rarely, can hit a stage where all the particles start to die, ex., in a slowly converging area, and the no. of alive particles fall below a certain critical level. At this stage, we analyze a set of static and dynamic parameters, both for RF and inertial modules (P_{Is}, P_{RFs} & P_{Id}, P_{RFd}) for the system at current location, calculate an independent location from the beacons, and fuse it with the latest inertial module's location as per following:

$$L_{fused} = (\sum_i p_{Is}^i + \sum_i p_{Id}^i) \cdot L_I + (\sum_i p_{RFs}^i + \sum_i p_{RFd}^i) \cdot L_{RF}$$

where L_{fused}, L_I and L_{RF} are the fused, pure inertial and pure beacon-based location respectively.

$$P_{Is} = \{1 - (|A_C^2|/|C_{ha}^2|),\ Map\ Accuracy\ Quotient\}$$

$$P_{Id} = \{Alive\ Particles/Maximum\ Particles, 1 - (NV_b/N_b)\}$$

$P_{RFs} = 1 - P_{Is}$ and $P_{RFd} = 1 - P_{Id}$, where N_b is the total no. of visible beacons, and NV_b is the no. of beacons with signal strength above a certain pre-defined level. This measure quantifies the closeness of the beacons in the area, hence the accuracy of the location obtained by trilateration. Map accuracy quotient defines the accuracy in the floor map geometry, with 0.5 as fully accurate, a lower value otherwise.

As shown in section V, InLoc's fusion module provides a fairly accurate location when the particle filter is stuck. The location might not be as accurate as that of inertial-based in some cases, but if allowed to propagate, can lead to location failure. This is a practical problem with most of the existing arts when used for tracking for an extended time using a particle filter. Fused location helps the particle filter to resume, and with an accuracy which is greater than that given by BLE beacons alone. So, fusion benefits both from the accuracy of inertial module, and ubiquitous availability of BLE beacons.

V. Experimental Setup and Results

We test InLoc in various complex office environments using various mobile devices. We post our results separately for each novel component of the system in the following sections.

A. Routing and Tracking

In order to test the efficiency of our vector map model for performing graph operations like routing, we used the map from our Bangalore premises, which is over $7000m^2$ in area. We made 50 shortest route queries of length between $50m - 120m$ from user's current location, which took an average time of $80ms$ per query with the standard Dijkstra's algorithm running on an Intel Quad Core system. A mobile screen-shot is shown in Figure 5 showing the shortest path from user's current location along with the distance to the destination, and Figure 6 showing the actual path traced by the user thereafter, derived using the inertial module. The map is pre-annotated with some key locations user might want to reach. Mean error in location for each step was recorded as $0.4m$, which is a big improvement over the results by Liu et. al. [5] $(1.6m)$

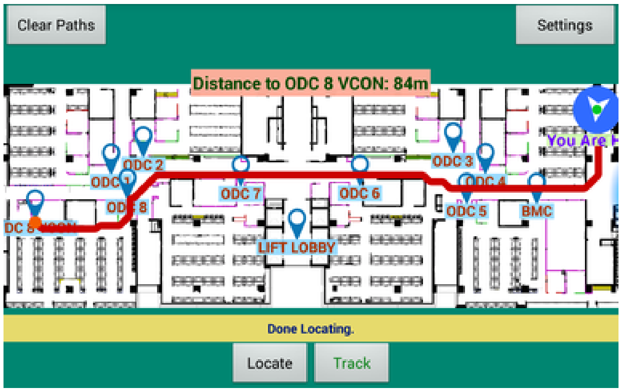


Fig. 5. Shortest Path to 'ODC 8 VCON' Displayed on Phone Screen

Fig. 6. User's Tracked Path (Blue) along the Suggested Shortest Path (Red)

B. RSSI to Distance Estimation

RSSI to distance mapping is one of the key contributions of this work. We compared our distance estimation method with that presented by Zhao et. al. [18], which employs a familiar attenuation model which maps the RSSI to distance. While conducting the experiments, we gathered the RSSI data from known distances with NOS/NLOS conditions at various instances. We confined our evaluation to $8m$. Table I illustrates the comparative analysis between Zhao et. al. [18] and our system. The lower errors and standard deviations for most of the distances clearly signify that InLoc's distance estimation is moderately stable, accurate and consistent.

TABLE I
ABSOLUTE ERROR COMPARISON IN THE FORM OF MEAN+/-STD

Actual Distance	Error (InLoc)	Error (Zhao et. al.[18])
2	1.2 ± 1.09	1.03 ± 0.17
3	$.8 \pm .44$	$1.7 \pm .6$
4	$1.2 \pm .83$	$1.2 \pm .52$
5	$1 \pm .70$	$1.9 \pm .55$
6	$1.6 \pm .54$	2.9 ± 2.5
7	$.6 \pm .54$	2.43 ± 1.8
8	$.2 \pm .4$	3.4 ± 1.22

C. Location Accuracy: BLE Vs. WiFi Fingerprinting

BLE beacons alone are used to estimate user's current location on the floor when InLoc starts (Figure 2). This location is also required for fusion as discussed in section IV-E. We analyzed the performance of the BLE location estimation by comparing it with RADAR, a very familiar WiFi fingerprinting system proposed by Bahl et. al.[27]. The experiments were conducted in our complex Kolkata premises of over $1200m^2$ in area, incorporating four WiFi access points. The floor plan of experimental set up, test points (in yellow), BLE beacons (in blue) and WiFi access point locations (in green) are shown in Figure 7. Cumulative distribution functions (CDF) of the location errors for location estimations using both BLE and WiFi are shown in Figure 8, where it is clearly visible that InLoc reaches high probability values faster, as its distance error is concentrated in small values. This can be attributed to a stable and accurate distance estimations from anchor beacons (section V-B).

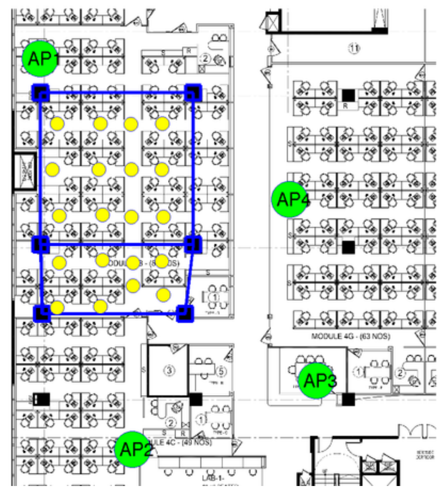


Fig. 7. Experimental Setup for BLE vs. WiFi Location Comparison

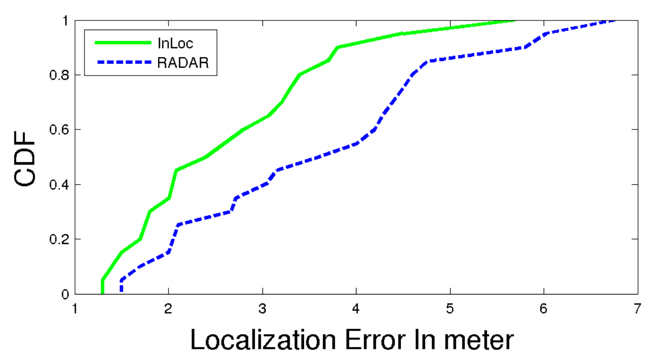


Fig. 8. CDF Comparison of Location Errors

D. Inertial+BLE Fusion

As discussed in section IV-E, fusion extracts the best of both inertial location, and location from BLE beacons when the no. of particles fall below a critical level. We traced such rare cases, and observed that our fusion method works expectedly, by providing a fused location more accurate than the BLE (and inertial too in most of the cases). Figure 9 shows some of the instances where fusion was required, and that our fusion method yields an improved location estimate in all the cases with a mean location error of $0.9m$ as compared to $2.6m$ by [20] using only BLE beacons.

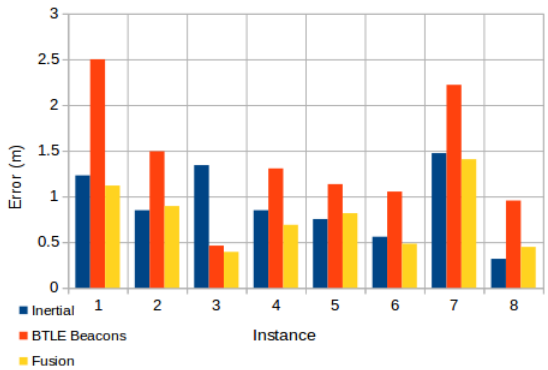


Fig. 9. Location Error Comparison

VI. Conclusion

We proposed a complete and easy to implement end-to-end InLoc system with integrated shortest path feature, by leveraging a simple and an efficient method for vectorizing commonly available indoor raster floor plans. We also devised an improved machine learning based method for estimating distance from the BLE beacons, which in turn helped us to achieve better estimation of the user's initial location fix. The system can be integrated with meeting planners etc. where the user can get path to his destination with a single click, and can track self in realtime, which can provide a big impact from business perspective. A technique for fusing inertial-based and RF-based locations was discussed, improving the overall system robustness. The system can be readily used for multi-floor tracking involving staircases and lifts, which we plan to demonstrate in our future endeavours, along with a turn-by-turn direction guidance while routing a user.

References

[1] "Gps-sps," http://www.gps.gov/systems/gps/performance/accuracy/, accessed: 2016-05-30.
[2] "Garmin products," http://www8.garmin.com/automotive/pdfs/drive.pdf, accessed: 2016-05-29.
[3] M. Siekkinen, M. Hiienkari, J. K. Nurminen, and J. Nieminen, "How low energy is bluetooth low energy? comparative measurements with zigbee/802.15. 4," in *Wireless Communications and Networking Conference Workshops (WCNCW), 2012 IEEE*. IEEE, 2012, pp. 232–237.
[4] Y. Zhou, "An efficient least-squares trilateration algorithm for mobile robot localization," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2009, pp. 3474–3479.
[5] Y. Liu, M. Dashti, and J. Zhang, "Indoor localization on mobile phone platforms using embedded inertial sensors," in *Positioning Navigation and Communication (WPNC), 2013 10th Workshop on*. IEEE, 2013, pp. 1–5.
[6] I. Constandache, R. R. Choudhury, and I. Rhee, "Towards mobile phone localization without war-driving," in *Infocom, 2010 proceedings ieee*. IEEE, 2010, pp. 1–9.
[7] M. Alzantot and M. Youssef, "Uptime: Ubiquitous pedestrian tracking using mobile phones," in *Wireless Communications and Networking Conference (WCNC), 2012 IEEE*. IEEE, 2012, pp. 3204–3209.
[8] W. Kang and Y. Han, "Smartpdr: smartphone-based pedestrian dead reckoning for indoor localization," *Sensors Journal, IEEE*, vol. 15, no. 5, pp. 2906–2916, 2015.
[9] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, and F. Zhao, "A reliable and accurate indoor localization method using phone inertial sensors," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM, 2012, pp. 421–430.
[10] J. Qian, J. Ma, R. Ying, P. Liu, and L. Pei, "An improved indoor localization method using smartphone inertial sensors," in *Indoor Positioning and Indoor Navigation (IPIN), 2013 International Conference on*. IEEE, 2013, pp. 1–7.
[11] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: zero-effort crowdsourcing for indoor localization," in *Proceedings of the 18th annual international conference on Mobile computing and networking*. ACM, 2012, pp. 293–304.
[12] J. A. B. Link, P. Smith, N. Viol, and K. Wehrle, "Footpath: Accurate map-based indoor navigation using smartphones." in *IPIN*. Citeseer, 2011, pp. 1–8.
[13] W.-Y. Shih and K. Lan, "Using smartphone with un-scaled map for indoor localization," in *Proceedings of the IEEE 9th Annual Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), Seoul, Korea*, 2012, pp. 18–21.
[14] C. S. Jensen, H. Lu, and B. Yang, "Graph model based indoor tracking," in *Mobile Data Management: Systems, Services and Middleware, 2009. MDM'09. Tenth International Conference on*. IEEE, 2009, pp. 122–131.
[15] M. Stroila, A. Yalcin, J. Mays, and N. Alwar, "Route visualization in indoor panoramic imagery with open area maps," in *Multimedia and Expo Workshops (ICMEW), 2012 IEEE International Conference on*. IEEE, 2012, pp. 499–504.
[16] L. Han, T. Zhang, and Z. Wang, "The design and development of indoor 3d routing system," *Journal of Software*, vol. 9, no. 5, pp. 1223–1228, 2014.
[17] K. Subbu, C. Zhang, J. Luo, and A. Vasilakos, "Analysis and status quo of smartphone-based indoor localization systems," *Wireless Communications, IEEE*, vol. 21, no. 4, pp. 106–112, 2014.
[18] X. Zhao, Z. Xiao, A. Markham, N. Trigoni, and Y. Ren, "Does btle measure up against wifi? a comparison of indoor location performance," in *European Wireless 2014; 20th European Wireless Conference; Proceedings of*. VDE, 2014, pp. 1–6.
[19] A. N. Raghavan, H. Ananthapadmanaban, M. S. Sivamurugan, and B. Ravindran, "Accurate mobile robot localization in indoor environments using bluetooth," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, May 2010, pp. 4391–4396.
[20] R. Faragher and R. Harle, "Location fingerprinting with bluetooth low energy beacons," *Selected Areas in Communications, IEEE Journal on*, vol. 33, no. 11, pp. 2418–2428, 2015.
[21] L. Chen, L. Pei, H. Kuusniemi, Y. Chen, T. Kröger, and R. Chen, "Bayesian fusion for indoor positioning using bluetooth fingerprints," *Wireless Personal Communications*, vol. 70, no. 4, pp. 1735–1745, 2013. [Online]. Available: http://dx.doi.org/10.1007/s11277-012-0777-1
[22] L. Zhang, X. Liu, J. Song, C. Gurrin, and Z. Zhu, "A comprehensive study of bluetooth fingerprinting-based algorithms for localization," in *Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on*. IEEE, March 2013, pp. 300–305.
[23] D. Marr and E. Hildreth, "Theory of edge detection," *Proceedings of the Royal Society of London B: Biological Sciences*, vol. 207, no. 1167, pp. 187–217, 1980.
[24] "Geos," https://trac.osgeo.org/geos/, accessed: 2016-05-30.
[25] E. Van der Laan, "Radio propagation aided indoor localization: Indoor localization by applying proportionate measurement localization (pml) using bluetooth low energy tags," Ph.D. dissertation, TU Delft, Delft University of Technology, 2014.
[26] V. Chandel, A. Sinharay, N. Ahmed, and A. Ghose, "Exploiting imu sensors for iot enabled health monitoring," in *Proceedings of the First Workshop on IoT-enabled Healthcare and Wellness Technologies and Systems*, ser. IoT of Health '16. New York, NY, USA: ACM, 2016, pp. 21–22. [Online]. Available: http://doi.acm.org/10.1145/2933566.2933569
[27] P. Bahl and V. N. Padmanabhan, "Radar: An in-building rf-based user location and tracking system," in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2. Ieee, 2000, pp. 775–784.