MEC-based UWB Indoor Tracking System

José Luis Carrera V., Zhongliang Zhao, Mischa Wenger, Torsten Braun
Institute of Computer Science, University of Bern, Switzerland

***Abstract*—Real-time localization is the underlying requirement for providing context-aware services in the Internet of Things (IoT). Although several methods have been proposed to provide indoor localization, most of them implement the running algorithms locally in the mobile device to be located. However, the limited computational resources of mobile devices make it difficult to run complex algorithms. As an alternative, Multi-Access Edge Computing (MEC) as a promising paradigm extends the traditional cloud computing capabilities towards the edge of the network. This enables accurate location-aware services. In this work, we present an indoor tracking system based on the MEC paradigm for ultra wide band devices. Our tracking algorithms fuse machine learning-based zone prediction, Ultra Wide Band (UWB) radio ranging, inertial measurement units, and floor plan information into an enhanced particle filter. The localization process is hosted in an Edge server, which performs the resource-demanding calculation that is offloaded from the client devices. Moreover, the client devices are also equipped with certain processing power to handle sensor data processing. Our system includes also a Cloud layer, which enables data storage and data visualization for multiple clients. We evaluate our system in two complex environments. Experiment results show that our tracking system can achieve the average tracking error of 0.49 meters and 90% accuracy of 0.6 meters in real-time.**

***Index Terms*—Internet of Things, Indoor localization, MEC computing, Cloud computing, particle filter.**

I. Introduction

The rising interest around IoT and context-aware applications has introduced a variety of technologies to deal with all the produced data in the field of the IoT and context-aware applications [21]. Context-aware applications must be able to timely react to the current physical context (i.e., environment) of mobile users. Thus, real-time localization becomes the underlying requirement of these applications. However, real-time indoor localization requires significant computation, which is typically running on the mobile devices with limited resources. Offloading the heavy computation to a third party server could resolve the problem, but the data transmission between the mobile devices and centralized server could lead to increased latency and unreliable performance, which is not tolerable for real-time applications.

Multi-Access Edge Computing (MEC) technology [17] has been proposed as an alternative solution to bring cloud computing capabilities to the network edge to meet application requirements of short latency and high resource-demanding. By deploying the indoor tracking system on the MEC paradigm, heavy computation tasks can be offloaded from client devices to near edge servers, while the short device-server distance could reduce the data transmission time to guarantee real-time localization performance. Therefore, by eliminating the distance and the time it takes to send data to one distributed server, performance of the indoor localization system can be improved.

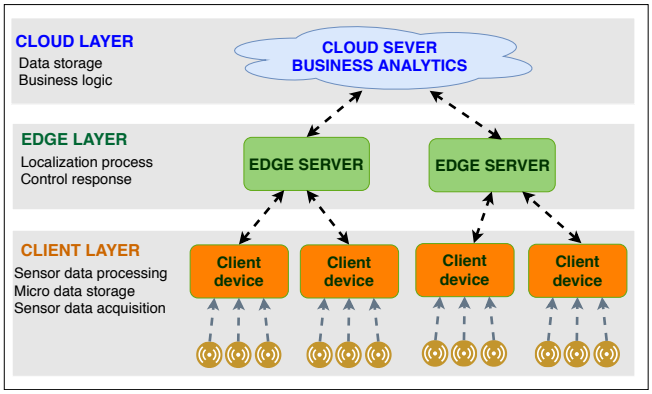


Figure 1: MEC-based Indoor Tracking System Architecture.

Due to the growing ubiquitousness of the IoT and the importance of context aware services, indoor localization has become an important research topic. Several indoor localization methods have been proposed. Radio Frequency (RF) technology is one of the promising solutions to provide real-time indoor localization. Compared to other RF technologies, ultra wide band (UWB) has received increased interests due to its capability to reduce the localization errors to lower than one meter. UWB is robust to multi-path effects because UWB radios are able to differentiate pulses reflected from different objects. Thus, UWB-based localization achieves better accuracy and reliability than other wireless technologies such as Wi-Fi or Bluetooth, which normally achieves localization accuracy of several meters [20].

Fingerprinting-based indoor localization systems usually consist of two phases: training phase (off-line) and localization phase (on-line). In the off-line phase, the fingerprint database is built by collecting various types of radio signals in the target indoor environments. The earth magnetic field (MF) in indoor environments presents distortions over space due to the presence of ferromagnetic materials. These MF distortion patterns can also be used to identify indoor locations [1]. MF and RF observations can be used as fingerprints to detect unique locations in indoor environments. In the on-line phase, the observed fingerprint at an unknown location is compared with the stored fingerprints in the fingerprint database to determine the closest match. Any single learning model can be applied. However, ensemble learning models usually allow better predictive performance compared to single

ISBN 978-3-903176-13-3

models [13]. Fingerprinting-based methods that build the classification model exclusively based on previously observed data (i.e., fingerprint database) are called discriminative learning methods.

Range-based approaches need to convert the measured RF parameters into range values. This process is called ranging. After ranging, multilateration methods can be adopted to derive the absolute position of the target. However, some RF technologies such as Wi-Fi are highly affected by multi-path effects especially in Non-Line of Sight (NLOS) conditions. Thus, in such technologies, range-based localization methods present high localization errors.

Recently, UWB technology has become popular for accurate indoor localization. Its main feature is the use of a large frequency band (500 MHz). UWB radios are able to precisely differentiate between pulses that are reflected from different objects. Thus, UWB is highly robust to multi-path effects. Therefore, using UWB technology assures high performance in dense multi-path environments.

In this work, we present a MEC-based indoor tracking system (InTrack) for UWB devices that is implemented based on the paradigm of MEC. MEC extends traditional mobile cloud computing capabilities towards the edge of the network and enables accurate location-aware services. Fig. 1 shows the architecture of the system. We provide high indoor tracking performance by fusing machine learning zone prediction information, UWB radio-based ranging, inertial measurement units (IMUs), and floor plan information in an enhanced particle filter. We define a zone as any subarea in the area of interest. The zone detection method is designed with an ensemble learning algorithm by combining Hidden Markov Models (HMM) and discriminative learning methods. Further, we integrate zone prediction results with an UWB radio-based ranging and floor plan information in a particle filter approach.

The rest of the paper is organized as follows. Section II presents related work. The architecture of our proposed tracking system is reviewed in Section III. Section IV presents the implementation of the MEC-based tracking system. Section V discusses the performance evaluation results. Section VI concludes the paper.

II. Related Work

Multi-Access Edge Computing (MEC) has been introduced to bring the cloud services and resources closer to end users by leveraging the available resources in the network edge [17]. MEC aims to enable resource-limited mobile devices to execute the real-time computation-intensive applications directly at the network edge [2]. Compared to cloud computing, the distinguishing features of MEC are its closeness to end-users, location awareness, mobility support, and low latency.

Indoor localization has attracted significant attention in recent years. Many localization solutions have been investigated, mostly using RF technology or IMU movement information. Due to the the fast development of IMUs in modern commodity mobile devices, Pedestrian Dead Reckoning (PDR) methods have attracted research interest. IMUs can be leveraged to detect pedestrian movement (i.e., heading orientation and step recognition, and stride displacement) [19]. PDR systems estimate the new location based on the previous location. Authors of [11] compute the heading orientation based on gyroscope measurements, whereas the displacement is estimated from accelerometer readings. Since PDR systems measure position changes rather than the absolute position, PDR positioning results in an accumulation of sensor errors over time. Thus, some additional information must be considered to deal with this cumulative errors.

RF technologies are often used to provide indoor localization. In [28], the authors proposed to use Wi-Fi received signal strength indicator (RSSI), whereas in [15] time information related to radio propagation is applied. Radio-based indoor localization can be classified as range-based and range-free methods. Range is defined as the propagation distance from the target to Anchor Nodes (AN). The first stage in range-based localization methods is to calculate the propagation distances (i.e., ranges). Then, different positioning algorithms can be used to estimate the absolute locations of the targets [14]. In some RF technologies such as Wi-Fi, range-based methods do not work well in indoor environments because of the multi-path effects produced by the presence of obstacles [9]. Therefore, range-free methods such as fingerprinting are often used. However, it is very time consuming to build up a radio map, which is required to locate the targets in fingerprinting. In [10], the localization systems achieve around 1 meter accuracy by relying on radio frequency identification (RFID) technology. However, many RFID ANs must be deployed to provide such localization performance. UWB technology is used in [8] to overcome limitations of other technologies and achieve high localization accuracy. Authors claim to achieve localization accuracy lower than 0.2 meters. However, the localization approaches were tested only in small areas of interest.

Hidden Markov Models can be considered to support indoor localization. In [26] authors employed HMM and radio propagation models to reduce calibration efforts. The system utilizes a discrete probability distribution to derive probable positions. Then, the position is estimated from the most probable estimated positions. In [16], authors propose to fuse IMU measurements with wireless signal readings. Then, the new candidate position is derived by determining the pedestrian motion pattern and the most probable wireless signal reading at that position. Although authors report good accuracy, the method to determine the transition probabilities is not explained. Moreover, the applicability of the solution is restricted to the fidelity of the pedestrian motion pattern recognition method. In [22], authors include movement measurements (e.g., heading orientation) in the proposed HMM. Thus, the reported accuracy is improved compared to [26]. In [27], authors propose to fuse a RSSI pattern recognition method and HMM to provide indoor localization. Then, the transition probabilities of the HMM are derived from the pedestrian trajectories and the pattern recognition method. The pattern recognition method relies on a beforehand built radio map

ISBN 978-3-903176-13-3

database. Thus, some reference locations are defined through the indoor environment in an off-line phase to collect reference samples. Such collection process could take several hours or days for small or big areas, which is very labor expensive and time consuming.

III. System Architecture

This section presents the design details of the proposed MEC-based indoor localization system. Figure 1 summarizes the system architecture, which includes three layers: Cloud layer, Edge layer, and Client layer. Details of each layer are described below.

A. Client Layer

The client layer includes mobile devices (MD) that are to be located. The mobile devices constantly collect data from on-board sensors, such as inertial measurement units, Wi-Fi or UWB radio interfaces, etc. Instead of sending raw data to the Edge layer directly, mobile devices process them locally to derive meaningful insights (i.e., movement directions and speeds, Wi-Fi and UWB fingereprints, and ranges) through three modules of PDR-velocity, fingerprint acquisition, and UWB-ranging. The derived information are then sent to the Edge layer via the data transmission module for further processing. This architecture leaves all the device-dependent data processing, such as Wi-Fi or UWB signal processing, to happen at the Client layer. A significant advantage of this design is that the Edge layer is completely independent of the client device specifications, which makes the system capable to support different device types. For instance, a smartphone or Rasperry Pi can be easily integrated into the system, without any modifications at Edge and Cloud layers, as far as relevant information can be generated from the raw data. Figure 2 shows the client layer architecture, whose core subcomponents are described below.

1) PDR-velocity: In order to estimate the velocity of mobile devices, we use the accelerometer gyroscope, and the magnetometer sensors, from which the heading direction and the speed can be computed. To estimate the heading direction, we rely on a digital compass developed from the magnetometer, gyroscope, and accelerometer sensors embedded in the MD. Digital compass measures the clockwise angle between the magnetic north and the Y axis of the smartphone at time t. This value is called Azimuth (α_t). Therefore, the heading orientation (θ_t) in the local coordinate system can be determined as follows:

$$\theta_t = X_{off} - \alpha_t, \tag{1}$$

where X_{off} is the clockwise angle between the X axis of our local coordinate system and the magnetic north.

To estimate the speed of the MD, we use the accelerometer sensor. Thus, the speed is computed by using Equation 2

$$\overline{v} = \int_{t_o}^{t_f} a \cdot dt, \tag{2}$$

Since accelerometer data contains huge non-zero mean noise, accelerometer data is smoothed by using low pass filters.

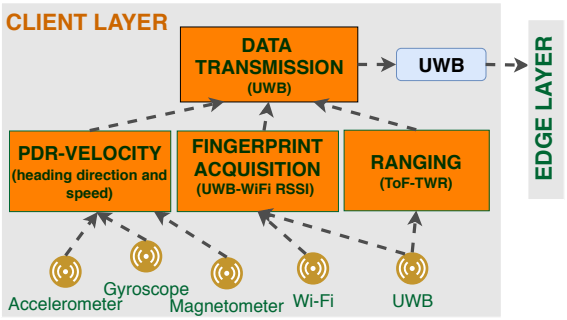


Figure 2: Client layer architecture

2) Fingerprint acquisition: Through the embedded Wi-Fi and UWB interface, the mobile device collects Wi-Fi and UWB RSSI values from surrounding Wi-Fi and UWB anchor nodes. Afterwards, this data is passed to the Edge layer as inputs for the zone recognition process.

3) UWB-ranging: The ranging process is conducted by the Two Way Ranging method (TWR). TWR determines the time of flight of the UWB radio frequency signal and then computes the distance between the nodes by multiplying the time by the speed of light [23].

4) Data transmission: When the PDR-velocity, fingerprint acquisition, and ranging processes are completed, the output data from these processes must be transmitted to the Edge layer for further processing. The transmission is conducted by using UWB technology.

B. Edge Layer

The Edge layer is responsible for running the computation-heavy localization algorithm to calculate the indoor locations in real-time. It includes three sub-modules and two interfaces. The zone prediction module is to estimate the indoor zone information using the received fingerprints. The space representation module is to constrain the location estimation ranges. The data fusion module is to apply advanced particle filter to fuse multiple inputs to estimate locations. The required information to feed the zone prediction and data fusion modules are periodically received from the Client layer via the UWB interface. Calculated indoor locations are sent to the Cloud layer via a web-socket. Figure 3 summarizes the processes of the Edge layer. Details of each components are given below.

1) Zone Prediction: We propose an ensemble learning method for zone prediction. The key idea of this method is to combine conceptually different individual machine learning algorithms in a HMM. Thus, the zone prediction method is also based on the concept of Markov localization [7], which can be described by estimating the state of the system with controllable state transitions. In HMM systems, the state is not directly visible. However some observations, dependent on the state, are visible. The sequence of observations generated by a HMM gives information about the sequence of states. In our zone prediction method, we define as observations the zone detection outputs given by the individual machine learning algorithms. Thus, the elements of the HMM of the zone prediction method are as follows:

ISBN 978-3-903176-13-3

- The set of n states $Z = \{z_1, z_2, ..., z_n\}$. Thus, the discrete random variable $s_t \in Z$ represents the hidden state at time t.
- The transition probability matrix A,

$$A = a_{1,1}, a_{1,2}, ..., a_{n,1}, ..., a_{n,n}$$

where a_{ij} is the probability of moving from zone z_i to zone z_j. Therefore, A can be written as follows:

$$A = \{a_{ij} = P(s_{t+1} = z_j \mid s_t = z_i)\}, \quad (3)$$

where A is a $n \times n$ matrix, a_{ij} represents the transition likelihood between zone z_i to zone z_j.
- The set of observations O,

$$O = \{(o_1, o_2, ...o_m)_1, ...(o_1, o_2...o_m)_r\}, \quad (4)$$

where o_i is the zone detection output of the i-th individual machine learning method. Thus, $q_t \in O$ represents an observation at time t.
- The emission probability matrix B,

$$B = b_{1,1}, b_{1,2}, ..., b_{1,r}, ..., b_{n,r}$$

where $b_{i,j}$ is the likelihood of an observation $(o_1, o_2...o_m)_j$ being generated at zone z_i. B contains the likelihoods of producing a particular set of observations q_j at zone z_i. Thus, B can be written as follows:

$$B = \{b_{ij} = P(q_j \mid z_i)\}, \forall q_j \in O \wedge z_i \in Z, \quad (5)$$

$P(q_j \mid z_i)$ can be computed assuming conditional independence among the prediction outcomes o_i given z_i. Our assumption is that the probability of obtaining the outcome o_i becomes independent if the value of z_i is known. The individual machine learning methods that constituent the ensemble zone prediction method are conceptually different and independent of each other. Therefore, it is reasonable to assume that their outcomes are conditionally independent given z_i. Thus, b_{ij} can be written as follows:

$$b_{ij} = \prod_{n=1}^{m} P(o_j \mid z_i)_n, \quad (6)$$

where $P(o_j \mid z_i)_n$ is the probability of predicting o_j at zone z_i by the n-th individual discriminative learning method. Therefore, $P(o_j \mid z_i)_n$ represents the prediction performance of individual predictors given the knowledge of the ground-truth class label (i.e., zone). Thus, $P(o_j \mid z_i)_n$ value can be obtained from the confusion matrix of the n-th individual machine learning algorithm part of the zone prediction method. Finally, the HMM of the zone detection method can be solved by applying the Viterbi algorithm [6].

2) Space Representation: To minimize the algorithmic complexity, our system defines a discrete structure to replace the conventional floor map. All the system states (i.e., indoor positions) are represented by a discrete set of locations by converting from a continuous state space to a discrete state space. Therefore, we consider the physical environment as a spatial data structure that defines space as an array of cells

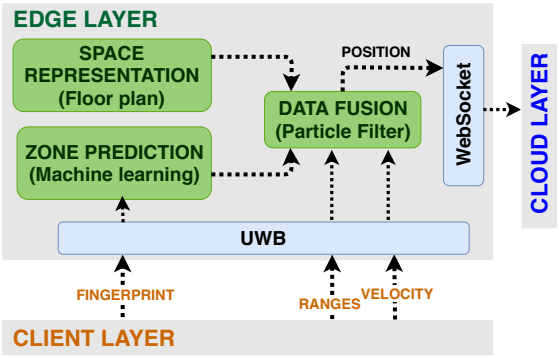


Figure 3: Edge layer architecture

arranged in rows and columns. Thus, each cell (i.e., location) belongs to a zone. We define two types of zones, enabled zones and not enabled zones. In the enabled zones the target object is allowed to move (i.e., corridors, offices, etc.). In the not enabled zones the target object is not allowed to move (i.e., walls, furniture, etc.). Therefore, the Space Representation component computes the map likelihood (i.e., allowed areas to spread particles), and the transition model (i.e., connections among zones).

3) Data Fusion: We consider indoor localization as a filtering problem, in which the position of the target can be computed from several noisy environmental observations. Thus, this work focuses on a particle filter approach to provide indoor localization.

In a particle filter approach, a belief of the target position is computed based on the observations, (i.e., posterior probability distribution). The posterior probability can be represented as a set of weighted particles. Particle filters estimate the posterior probability distribution of the system state based on some measurements q_t at time t [3]. At time t, the system state vector X_t is defined as:

$$X_t = [x_t, y_t, \theta_t, \ell_t, z_t], \quad (7)$$

where (x_t, y_t) are the Cartesian coordinates of the target object, θ_t is the heading orientation, ℓ_t is the displacement length and z_t is the zone in which the target is located.

Since locations belong to a zone, the zone z_t can be computed from the current Cartesian coordinates (x_t, y_t). Therefore, function $z_t = f(x_t, y_t)$ derives the current zone z_t. Thus, z_t can be written as:

$$z_t = f(x_{t-1} + \ell_t \cdot cos(\theta_t), y_{t-1} + \ell_t \cdot sin(\theta_t)) \quad (8)$$

Therefore, the particle filter prediction function can be written as:

$$X_t = \begin{pmatrix} x_{t-1} + \ell_t \cdot cos(\theta_t) \\ y_{t-1} + \ell_t \cdot sin(\theta_t) \\ \theta_t \\ \ell_t \\ z_t = f(x_t, y_t) \end{pmatrix} \quad (9)$$

Both θ_t and ℓ_t values are calculated by PDR methods, whereas z_t is a discrete random variable that identifies the

zone where the particle is located at time t. State vector X_t^i of each particle is updated from the particles at the previous time interval X_{t-1}^i based on Equation (9). Thus, the new set P_t is calculated from P_{t-1}. Particles are allowed to move only through non-restricted areas, (e.g., movement through walls is not allowed).

After updating particles using Equation (9), the associated weight w_t^i of the propagated particles must be corrected. The associated weight update is based on the likelihood of the observations conditioned on each particle state $P(q_t \mid X_t^i)$ at time t. The observation vector is defined by the estimated ranges to different ANs and the estimated zone information. Thus, the observation vector at time t is defined as $q_t = [\hat{d}_t, \hat{s}_t]$, where $\hat{d}_t$ contains ranges to different ANs and $\hat{s}_t$ contains the observations related to the predicted zone.

Since the ranging method (i.e., the method to estimate ranges) and the zone prediction method are different, we assume that range and zone prediction information are independent of each other. Therefore, the probability $P(q_t \mid X_t^i)$ can be determined as follows:

$$P(q_t \mid X_t^i) = P(d_t \mid X_t^i) \cdot P(s_t \mid X_t^i) \tag{10}$$

Hereafter, we refer to $P(d_t \mid X_t^i)$ as the ranging likelihood, and $P(s_t \mid X_t^i)$ as the zone likelihood. The associated weight w_t^i of each particle is given by the ranging and zone prediction information. The particle at the absolute position (x_t, y_t) with low probability to observe d_t^j will be assigned a small ranging likelihood. Particles positioned at zones with low probability of observing s_t will be assigned small zone likelihood values.

Since ANs are programmed to operate independently, we can assume that the ranges to different ANs are independent from each other. Therefore, the ranging likelihood can be defined as follows:

$$P(d_t \mid X_t^i) = \prod_{j=1}^{M} P(\hat{d}_{j,t} \mid X_t^i), \tag{11}$$

where $\hat{d}_{j,t}$ is the measured distance to AN j at time t. Hereafter, $P(\hat{d}_{j,t} \mid X_t^i)$ will be referred as the individual ranging likelihood, which can be further written as:

$$P(\hat{d}_{j,t} \mid X_t^i) = \frac{1}{\sigma_j \sqrt{2\pi}} \exp^{-\frac{[\hat{d}_{j,t} - \sqrt{(x^i - x_j)^2 + (y^i - y_j)^2}]^2}{2\sigma_j^2}}, \tag{12}$$

where (x_j, y_j) are the coordinates of the jth ranging AN.

Zone likelihood refers to the probability of observing s_t in the current particle state X_t^i. Therefore, $P(s_t \mid X_t^i)$ can be written as:

$$P(\hat{s}_t \mid X_t^i) = \frac{P(X_t^i \mid \hat{s}_t) \cdot P(\hat{s}_t)}{P(X_t^i)}, \tag{13}$$

where $\hat{s}_t$ is the zone related set of observations at time t. Since the $P(X_t^i)$ and $P(\hat{s}_t)$ are constant, $P(s_t \mid X_t^i)$ depends only on $P(X_t^i \mid \hat{s}_t)$. Therefore, $P(s_t \mid X_t^i) \propto P(X_t^i \mid \hat{s}_t)$. Applying Equation 8, $P(X_t^i \mid \hat{s}_t)$ can be written as follows:

$$P(z_t^i \mid \hat{s}_t) = \frac{P(\hat{s}_t \mid z_t^i) \cdot P(z_t^i)}{P(z_t^i)} \tag{14}$$

Since O is the set of observations related to the zone prediction process (see Equation 4), we can define $\hat{s}_t$ as an element of O ($\hat{s}_t \in O$). Therefore, $P(z_t^i \mid \hat{s}_t)$ is computed by zone prediction method.

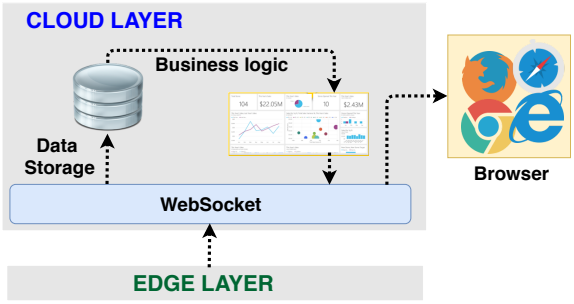


Figure 4: Cloud layer architecture

Algorithm 1 Data Fusion

1: Spread particles:
$x_0^i = q(X_0), i = 1, ..., N$;
$W_0^i = 1/N$;
2: Update X_t based on Equation 9:
3: Calculate the ranging likelihood:
$P(\hat{d}_{j,t} \mid x_t^i) = \frac{1}{\sigma_j \sqrt{2\pi}} \exp^{-\frac{[\hat{d}_{j,t} - \sqrt{(x^i - x_j)^2 + (y^i - y_j)^2}]^2}{2\sigma_j^2}}$;
4: Calculate the zone likelihood:
$P(s_t \mid X_t^i) = \frac{P(X_t^i \mid \hat{s}_t) \cdot P(\hat{s}_t)}{P(X_t^i)}$;
5: Compute unnormalized weights:
$\hat{w}_t^i = P(X_t^i \mid \hat{s}_t) \cdot \prod_{j=1}^{M} P(\hat{d}_{j,t} \mid X_t^i)$
6: Normalize weights: $w_t^i = \frac{\hat{w}_t^i}{\sum_{n=1}^{N} \hat{w}_n^i}$;
7: Resample the particles;
8: Compute the estimated state:$X_t = \sum_{i=1}^{N} w_t^i \cdot x_t^i$;
9: Go to step 2 for next iteration;

C. Cloud Layer

The Cloud layer is responsible for the storage of historical localization information. This information is related to users and the localization process along multiple areas of interest. The information is stored in a structured data base in the Cloud server. Thus, the Cloud layer enables high-order queries over the historical localization information to provide predictive analysis and business control. Therefore, allowing data collection from multiple scenarios and mobile devices (i.e. clients) and making this data accessible anywhere in the world are the main advantages of the Cloud layer.

Client devices gather data from on-site, then they pass this data to the Edge layer for processing (i.e., localization). Processed data is then passed to the Cloud layer, which is typically in a different geographical location. Thus, the cloud layer benefits from client devices by receiving their

ISBN 978-3-903176-13-3

Table I: Fog-edge localization components

Layer	Component	Specifications
Cloud	**Cloud Server**	**Model:** HP EliteBook **CPU:** 2.30 GHz Intel Core i5-5300U **OS:** Windows 10 Enterprise **RAM:** 8 GB
Edge	**Edge Server**	**Model:** HP EliteBook **CPU:** 2.30 GHz Intel Core i5-5300U **OS:** Windows 10 Enterprise **RAM:** 8 GB **UWB Interface:** Sequitur Pi (InGPS litle)
Client	**Client Device**	**Model:** Raspberry Pi Model B **CPU:** Quad Core 1.2GHz **OS:** Raspbian 4.14 **WLAN:**WiFi b/g/n **UWB-IMU:** Sequitur Pi (InGPS litle)
	Wi-Fi-AN	**Model:** D-Link (D-635 and DAP-2553)
	UWB-AN	**Model:** Raspberry Pi Model B **CPU:** Quad Core 1.2GHz **OS:** Raspbian 4.14 **WLAN:**WiFi b/g/n **UWB:** Sequitur Pi (InGPS litle)

data through the other layers. Figure 4 shows the internal architecture of the Cloud layer.

IV. Implementation

Our MEC-based tracking system comprises five main components: a client mobile device (MD), some commercial Wi-Fi access points (Wi-Fi-AN), some UWB anchor nodes (UWB-AN), a Edge server (ES), and a cloud server (CS). The MD is the device to be localized. Positions of UWB-AN are chosen to provide the maximum coverage inside the area of interest. Table I summarizes the specification of each component. Communication between the Cloud and the Edge layer was implemented by using WebSocket technology. WebSocket is a computer communication protocol, which allows two or more connected devices to communicate with one another in both directions through a single TCP connection. It is supported by many platforms. WebSocket technology uses the HTTP upgrade header to change from the HTTP to the WebSocket protocol [5]. Thus, Tornado [24] was used to provide web server and WebSocket server in the cloud layer.

The system requires information related with zones' distribution and physical connections among zones (i.e., zone transition information). Therefore, it is necessary to have coarse-grained information about the area of interest. We define 14 zones in our environment. Each zone is a wall separated area (i.e., rooms, corridor).

In the zone prediction method, we setup three conceptually different machine learning algorithms (KStar, Multilayer Perceptron (MLP) and CART). Python Scikit-learn library [12] was used to implement the individual machine learning algorithms. To build the zone fingerprinting database, we collected 9800 fingerprint instances, approximately 700 in each zone. The structure of a fingerprint instance consists of Wi-Fi and UWB RSS readings. Zone fingerprint database entries were collected equally distributed over the whole area in each zone. The data collection rate is only constrained by computational capabilities of the Wi-Fi sensor of the MT. Thus, in our experiments every fingerprinting entry was collected at a rate of 3 entries/second. Since our approach does not need to predefine any survey point, the time needed to build the fingerprinting database is proportional to the number of collected instances multiplied by the instance collection rate. Since hyperparameters have significant impact on the performance of the machine learning algorithm, we use a nested cross validation technique to adjust them [18]. Finally, to reduce the negative impact of environmental changes and different hardware, we use differential Wi-Fi RSS instead of absolute raw values.

Table II: Tracking performance summary.

Scenario	Tracking system	Mean error	S.D	90% Acc.
1	InTrack	1.53m	1.09m	3.1m
	Commercial	1.69m	1.68m	3.1m
2	InTrack	0.44m	0.14m	0.6m
	Commercial	0.47m	0.28m	0.8m
	Edge-based	1.42m	0.61m	2.1m

V. Performance Evaluation

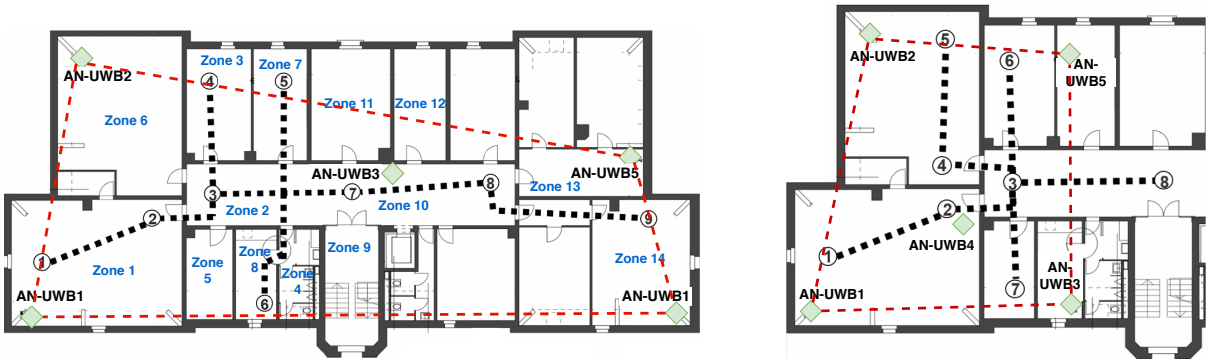
We tested our system in two office-like indoor scenarios along complex trajectories. Experiments were conducted in the third floor of the building of the Institute of Computer Science at the University of Bern. In the first scenario, we deployed 5 UWB-ANs in an area of $702m^2$ ($39m$ x $18m$). In the second scenario we increased the UWB-AN density by deploying 5 UWB-ANs in an area of $342m^2$ ($19m$ x $18m$). To determine the localization error, some checking points are defined along each trajectory. In the trajectory of scenario 1, we defined 9 checking points, whereas 8 checking points are defined along the trajectory in scenario 2. Distribution of the checking points can be seen in Figures 5a and 5b. Experiments were repeated five times. Therefore, 45 checking points were analyzed in scenario 1 and 40 checking points in scenario 2. The localization error is computed by the Euclidian distance between the position calculated by the system and the ground truth position in each checking point. We compare our indoor tracking (InTrack) approach to the commercial solution Sequitur InGPS Lite [25]. Hereafter, we will refer to Sequitur InGPS as the commercial approach. Additionally, we compared InTrack with an client-based tracking solution [4], which is referred to as client-based tracking.

A. Experiment Results

Figure 6 shows the CDF of localization error for the tracking systems. InTrack and the commercial tracking system show similar localization performance. However, as shown in Figure 6, InTrack approach achieves higher accuracy and more stable performance compared to the commercial system. Table II summarizes the average of tracking errors, standard deviation and 90% accuracy.

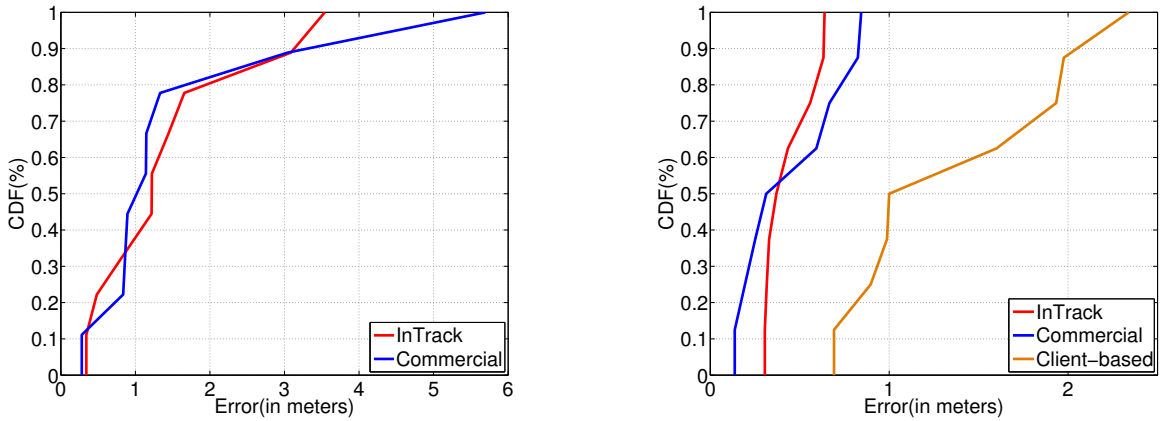
In scenario 1, both InTrack and the commercial system achieve around $3.1m$ for 90% accuracy. The mean error and standard deviation are also similar for both systems in this scenario. However, InTrack overcomes the commercial system by $0.59m$, and $0.16m$ by considering standard deviation and mean tracking error respectively.

ISBN 978-3-903176-13-3



(a) Scenario 1 (b) Scenario 2

Figure 5: Scenarios for experiments. Trajectories (black dotted lines), UWB-AN Bounding box (area inside the red dotted lines), UWB-AN distribution (Diamond green points), checking points (numerated circles), and zone definition.



(a) Empirical CDF of tracking error in Scenario 1. (b) Empirical CDF of tracking error in Scenario 2.

Figure 6: Tracking error performance in scenario 1 and scenario 2.

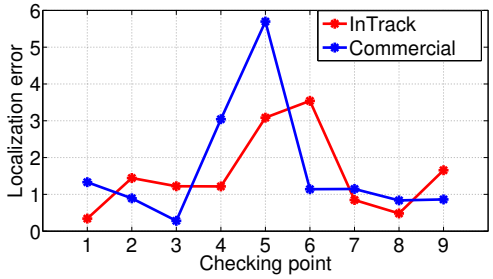


Figure 7: Localization errors along trajectory in Scenario 1

Although localization approaches based on UWB ranging present sub-meter localization errors, we observe that the tracking errors of InTrack and the commercial systems are higher than $1m$ in some checking points in the trajectory of scenario 1. Figure 7 presents the observed mean tracking error in each checking point in scenario 1. To find the reasons for this low tracking performance, it is necessary to consider the underlying implementation of the systems. Thus, two main issues are identified:

- Trajectories outside the UWB-AN area defined by linear straight connections among UWB-ANs. This area is called the bounding box. Figure 5 shows the UWB-AN bounding boxes in scenario 1 and 2.
- Failures in the UWB-based communication among UWB-AN and the MD.

Errors in the ranging process lead to decreased localization performance. Locating a position placed outside the bounding box is prone to significant ranging errors. In scenario 1, checking points 1, 4, 5, 6 are located either outside or close to the border of the bounding box. These locations present higher localization error than locations inside the UWB-AN bounding box. It can be seen in Figure 7 that localization errors at these points are higher than $1m$.

Since the localization algorithms are running on the Edge layer, InTrack's performance depends on the quality of links between client devices and Edge servers. Both, InTrack and the commercial system, use UWB-based communication links to transmit data. Therefore, a low quality UWB link connection leads to increased tracking errors for InTrack and the commercial system. This behaviour is evident in scenarios with low UWB-AN density (i.e., low amount of UWB-AN in a big area of interest), such as scenario 1, where we observe data UWB communication problems due to long transmission distances.

ISBN 978-3-903176-13-3

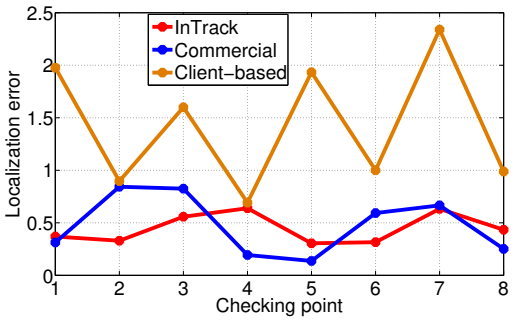


Figure 8: Localization errors along trajectory in scenario 2

In scenario 2, InTrack and the commercial system show similar performance. However, as it can be seen in Figure 6b and Table II, InTrack overcomes the commercial and the client-based systems. To tackle the UWB communication issues observed in scenario 1, we conducted the experiment in scenario 2 by increasing the UWB-AN density. We increased the UWB-AN density of 5 UWB-AN in an area of $702m^2$ to 5 UWB-AN in an area of $342m^2$. Thus, the quality in the UWB communication link was improved. As shown in Figure 8, the localization error in each checking point is significantly reduced compared to scenario 1. It proves the importance of the UWB-AN density in the localization performance.

Finally, we can observe in Figure 6b and Table II that InTrack overcomes the client-based system by 69%, 77%, and 71% considering mean tracking error, standard deviation and 90% accuracy respectively. It is due to heavy computations (i.e., localization algorithms) are offloaded from the client device to one Edge server. Moreover, reducing the device-server distance reduce the data transmission time. This allows real-time localization performance. Thus, it proves that in real-time localization applications, processing time influences the accuracy performance of the localization system.

VI. Conclusions

This work exploits a MEC architecture to implement an accurate real-time indoor tracking system. Our approach fuses UWB radio signals, machine learning for zone prediction information, inertial sensors and physical information of the environment, to achieve high localization accuracy in complex indoor scenarios. Experiments results show that our approach can achieve an average tracking error of $0.44m$ and 90% accuracy is $0.6m$. It outperforms some commercial products and client-based tracking systems. Thus, by bringing cloud computing capabilities to the network edge, our MEC-based approach is more accurate and robust than traditional and commercial indoor localization methods.

References

[1] H. Abdelnasser, R. Mohamed, A. Elgohary, M. Alzantot, H. Wang, S. Sen, R. Choudhury, and M. Youssef, "Semanticslam: Using environment landmarks for unsupervised indoor localization." *IEEE Transactions on Mobile Computing*, vol. 15, pp. 1770–1782, 2016.

[2] E. Ahmed and M. H. Rehmani, "Mobile edge computing: Opportunities, solutions, and challenges," *Future Generation Computer Systems*, vol. 70, pp. 59 – 63, 2017.

[3] J. Carrera, Z. Li, Z. Zhao, and T. Braun, "A real-time indoor tracking system in smartphones." *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, November 2016.

[4] J. Carrera, Z. Zhao, T. Braun, Z. Li, and A. Neto, "A real-time robust indoor tracking system in smartphones," *Elsevier Computer Communications*, vol. 117, pp. 104–115, February 2018.

[5] I. Fette and A. Melnikov, "The websocket protocol," *Internet Engineering Task Force (IETF)*, 2011.

[6] G. Forney, "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, pp. 268–278, March 1973.

[7] D. Fox, W. Burgard, and S. Thrun, "Markov localization for mobile robots in dynamic environments," *J. Artif. Int. Res.*, vol. 11, no. 1, pp. 391–427, Jul. 1999.

[8] S. Gezici, Z. Tian, G. Giannakis, H. Kobayashi, A. Molisch, H. Poor, and Z. Sahinoglu, "Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks," *IEEE Signal Processing Magazine.*, vol. 22, pp. 70–84, July 2005.

[9] S. He and G. Chan, "Wi-fi fingerprint-based indoor positioning:recent advances and comparisons," *IEEE Communications Survey Tutorials*, 2016.

[10] J. Hightower, G. Borriello, and R. Want, "Spoton: An indoor 3d location sensing technology based on rf signal strength," *UW CSE Technical Report*, Feb 2000.

[11] F. Hong, H. Chu, L. Wang, Y. Feng, and Z. Guo, "Pocket mattering: Indoor pedestrian tracking with commercial smartphone," *International Conference on Indoor Positioning and Indoor Navigation*, 2012.

[12] INRIA. (2018) Scikit-learn. machine learning in python. [Online]. Available: http://scikit-learn.org/stable/

[13] Z. Li, T. Braun, X. Zhao, and Z. Zhao, "A narrow-band indoor positioning system by fusing time and received signal strength via ensemble learning," *IEEE Access 2018*, pp. 9936–9950, 2018.

[14] Z. Li, D. Dimitrova, and T. Braun, "A passive wifi source localization system based on fine-grined power-based trilateration." *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2015.

[15] ——, "A time-based passive source localization system for narrow-band signal," *The IEEE International Conference on Communications (ICC)*, vol. 39, pp. 4599–4605, June 2015.

[16] J. Liu, L. P. R. Chen, R. Guinness, and H. Kuusniemi, "A hybrid smartphone indoor positioning solution for mobile lbs," *Sensors*, 2012.

[17] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1628–1656, thirdquarter 2017.

[18] K.-R. Mller, M. Krauledat, G. Dornhege, G. Curio, and B. Blankertz, "Machine learning techniques for brain-computer interfaces." *BIOMEDICAL ENGINEERING*, pp. 11–22, 2004.

[19] K. N. and S. Kamijo, "Pedestrian dead reckoning for mobile phones through walking and running mode recognition," *The 16th International IEEE Conference on Intelligent Transportation Systems*, Oct 2013.

[20] M. Ridolfi, S. Van de Velde, H. Steendam, and E. De Poorter, "Analysis of the scalability of uwb indoor localization solutions for high user densities," *Sensors*, vol. 18, no. 6, 2018.

[21] B. Schilit, N. Adams, and R. Want. (2018) Context-aware computing applications.

[22] J. Seitz, T. Vaupel, S. Meyer, B. Gutirrez, and J. Thielecke, "A hidden markov model for pedestrian navigation." *in Proceedings of the positioning navigation and communication*, pp. 120–127, march 2012.

[23] Sewio. (2018) Uwb: Technology-two way ranging. [Online]. Available: https://www.sewio.net/uwb-technology/two-way-ranging/

[24] T. Tornado-Authors. (2018) Tornado: Running and deploying. [Online]. Available: http://www.tornadoweb.org/ en/stable/guide/running.html

[25] UNISET, "Uniset company indoor tracking solution." [Online]. Available: http://www.unisetcompany.com/sequitur-family-en/

[26] M. Wallbaum and O. Spaniol, "Indoor positioning using wireless local area networks," *in Proceedings of the IEEE John Vincent Atanasoff International Symposium on Modern Computing.*, pp. 17–26, Oct 2006.

[27] N. Yepeng, L. Jianbo, A. Liu, and Y. Bai, "An indoor pedestrian positioning method using hmm with a fuzzy pattern recognition algorithm in a wlan fingerprint system," *Sensors*, Sep 2016.

[28] M. Youseff and A. Agrawala, "The horus wlan location determination system," *Proceedings of the International Conference on Mobile Systems, Applications, and Services (Mobisys05)*, pp. 205–218, 2005.