BluePIL: a Bluetooth-based PassIve Localization Method

Bruno Rodrigues, Cyrill Halter, Muriel Franco, Eder J. Scheid, Christian Killer, Burkhard Stiller
Communication Systems Group CSG, Department of Informatics IfI, University of Zürich UZH
Binzmühlestrasse 14, CH—8050 Zürich, Switzerland
E-mail: [rodrigues,franco,scheid,killer,stiller]@ifi.uzh.ch, cyrill.halter@uzh.ch

***Abstract*—Indoor location services are of great relevance for several applications ranging from mobile marketing and navigation to healthcare technologies. Wireless technologies provide a method of gauging performance indicators through the capturing of signals emitted by mobile devices. Since the Bluetooth technology has seen a steady growth over the years, providing a viable alternative to 802.11 protocols, this paper presents *BluePIL*, a fully passive system for Bluetooth device identification and localization designed as a distributed streaming architecture delivering results in near-real-time. This approach enables a contact and activity tracing with different data privacy dimensions.**

***BluePIL* relies on parts of the Bluetooth address for device identification and a modified multi-lateration algorithm using a path loss model for device localization. The results were obtained using Ubertooth Bluetooth sensors and low-cost hardware (ASUS Tinkerboard devices), showing that the approach achieves localization accuracies of $1\,\mathrm{m}$ to $1.4\,\mathrm{m}$ within a space of around $12\,\mathrm{m}^2$ and $25\,\mathrm{m}^2$, respectively.**

I. Introduction

Measuring the interest in a product or service in a public space, such as a trade show or a sales floor, is fundamentally important for the evaluation of the marketing strategies of businesses. Thus, the increasing adoption of portable devices allows indoor positioning systems to be implemented [19]. The analysis of such signals emitted by portable devices, such as smartphones, laptops, and tablets, enables the extraction of positional data providing a method of gauging Key Performance Indicators (KPI) through their passive or active capture [36].

Localization enables advanced security measures, such as emergency routes, which can be planned dynamically by analyzing the human crowd behavior. The efficient planning of marketing strategies of business events or campaigns [31] becomes possible due to an improved planning of public spaces [36] or especially in the context of the SARS-CoV-2 outbreak, a proximity detection, contact, and activity tracking relying on Bluetooth (BT) technology was designed [11], [13].

Typical tracking approaches rely on Wi-Fi Received Signal Strength Indication (RSSI), such as 802.11g (2.4 GHz) or 802.11a (5 GHz), and on one hand driven by Wi-Fi Media Access Control (MAC) addresses the calculation of distances and an identification of individual devices, respectively [29], [19], [6], becomes possible. On the other hand, to prevent a tracking of devices and strengthen user privacy, MAC-randomization strategies [30] have been proposed. It prevents the building of a history of devices based on their MAC addresses. Thus, strategies within Wi-Fi make the step of a device identification complex. The fact that MAC-randomization does not exist in classic BT [4], spawned the hypothesis that BT may provide a viable alternative to Wi-Fi. While a fair amount of research exists for the areas of unique identification and localization of BT devices [1], [34], [17], [20], [31], most approaches up to now do not allow for a system to be entirely passive, *i.e.*, not requiring any knowledge of and collaboration with the target devices. This is the case especially for the area of device localization, where approaches based on fingerprinting and those based on a path loss model require prior calibration using a target device.

By passively measuring at a given point in time the RSSI it is possible to determine the uniqueness of devices tracked with a certain likelihood. This is done by comparing the current RSSI with a radio-propagation map, which contains expected RSSI values [26] with a localization via the smallest error distance. Most approaches estimate the device's position in space based on measurements of at least three observing points (*i.e.*, tri-lateration), whereas, for multiple observing points, a multi-lateration technique is employed [1]. For example, a Log-Distance Path Loss model can be used (*cf.* Section II-D) to calculate radio signal decay over distance [2] and to estimate the position of BT devices by modeling the finding that the decay of a signal over distance is approximated by a logarithmic function.

BluePIL introduces a distributed streaming architecture that uses a node-sink topology to deliver near-real-time positioning estimates of BT devices. It defines a data processing pipeline accomplishing identification and localization tasks through passively captured BT Basic Rate/Enhanced Data Rate (BT-BR/EDR) packets in several steps, *i.e.*, device identification, signal strength filtering, signal strength merging, the localization algorithm, and location filtering. *BluePIL* is based on a Python implementation running on low-cost hardware (*e.g.*, Ubertooth Devices [16] and small computers, such Raspberry Pi's or ASUS Tinkerboard), and requires a minimal setup, since configurations are handled automatically.

The remainder of this paper is organized as follows. Section II overviews fundamentals. While Section III describes *BluePIL*'s design and implementation, Section IV details the evaluation. Related work is described in Section V followed by Section VI summarizing the paper's final considerations.

978-3-903176-32-4

II. Background

Basic technical background covers BT, the BT sniffer, multi-lateration, and the log-distance path loss modell as follows.

A. Bluetooth (BT) Packet Relevant Fields

Bluetooth (BT) is defined as a short-range communications system intended to replace the cable(s) connecting portable and/or fixed electronic devices [7]. Operating in the unlicensed 2.4 GHz Industrial, Scientific, and Medical (ISM) frequency band, BT devices typically transmit up to a distance of $10\,\mathrm{m}$ to serve this purpose.

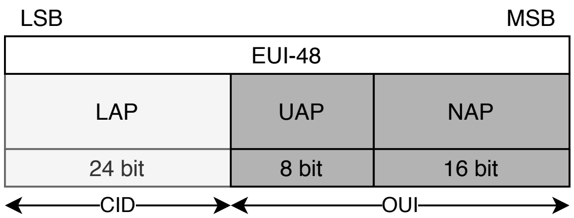

Fig. 1: The Composition of BT Addresses [7]

A BT packet is composed of a 24-bit Company ID (CID) and a 24-bit Organizationally Unique ID (OUI) (*cf.* Figure 1). Every BT device has a unique BT address, which is constructed as a 48-bit Extended Unique Identifier (EUI-48) according to the IEEE Standard for Local and Metropolitan Area Networks [22]. The CID is vendor-assigned, whereas the OUI has to be obtained from the IEEE Registrations Authority and is assigned to individual organizations, manufacturers, or vendors of BT technology. For BT networking, parts of the BT address are differentiated by the Lower Address Part (LAP), corresponding to the CID, and the 16-bit non-significant and 8-bit upper address parts, forming the OUI.

B. Ubertooth BT Sniffer

The Project Ubertooth is the passive sensing device used for tracking BT packets. Ubertooth is a fully open-source hardware and software package for wireless development [16], suitable for experimentation with both BT Low Energy (BTLE) and classic BT. Ubertooth allows access to lower layers of Bluetooth protocols, which are normally hidden in off-the-shelf Bluetooth modules, being available at low cost.

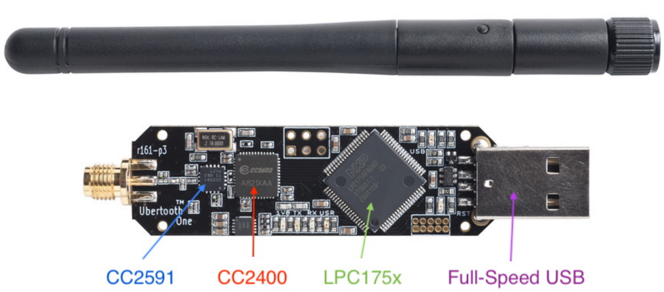

Fig. 2: The Ubertooth One Sniffer [18]

BT is a spread spectrum technology, meaning that it moves over a wide range of frequencies during the transmission of data. Most BT monitoring hardware, therefore, implements an array of transceivers to observe all channels used by the BT protocol simultaneously. Ubertooth One used in this paper (*cf.* Figure 2), however, only uses a single transceiver, opting instead to try and hop along with the hopping pattern of ongoing BT connections in order to eavesdrop data being transmitted [16]. Consequently, only BT classic is fully supported by Ubertooth One.

C. Multi-lateration

Multi-lateration defines the process of geometrically estimating an object's position in space through distance measures to at least three points. For the case where exactly three points are used, it is called tri-lateration. Mathematically, this corresponds to solving the following non-linear system, with (x_i, y_i, z_i) the position of the i-th point, (x, y, z) the position of the object, and d_i the distance of the object to the i-th point. For planar problems, this can be simplified further, leading to the following system in two variables:

$$
\begin{aligned}
(x - x_1)^2 + (y - y_1)^2 &= d_1^2 \\
(x - x_2)^2 + (y - y_2)^2 &= d_2^2 \\
(x - x_3)^2 + (y - y_3)^2 &= d_3^2
\end{aligned} \tag{1}
$$

This system is then often linearized by subtracting the last equation from the other two, leading to the following determined linear system of equations [25]. The solution is reached for any higher order multilateration problems similarly. In practice, distance measures are often imperfect and the calculation of a solution for Equation 1 using a non-linear optimization lead to better results.

D. The Log-Distance Path Loss Model

The Log-Distance Path Loss Model is a popular model for radio signal decay over distance [2]. It models the finding that the decay of a signal over distance can be approximated by a logarithmic function. With $RSS(d)$ the received signal strength at distance d, d_0 a reference distance, n the path-loss coefficient, and $\mathcal{X}_\sigma$ a zero-mean Gaussian random variable, it can be defined as follows:

$$RSS(d) = RSS(d_0) - 10n \log\left(\frac{d}{d_0}\right) + \mathcal{X}_\sigma \tag{2}$$

In practice, the reference distance d_0 is often set to $1\,\mathrm{m}$ and noise is ignored for the calculation, simplifying the model even further. With RSS_C, the received signal strength at $1\,\mathrm{m}$, it can then be expressed as follows:

$$RSS(d) = RSS_C - 10n \log(d) \tag{3}$$

RSS_C depends on each individual device and has to be calibrated. The path-loss coefficient n is a factor depending on the environment. For free-space, it is often chosen at $n = 2$.

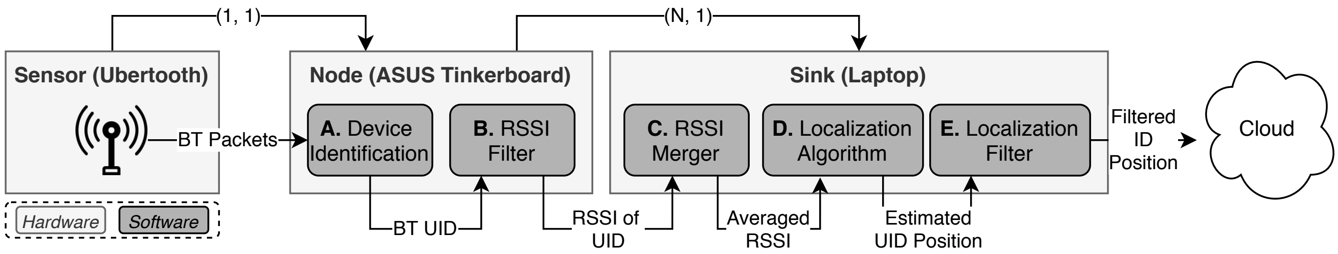

Fig. 3: *BluePIL*'s Data Stream Pipeline

III. BLUEPIL DESIGN

Figure 3 describe the *BluePIL* data pipeline, designed toward a flexible deployment and operation of individual hardware components (*i.e.*, sensors, nodes, and sink). In general, *BluePIL* is set up as a streaming data processing pipeline. It allows for physical or virtual logical processing entities in a system deployed to be configured in different ways. *BluePIL* is based on a distributed node-sink setup (*i.e.*, , a system where many physical nodes send data to a single physical sink), which is responsible for forwarding data to an entity, where it can be stored or processed (*e.g.*, the cloud). Computations are performed as early as possible to avoid bottlenecks downstream and to reduce the amount of data forwarded by the sink.

A. Device Identification

The identification step allows the system to profit from the fact that BTBR/EDR lacks any sort of MAC randomization and avoids building a complex systems for fingerprinting by using a unique identifier that is already available: the BT address. The BT address consists of the LAP, the NAP, and the UAP. While LAP is not not globally unique, it is sufficient to identify devices under certain circumstances [32], [7], [10]. As *BluePIL*'s goal is to identify mobile devices, such as smartphones or tablets, it is important to account that the five biggest smartphone manufacturers share 72% of the smartphone market among them (as of the first quarter of 2020 [23]). Considering a the probability of encountering a LAP collision as $P(col)$, b the probability of encountering a different OUI as $P(dO)$, and c the probability of encountering the same CID $P(sC)$, it can be stated that encountering the same LAP twice means that their OUI is different, since BT addresses are globally unique. Thus, the probability of a LAP collision is defined in terms of $P(col) = P(sC) * P(dO)$. Even without any further optimization, this gives a fairly small probability of around $P(col) \approx 5.96\mathrm{e}^{-8}$.

Assuming that the 20 largest smartphone manufacturers share (almost) the entire market, $P(dO) \approx \frac{19}{20}$ and $P(col) \approx 5.66\mathrm{e}^{-8}$ holds. Thus, if in a certain environment the system would register 10,000 different BT addresses, for example, the probability for a LAP collision would still only be at $1 - (1 - P(col))^{10{,}000} \approx 0.06\%$. This is sufficient for the potential use cases of *BluePIL* and, therefore, the LAP as computed in [32] is used as a quasi-unique identifier. This identifier is also suitable to identify individuals carrying a BT device, since all devices in a piconet use the master device's LAP for the construction of the access code, *i.e.*, two connected devices, such as a smartphone and a pair of BT headphones, do not produce two separate identifiers.

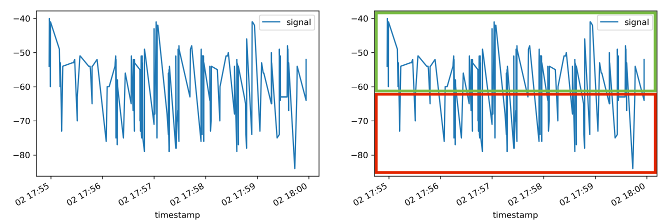

Fig. 4: (Left) RSSI Measurements for a Static Device Over a Period of 5 mins Using an *Ubertooth* Sensor, and (Right) RSSI Values Potentially Useful (Top Box/Green) and Those Probably Caused by Multi-path Fading (Lower Box/Red)

B. RSSI Filter

RSSI values obtained from the sensor are pre-processed as a first step. Figure 4 shows an example of RSSI measurements for a static device over a period of five minutes. This example illustrates the large amounts of high-frequency, high-variance noise that must be taken into account, when working with this type of data.

Existing research suggests that noisy parts of RSSI values correspond to the lower set of values in the RSSI distribution. [9], defines a unidirectional outlier filter to be effective. It eliminates values that deviate from the maximum value by a certain degree. [15] determines the maximum to be the most effective filter for a pre-processing of RSSI values for localization purposes. The signal that travels along the line of sight, *i.e.*, that is not influenced by multi-path fading, covers the smallest distance and, thus, arrives at the sensor with the highest strength. A combination of a maximum filter followed by a mean filter is, thus, used in *BluePIL*. To account for the streaming paradigm, these filters work in a purely retrospective way, *i.e.*, work with a local subset of the data that only uses values from the past. To this end, a rolling time window is implemented only containing values from the interval $[t_c - \Delta t, t_c]$, with t_c being the current time and Δt the window size, which are determined by the update frequency of the sensor and the expected variance of the data.

C. RSSI Merger

To compute a location from pre-processed RSSI values, a strategy has to be determined to merge data streams. This part of the processing pipeline deals with two problems: First, update cycles may differ between sensors, *i.e.*, it cannot be assumed that all sensors will have the same amount of data available at a specific point in time. Second, data delivered by sensors may be fairly sparse. This may be due to the quality and capabilities of sensors themselves, due to environmental factors or due to characteristics of the target device. The goal of this step is to handle these problems, taking into account the streaming paradigm implemented for *BluePIL*.

Interpolation is able to help with both the problem of differing update cycles and sparsity of data. In general, *BluePIL* builds upon the assumption that update cycles of individual sensors are short enough to legitimize the linear interpolation between two data points as a valid estimation of the true state of the system. To enable the inference of RSSI values at a certain point in time through interpolation, measured values must be available preceding and succeeding said point. The signal strength merger will, therefore, delay the emission of a value from a sensor until data is available from all other sensors before and after the point in time, where the value was received.

D. Localization Algorithm

Since *BluePIL* is completely passive, information is limited to the signal strengths detected on an external sensor from any ongoing BT connection. This rules out any fingerprinting-based approaches, since they require the creation of a radio map with the devices involved beforehand, leaving the path-loss-model-based approaches. A path loss model requires parameters n and RSS_C to be defined beforehand in order to calculate a distance from a signal strength value. Based on existing research [12], [33], [9], it is viable to set n to a fixed value based on the environment *BluePIL* is working in, as long as this does not change drastically. n is dependent on environmental factors and does not vary between devices. The issue with RSS_C, however, is not so easy to solve. Transaction strengths may vary between BT devices. Due to adaptive power control, they may even change over time for the same device [7]. The choice of a fixed value for RSS_C is, therefore, not an option.

To approach this, a method was designed that dynamically estimates the location of a BT device, and the necessary channel parameters of the path loss model. With k the number of sensors, (x_i, y_i) the location of the i-th sensor, (x, y) the location of the target device and d_i the distance between the i-th sensor and the target device, the following multilateration problem is defined:

$$(x - x_i)^2 + (y - y_i)^2 = d_i^2, i \in 1..k \tag{4}$$

Since it is not possible to compute d_i from the path loss model directly due to the issues mentioned before, the path loss model equation for distance is solved and then combined with the multilateration problem above:

$$\begin{aligned} RSS(d) &= RSS_C - 10n\log(d) \\ d &= 10^{\frac{RSS_C - RSS(d)}{10n}} \end{aligned} \tag{5}$$

$$(x - x_i)^2 + (y - y_i)^2 = 10^{\frac{RSS_C - RSS(d_i)}{5n}}, i \in 1..k \tag{6}$$

A non-linear set of k minimizable equations is defined in terms of x, y and RSS_C, with RSS_C the calibration signal strength $1\,\mathrm{m}$ away from the target device and RSS_i the RSS measurement for the i-th sensor. This corresponds to a problem that can be solved using a non-linear optimization algorithm. *BluePIL* uses *Levenberg-Marquardt (LM)*, an iterative minimizer that can be described as a combination of the *Steepest Descent* and the *Gauss-Newton* methods [27] [28]. With $\mathbf{p} = (x, y, RSS_C)$, the parameter vector, the vector $\mathbf{p}^+$ is determined where $f_i(\mathbf{p}^+)$ is minimal for all i. LM works through a local linearization of the non-linear set of equations at a certain area of interest according to the statement $f(\mathbf{p} + \delta_p) \approx f(\mathbf{p}) + \mathbf{J}\delta_p$, where $\mathbf{J}$ is the *Jacobian* matrix. For *BluePIL*'s set of equations, the Jacobian matrix is defined as:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial RSS_C} \\ \vdots & \vdots & \vdots \\ \frac{\partial f_i}{\partial x} & \frac{\partial f_i}{\partial y} & \frac{\partial f_i}{\partial RSS_C} \\ \vdots & \vdots & \vdots \\ \frac{\partial f_k}{\partial x} & \frac{\partial f_k}{\partial y} & \frac{\partial f_k}{\partial RSS_C} \end{bmatrix} =$$

$$\begin{bmatrix} 2(x - x_1) & 2(y - y_1) & -\frac{log10}{5n} * 10^{\frac{RSS_C - RSS_1}{5n}} \\ \vdots & \vdots & \vdots \\ 2(x - x_i) & 2(y - y_i) & -\frac{log10}{5n} * 10^{\frac{RSS_C - RSS_i}{5n}} \\ \vdots & \vdots & \vdots \\ 2(x - x_k) & 2(y - y_k) & -\frac{log10}{5n} * 10^{\frac{RSS_C - RSS_k}{5n}} \end{bmatrix} \tag{7}$$

With the Jacobian defined, LM then iteratively adjusts $\mathbf{p}$ by δ_p in a descending direction until convergence is reached. To ensure that this convergence is to a global minimum, an appropriate starting point $\mathbf{p}_0$ has to be defined. For the problem posed, it is important that the minimum is found in the area of overlap of all sensors. To guaratee this, $\mathbf{p}_0$ is chosen at the center of the area spanned by the sensors and with a value for RSS_C that approximates the range of values that are expected from the relevant device class.

$$\mathbf{p}_0 = (\frac{\sum_i x_i}{k}, \frac{\sum_i y_i}{k}, -30) \tag{8}$$

Due to the limited resources available, the problem is generally limited to four sensors, *i.e.* $k = 4$. The localization algorithm may, therefore, also be referred to as a *quadlateration* algorithm in the following.

E. Localization Filter

After having calculated a location in the previous step, the knowledge of the motion of a person carrying a BT device can be used to improve these results further. Kalman filters are a popular method for the improvement of positioning calculations and have been used in many path-loss-based localization approaches [9] [37] [25]. They combine models for the state of the system, the knowledge of previous observations, and models for the observation of states to estimate the most plausible state of a system captured through noisy observations.

To use a Kalman filter, it is necessary to define the following: the state transition model F_k, the observation model H_k, the process noise covariance Q_k, and the observation noise covariance R_k. *BluePIL* uses a simple kinematic model with (x_k, y_k) being current location's coordinates, and $(\dot{x_k}, \dot{y_k})$ the current velocity in x and y direction. The state vector is designed in a similar manner to [37]:

$$s_k = \begin{bmatrix} x_k \\ \dot{x_k} \\ y_k \\ \dot{y_k} \end{bmatrix} \tag{9}$$

With Δt_k being the time difference to the last state estimate s_{k-1}, the following state-transition matrix is defined:

$$F_k = \begin{bmatrix} 1 & \Delta t_k & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t_k \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{10}$$

This expresses a belief that the subject carrying a BT device will have moved in the direction gathered from the last measurement and that the velocity of said movement will not have changed abruptly. For *BluePIL*'s process noise covariance, it is used a discrete white noise as suggested in [24] and [3], under the assumption that the noise is a Wiener process, *i.e.*, is independent from previous time intervals and constant over a time interval. With the variance $\sigma_v^2 = 0.001$ [24], it is defined as follows:

$$Q_k = \begin{bmatrix} \frac{1}{4}\Delta t_k^4 & \frac{1}{2}\Delta t_k^3 & 0 & 0 \\ \frac{1}{2}\Delta t_k^3 & t_k^2 & 0 & 0 \\ 0 & 0 & \frac{1}{4}\Delta t_k^4 & \frac{1}{2}\Delta t_k^3 \\ 0 & 0 & \frac{1}{2}\Delta t_k^3 & t_k^2 \end{bmatrix} * \sigma_v^2 \tag{11}$$

Values obtained from the previous step in the pipeline are used as observations, *i.e.*, location estimates calculated through the modified multi-lateration method. Therefore, the following observation vector is used:

$$z_k = \begin{bmatrix} x_k \\ y_k \end{bmatrix} \tag{12}$$

Then, the following observation matrix is defined to express that the observation corresponds to the x and y coordinates of the state vector:

$$H_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{13}$$

Finally, the observation noise covariance matrix is defined. Those values used were determined experimentally and work well with sensors used for this setting, while for a different set of sensors, these values might have to be adjusted.

$$R_k = \begin{bmatrix} 0.3 & 0 \\ 0 & 0.3 \end{bmatrix} \tag{14}$$

Fig. 5: Evaluation Environments: (Left) Indoor and (Right) Outdoor

Using the Kalman filter allows for the improvement of values calculated in the previous step using the information contained in previous values and the knowledge of the system dynamics. It eliminates outliers and smooths these results simultaneously, using plausibility as a determining factor.

IV. BLUEPIL EVALUATION

Experiments in an indoor and an outdoor space were conducted to evaluate the effectiveness of *BluePIL*'s device localization method. The respective scenes are shown in Figure 5. The indoor experiment was performed in a room at that time being empty. This was ideal, since it allowed to keep the amount of signal interferences as low as possible. The outdoor experiment was done on a private terrace in a residential area of Zürich. This allowed to decrease the amount of signal interference from multi-path fading, since more space was available. A $4.2\,\mathrm{m} \times 2.9\,\mathrm{m}$ area was designated to perform the experiments. An Ubertooth One sensor was placed at each of the four corners of this space and connected to a MacBook Pro via a 2.5 m USB cable. Nine points were defined, where static measurements would be carried out.

A. Experiment 1 — Indoor

This experiment was then conducted via a *Nokia 7 Plus* smartphone being connected to a pair of *JBL Reflect Flow* Bluetooth headphones. Music was streamed over said connection throughout the experiment to generate traffic that could be captured passively. In order to keep the conditions as realistic as possible, the headphones were placed in a test subject's ears and the smartphone in their front right pant pocket. The test subject then stood for 5 min at each of the nine points indicated in Figure 6. The four Ubertooth One sensors were configured to record any packets that could be intercepted during that time interval.

Experiment 1 used a static version of the localization algorithm, which allowed for simplifying the merging of data sets from individual sensors. The streaming interpolation method used in the final processing pipeline could be omitted and the data could be merged using a static interpolation and re-sampling process. While this initial approach differs slightly from the final system, it does not invalidate results of this experiment as an evaluation of the device localization method. This experiment also included an evaluation of the filtering methods used in the *BluePIL* processing pipeline,

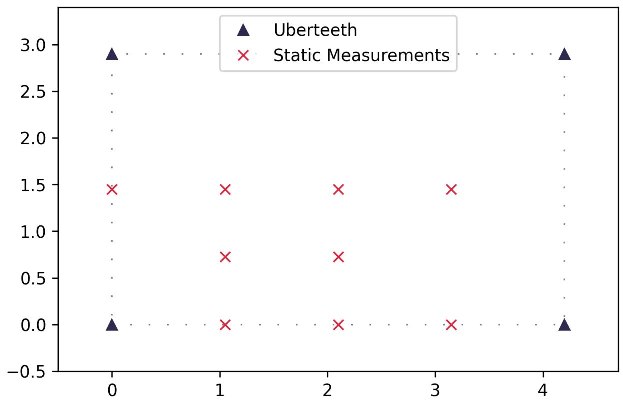

Fig. 6: Indoor Setup (x and y Axes in Meter)

TABLE I: Indoor Environment Experiment Results

True Point (m)	**Mean Estimated Point** (m, rounded)	**Mean Error** (m, rounded)	**Mean No. Meas/Sensor**
$(2.10, 0.00)$	$(1.798, 0.826)$	0.888	258.5
$(1.05, 0.00)$	$(1.695, 1.692)$	1.856	253.25
$(3.15, 0.00)$	$(3.193, 1.629)$	1.718	206.25
$(2.10, 1.45)$	–	–	–
$(1.05, 1.45)$	$(0.942, 1.522)$	0.612	231.0
$(3.15, 1.45)$	$(2.761, 1.965)$	0.671	299.5
$(0.00, 1.45)$	$(0.556, 1.238)$	0.682	249.75
$(1.05, 0.73)$	$(1.118, 1.412)$	0.822	190.75
$(2.10, 0.73)$	$(1.931, 1.809)$	1.239	228.0
Overall Mean Error: 1.061		**Overall Mean/Sensor**: 293.63	

namely the method used for signal strength filtering and location filtering. The following variants were included for signal strength filtering: a simple rolling mean filter, a rolling maximum followed by a rolling mean filter, and a rolling maximum followed by a rolling median filter. With regards to location filtering, the improvement gained by the Kalman filter was analyzed.

Tables I show the results of the indoor experiments. The average location estimation, the average localization errors, and the average number of measurements per sensor for the Nokia smartphone's LAP are shown. Data for the point $(2.1, 1.45)$ is missing in these results from the indoor experiments, due to the failure of one of the sensors that was only noticed after the completion of the experiment.

B. Experiment 2 — Outdoor

A second experiment was designed to evaluate the *BluePIL* device localization method under more challenging conditions and to evaluate the system's performance in its final streaming architecture. A space of $5\,\mathrm{m} \times 5\,\mathrm{m}$ was designated to perform the experiment. An Ubertooth One sensor was placed at each corner of the space, connected to an Asus Tinkerboard. A MacBook Pro was used to control these nodes. 32 equally spaced points were chosen in the $5\,\mathrm{m} \times 5\,\mathrm{m}$ space (*cf.* Figure 7). A test subject holding a Nokia smartphone, which was streaming audio to a pair of Bluetooth headphones, traversed these points, resting at each one for one minute. During this minute, data was captured by Ubertooth sensors.

TABLE II: Outdoor Environment Experiment Results

True Point (m)	**Mean Estimated Point** (m, rounded)	**Mean Error** (m, rounded)	**No. Localizations**
$(1, 1)$	$(0.989, 1.942)$	1.263	36
$(1, 4)$	$(1.160, 3.108$	1.255	33
$(4, 1)$	$(3.465, 3.188)$	2.320	26
$(4, 4)$	$(3.227, 3.766)$	1.065	26
$(2.5, 2.5)$	$(3.257, 1.703)$	1.129	22
Overall Mean Error: 1.406		**Mean No. Localizations**: 28.6	

A first approach repeated Experiment 1 under more challenging conditions. 32 equally spaced points were chosen in the $5\,\mathrm{m} \times 5\,\mathrm{m}$ space. They are shown in Figure 7. Experiment 2 follows the same testing procedure as Experiment 1 (*i.e.*, same test subject and moving pattern). This data was analyzed ex-post with the same static version of the pipeline used in Experiment 1.

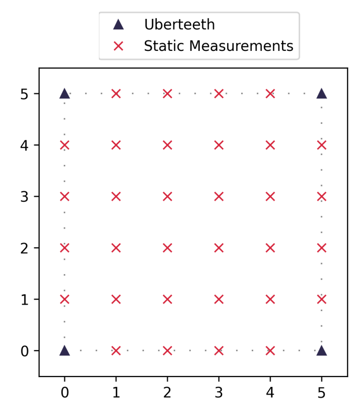

Fig. 7: Outdoor Setup (x and y Axes in Meter)

This second step tested the system in its full streaming implementation. To this end, five points were chosen in the $5\,\mathrm{m} \times 5\,\mathrm{m}$ space. Again, the Nokia smartphone and the JBL headphones were used to stream audio over BT. The near real-time positioning pipeline was then run for 15 mins. During this time, the test subject covered each of the five points, resting at each point for 2 mins and taking a maximum of one minute for the change between points. One minute of buffering time was included at the beginning. These points were traversed according to the following order: $(1, 1) \rightarrow (1, 4) \rightarrow (4, 1) \rightarrow (4, 4) \rightarrow (2.5, 2.5)$.

Experiment 2 revealed concerns with the Ubertooth sensors used. Most importantly, the performance regarding the number of packets captured deteriorated significantly from Experiment 1. During the first part of the experiment, the number of packets captured per second and per sensor was reduced to about 0.38 compared to 1.0 from before. This created problems in the location computation, especially in the first part of the experiment: Due to the decreased number and the fact that captured packets were not evenly spread throughout the time interval, it occurred that, for some of the points, there was no overlap between these points in time of the RSSI measurements. Consequently, it was impossible to merge the RSSI value streams between sensors. Only points that

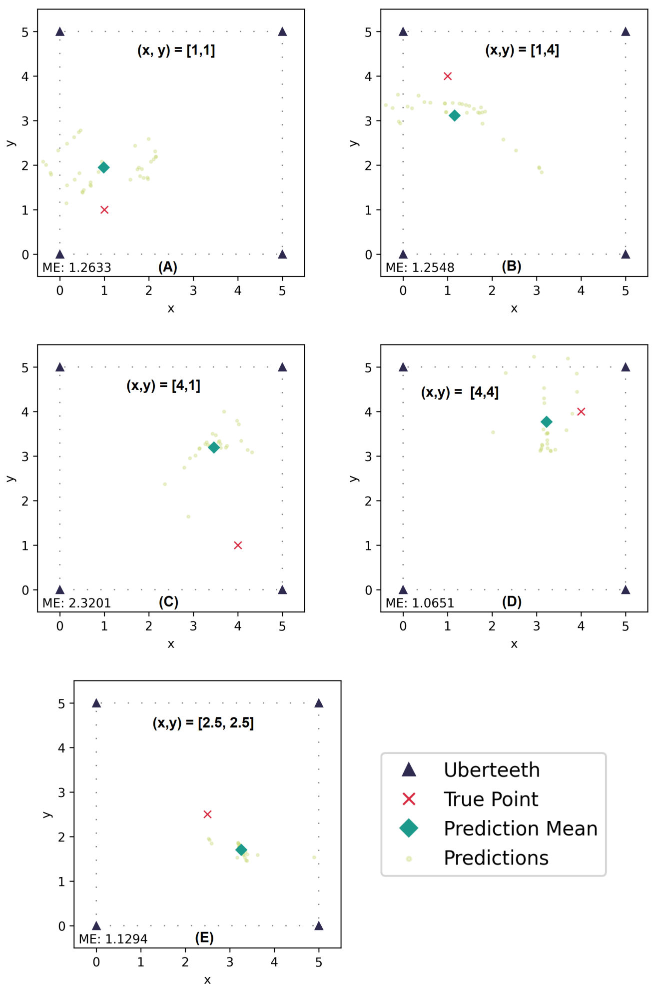

Fig. 8: Outdoor Experiment. (a) Point [1,1], (b) Point [1,4], (c) Point [4,1], (d) Point [4,4], and (e) Point [2.5,2.5]

produced a sufficient overlap of a minimum of 10 s were, therefore, analyzed for the first part of the experiment.

Table II and Figure 8 show results for the second part of the experiment. It is noticeable that the streaming processing pipeline handles the sparsity of sensor measurements better than individual evaluations performed in part one of the experiment. This is mostly due to the fact that the processing stream is able to use values for interpolation that lie outside time intervals defined as resting periods at each point. These results are more accurate than for the first part of the experiment, but worse than in Experiment 1 with an overall mean error of $1.406\,\mathrm{m}$. It should be noted, however, that the mean error value for point $(4, 1)$ forms an outlier, differing from the next lower value by $1.057\,\mathrm{m}$, more than five times the difference between any other two points ($0.198\,\mathrm{m}$). This corresponds to a pattern that was already observed in Experiment 1.

C. Discussion of Results and Findings

All evaluations were conducted in real-world environment containing externalities such as multipath fading, interference with other objects in the room, and other wireless signals. Overall, the positioning accuracy in both experiments was fairly similar, with an average error of $1.04\,\mathrm{m}$ and $1.061\,\mathrm{m}$ for the outdoor and the indoor experiments, respectively. The error values ranged from $0.398\,\mathrm{m}$ to $1.703\,\mathrm{m}$ for the outdoor and $0.612\,\mathrm{m}$ to $1.856\,\mathrm{m}$ for the indoor experiment. While the overall sensor performance was quite similar in the indoor and the outdoor experiment, collecting around one measurement per second, it was more stable in the indoor environment, where the mean number of measurements per sensor for the Nokia smartphone's LAP ranged from 190.75 to 299.5 compared to 132.75 to 439.5 in the outdoor case. Both sets of results show some outliers in the upper range of the error values, most notably point $(1.05, 0.00)$ for the outdoor experiment and points $(1.05, 0.00)$ and $(3.15, 0.00)$ for the indoor experiment.

Device localization in Experiment 1 produces results at around $1\,\mathrm{m}$ accuracy on average, both in indoor and outdoor spaces without a prior calibration of the system to estimate RSS_C and, in a completely passive manner, setting it apart from existing approaches. Results for the indoor space were slightly worse, which may be explained by higher amounts of noise from multi-path fading due to the more constrained dimensions. The first part of Experiment 2 did not show the same level of success. The decreased timespan allocated for each measurement combined with the deteriorated performance of the sensors led to large parts of the data being unusable. The performance for remaining data points was significantly worse than in the first experiment. The second part of the second experiment, however, showed the effectiveness of the streaming architecture, most notably the signal strength merger component, which dealt well with the additional sparsity of signal strength measurements encountered. The localization performance was good, apart from a single outlier, a pattern which was also observed during the first experiment. These results are comparable to existing research, which is advantageous given that the *BluePIL* system estimates locations and channel parameters simultaneously.

The existence of negative outliers in localization results in both experiments warrants discussion. An inspection of the raw data revealed that these outliers were not caused by a failure in the processing pipeline, but were already present in the data received from the sensors. A plausible explanation for this is that environmental factors, or possibly a combination thereof, will have lead to RSSI values not representing the location of the test subject accurately. Consequently, influencing factors, such as background noise, the topology of the space, temperature, humidity, could not be controlled and may have led to the disturbance of the signal. The test subject carrying the test devices may itself have had an effect on the signal strengths measured, as is had been suggested by the investigation of the influence of the human body on RSSI measurements [34].

The deterioration of sensor performance in Experiment 2 presented a further challenge. Since the environment was identical to the outdoor space used in the first experiment and the frequency of measurements was lower regardless of the distance from the sensor, these two factors do not explain the degradation. One potential explanation could be the change in weather conditions. While the first experiment took place in spring, the second one was performed in mid-summer, with temperatures exceeding $30°C$ on an asphalt surface. The sensors may have overheated, leading to the decreased frequency and lower accuracy of measurements in the radio components. Existing research suggests a strong influence of operating temperature on RSSI measurements for chips very similar to the one used in the Ubertooth One [8]. The unpredictability of these results may have been exacerbated by shade reaching some of the sensors over the duration of the experiment, breaking the assumption that all four sensors exhibit the same radio characteristics. This hypothesis could also explain the better performance in the second part of the experiment, which was executed at a later time on the same day, when temperatures were lower and the entire environment was covered by shade.

V. Related Work

Fingerprinting-based approaches, *e.g.*, [15] [5], require the construction of a radio map of the area of interest, *i.e.*, a set of sensor measurements for signal strength throughout the space, as a preliminary step. During localization, the signal strength measured for a mobile device within this space is correlated with individual points in the radio map. The best match is assumed to correspond to the actual location of the device. [5], [35] implements a BTBR/EDR-beacon-based fingerprinting approach. In addition, it compares the performance of k-Nearest-Neighbors and Naïve Bayes classifiers for fingerprint correlation.

The path-loss-based approaches, such as [9], [21], [25], [12], [34], [14], rely on a model that enables them to convert signal strengths to distances. [9] implements a log-distance path loss model to calculate distances for BTLE beacon advertising packets, which are used to perform tri-lateration in order to compute a location for the target device. In addition to this, they use a Kalman Filter to pre-process signal strength measurements to receive a smoother result. [21] takes a similar approach with a combination of the log-distance path loss model and tri-lateration. [25] also uses tri-lateration, but implements a neural-network-based path loss model that uses empirical data to infer a mathematical representation of path loss characteristics. Signal strengths are pre-processed using a Kalman filter. [12] compares the performance of the log-distance path loss model and a particle filter for the localization problem. The tri-lateration step is avoided by heuristically reducing the localization problem to a one-dimensional one through assumptions on the topology of the environment they are working in, *i.e.* a shopping mall. [34] uses BTBR/EDR advertising packets to perform localization using the log-distance path loss model and tri-lateration. This work includes the investigation of influences of the human body on signal strength measurements. [12] also work with the log-distance path loss model and tri-lateration, but the calculation of a location is not in the focus. Instead, the location is assumed to be known and this information is used to estimate channel parameters RSS_C and n.

Finally, [37] presents an approach that fuses fingerprinting and a polynomial regression path loss model. Distance estimates from the fingerprinting algorithm and the path loss model are averaged and fed to an extended Kalman filter, which performs the localization and smooths the result. Additionally, multiple layers of outlier detection are implemented in order to improve the overall localization accuracy.

VI. Summary and Future Work

This paper presented *BluePIL*, a distributed, near real-time, streaming system using a node-sink topology. To the best of the authors' knowledge, *BluePIL* is the first fully passive approach for the identification and localization of Bluetooth (BT) devices. It defines a data processing pipeline accomplishing the tasks of identification and localization through passively captured BT packets in several steps, *i.e.*, device identification, signal strength filtering, signal strength merging, the localization algorithm and location filtering.

The experiments demonstrated the effectiveness of *BluePIL* in different scenarios, indoor and outdoor. The indoor showed that the device localization method is sound and accurate in controlled environment, producing results with an accuracy of around $1\,\mathrm{m}$ in a $12\,\mathrm{m}^2$ area. The outdoor experiment evaluated the system's design and tested the localization method under more challenging conditions. It showed that the streaming architecture selected for the system is valid and that it can localize devices with an accuracy of $1.4\,\mathrm{m}$ in a $25\,\mathrm{m}^2$ area. This leads to the statement that contact and activity tracing can be performed passively in a BT device setting. However, purely BT-based tracing approaches have a relatively smaller range (at about 5 to $10\,\mathrm{m}$) than those based on Wi-Fi 802.11, suggesting that the combination of BT and Wi-FI data and their correlation may result in higher total accuracy of the tracing approach. Lastly, *BluePIL* is open source[1] based on a Python implementation running on low-cost hardware requiring minimal configuration, since most configuration is handled automatically.

As next steps in future work, the addition of further sensors could allow for either the extension of said range or for the improvement of the localization accuracy. While algorithms used for filtering and localization are theoretically capable of working with more than four measurements, approaches to extend the possible range would have to include strategies for the selection of the most useful measurements during the signal strength merger stage. Furthermore, during the evaluation, a variety of problems with the Ubertooth One sensors used were discovered. Running the system with an alternative sensor could yield information on the source of outliers discovered in those localization results. In addition, a more precise BT sensor may enable an improvement of the accuracy of the system overall.

[1] *BluePIL*'s code is available at: https://gitlab.ifi.uzh.ch/rodrigues/bluepil

Acknowledgments

This work has been supported by Innosuisse, the Swiss Agency for Innovation Demands under Grant No. 42193.1 IP-ICT. Also, the authors would also like to thank Simon Tuck, Livealytics AG, who supported the research and development during those experiments in Corona times of limited public accessibility and with many discussions on different setups and interpretations.

References

[1] A. Alhamoud, A. A. Nair, C. Gottron, D. Böhnstedt, and R. Steinmetz, "Presence Detection, Identification and Tracking in Smart Homes Utilizing Bluetooth Enabled Smartphones," in *39th Annual IEEE Conference on Local Computer Networks Workshops*. Edmonton, Canada: IEEE, 2014, pp. 784–789.

[2] J. B. Andersen, T. S. Rappaport, and S. Yoshida, "Propagation Measurements and Models for Wireless Communications Channels," *IEEE Communications Magazine*, Vol. 33, No. 1, pp. 42–49, 1995.

[3] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*. John Wiley & Sons, 2004.

[4] J. K. Becker, D. Li, and D. Starobinski, "Tracking Anonymized Bluetooth Devices," Vol. 2019, No. 3. Stockholm, Sweden: Sciendo, 2019, pp. 50–65.

[5] A. Bekkelien, M. Deriaz, and S. Marchand-Maillet, "Bluetooth Indoor Positioning," Master's thesis, University of Geneva (Unige), Geneva, Switzerland, 2012.

[6] S. Bhatti and J. Xu, "Survey of Target Tracking Protocols Using Wireless Sensor Network," in *5th International Conference on Wireless and Mobile Communications (WMC 2009)*. Cannes, France: IEEE, 2009, pp. 110–115.

[7] *Bluetooth Core Specification v5.2*, Bluetooth SIG, Dec. 2019.

[8] C. A. Boano, N. Tsiftes, T. Voigt, J. Brown, and U. Roedig, "The Impact of Temperature on Outdoor Industrial Sensornet Applications," *IEEE Transactions on Industrial Informatics*, Vol. 6, No. 3, pp. 451–459, 2009.

[9] S. Chai, R. An, and Z. Du, "An Indoor Positioning Algorithm using Bluetooth Low Energy RSSI," in *2016 International Conference on Advanced Materials Science and Environmental Engineering*. Atlantis Press, 2016.

[10] M. Chernyshev, "An Overview of Bluetooth Device Discovery and Fingerprinting Techniques–assessing the Local Context," in *13th Australian Digital Forensics Conference*. RI Security Research Institute, Edith Cowan University, Perth, Western Australia, 2015.

[11] A. De Carli, M. Franco, A. Gassmann, C. Killer, B. Rodrigues, E. Scheid, D. Schoenbaechler, and B. Stiller, "WeTrace–A Privacy-preserving Mobile COVID-19 Tracing Approach and Application," *arXiv preprint arXiv:2004.08812*, 2020.

[12] P. Dickinson, G. Cielniak, O. Szymanezyk, and M. Mannion, "Indoor Positioning of Shoppers using a Network of Bluetooth Low Energy Beacons," in *International Conference on Indoor Positioning and Indoor Navigation (IPIN 2016)*. Madrid, Spain: IEEE, 2016, pp. 1–8.

[13] DP-3T, "Dp3t - decentralized privacy-preserving proximity tracing," https://github.com/DP-3T/documents, 2020, accessed: 2020-07-27.

[14] J. Du, J.-F. Diouris, and Y. Wang, "A RSSI-based Parameter Tracking Strategy for Constrained Position Localization," *EURASIP Journal on Advances in Signal Processing*, Vol. 2017, No. 1, pp. 1–10, 2017.

[15] R. Faragher and R. Harle, "Location Fingerprinting with Bluetooth Low Energy Beacons," *IEEE Journal on Selected Areas in Communications*, Vol. 33, No. 11, pp. 2418–2428, 2015.

[16] Great Scott Gadgets, "Ubertooth," https://github.com/greatscottgadgets/ubertooth, 2020, revision: c2cc373, accessed: 2020-07-19.

[17] M. Haase, M. Handy *et al.*, "BlueTrack–Imperceptible Tracking of Bluetooth Devices," in *Ubicomp Poster Proceedings*, Nottingham, England, 2004, p. 2.

[18] Hacker Warehouse, "Ubertooth one," https://hackerwarehouse.com/product/ubertooth-one/, 2020, accessed: 2020-08-18.

[19] A. Hameed and H. A. Ahmed, "Survey on Indoor Positioning Applications based on Different Technologies," in *12th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS)*, Karachi, Pakistan, 2018, pp. 1–5.

[20] J. Huang, W. Albazrqaoe, and G. Xing, "Blueid: A Practical System for Bluetooth Device Identification," in *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*. Toronto, Canada: IEEE, 2014, pp. 2849–2857.

[21] J.-H. Huh and K. Seo, "An Indoor Location-based Control System using Bluetooth Beacons for IoT Systems," *Sensors*, Vol. 17, No. 12, p. 2917, 2017.

[22] IEEE, "IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture," *IEEE Std 802-2014 (Revision to IEEE Std 802-2001)*, pp. 1–74, 2014.

[23] International Data Corporation (IDC), "Smartphone Market Share," https://www.idc.com/promo/smartphone-market-share/vendor, 2020, accessed: 2020-07-30.

[24] R. Labbe, "Kalman and Bayesian Filters in Python," https://github.com/rlabbe/Kalman-and-Bayesian-Filters-in-Python, Commit 91f8010bee8cd7e07bdca338ecd90c3dd7735e92, 2014, GitHub Repository.

[25] G. Li, E. Geng, Z. Ye, Y. Xu, J. Lin, and Y. Pang, "Indoor Positioning Algorithm Based on the Improved RSSI Distance Model," *Sensors*, Vol. 18, No. 9, p. 2820, 2018.

[26] A. Lindemann, B. Schnor, J. Sohre, and P. Vogel, "Indoor Positioning: A Comparison of WiFi and Bluetooth Low Energy for Region Monitoring," in *International Joint Conference on Biomedical Engineering Systems and Technologies*. Rome, Italy: SCITEPRESS-Science and Technology Publications, Lda, 2016, pp. 314–321.

[27] M. Lourakis, "A Brief Description of the Levenberg-Marquardt Algorithm Implemented by Levmar," *Foundation of Research and Technology*, Vol. 4, No. 1, pp. 1–6, 2005.

[28] K. Madsen, H. Nielsen, and O. Tingleff, "Methods for Non-Linear Least Squares Problems (2nd ed.)," p. 60, Jan. 2004.

[29] L. Mainetti, L. Patrono, and I. Sergi, "A Survey on Indoor Positioning Systems," in *22nd International Conference on Software, Telecommunications, and Computer Networks (SoftCOM 2014)*. Split, Croatia: IEEE, 2014, pp. 111–120.

[30] J. Martin, T. Mayberry, C. Donahue, L. Foppe, L. Brown, C. Riggins, E. C. Rye, and D. Brown, "A Study of MAC Address Randomization in Mobile Devices and When it Fails," Vol. 2017, No. 4. Boston, USA: Sciendo, 2017, pp. 365–383.

[31] D. Oosterlinck, D. F. Benoit, P. Baecke, and N. Van de Weghe, "Bluetooth Tracking of Humans in an Indoor Environment: An application to Shopping Mall Visits," *Applied Geography*, Vol. 78, pp. 55–65, 2017.

[32] D. Spill and A. Bittau, "BlueSniff: Eve Meets Alice and Bluetooth," *WOOT*, Vol. 7, pp. 1–10, 2007.

[33] M. Versichele, L. De Groote, M. C. Bouuaert, T. Neutens, I. Moerman, and N. Van de Weghe, "Pattern Mining in Tourist Attraction Visits Through Association Rule Learning on Bluetooth Tracking Data: A Case Study of Ghent, Belgium," *Tourism Management*, Vol. 44, pp. 67–81, 2014.

[34] Y. Wang, X. Yang, Y. Zhao, Y. Liu, and L. Cuthbert, "Bluetooth Positioning Using RSSI and Triangulation Methods," in *2013 IEEE 10th Consumer Communications and Networking Conference (CCNC)*. Las Vegas, USA: IEEE, 2013, pp. 837–842.

[35] Z. Wu, Q. Xu, J. Li, C. Fu, Q. Xuan, and Y. Xiang, "Passive Indoor Localization based on CSI and Naive Bayes Classification," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 48, No. 9, pp. 1566–1577, 2017.

[36] S. N. Zellner, M. J. Enzmann, and R. T. Moton Jr, "System and Method for Tracking Movement of a Wireless Device," Sep. 28 2004, US Patent 6,799,049.

[37] Y. Zhuang, J. Yang, Y. Li, L. Qi, and N. El-Sheimy, "Smartphone-based Indoor Localization with Bluetooth Low Energy Beacons," *Sensors*, Vol. 16, No. 5, p. 596, 2016.