A Compressive Sensing Approach to Detect the Proximity Between Smartphones and BLE Beacons

Pai Chet Ng, James She, *Member, IEEE*, and Rong Ran, *Member, IEEE*

***Abstract*—Bluetooth low energy (BLE) beacons have been widely deployed to deliver proximity-based services (PBSs) to user's smartphones when users are in the proximity of a beacon. Conventional proximity detection simply uses the received signal strength (RSS) to infer the proximity, and then retrieves the PBS by mapping the beacon ID with the corresponding service in the cloud database. Such an approach suffers two major issues: 1) the severe RSS fluctuation might confuse the smartphone during the detection and 2) a malicious PBS can be delivered by manipulating the same beacon ID. This paper proposes RF fingerprinting to label a beacon with an *N*-dimensional fingerprint vector, which consists of *N* RSS values from *N* deployed beacons. The contribution of our proposed method is twofold: 1) we infer the proximity based on the fingerprint vector instead of relying solely on the single RSS value and 2) we retrieve the PBS by mapping the fingerprint vector instead of the hard-coded beacon ID. The challenge with our proposed approach is the incomplete fingerprint observation during real-time detection, resulting in an underdetermined proximity detection problem. To this end, we exploit the compressive sensing (CS) approach based on the differential evolutional algorithm to address such an underdetermined problem. Extensive simulations with real-world datasets show that our proposed approach outperforms the legacy machine learning techniques with substantial performance gains.**

***Index Terms*—Bluetooth low energy (BLE) beacon, compressive sensing (CS), differential evolution (DE), Internet of Things, proximity detection.**

I. Introduction

DIFFERENT to location-based services (LBS) which provide services based on users' locations [1], proximity-based services (PBSs), on the other hand, deliver their services to users' smartphones when users are in the proximity of a target item [2]. In other words, the service is delivered by detecting the proximity between the target item and the user's smartphone, which is not necessarily confined to a specific location. PBS has received a lot of interests these days following the introduction of Bluetooth low energy (BLE) Beacons. The active involvements of Apple and Google with their iBeacon and Eddystone, respectively, have led to massive beacon deployment in many public spaces, including shopping malls, airports, museums and the like.

In this paper, we define a thing of interest (ToI) to refer to a target item with a beacon attached for PBS purposes. The ToI can be either a moving item or a static item located at a fixed location. Most of the commercial PBS applications simply use the received signal strength (RSS) to infer the proximity between the smartphone and the ToI. After that, the mobile App can retrieve the corresponding service by mapping the beacon ID with a list of PBS stored in the cloud database. Such an approach suffers two drawbacks: 1) the always fluctuating RSS can confuse the smartphone and eventually causing false detection and 2) the hard-coded beacon ID can be manipulated for a malicious service. Furthermore, these commercial PBS applications always assume that the beacon attached with each ToI is always working, and the smartphone is able to receive all the signals successfully, as illustrated in Fig. 1(a). Based on such ideal assumptions, the proximity detection can be solved by simply identifying the beacon which contributes the strongest RSS. However, it is very unlikely for the smartphone to receive all the signals in practical scenarios owing to unpredictable signal loss and faulty beacons, as illustrated in Fig. 1(b).

Motivated by the above limitations, this paper proposes a proximity detection method with RF fingerprinting. Specifically, we label each ToI with a fingerprint vector $\Phi \in \mathbb{R}^N$, which consists of N RSS values from N beacons. Then, the proximity detection can be solved by comparing the similarity of the real-time fingerprint observation with a list of fingerprint vectors in the database. The contribution of our proposed RF fingerprinting is twofold: 1) we infer the proximity based on a fingerprint vector instead of relying solely on the single RSS value and 2) we retrieve the service by mapping the fingerprint with a list of PBS in the cloud database instead of using the hard-coded beacon ID. However, our proposed RF fingerprinting also suffers the same issues illustrated in Fig. 1(b). The unpredictable signal loss and faulty beacon lead to incomplete fingerprint observation, in which the dimension of the observation vector $\Phi_y \in \mathbb{R}^M$ is far smaller compared to the fingerprint vector $\Phi \in \mathbb{R}^N$ registered in the database. Such a dimensional mismatch causes an underdetermined proximity detection problem, in which $M << N$.

To this end, this paper leverages compressive sensing (CS) approach [3] to model the underdetermined problem described above. While some works have employed CS for target localization [4], [5], to the best of our knowledge, no work exploits

Manuscript received January 6, 2019; revised March 29, 2019; accepted April 30, 2019. Date of publication May 3, 2019; date of current version July 31, 2019. This work was supported by the Ajou University Research Fund and the frame of International Cooperation Program managed by the National Research Foundation of Korea under Grant 2017K2A9A2A06016102. *(Corresponding author: Rong Ran.)*

P. C. Ng and J. She are with the Department of Electronics and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong (e-mail: pcng@ust.hk; eejames@ust.hk).

R. Ran is with the School of Electrical and Computer Engineering, Ajou University, Suwon 16499, South Korea (e-mail: sunnyran@ajou.ac.kr).

Digital Object Identifier 10.1109/JIOT.2019.2914733

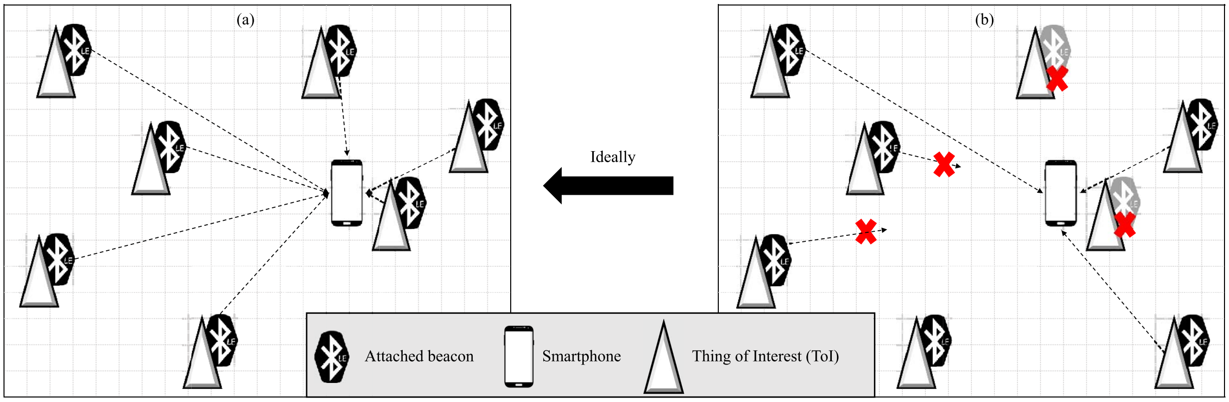


Fig. 1. Most commercial PBS applications always assume each ToI is attached with a working beacon and the smartphone is able to receive all the signals. Such an ideal assumption is invalid in a practical scenario with frequent signal loss and faulty beacons.

CS for the proximity detection problem. Proximity detection based on CS model is nonconvex with many local optimums. To efficiently deal with the nonconvex problem, we use correlation-based filter (CF) to narrow down the search scope before implementing differential evolution (DE) algorithm to search for the best solution. The main contributions of this paper are summarized as follows.

1) Our fingerprint vector is constructed with higher-order moments to better capture any statistical variations. Furthermore, we use time average to capture the RSS variations in the temporal domain and space average to capture the RSS variations in the spatial domain.
2) Our proposed correlation-based filter with differential evolution (CF+DE) is able to search for the optimum solution in less computational time compared to legacy machine learning techniques, while still outperforms these techniques with high detection accuracy.
3) Our experiments were conducted with real RSS measurements collected from a practical testbed. Furthermore, we demonstrate the feasibility of our approach with a real PBS use case, i.e., providing a parking service to users when they are approaching a parking space.

The rest of this paper is organized as follows. Section II reviews the related works. Section III provides an empirical analysis regarding the RSS values measured from all the beacons. A statistical fingerprint construction is then proposed in light of the empirical observation. Section IV leverages CS approach to model the proximity detection problem, and deduces a multiobjective function subject to the sparsity constraint. Section V presents our proposed CF+DE. Section VI describes our experiments and discusses the results. Section VII demonstrates the feasibility of our proposed approach with a practical implementation in a multistorey car park setting. Section VIII concludes this paper.

II. Related Works

Proximity detection is a method that uses wireless technologies to infer if two devices are in proximity to each other [6]. While RFID has been a prominent technology for proximity detection [2], [7], Bluetooth technology has emerged as another best wireless technology for proximity detection in consequences with the pervasive use of Bluetooth in smartphones. Many works have leveraged Bluetooth technology to detect the proximity between human [8]–[10] and to encourage proximity-based sharing and interaction [11], [12]. In general, these works infer the proximity by estimating the signal strength and the communication distance between two smartphones. None of them study the proximity between a smartphone to a physical thing in a particular space.

Recently, BLE beacon has been introduced to promote Internet of Things development using proximity detection [13]. Many commercial applications have started to adopt beacon to boost their applications' service. For example, beacons have been used in retail stores such as Walmart and Tesco to provide the consumers the details about a certain product [14]. Beacon also has been used in the art gallery/museum to assist visitors in browsing their favorite artworks and promoting further interaction [15], [16]. The simplest way to detect the proximity between a beacon and smartphone is by measuring the RSS values [17]. These RSS values can be measured by a smartphone equipped with Bluetooth connectivity. Furthermore, many beacon manufacturers have provided their software development kits (SDKs) freely to escalate the application development. Most of these SDKs offer a peak detection (PD) technique to infer the proximity. That is, a smartphone is considered to be in proximity with beacon A when beacon A returns a higher RSS than beacon B. Such a naive approach fails to consider the possible false detection owing to the signal fluctuation and faulty beacon.

A number of works have been proposed to enhance the accuracy of using beacon for PBS [18] While there are a number of works exploiting RF fingerprinting approach for localization purposes [19], [20], there is no work employs RF fingerprint for proximity detection. Despite the application purposes of using RF fingerprinting, RF fingerprinting generally suffers from the same drawbacks. First, the fingerprinting process is labor intensive and required frequent calibration to maintain the database [21], [22]. Second, the real-time fingerprint observation is always incomplete, resulting in an underdetermined proximity detection problem.

Fig. 2. BLE beacon by WuXi Ghostyu Electronics Company, Ltd. [Online]. Available: http://ghostyu.com.

III. Bluetooth Low Energy Beacon

BLE is a wireless technology operates in 2.4-GHz ISM band. BLE defines a total of 40 channels with 2-MHz spacing between each adjacent channel. BLE beacon only uses three channels, that is channels 38–40 which are strategically allocated to avoid interference with WiFi signals [23], [24]. According to the BLE specifications, there are four modes of a BLE device as a central, a peripheral, an observer, or an advertiser. BLE beacon generally works as an advertiser or a peripheral. A beacon in an advertiser mode is nonconnectable and is designed to broadcast its packet according to the predefined advertising interval T_a; whereas a beacon in a peripheral mode is connectable which can accept connection request upon receiving a response (known as scanResponse) from the receiver. BLE beacons have been widely deployed in many public locations for proximity detection and interaction [17]. Furthermore, BLE beacons are the potential infrastructure for RF fingerprint in comparison to the WiFi access points, as proven by Faragher and Harle [19]. Specifically, the small form factor of BLE beacon, as shown in Fig. 2 allows the beacon to be deployed in a remote area which is not easily accessible by the commodity WiFi device.

As discussed, the beacon broadcast their advertising packet periodically according to T_a. The advertising packet is a packet which can be formatted according to a certain protocol data unit (PDU). There are two very popular PDUs: iBeacon from Apple[1] and Eddystone from Google.[2] iBeacon allocates 2 bytes major and 2 bytes minor as the identifier, whereas Eddystone allocates at least 20 bytes for their Eddystone frame. Generally, these two PDUs provides a set of design rules for the developer to hard-coded the beacon with a fixed identifier. The corresponding service can be retrieved by mapping the identifier with a list of services stored in the online database.

RSS value can be measured by the receiver upon receiving the packet. In general, RSS values always suffer severe fluctuations due to various environmental factors, such as shadowing and multipath [25], [26]. This section first provides an empirical analysis regarding the RSS values measured by a smartphone at multiple positions around the same ToI and then presents our fingerprint vector constructed with higher-order moments to better capture the statistical variations observed from the empirical analysis.

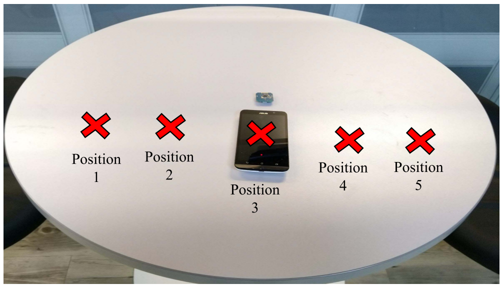


Fig. 3. Smartphone was placed at five different positions indicated by the "x" marker to measure the RSS values.

A. Variations of Received Signal Strength

The RSS value is a measurement in dBm scale [i.e., $P_{r,(\text{dBm})} = 10\log([P_{r,(\text{watt})}]/1\ \text{mW})$], ranging from -20 to -90 dBm subject to the distance between the beacon and the smartphone. According to the inverse square law, RSS is inversely proportional to the square of distance (i.e., $P_r \propto d^2$ [27], [28]). Hence, conventional proximity detection always infers the proximity between the smartphone and the ToI by identifying the ToI which contributes the strongest RSS. Such an intuition is valid if and only if: 1) the smartphone can measure all the RSS values from all of the beacons and 2) the RSS values from each beacon are distinguishable. However, RSS is proven to be unreliable and subject to heavy fluctuations owing to multipath and shadowing effects [29], [30].

We conducted an experiment to investigate the RSS fluctuations with respect to time and space. The experiment was conducted in a controllable manner by strategically placing the smartphone at five different positions around the same ToI, as depicted in Fig. 3. We performed the experiment at around midnight to minimize the external noise such that we can observe the inherent nature of RSS fluctuations rather than fluctuations caused by external factors.

Fig. 4 shows the RSS values measured by the smartphone from 0 to 150 s at the five positions. By examining the five graphs horizontally at any fixed position, we can see that the RSS values vary across the temporal domain, whereas by examining the five graphs vertically at any fixed time, the RSS values vary across the spatial domain. Hence, it is unreliable to detect the proximity by relying solely on the RSS values because the smartphone might make a false detection when the RSS values from the target ToI drop below the RSS values from the adjacent ToIs.

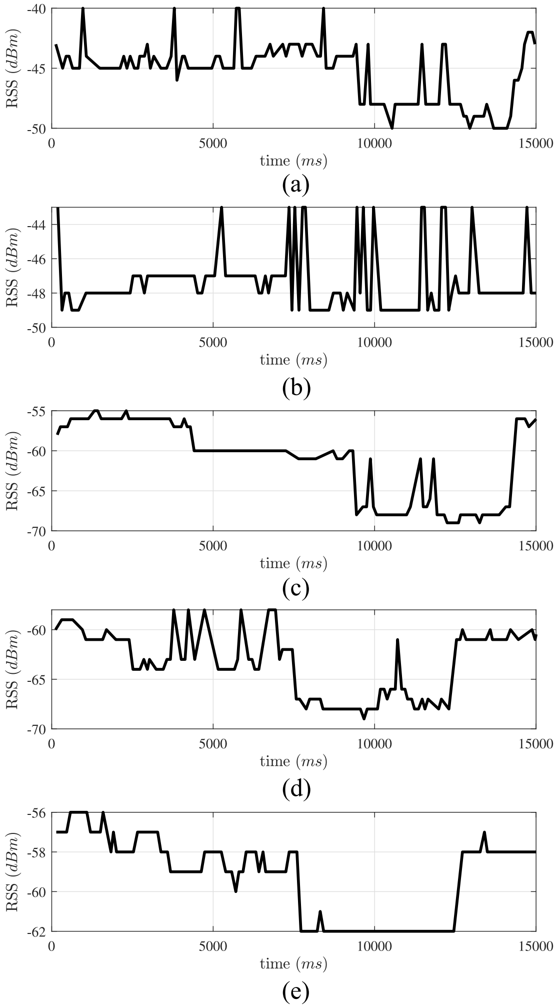
B. Fingerprint Construction Based on Higher-Order Moments

Let $\mathcal{B} = \{1, 2, \ldots, j, \ldots, N\}$ be a set of beacons attached with N ToIs, then a fingerprint vector $\Phi \in \mathbb{R}^N$ can be constructed by averaging the RSS values measured from all the $N = |\mathcal{B}|$ beacons during the deployment phase. Specifically, the fingerprint vector for jth ToI can be expressed as follows:

$$\Phi_j = \begin{pmatrix} \phi_1^{(j)} & \phi_2^{(j)} & \ldots & \phi_i^{(j)} & \ldots & \phi_N^{(j)} \end{pmatrix}^T \quad \forall j \in \mathcal{B} \tag{1}$$

[1] "iBeacon for Developers." [Online]. Available: https://developer.apple.com/ibeacon/

[2] "Eddystone." [Online]. Available: https://developers.google.com/beacons



(a)

(b)

(c)

(d)

(e)

Fig. 4. It is obvious that the RSS values vary across time regardless of the positions. By examining the RSS values from positions 1 to 5 at any fixed time, we can see that the values also changing across different positions.

and $\phi_i = (1/a)\sum_{t=t_1}^{t_a} P_r^{(i)}(t)$ is the time average RSS measured from ith beacon. $P_r^{(i)}(t)$ denotes the received power in dBm, measured at discrete time t_a for all $t_a \leq T_s$ and T_s is the scanning duration. Note that T_s is not fixed during fingerprint registration, instead we perform the scanning continuously until sufficient signals are sampled.

Note that (1) describes the common RF fingerprinting approach employed by most prior works [19], [31], [32]. However, from the previous experiment, we observe that the RSS always fluctuates even though the smartphone remained stationary at a fixed location. Furthermore, the smartphone is still considered to be in proximity with ToI 1 regardless of the different positions of the smartphone to the ToI 1, as illustrated with the red "x" marker in Fig. 3. In other words, the RSS variations happen across temporal and spatial domains. In light of the above observations, we construct the fingerprint vector with both time and space average. Furthermore, the RSS distribution does not follow the Gaussian distribution, as shown in Fig. 5, rather it is skewed to the right or left in connection to the positions of where the RSS is measured. Hence, we also introduce third and fourth order moments in constructing the fingerprint vector.

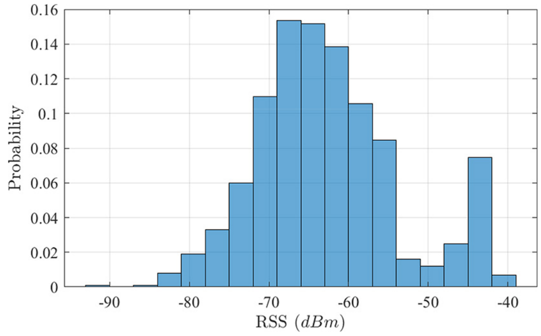


Fig. 5. RSS distribution does not follow the Gaussian distribution exactly.

In fact, the moment-based approach has been adopted in many research fields to address the noise issue caused by multipath, shadowing and other environmental factors [33]–[35]. Cardoso [36] suggested that higher-order moments can better discriminate each individual source, such a discrimination feature is indeed beneficial for the ToI detection. Hence, we further refine the conventional fingerprint vector described by (1) with the following fingerprint matrix $\Phi_j \in \mathbb{R}^{N\times 5}$:

$$\Phi_j = \begin{pmatrix} \phi_{1,1}^{(j)} & \phi_{1,2}^{(j)} & \phi_{1,3}^{(j)} & \phi_{1,4}^{(j)} & \phi_{1,5}^{(j)} \\ \phi_{2,1}^{(j)} & \phi_{2,2}^{(j)} & \phi_{2,3}^{(j)} & \phi_{2,4}^{(j)} & \phi_{2,5}^{(j)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \phi_{i,1}^{(j)} & \phi_{i,2}^{(j)} & \phi_{i,3}^{(j)} & \phi_{i,4}^{(j)} & \phi_{i,5}^{(j)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \phi_{N,1}^{(j)} & \phi_{N,2}^{(j)} & \phi_{N,3}^{(j)} & \phi_{N,4}^{(j)} & \phi_{N,5}^{(j)} \end{pmatrix} \quad \forall j \in \mathcal{B} \quad (2)$$

where the first column of Φ_j is the time average vector similar to (1), the second column ($\phi_{i,2}$) represents the space average vector, the third column ($\phi_{i,3}$) the variance vector, the fourth column ($\phi_{i,4}$) the skew vector, and the fifth column ($\phi_{i,5}$) the kurtosis vector. To facilitate the CS formulation described in the later section, we further flatten the above matrix into a kN-dimensional fingerprint vector, where $k = 5$.

Furthermore, we also introduce a dynamic fingerprint updating mechanism for the case when a new ToI is added to the system, as illustrated in Fig. 6. Initially, all the fingerprints were registered at a few random positions around the ToI when it is first deployed. After that, we can update the fingerprint from time to time when we are confident about the detection output. More precisely, we use confidence interval to determine how confident we are regarding the detection output, and only update the fingerprint matrix when the confidence level of the detection output is more than the minimum confidence threshold (γ). Based on the RSS measurements acquired during the detection phase, the space average can be updated as follows:

$$\phi_{i,2} = \begin{cases} \phi_{i,1}(r=1) + \frac{1}{a}\sum_{t=t_1}^{t_a} P_r^{(i)}(t,r), & \text{if } \phi_{i,2} = \phi_{i,1} \\ \phi_{i,2} + \frac{1}{a}\sum_{t=t_1}^{t_a} P_r^{(i)}(t,r), & \text{if } \phi_{i,2} \neq \phi_{i,1} \end{cases} \quad (3)$$

where r is the random position of the smartphone around the ToI and $r = 1$ refers to the initial position during the fingerprint registration. Similarly, the variance, skew and kurtosis are updated accordingly.

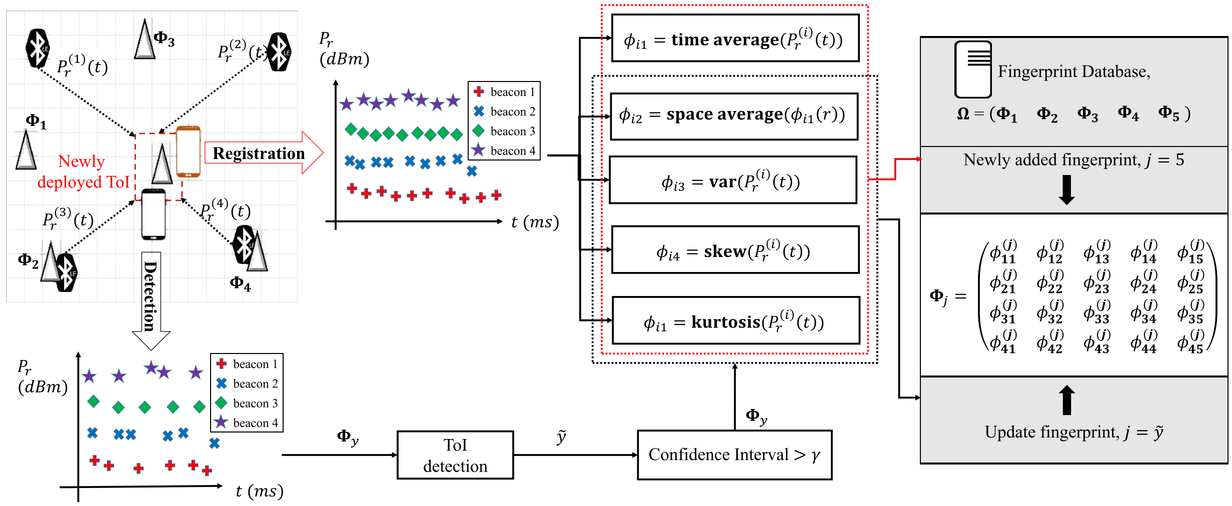

$P_r^{(1)}(t)$
Φ_3
$P_r^{(2)}(t)$
Φ_1
Newly deployed ToI
Registration
$P_r^{(3)}(t)$
$P_r^{(4)}(t)$
Φ_2
Φ_4
Detection
P_r (dBm)
beacon 1
beacon 2
beacon 3
beacon 4
$t\ (ms)$
ϕ_{i1} = time average$(P_r^{(i)}(t))$
ϕ_{i2} = space average$(\phi_{i1}(r))$
ϕ_{i3} = var$(P_r^{(i)}(t))$
ϕ_{i4} = skew$(P_r^{(i)}(t))$
ϕ_{i1} = kurtosis$(P_r^{(i)}(t))$
Fingerprint Database,
$\Omega = (\Phi_1\ \Phi_2\ \Phi_3\ \Phi_4\ \Phi_5)$
Newly added fingerprint, $j = 5$
$\Phi_j = \begin{pmatrix} \phi_{11}^{(j)} & \phi_{12}^{(j)} & \phi_{13}^{(j)} & \phi_{14}^{(j)} & \phi_{15}^{(j)} \\ \phi_{21}^{(j)} & \phi_{22}^{(j)} & \phi_{23}^{(j)} & \phi_{24}^{(j)} & \phi_{25}^{(j)} \\ \phi_{31}^{(j)} & \phi_{32}^{(j)} & \phi_{33}^{(j)} & \phi_{34}^{(j)} & \phi_{35}^{(j)} \\ \phi_{41}^{(j)} & \phi_{42}^{(j)} & \phi_{43}^{(j)} & \phi_{44}^{(j)} & \phi_{45}^{(j)} \end{pmatrix}$
Update fingerprint, $j = \tilde{y}$
Φ_y
ToI detection
$\tilde{y}$
Confidence Interval > γ
Φ_y

Fig. 6. Fingerprint is registered when a ToI is deployed, and dynamic update will be performed from time to time to capture the measurement variations.

IV. Problem Formulation

Proximity indicator vector **b** is defined to indicate the proximity of the smartphone to the ToI. Conventionally the proximity vector is an N dimensional one-sparse vector, i.e., the vector consists of only one nonzero entry indicating the proximity of the smartphone to the ToI. However, our proposed fingerprint is an kN dimensional vector by flattening the $N \times k$ matrix described in (2). Hence, we define a k-sparse proximity indicator vector $\mathbf{b} \in \mathbb{R}^{kN}$, which consists of k number of nonzero entries. This section presents our k-sparse problem formulation based on CS theory.

A. k-Sparse Problem

Proximity detection is a process for detecting the target ToI given the observation vector $\Phi_y \in \mathbb{R}^{kM}$ (i.e., fingerprint observed during detection phase). Suppose there are N registered fingerprints in the database, we can define a matrix $\Omega \in \mathbb{R}^{N\times kN}$ by appending the fingerprint matrix described in (2). Then, the proximity detection problem can be described as follows:

$$\Phi_y = \Omega\mathbf{b} + \mathbf{v} \tag{4}$$

where $\mathbf{v}$ is the measurement noise. Note that the above equation is only valid when the dimension of the observation vector Φ_y is the same as the number of columns in Ω, i.e., $kM = kN$.

However, the dimension of Φ_y is always much smaller, i.e., $kM << kN$, in consequences to the incomplete observation during the detection phase. In other words, the smartphone is only able to measure the RSS values from a portion of beacons. While it is possible to increase the scanning duration to ensure that the smartphone can measure RSS values from all of the deployed beacons, it is impractical for most PBS applications which require real-time service delivery. The other approach, which has been widely used by most machine learning techniques such as kNN, is to assign an artificial low RSS value to the missing element. However, such an approach is dangerous when the artificial RSS value cannot reflect the true RSS value.

In this paper, we exploit CS theory to deal with such an underdetermined problem. Let $\mathcal{D} = \{j : f(j) = 1, j \in \mathcal{B}\}$ be a set of indices indicating the beacons which can be observed by the smartphone during the detection phase, and $f(\cdot)$ is the mapping function in which $f(j) = 1$ when j ToI has a working beacon attached, otherwise $f(j) = 0$. Then, a sparsify matrix $\Psi \in \mathbb{R}^{M\times N}$ can be deduced as follows:

$$\Psi = \begin{pmatrix}\mathbf{c}_1 & \mathbf{c}_2 & \cdots & \mathbf{c}_M\end{pmatrix} \tag{5}$$

where $\mathbf{c}_i = (f(1) = 0, \ldots, f(i) = 1, \ldots, f(N) = 0)^T$, $\forall i \in \mathcal{D}$ and the element $f(i) = 1$ if $i = \mathcal{D}_i$ and 0 otherwise.

Given the sparsify matrix Ψ, the $N \times kN$ matrix Ω can be reduced to a $M \times kN$ matrix, i.e.,

$$\tilde{\Omega} = \Psi\Omega \tag{6}$$

Now, the proximity detection problem $\Phi_y = \tilde{\Omega}\mathbf{b} + \mathbf{v}$ is still a valid problem even though $kM << kN$. In our previous work [37], the proximity indicator vector $\mathbf{b} \in \mathbb{R}^N$ is a one-sparse vector, i.e.,

$$\mathbf{b}_{\{N\times 1\}} = \begin{pmatrix}0 & 0 & \ldots & 1 & \ldots & 0 & 0\end{pmatrix}^T \tag{7}$$

where the entry with value 1 indicates the proximity of the smartphone to the target ToI. Note that 1 is an ideal value of which we are very confident with the detection output. For the general case, the nonzero value is within the interval of (0, 1]. In this paper, our fingerprint is constructed with higher-order moments [defined by k columns in (2)], hence, we extend the one-sparse proximity indicator vector to k-sparse, i.e.,

$$\begin{aligned}\mathbf{b}_{\{kN\times 1\}} &= \begin{pmatrix}0 & 0 & \ldots & x_1 & x_2 & \ldots & x_k & \ldots & 0 & 0\end{pmatrix}^T \\ &= \left(\left\{\mathbf{x}^{(1)}_{\{k\times 1\}}\right\} \quad \ldots \quad \left\{\mathbf{x}^{(j)}_{\{k\times 1\}}\right\} \quad \ldots \quad \left\{\mathbf{x}^{(N)}_{\{k\times 1\}}\right\}\right)^T\end{aligned} \tag{8}$$

where $\mathbf{b} \in \mathbb{R}^{kN}$ is a k-sparse vector with all elements being zeros excepts for the k number of entries matching to the five columns of the fingerprint matrix described in (2). In other words, $\mathbf{b}$ is a concatenation of N number of vectors $\mathbf{x} \in \mathbb{R}^k$, in which $N - 1$ of them should be zero vectors.

For an ideal case with no temporal variation, the only nonzero vector $\mathbf{x}$ should have 1 at its first entry, i.e., $\mathbf{x} = (1 \quad 0 \quad 0 \quad 0 \quad 0)$, which indicates an exact match. However, such an ideal case is very unlikely considering the uncertainties imposed by both temporal and spatial variations. For the practical case, the vector $\mathbf{x}$ might consist of k number of nonzero values which leads to a k-sparse problem. Since k is far less than N, which satisfies the condition required by CS, then the k-sparse problem can be solved with a very high probability. Mathematically, the k-sparse problem can be formulated as follows:

$$\widetilde{\mathbf{b}} = \arg\min \|\mathbf{b}\|_0, \quad \text{s.t.} \quad \left\|\Phi_y - \tilde{\Omega}\mathbf{b}\right\|_2^2 < \epsilon. \tag{9}$$

Remark 1: CS theory [3] states that as long as the size of the real-time fingerprint observation $\Phi_y \in \mathbb{R}^{kM}$ is greater than $ck\log(N)$, i.e., $M >= ck\log(N)$, then the optimization problem defined by (9) can be solved uniquely with high probability. Since $k = 5$ in our case, a unique solution $\widetilde{\mathbf{b}}_{\{kN\times 1\}}$ can be obtained when kM is at least $5 \times 4 = 20$ for $N \leq 100$.

Remark 2: According to the restricted isometric property (RIP) [38] defined as follows:

$$(1 - \delta_k)\|\mathbf{b}\|_2^2 \leq \|\Omega\mathbf{b}\|_2^2 \leq (1 + \delta_k)\|\mathbf{b}\|_2^2 \tag{10}$$

where $\delta \in (0, 1]$, then we can argue that there exists a unique solution $\widetilde{\mathbf{b}}_{\{kN\times 1\}}$ when the Euclidean norm of $\widetilde{\mathbf{b}}_{\{kN\times 1\}}$ is preserved under the action of the measurement matrix Ω. In other words, the Euclidean norm of the observation vector $\Phi_y \in \mathbb{R}^{kM} = \Omega\mathbf{b}$ should be very similar to the Euclidean norm of $\mathbf{b}$, with acceptable variation δ.

B. Constrained Multiobjective Model

Given the incomplete observation vector Φ_y, our objective is to search for the vector $\mathbf{b}$ such that the resultant nonzero vector $\mathbf{x}$ has its x_1 maximized while minimizing the summation of x_2 to x_k. To ensure $\tilde{y}$ can be retrieved with high probability, we define a multiobjective function to model the k-sparse problem and impose a constraint on those k elements in nonzero vector $\mathbf{x}$. The intuition here is that the estimated k-sparse vector $\widetilde{\mathbf{b}}$ should have its first nonzero entry closely approximate to the fingerprint registered during the initial deployment phase with minimum measurement variation. If the nonzero vector $\mathbf{x}$ reflects a large measurement variation, it is very likely that the ToI indicated by $\mathbf{b}$ might not be the target ToI we are looking for. Mathematically, the multiobjective problem is defined as follows:

$$\begin{aligned}
\widetilde{\mathbf{b}} &= \arg\min_{\mathbf{b}} \ \|\Phi_y - \tilde{\Omega}\mathbf{b}\|_2^2 + \lambda\|\mathbf{b}\|_1 \\
\widetilde{\mathbf{x}} &= \arg\max_{\mathbf{x}} \ |x_1 - \sum_{k=2}^{5} x_k|^2 \\
&\qquad \text{s.t.} \ \left\|\Phi_y - \tilde{\Omega}\mathbf{b}\right\|_2^2 < \epsilon \\
&\qquad\qquad \sum_{k=1}^{5} x_k = 1, \quad x_k \in [0, 1]
\end{aligned} \tag{11}$$

where λ is the regularization parameter. To retrieve the index of the target ToI $\tilde{y}$, we multiply $\widetilde{\mathbf{b}}$ with a k-shrinking matrix $\mathbf{S} \in \mathbb{R}^{5N\times N}$, in which each row of $\mathbf{S}$ consists of k numbers of ones at the entry of $i = j$ up to $i = j + k - 1$. The rests are all 0s

$$s_{ij} = \begin{cases} 1, & i = j + k - 1 \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

where k varies from 1 to 5. Then, $\tilde{y}$ can be retrieved as follows:

$$\tilde{y} = \mathcal{B} \times \text{round}\big(\mathbf{S}\widetilde{\mathbf{b}}\big). \tag{13}$$

Note that finding an optimal solution given (11) can take years since there are many possible solutions. In the next section, we present our CF to narrow down our search scope and use DE to search for the optimal solution heuristically.

V. Proposed Correlation-Based Filter With Differential Evolution

This section first describes CF before putting forward the logic flow of DE.

A. Correlation-Based Filter

This paper considers the relationship between the observation vector Ω_y and all the fingerprint matrices stored in the database by computing the correlation between Ω_y and the mean of the first and second column of the registered fingerprint matrix. Mathematically, the correlation between $\Phi_y \in \mathbb{R}^M$ and $\Phi_j \in \mathbb{R}^{N\times 5}$ can be described as follows:

$$\Upsilon_j = \text{corr}\big(\Phi_y, \ddot{\Phi}_j\big) \tag{14}$$

where $\ddot{\Phi}_j$ consists of the mean values of all the elements in first and second column of Φ_j which is further conditioned on the sparsify matrix Ψ, i.e.,

$$\ddot{\Phi}_j = \frac{1}{2}\Psi_{\{M\times N\}}\left(\begin{pmatrix} \phi_{1,1}^{(j)} \\ \phi_{2,1}^{(j)} \\ \vdots \\ \phi_{N,1}^{(j)} \end{pmatrix} + \begin{pmatrix} \phi_{1,2}^{(j)} \\ \phi_{2,2}^{(j)} \\ \vdots \\ \phi_{N,2}^{(j)} \end{pmatrix}\right). \tag{15}$$

The resultant Υ_j ranges between -1 and 1, with $\Upsilon_j = 1$ indicates an exact match to the fingerprint registered by jth ToI. By computing the correlation for all ToIs, we have a correlation vector $\Upsilon = \big(\Upsilon_1 \quad \Upsilon_2 \quad \ldots \quad \Upsilon_j \quad \ldots \quad \Upsilon_N\big)^T, \quad \forall j \in \mathcal{B}$. Then, the set of ToI's indices to be filtered can be formulated as follows:

$$\mathcal{F} = \big\{j : \Upsilon_j < 0, \forall j \in \mathcal{B}\big\}. \tag{16}$$

Furthermore, the fingerprints which belong to $\mathcal{F}$ are excluded from the matrix Ω, and thus we obtain a filtered matrix $\widehat{\Omega}$ as follows:

$$\widehat{\Omega} = \big\{\Phi_j : \forall j \in \mathcal{B} \setminus \mathcal{F}\big\} \tag{17}$$

where $\mathcal{B} \setminus \mathcal{F}$ denotes all the indices in $\mathcal{B}$ except those in $\mathcal{F}$.

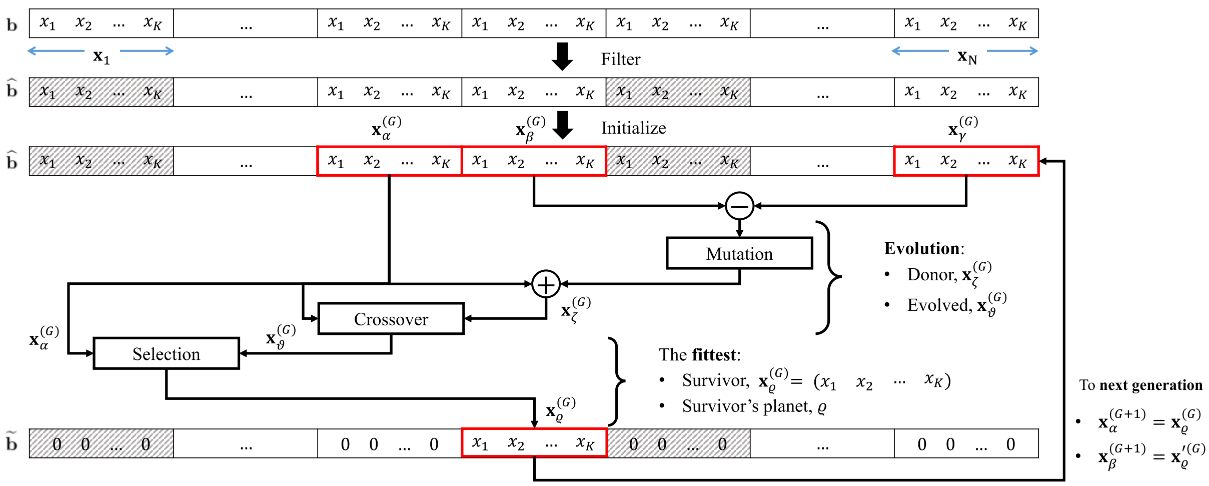

$\mathbf{b}$
x_1 x_2 ... x_K
$\mathbf{x}_1$
$\mathbf{x}_N$
Filter
$\widehat{\mathbf{b}}$
Initialize
$\mathbf{x}_\alpha^{(G)}$
$\mathbf{x}_\beta^{(G)}$
$\mathbf{x}_\gamma^{(G)}$
Mutation
Evolution:
Donor, $\mathbf{x}_\zeta^{(G)}$
Evolved, $\mathbf{x}_\vartheta^{(G)}$
Crossover
$\mathbf{x}_\zeta^{(G)}$
$\mathbf{x}_\vartheta^{(G)}$
Selection
The fittest:
Survivor, $\mathbf{x}_\varrho^{(G)} = (x_1 \ x_2 \ \cdots \ x_K)$
Survivor's planet, ϱ
$\mathbf{x}_\varrho^{(G)}$
$\widetilde{\mathbf{b}}$
0 0 ... 0
To next generation
$\mathbf{x}_\alpha^{(G+1)} = \mathbf{x}_\varrho^{(G)}$
$\mathbf{x}_\beta^{(G+1)} = \mathbf{x}_\varrho'^{(G)}$

Fig. 7. General flow of DE to search for the optimal *k*-sparse proximity indicator vector.

B. Differential Evolution

Given the filtered matrix $\widehat{\Omega}$, the relation between Φ_y and $\mathbf{b}$ described in (4) is further refined to $\Phi_y = \widehat{\Omega}\widehat{\mathbf{b}} + \mathbf{v}$. For the purpose of the matrix operation, the proximity indicator vector $\mathbf{b}$ needs to be confined to $\widehat{\mathbf{b}}$ such that its height equals to the length of $\widehat{\Phi}$. We can apply the same filter operation on $\widehat{\mathbf{b}}$, i.e.,

$$\widehat{\mathbf{b}} = \left\{\mathbf{x}_j : \forall j \in \mathcal{B} \setminus \mathcal{F}\right\}. \tag{18}$$

The resultant $\widehat{\mathbf{b}}$ is illustrated in Fig. 7 with the shaded blocks indicating the filtered vector $\mathbf{x}$.

Given the filtered output $\widehat{\mathbf{b}}$ and $\widehat{\Omega}$, DE iteratively search for the optimal nonzero vector $\mathbf{x}$ which best fits the multiobjective function described in (11). The intuition behind DE is to search for the survivor (the fittest solution) from the current population, and others which fail to survive will continue to evolve to the next generation. The basic idea is borrowed from the general concept of evolutionary programming [39], [40], and we further extend DE by considering the best planet that provides a better habitat for the survivors.

Referring to Fig. 7, three children are initialized randomly from the population $\widehat{\mathbf{b}}$, i.e., $\mathbf{x}_\alpha$, $\mathbf{x}_\beta$, and $\mathbf{x}_\gamma$. In contrast to most conventional matching pursuit algorithms, which tries to match the observation to every single column in the measurement matrix; DE picks any three random solutions and processes them in parallel to locate the survivor. Furthermore, DE returns the solution immediately once the solution converges. To ensure unit length, we imposed a normalization onto each child

$$\bar{\mathbf{x}}_u = \mathbf{x}_u^2 \frac{1}{\sum_{k=1}^{5} x_{u_k}^2} \qquad \forall u = \{\alpha, \beta, \gamma\}. \tag{19}$$

The target child $\bar{\mathbf{x}}_\alpha$ at the current generation G can be chosen with the following operation:

$$\left[\bar{\mathbf{x}}_\alpha^{(G)}, \rho_\alpha^{(G)}\right] = \max_{u,j\in\widehat{\mathcal{B}}}\left(\bar{\mathbf{x}}_u^T \Phi_j\right) \tag{20}$$

where $\widehat{\mathcal{B}} \subseteq \mathcal{B}$ is the filtered set of indices. Another variable returned by (20) is the best planet to accommodate the $\bar{\mathbf{x}}_\alpha^{(G)}$. We denote this best planet at the current generation as $\rho_\alpha^{(G)}$. The other two children which are not chosen at the current generation (i.e., $\bar{\mathbf{x}}_\beta^{(G)}$ and $\bar{\mathbf{x}}_\gamma^{(G)}$) are then subject to the following evolution.

1) *Mutation:* Gene alteration is imposed by multiplying the difference between $\bar{\mathbf{x}}_\beta^{(G)}$ and $\bar{\mathbf{x}}_\gamma^{(G)}$ with a mutation ratio and then adding it up with the target child to form a donor gene. Mathematically, the mutation phase can be expressed as follows:

$$\mathbf{x}_\zeta^{(G)} = \bar{\mathbf{x}}_\alpha^{(G)} + \mathcal{M}\left(\bar{\mathbf{x}}_\beta^{(G)} - \bar{\mathbf{x}}_\gamma^{(G)}\right) \tag{21}$$

where $\mathcal{M}$ is the mutation ratio ranging from 0 to 2 according to [41].

2) *Crossover:* The resultant donor gene is exchanged with the target child according to a predefined crossover rate to produce an evolved child. Specifically, when the randomly generated number is equal or greater than the crossover rate, then the evolved child inherits the gene from $\mathbf{x}_\zeta^{(G)}$; otherwise, the evolved child inherits the gene from $\bar{\mathbf{x}}_\alpha^{(G)}$. In general, a crossover vector $\mathbf{O} \in \mathbb{R}^k$ which consists of only 1s or 0s subject to the random generated number and the crossover rate can be obtained as follows:

$$o_i = \begin{cases} 1, & R_i(n) \geq \mathcal{C} \\ 0, & R_i(n) < \mathcal{C} \end{cases} \qquad \forall i \in [1, K] \tag{22}$$

where $R_i(n)$ is the random number generation function for element i. The evolved child based on the crossover vector $\mathbf{O}$ can be produced as follows:

$$\mathbf{x}_\vartheta^{(G)} = \mathbf{O}^T \mathbf{x}_\zeta^{(G)} + \mathbf{O}'^T \bar{\mathbf{x}}_\alpha^{(G)} \tag{23}$$

where $\mathbf{O}'$ is the complement vector to those elements in $\mathbf{O}$.

The evolved child is then compared to the target child using the similar operation described in (20). According to Fig. 7, the survivor which defeats its peers will continue its journey to the next generation. Mathematically, the survivor is selected as follows:

$$\left[\bar{\mathbf{x}}_\varrho^{(G+1)}, \varrho\right] = \max_{j\in\widehat{\mathcal{B}}}\left(\left\{\bar{\mathbf{x}}_\alpha^T \Phi_j\right\}, \left\{\bar{\mathbf{x}}_\vartheta^T \Phi_j\right\}\right) \tag{24}$$

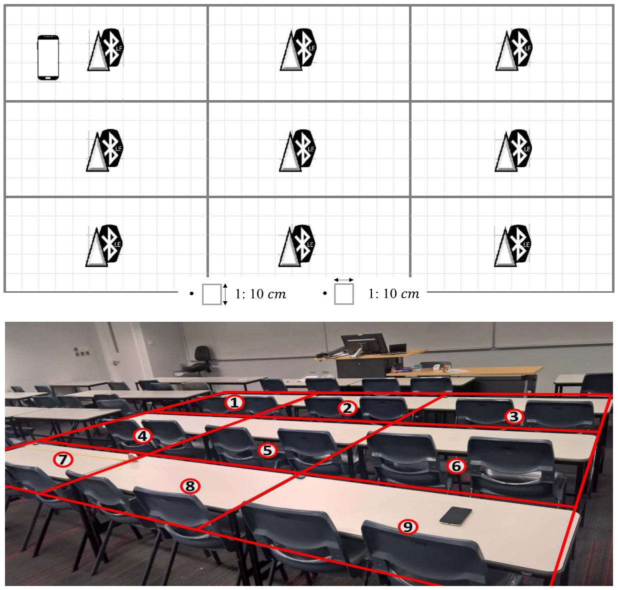


Fig. 8. Classroom testbed with nine equally spaced beacons was used for data collection purposes.

where ϱ denotes the best planet for the current survivor to settle. Suppose that the best survivor $\mathbf{x}_\varrho$ across several generations is found, then the estimated proximity indicator vector $\widetilde{\mathbf{b}}$ can be concluded to

$$\widetilde{\mathbf{b}} = \left\{\mathbf{x}_1^T, \mathbf{x}_2^T, \ldots, \mathbf{x}_j^T, \ldots, \mathbf{x}_N^T\right\} = \begin{cases} \mathbf{x}_j = \mathbf{0}, & \text{if } j \neq \varrho \\ \mathbf{x}_j = \bar{\mathbf{x}}_\varrho, & \text{if } j = \varrho \end{cases} \tag{25}$$

where $\mathbf{0} \in \mathbb{R}^k$ is the zero vector. Given the estimated $\widetilde{\mathbf{b}} \in \mathbb{R}^k N$, the index of target ToI $\tilde{y}$ can be retrieved via the same operation, as described in (13).

VI. Performance Evaluation

We performed our experiment with real-world dataset collected from different environments. The testbed for data collection and the corresponding experimental results are discussed in Sections VI-A and VI-C, respectively.

A. Experimental Testbed

Rather than generate the signals artificially, we collected real RSS values from different environments with different settings at different times. In particular, a few empty classrooms were used as the testbeds. The testbed setting consists of nine equally spaced beacons, as shown in Fig. 8. Note that such a classroom setting is generic and applicable to diverse PBS use cases. All the RSS values contributed by all the beacons were collected with an Android-based smartphone (i.e., Asus Zenphone Deluxe). In each ToI, approximately 1000 RSS values from each beacon were measured. We repeated the same measurement at five different positions around the same ToI resulting in more than 5000 data from each ToI. Similar steps were repeated in all the nine ToIs, which contributes to a total of $> 45k$ data.

TABLE I
Performance Comparison Between OMP and CoSaMP

Performance Measure	OMP	CoSaMP
Average Detection Performance	0.6555	0.713
Average Runtime	0.0898	0.0079

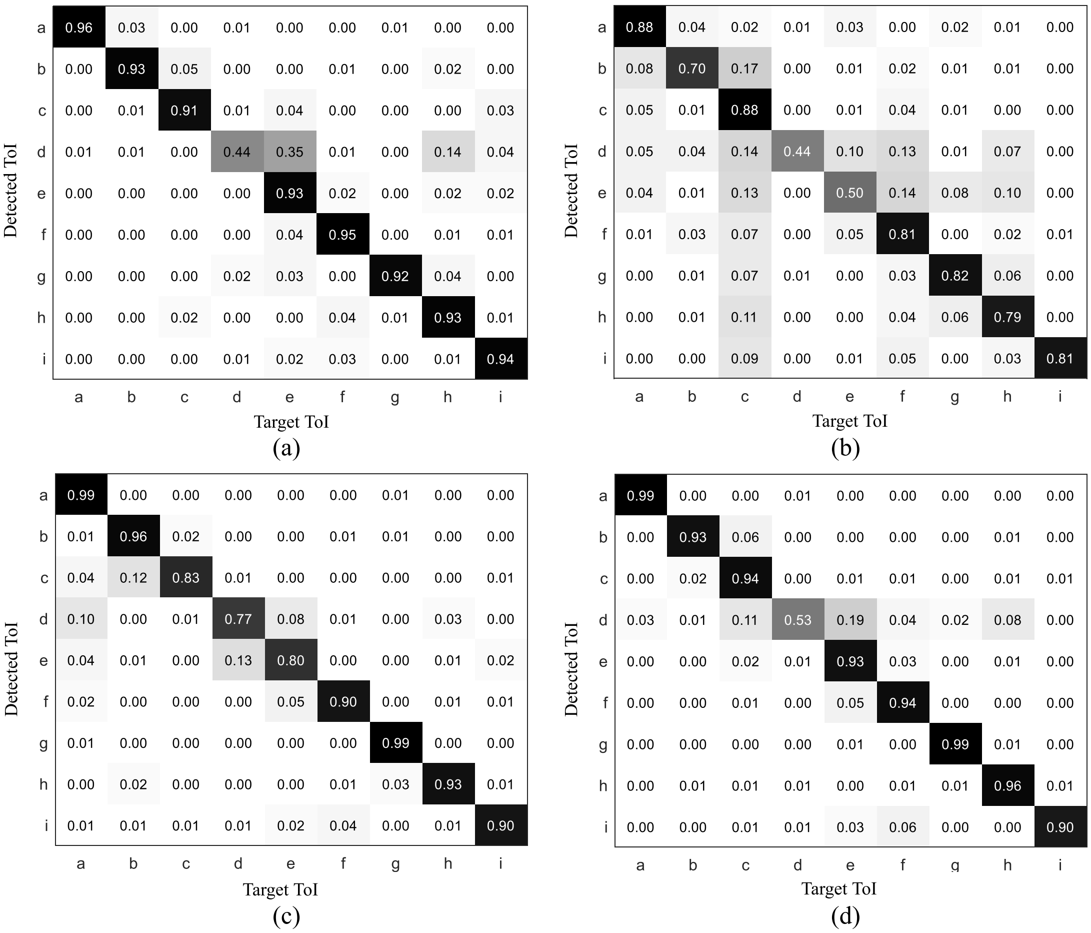
B. Baseline Algorithms

Three commonly used algorithms for proximity detection problem are chosen as baselines.

1) Peak Detection: This straight forward approach is offered as a free open source SDK by many beacon manufacturers to fasten the development cycle of PBS applications. As its name implies, PD examines the observation vector Φ_y, and returns the element that contributes the strongest RSS. Mathematically, we can find the index of the target ToI by computing $\tilde{y} = \arg\max_{\phi_{i,y} \in \Phi_y} \Phi_y$, where $\phi_{i,y}$ is the time average RSS measured from the beacon attached with ith ToI during the scanning duration T_s.

2) kNN-Based Fingerprinting: Having constructed the radio map (i.e., the matrix consists of only the RSS values), this machine learning technique selects the top k ToIs by computing the Euclidean-based similarity between the observation vector Φ_y to all the RSS-based fingerprints registered in the radio map $\Omega_{\mathcal{R}}$ [42], [43]. Since the height of vector Φ_y might not be equal to the length of the radio map $\Omega_{\mathcal{R}}$, the empty element of Φ_y, if any, is assigned with an artificial RSS value, i.e., $\phi_{i,y} = \{-100 : i \in \mathcal{B} \vee \phi_{i,y} = \emptyset\}$. Now, given $\Omega_{\mathcal{R}} = \begin{pmatrix} \Phi_{\mathcal{R}}^1 & \Phi_{\mathcal{R}}^2 & \ldots & \Phi_{\mathcal{R}}^j & \ldots & \Phi_{\mathcal{R}}^N \end{pmatrix} \in \mathbb{R}^{m \times N}$ and $\Phi_y \in \mathbb{R}^m$, the index of the target ToI $\tilde{y}$ can be computed with $\tilde{y} = \arg\min_{j \in \mathcal{B}} \|\Phi_y - \Phi_{\mathcal{R}}^j\|_2^2$.

3) Compressive Sampling Matching Pursuit: Two of the most widely used approaches in CS framework are orthogonal matching pursuit (OMP) [44] and compressive sampling matching pursuit (CoSaMP) [45]. Both OMP and CoSaMP are greedy algorithms which search for the sparse solution iteratively. Table I compares the experimental results between OMP and CoSaMP. In this experiment, we used a small subset of data from the testbed illustrated in Fig. 8, and randomly discarded the some beacons' signals to emulate the phenomenon of incomplete observations. We performed the experiments for 1370 times for data collected at different ToI and computed the average results. It is clear that CoSaMP achieves better performance than OMP in terms of detection performance and runtime. Hence, we choose CoSaMP as a baseline algorithm to compare with our proposed CF+DE. The general flow of CoSaMP is as follows: first, CoSaMP transforms $\Psi\Omega_{\mathcal{R}}$ into an orthonormal basis, i.e., $Q = \text{orth}(\Psi{\Omega_{\mathcal{R}}}^T)^T$, and then imposes a signal preprocessing operator $T = Q(\Phi_y)^\dagger$ to transform the observation vector, i.e., $\Phi_y' = T\Phi_y$. Based on the orthonormal basis Q and transformed observation vector Φ_y', the k-sparse proximity indicator vector can be estimated. Note that in our case, the sparsity level k is known. Hence, CoSaMP can be applied to obtain $\mathbf{b}$ efficiently. In case k is unknown, Candes and Wakin [3] suggested that we can use a phase transition analysis method to estimate the unknown k.



	a	b	c	d	e	f	g	h	i
a	0.96	0.03	0.00	0.01	0.00	0.00	0.01	0.00	0.00
b	0.00	0.93	0.05	0.00	0.00	0.01	0.00	0.02	0.00
c	0.00	0.01	0.91	0.01	0.04	0.00	0.00	0.00	0.03
d	0.01	0.01	0.00	0.44	0.35	0.01	0.00	0.14	0.04
e	0.00	0.00	0.00	0.00	0.93	0.02	0.00	0.02	0.02
f	0.00	0.00	0.00	0.00	0.04	0.95	0.00	0.01	0.01
g	0.00	0.00	0.00	0.02	0.03	0.00	0.92	0.04	0.00
h	0.00	0.00	0.02	0.00	0.00	0.04	0.01	0.93	0.01
i	0.00	0.00	0.00	0.01	0.02	0.03	0.00	0.01	0.94

(a)

	a	b	c	d	e	f	g	h	i
a	0.88	0.04	0.02	0.01	0.03	0.00	0.02	0.01	0.00
b	0.08	0.70	0.17	0.00	0.01	0.02	0.01	0.01	0.00
c	0.05	0.01	0.88	0.00	0.01	0.04	0.01	0.00	0.00
d	0.05	0.04	0.14	0.44	0.10	0.13	0.01	0.07	0.00
e	0.04	0.01	0.13	0.00	0.50	0.14	0.08	0.10	0.00
f	0.01	0.03	0.07	0.00	0.05	0.81	0.00	0.02	0.01
g	0.00	0.01	0.07	0.01	0.00	0.03	0.82	0.06	0.00
h	0.00	0.01	0.11	0.00	0.00	0.04	0.06	0.79	0.00
i	0.00	0.00	0.09	0.00	0.01	0.05	0.00	0.03	0.81

(b)

	a	b	c	d	e	f	g	h	i
a	0.99	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00
b	0.01	0.96	0.02	0.00	0.00	0.01	0.01	0.00	0.00
c	0.04	0.12	0.83	0.01	0.00	0.00	0.00	0.00	0.01
d	0.10	0.00	0.01	0.77	0.08	0.01	0.00	0.03	0.00
e	0.04	0.01	0.00	0.13	0.80	0.00	0.00	0.01	0.02
f	0.02	0.00	0.00	0.00	0.05	0.90	0.00	0.01	0.01
g	0.01	0.00	0.00	0.00	0.00	0.00	0.99	0.00	0.00
h	0.00	0.02	0.00	0.00	0.00	0.01	0.03	0.93	0.01
i	0.01	0.01	0.01	0.01	0.02	0.04	0.00	0.01	0.90

(c)

	a	b	c	d	e	f	g	h	i
a	0.99	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00
b	0.00	0.93	0.06	0.00	0.00	0.00	0.00	0.01	0.00
c	0.00	0.02	0.94	0.00	0.01	0.01	0.00	0.01	0.01
d	0.03	0.01	0.11	0.53	0.19	0.04	0.02	0.08	0.00
e	0.00	0.00	0.02	0.01	0.93	0.03	0.00	0.01	0.00
f	0.00	0.00	0.01	0.00	0.05	0.94	0.00	0.00	0.00
g	0.00	0.00	0.00	0.00	0.01	0.00	0.99	0.01	0.00
h	0.00	0.01	0.01	0.01	0.00	0.01	0.01	0.96	0.01
i	0.00	0.00	0.01	0.01	0.03	0.06	0.00	0.00	0.90

(d)

Fig. 9. Detection performances of PD, kNN-F, CoSaMP, and CF+DE when all the attached beacons are working perfectly.

TABLE II
DETECTION PERFORMANCE OF EACH APPROACH

Approach	Precision	Recall	Accuracy
PD	0.8928	0.8774	0.8788
kNN-F	0.7759	0.7372	0.7367
CoSaMP	0.8999	0.8973	0.8967
CF+DE	**0.9076**	**0.8997**	**0.9011**

C. Experimental Results

We first evaluate the detection performance when all the beacons are working correctly, but the RSS values varied across the temporal and spatial domains. The detection performance of each approach is presented with a confusion matrix, as shown in Fig. 9. The *x*-axis is the target ToI and the *y*-axis indicates the detected ToI. Hence, the diagonal elements indicate a correct detection. The overall precision, recall, and accuracy achieved by each approach are further summarized in Table II. While most approaches were able to return a high detection performance, our proposed CF+DE outperforms the rest with an average of 90.11% accuracy.

Next, a few beacons were intentionally shut off to emulate a scenario with faulty beacons. Fig. 10(a)–(c) illustrates the three scenarios with different faulty beacons. The same measuring procedures were repeated for all scenarios, resulting in a total of $> 135k$ data. The detection performances reported by the baselines and our proposed CF+DE are described below.

1) Scenario 1: Fig. 11 shows that PD only works when the ToI is attached with a working beacon. It suffers severe performance degradation when the attached beacons are not working. Even though both kNN-based fingerprinting (kNN-F) and CoSaMP are comparatively better than PD, their detection accuracy, however, is still below average. CF+DE, on the other hand, achieves a good detection performance with more than 80% precision and recall, on average.

2) Scenario 2: Again, Fig. 11 shows that PD fails to produce a correct detection at those ToIs without an attached beacon (i.e., ToIs 2, 4–6, and 8). The performances of both kNN-F and CoSaMP are still below average; whereas CF+DE achieves a very good performance with above 70% precision and recall at those ToIs without a working beacon attached and above 90% in the ToIs with a working beacon attached.

3) Scenario 3: When only three ToIs are attached with a working beacon, the detection performance of both CoSaMP and kNN-F degrades severely. On the other hand, Fig. 11 shows that our proposed CF+DE is still able to maintain its detection performance. This indicates that CF+DE is robust and can guarantee a reliable detection performance even though the number of faulty beacons increases.

4) Unpredictable Scenario: Practically, the attached beacons might stop working at an unexpected time. Furthermore,

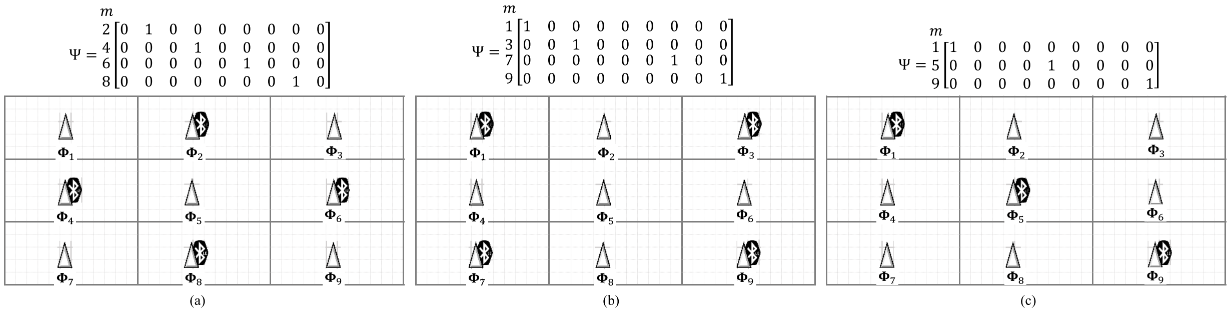


Fig. 10. Three different scenarios to illustrate the faulty beacons attached with (a) ToIs 2, 4, 6, and 8, (b) ToIs 1, 3, 7, and 9, and (c) ToIs 1, 5, and 9.

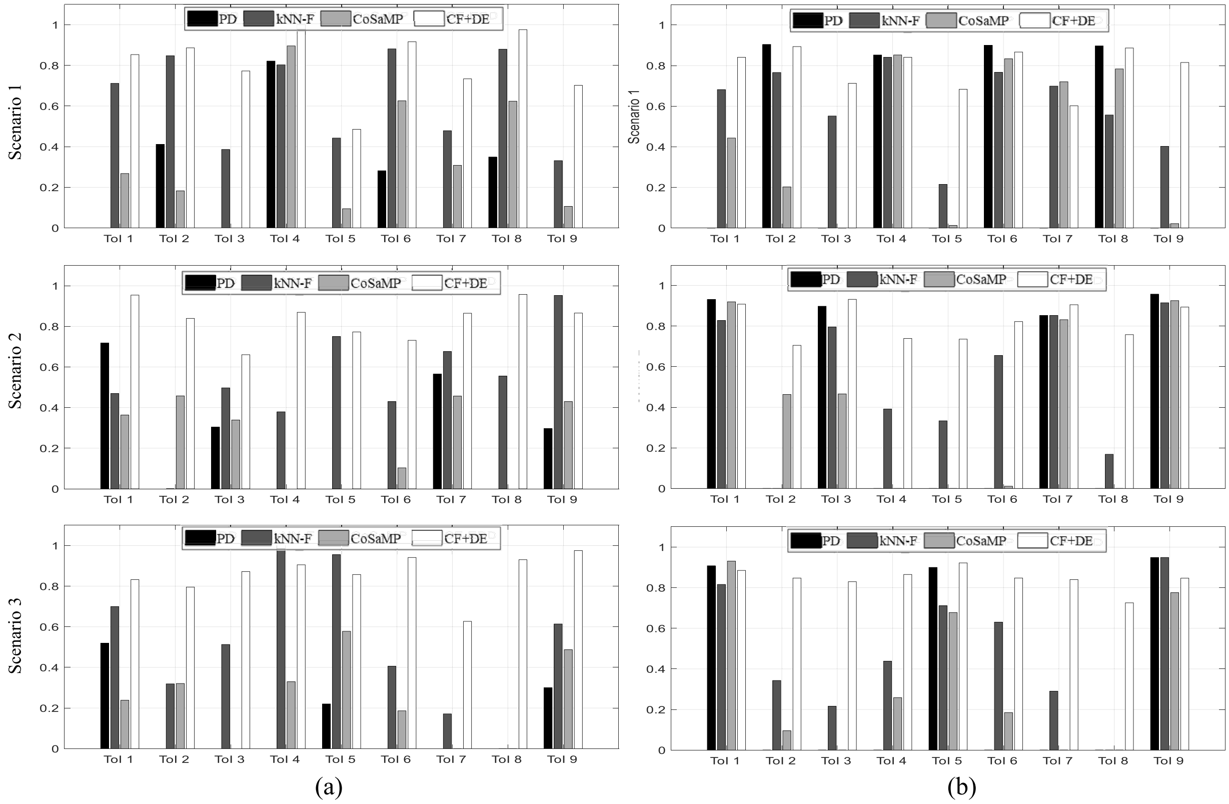


Fig. 11. Detection performances of PD, kNN-F, CoSaMP, and CF+DE for all three scenarios illustrated in Fig. 10. (a) Precision. (b) Recall.

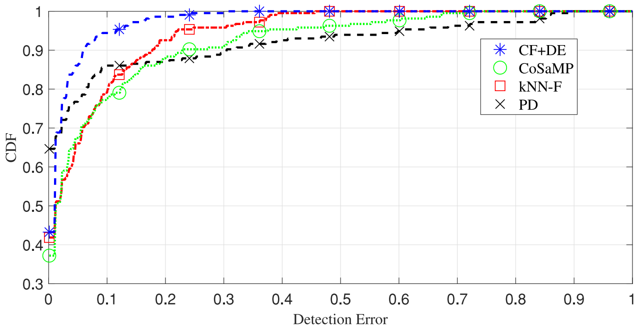


Fig. 12. Cumulative distribution of the detection error.

due to various environmental factors, such as a change in weather, or the movement of the human body, might affect the signal reception rate in the smartphone. All these factors lead to an unpredictable scenario, i.e., at a certain point in time, the ToI which gives stable signals might appear to be unavailable at another time. To better capture all these uncertainties, we performed another simulation by randomly discarding some beacons' signals. The detection errors produced by each approach is computed, and the results are shown in Fig. 12. Clearly, our proposed CF+DE outperforms the rest with very low detection error, i.e., < 0.15 error for $> 90\%$ of the time.

VII. Practical Implementation

This section presents a practical implementation in a multistorey car park setting. We performed two experiments: first experiment consists of 18 beacons attached to 18 known ToIs (i.e., the parking lots) and second experiment consists of the same 18 ToIs plus some randomly added unknown ToIs.

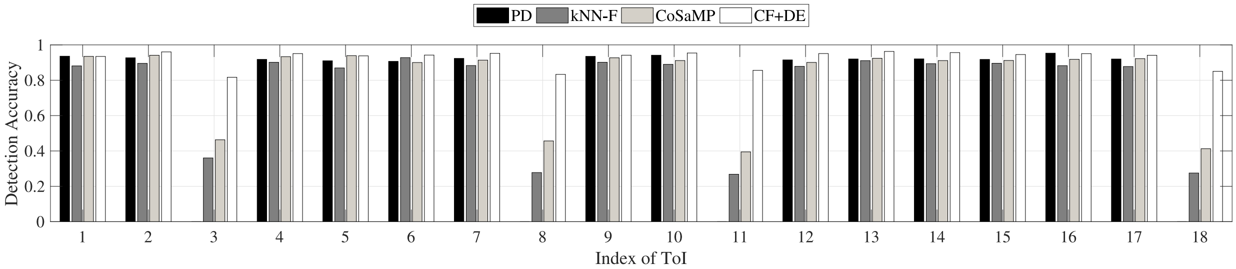


Fig. 13. Detection performance of each approach for the car park scenario illustrated in Fig. 14.

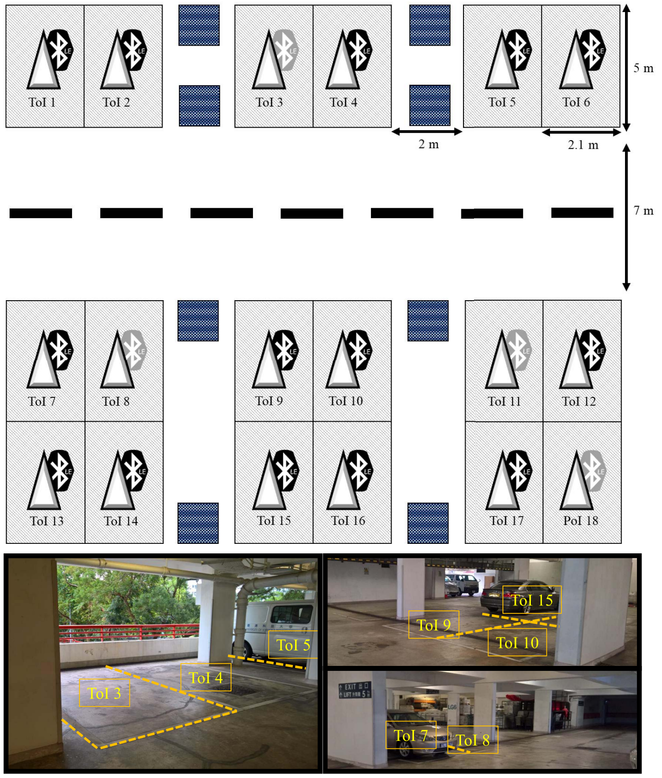


Fig. 14. Practical experiment with 18 ToIs deployed in a multistorey car park space.

A. Fixed ToIs With Parking Lots

Fig. 14 depicts the deployment of the 18 ToIs. Each ToI represents a parking lot, in which each parking lot is designated with a specific service. For example, one parking lot might contain the details of a reservation or provide an interactive service for the driver to pay for the parking fee if it is available. Each beacon was configured to broadcast their signals at an interval of 100 ms. An Android-based smartphone was implemented with our proposed CF+DE as well as the other three baseline approaches for on-the-spot experiments. We performed the detection from one parking lot to another in a descending sequence. The same experiment was repeated 20 times by intentionally shutting off the beacons attached to ToIs 3, 8, 11, and 18. The detection outputs were recorded; given the detection outputs, the detection accuracy is computed and the results are plotted in Fig. 13.

From Fig. 13, it is obvious that PD fails to provide correct detection at ToIs 3, 8, 11, and 18. However, the other three approaches are still able to detect these four ToIs even when their attached beacons were not working. On average, the detection accuracies for PD, kNN-F, CoSaMP, and CF+DE are 71.92%, 75.94%, 81.20%, and 92.43%, respectively. Clearly, our proposed CF+DE outperforms the rest with very high detection accuracy, such a significant performance is even more obvious in those ToIs without a working beacon attached.

B. Randomly Added ToIs

With the same set of 18 beacons in the multistory car park space, we randomly added a few beacons from time to time. The added beacons will increase the size of the observation at the smartphone. These random beacons are assumed to attach with unknown ToIs. This is valid for many practical scenarios where ToI can be any moving object. In a car park space, for example, we might have cars coming in and out, and some cars might be attached with some beacons for personalized private service. One example is that Tesla car has installed a beacon to provide automatic car unlock service to the owner when the owner approaching their own car. The main objective in this experiment is that can we still detect the same 18 fixed ToIs in the presence of random and unknown ToIs. Note that the presence of unknown ToIs will increase the size of the observation vector and possibly confuse the detection process.

In this experiment, we first examined the algorithms' runtime and then evaluate their detection performance. In general, we logged the timestamp when the smartphone starts the detection process and also the timestamp when the detection output is returned. Based on these starting and stopping timestamp, the runtime can be computed. Similarly, we logged the detection output in every step. All the information is stored in the local smartphone storage as a ".csv" file.

1) Algorithms' Runtime Versus M: In general, CoSaMP requires more computation in comparison to the other three algorithms; whereas PD requires the least computation. Fig. 15(a) shows the runtime of the four algorithms. Even though our proposed CF+DE has a longer runtime compared to kNN-F and PD, it is still faster than CoSaMP. Furthermore, CoSaMP shows an increment on runtime when the size of the observation vector increases from 0 to 10. However, our CF+DE has maintained a consistent runtime (i.e., about 2.148 μs in average) regardless of the size of the observation.

2) Detection Accuracy Versus M: Even though the runtime of PD is the fastest compared to the rest, as shown in Fig. 15(b), it failed to return a good detection performance

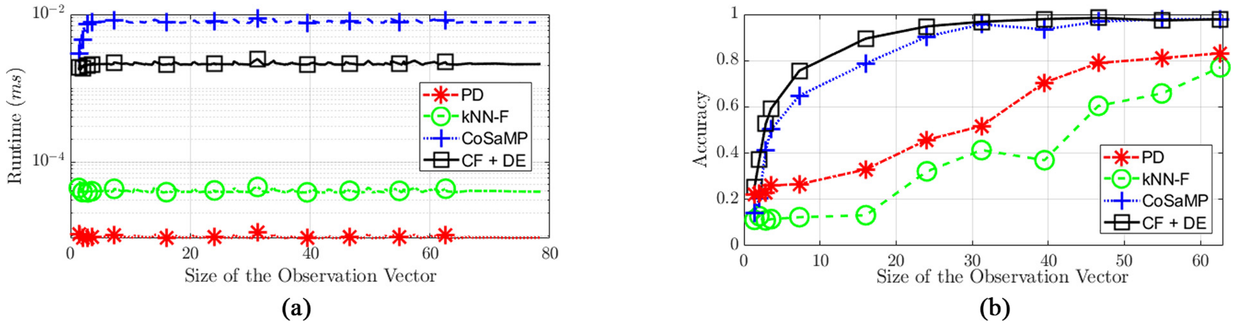


Fig. 15. (a) Runtime of each algorithm with respect to the observation's size. (b) Detection accuracy with respect to the observation's size.

when there is not enough observation for its to detect the closest ToI. Furthermore, the unpredictable RSS fluctuations may also confuse PD in making the right detection when the size of observation increases. On the other hands, our proposed CF+DE achieves very good performance even when the size of the observation is very small. The detection performance increases when the size of the observation increases. In particular, CF+DE achieves at least 80% accuracy when $M = 10$; whereas CoSaMP requires at least $M > 18$ to achieve similar performance.

While the increment of the observation might due to the presence of the unknown ToIs, the observation contributed by these unknown ToIs can, in fact, provide some useful information to narrow down the search space, and thus increase the detection performance. Intuitively, when there are more observations, we can have more linear equations and thus reducing the vagueness of those unknown coefficients. In other words, if our algorithm can solve the unknown coefficients with least number of linear equations with high probability, then our algorithm can definitely perform much better when it is provided with more linear equations.

VIII. Conclusion

This paper exploits CS approach to address the underdetermined proximity detection problem. More specifically, given an incomplete observation, our goal is to search for a proximity indicator vector which helps the smartphone to identify the target ToI. To ensure fast and reliable detection, this paper proposes a CF+DE to first narrow down the search scope before searching for the optimal solution. Extensive experiments with real-world data show that our proposed CF+DE outperforms other common algorithms [including the algorithms widely used by commercial applications (PD), machine learning (kNN), and other CS-based problems (CoSaMP)]. A real implementation in a multistorey parking space demonstrates the feasibility of our proposed approach for diverse practical PBS use cases involving smartphones and physical objects/spaces.

References

[1] H. Zou *et al.*, "BlueDetect: An iBeacon-enabled scheme for accurate and energy-efficient indoor-outdoor detection and seamless location-based service," *Sensors*, vol. 16, no. 2, p. 268, 2016.

[2] M. Bolic, M. Rostamian, and P. M. Djuric, "Proximity detection with RFID: A step toward the Internet of Things," *IEEE Pervasive Comput.*, vol. 14, no. 2, pp. 70–76, Apr./Jun. 2015.

[3] E. J. Candes and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, Mar. 2008.

[4] M. Leigsnering, F. Ahmad, M. G. Amin, and A. M. Zoubir, "Compressive sensing-based multipath exploitation for stationary and moving indoor target localization," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 8, pp. 1469–1483, Dec. 2015.

[5] C. Feng, W. S. A. Au, S. Valaee, and Z. Tan, "Received-signal-strength-based indoor positioning using compressive sensing," *IEEE Trans. Mobile Comput.*, vol. 11, no. 12, pp. 1983–1993, Dec. 2012.

[6] Y. Agata, J. Hong, and T. Ohtsuki, "Room-level proximity detection based on RSS of dual-band Wi-Fi signals," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2016, pp. 1–6.

[7] M. J. Hussain, L. Lu, and S. Gao, "An RFID based smartphone proximity absence alert system," *IEEE Trans. Mobile Comput.*, vol. 16, no. 5, pp. 1246–1257, May 2017.

[8] A. Montanari, S. Nawaz, C. Mascolo, and K. Sailer, "A study of Bluetooth low energy performance for human proximity detection in the workplace," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. (PerCom)*, Kona, HI, USA, Mar. 2017, pp. 90–99.

[9] M. M. Scheunemann, K. Dautenhahn, M. Salem, and B. Robins, "Utilizing Bluetooth low energy to recognize proximity, touch and humans," in *Proc. 25th IEEE Int. Symp. Robot Human Interact. Commun. (RO-MAN)*, New York, NY, USA, Aug. 2016, pp. 362–367.

[10] S. Liu, Y. Jiang, and A. Striegel, "Face-to-face proximity estimationusing Bluetooth on smartphones," *IEEE Trans. Mobile Comput.*, vol. 13, no. 4, pp. 811–823, Apr. 2014.

[11] E. Novak and Q. Li, "Near-pri: Private, proximity based location sharing," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Toronto, ON, Canada, Apr. 2014, pp. 37–45.

[12] M. Li, S. Yu, N. Cao, and W. Lou, "Privacy-preserving distributed profile matching in proximity-based mobile social networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 5, pp. 2024–2033, May 2013.

[13] N. Allurwar, B. Nawale, and S. Patel, "Beacon for proximity target marketing," *Int. J. Eng. Comput. Sci*, vol. 5, no. 5, pp. 16359–16364, 2016.

[14] A. Thamm, J. Anke, S. Haugk, and D. Radic, "Towards the omni-channel: Beacon-based services in retail," in *Proc. Int. Conf. Bus. Inf. Syst.*, 2016, pp. 181–192.

[15] P. C. Ng, J. She, and S. Park, "Notify-and-Interact: A beacon-smartphone interaction for user engagement in galleries," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Hong Kong, Jul. 2017, pp. 1069–1074.

[16] S. Alletto *et al.*, "An indoor location-aware system for an IoT-based smart museum," *IEEE Internet Things J.*, vol. 3, no. 2, pp. 244–253, Apr. 2016.

[17] K. E. Jeon, J. She, P. Soonsawad, and P. C. Ng, "BLE beacons for Internet of Things applications: Survey, challenges, and opportunities," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 811–828, Apr. 2018.

[18] F. Zafari, I. Papapanagiotou, M. Devetsikiotis, and T. J. Hacker, "Enhancing the accuracy of iBeacons for indoor proximity-based services," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–7.

[19] R. Faragher and R. Harle, "Location fingerprinting with Bluetooth low energy beacons," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 11, pp. 2418–2428, Nov. 2015.

[20] S. Yiu and K. Yang, "Gaussian process assisted fingerprinting localization," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 683–690, Oct. 2016.

[21] B. Wang, Q. Chen, L. T. Yang, and H.-C. Chao, "Indoor smartphone localization via fingerprint crowdsourcing: Challenges and approaches," *IEEE Wireless Commun.*, vol. 23, no. 3, pp. 82–89, Jun. 2016.

[22] S. He, B. Ji, and S.-H. G. Chan, "Chameleon: Survey-free updating of a fingerprint database for indoor localization," *IEEE Pervasive Comput.*, vol. 15, no. 4, pp. 66–75, Oct./Dec. 2016.

[23] J. Decuir, "Bluetooth smart support for 6LoBTLE: Applications and connection questions," *IEEE Consum. Electron. Mag.*, vol. 4, no. 2, pp. 67–70, Apr. 2015.

[24] M. Collotta, G. Pau, T. Talty, and O. K. Tonguz, "Bluetooth 5: A concrete step forward toward the IoT," *IEEE Commun. Mag.*, vol. 56, no. 7, pp. 125–131, Jul. 2018.

[25] M. Ayadi and A. B. Zineb, "Body shadowing and furniture effects for accuracy improvement of indoor wave propagation models," *IEEE Trans. Wireless Commun.*, vol. 13, no. 11, pp. 5999–6006, Nov. 2014.

[26] J. Yang, X. Wang, S. I. Park, and H. M. Kim, "Optimal direct path detection for positioning with communication signals in indoor environments," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2012, pp. 4798–4802.

[27] J. Talvitie, M. Renfors, and E. S. Lohan, "Distance-based interpolation and extrapolation methods for RSS-based localization with indoor wireless signals," *IEEE Trans. Veh. Technol.*, vol. 64, no. 4, pp. 1340–1353, Apr. 2015.

[28] C. Liu *et al.*, "RSS distribution-based passive localization and its application in sensor networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 4, pp. 2883–2895, Apr. 2016.

[29] X. Tian, R. Shen, D. Liu, Y. Wen, and X. Wang, "Performance analysis of RSS fingerprinting based indoor localization," *IEEE Trans. Mobile Comput.*, vol. 16, no. 10, pp. 2847–2861, Oct. 2017.

[30] M. S. Aman, H. Jiang, C. Quint, K. Yelamarthi, and A. Abdelgawad, "Reliability evaluation of iBeacon for micro-localization," in *Proc. IEEE Annu. Ubiquitous Comput. Electron. Mobile Commun. Conf. (UEMCON)*, New York, NY, USA, 2016, pp. 1–5.

[31] C. Xiao, D. Yang, Z. Chen, and G. Tan, "3-D BLE indoor localization based on denoising autoencoder," *IEEE Access*, vol. 5, pp. 12751–12760, 2017.

[32] F. Yin, Y. Zhao, F. Gunnarsson, and F. Gustafsson, "Received-signal-strength threshold optimization using Gaussian processes," *IEEE Trans. Signal Process.*, vol. 65, no. 8, pp. 2164–2177, Apr. 2017.

[33] M. Alvarez-Diaz, R. Lopez-Valcarce, and C. Mosquera, "SNR estimation for multilevel constellations using higher-order moments," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1515–1526, Mar. 2010.

[34] A. Stephenne, F. Bellili, and S. Affes, "Moment-based SNR estimation for SIMO wireless communication systems using arbitrary QAM," in *Proc. Conf. Rec. 41st Asilomar Signals Syst. Comput.*, Nov. 2007, pp. 601–605.

[35] R. Lopez-Valcarce and C. Mosquera, "Sixth-order statistics-based non-data-aided SNR estimation," *IEEE Commun. Lett.*, vol. 11, no. 4, pp. 351–353, Apr. 2007.

[36] J.-F. Cardoso, "Source separation using higher order moments," in *Proc. Int. Conf. Acoust. Speech Signal Process.*, vol. 4, May 1989, pp. 2109–2112.

[37] P. C. Ng, L. Zhu, J. She, R. Ron, and S. Park, "Beacon-based proximity detection using compressive sensing for sparse deployment," in *Proc. 18th IEEE Int. Symp. World Wireless Mobile Multimedia Netw.*, 2017, pp. 1–6.

[38] E. J. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5406–5425, Dec. 2006.

[39] Z. Michalewicz, D. Dasgupta, R. G. Le Riche, and M. Schoenauer, "Evolutionary algorithms for constrained engineering problems," *Comput. Ind. Eng.*, vol. 30, no. 4, pp. 851–870, 1996.

[40] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 82–102, Jul. 1999.

[41] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4–31, Feb. 2011.

[42] D. Li, B. Zhang, and C. Li, "A feature-scaling-based *k*-nearest neighbor algorithm for indoor positioning systems," *IEEE Internet Things J.*, vol. 3, no. 4, pp. 590–597, Aug. 2016.

[43] C. Zhou, H. Xie, and J. Shi, "Wi-Fi indoor location technology based on *k*-means algorithm," in *Proc. LISS*, 2015, pp. 765–770.

[44] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.

[45] M. A. Davenport, D. Needell, and M. B. Wakin, "Signal space CoSaMP for sparse recovery with redundant dictionaries," *IEEE Trans. Inf. Theory*, vol. 59, no. 10, pp. 6820–6829, Oct. 2013.

Pai Chet Ng received the B.S. degree in telecommunication engineering from Multimedia University, Cyberjaya, Malaysia. She is currently pursuing the Ph.D. degree at the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology (HKUST), Hong Kong.

She was a Research Engineer prior to joining the HKUST-NIE Social Media Laboratory. Her current research interests include Internet of Things analytics, low power wireless sensing, and networking.

James She (M'12) is an Assistant Professor with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology (HKUST), Hong Kong, and a Visiting Research Fellow with the University of Cambridge, Cambridge, U.K. He is also the Founding Director of Asia's first social media laboratory, HKUST-NIE Social Media Laboratory. His current research interests include analytics and systems for social media and multimedia big data, smart/wearable device-based interactive technologies in cyber-physical systems or Internet of Things infrastructures, and new media technologies and productions for art and culture.

Prof. She has been an Associate Editor for the *ACM Transactions on Multimedia Computing, Communications, and Applications* since 2016.

Rong Ran (S'05–M'10) received the Ph.D. degree from Yonsei University, Seoul, South Korea, in 2009.

In 2009, she joined ETRI, Daejeong, South Korea, and performed research on the IEEE 802.16m standardization. She was a Research Associate with the Hong Kong University of Science and Technologies, Hong Kong, in 2010. She is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Ajou University, Suwon, South Korea. Her current research interests include wireless communication and networks, compressive sensing, and machine learning.