Accurate and Efficient Indoor Location by Dynamic Warping in Sequence-Type Radio-map

XUEHAN YE, Renmin University of China
YONGCAI WANG[†], Renmin University of China
YUHE GUO, Renmin University of China
WEI HU, Princeton University
DEYING LI, Renmin University of China

An efficient way to overcome the calibration challenge and RSS dynamics in radio-map-based indoor localization is to collect *radio signal strength (RSS)* along indoor paths and conduct localization by sequence matching. But such sequence-based indoor localization suffers problems including indoor path combinational explosion, random RSS miss-of-detection during user movement, and user moving speed disparity in online and offline phases. To address these problems, this paper proposes an undirected graph model, called *WarpMap* to efficiently calibrate and store the sequence-type radio-map. It reduces RSS sequence signature storage complexity from $O(2^N)$ to $O(N)$ where N is the number of path crosses. An efficient on-line candidate path extraction algorithm is developed in it to find a set of the most possible candidate paths for matching with the on-line collected RSS sequence. Then, to determine the user's exact location, a sub-sequence dynamic time warping (SDTW) algorithm is proposed, which matches the online collected RSS sequence with the sequential RSS signatures of the candidate paths. We show the SDTW algorithm is highly efficient and adaptive, which localizes user without backtracking of warping path. Extensive experiments in office environments verified the efficiency and accuracy of WarpMap, which can be calibrated within thirty minutes by one person for $1100m^2$ area and provides overall nearly 20% accuracy improvements than the state-of-the-art of radio-map method.

Additional Key Words and Phrases: Sequence-type radio-map, Dynamic Time Warping, Indoor Location

ACM Reference format:
Xuehan Ye, Yongcai Wang[†], Yuhe Guo, Wei Hu, and Deying Li. 2018. Accurate and Efficient Indoor Location by Dynamic Warping in Sequence-Type Radio-map. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 1, Article 50 (March 2018), 22 pages.
https://doi.org/10.1145/3191782

1 INTRODUCTION

Radio-map based indoor locating, which characterizes each location-of-interest by the radio signal strength (RSS) signature at that location, has attracted great attentions [2][5][38][12]. Compared with other locating techniques, it provides key advantages including: 1) purely software-implementable on mobile phones without requiring

[†]Corresponding author: Yongcai Wang, e-mail: ycw@ruc.edu.cn.

This work was supported in part by the National Natural Science Foundation of China Grant No. 11671400, 61672524; the Fundamental Research Funds for the Central University, and the Research Funds of Renmin University of China, 2015030273.

2474-9567/2018/3-ART50 $15.00
https://doi.org/10.1145/3191782

additional hardware infrastructure; 2) privacy protection for working in navigation mode; and 3) reasonable accuracy with errors around 2 - 3 meters .

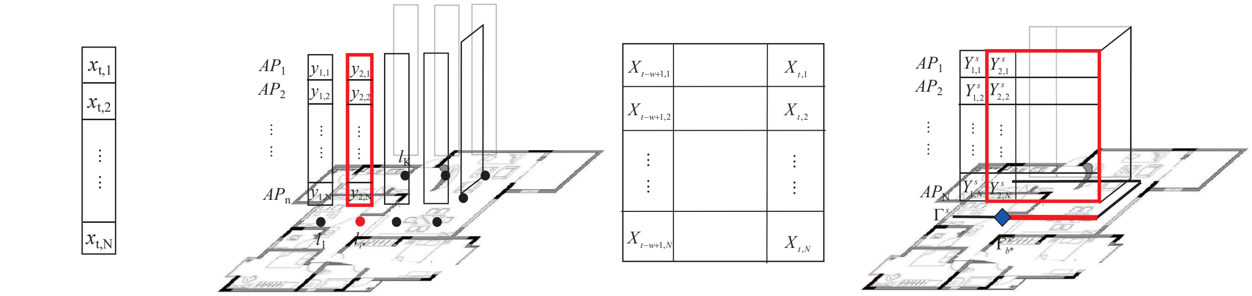
Despite of the advantages, several practical issues must be considered for practical applications: 1) it is very laborious to calibrate a fine-grained radio-map, especially for large application areas; 2) the point-type matching at discrete locations is sensitive to random online RSS noises and environment dynamics. To overcome these problems, previous works have devoted deliberated efforts to exploit different kinds of information to reduce radio-map calibration cost and to improve the positioning accuracy and robustness.

A major approach to reduce calibration cost is to exploit environment signature [31], including magnetic fluctuation, illumination intensity at specific spots as internal landmarks [29], and used dead reckoning by mobile phones [6] to track the inertial landmarks to conduct locating, so as to avoid the manually radio-map calibration. Another major approach exploited automatic labeling [35], which leveraged dead reckoning by motion sensors to construct a radio-map firstly in the radio space, then stretch and associate the radio-space geometry to physical space by geometrical matching using the path information from the floor-plan. Other approaches also exploited Expectation-Maximisation (EM) and Manifold methods to learn radio-map parameters by training parametric radio-map models [22][36]. These methods, however, generally require additional sensors or depend heavily on the accuracy of dead reckoning, which is known inaccurate by using commodity mobile phones [6][21]. The inaccuracy of radio-map model may also degrade the performance of locating.

Compared with these approaches, a more straightforward way to tailor RSS noises and to reduce calibration cost is to collect RSS sequence along indoor paths to build sequence-type radio-map. This sequence-matching idea was firstly exploited in [26], which proposed geomagnetic sequence collected on a path as the path's digital signature. They showed that sequence matching outperformed point matching in both accuracy and reliability. However, as stated in the paper, *scalability* is a key challenge for sequence-based localization, and the paper considered only a small number of paths. When indoor navigation on all paths is considered, the number of indoor paths grow exponentially with the number of path crosses, i.e., in the order of $O(2^N)$ where N is the number of path crosses (See Section II). This requires efficient model for calibrating and storing the sequential signatures for all paths. Secondly, the moving speeds and moving patterns of users in offline sequence collection phase and online navigation phase are generally different, leading to sequence misalignment during matching. Adaptive and efficient sequence matching algorithm to tolerate the moving pattern differences is required. Thirdly, WiFi RSS is widely available and more descriptive signature for characterizing the indoor paths. How to exploit RSS sequence signature for scalable and accurate indoor localization remains a promising and challenging problem.

This paper proposes to model the RSS sequence signatures of indoor path by an undirected *trace-graph*. Each vertex in the graph models the RSS sequence of a path segment, and the edges model the spatial adjacency of the vertexes. This trace-graph can be trained by a user traversing the indoor paths once and overcomes the path combinatorial explosion problem. The sequential RSS signatures of a map with at most $O(2^N)$ paths can be stored by a graph of $O(N)$ nodes. Based on the trace-graph, an efficient candidate path extraction algorithm is proposed, which extracts $O(1)$ number of candidate paths based on the on-line collected RSS sequence. Finally, to determine the exact location of the user, a *subsequence dynamic time warping (SDTW)* algorithm is proposed to conduct on-line localization. The SDTW algorithm can adaptively localize the user to an end point of a path segment in the candidate path set, without the need of traditional DTW's backtracking. Further, the random RSS missing problem is addressed by *collaborative filter (CF)*, to smooth the measured RSS values, and to fill the missing RSS values.More specifically, our contributions include:

(1) A WarpMap model represented by undirected graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, where $\mathbf{V}$ represents RSS signatures on path segments and $\mathbf{E}$ represents adjacency of path segments.
(2) A fast calibration process to build $\mathbf{G}$ by the user traversing indoor routes for once, and Collaborative Filter method to smooth the noisy RSS data.



(a) Online measuring RSS vector x_t

(b) Point radio-map & vector matching

(c) Online measuring RSS matrix X

(d) Sequence radio-map & Matrix matching

Fig. 1. Comparison between point-type radio-map and sequence-type radio-map.

(3) In online phase, a potential path extraction algorithm to extract from **G** the potential paths that the target maybe undergoing, based on the real-time detected *access points (APs)*, which is called *candidate sequence set (CSS)*.

(4) A *subsequence dynamic time warping (SDTW)* algorithm to find a subsequence in CSS which has the least warping distance to the online measured RSS sequence within a short time window.

(5) Investigations of different warping distance functions and different CSS selection methods including Principle Component Analysis (PCA) etc.

(6) System implementation and extensive experiments in different environments using different brands of cell phones, which verified the efficiency and accuracy improvement than the state-of-the-art (point-type radio-map + K-nearest neighbor/particle filter) locating method.

The rest of this paper is organized as follows. Background and problem model are introduced in Section 3. The trace-graph construction is introduced in Section 4. Online locating by SDTW is introduced in Section 5. Performance evaluation by a prototype system is presented in Section 7. The paper is concluded in Section 8.

2 RELATED WORK

Radio-map based locating is essentially a pattern-matching based approach. The seminal work is RADAR [2], proposed in 2001, to use radio frequency identification (RFID) signature for indoor locating. After that, various efforts have been devoted into this area. The major related works fall into three research categories: 1) reducing the radio-map calibration efforts; 2) improving the location accuracy; 3) improve radio-map adaptivity.

2.1 Reduce the radio map calibration efforts

One key problem in indoor locating is how to reduce the radio-map calibration cost, because it is very laborious to train the radio-map, especially for the large environment. A major approach to reduce calibration cost is unsupervised indoor locating method [31], which exploited environment signature. Scholl et al. [24] proposed fast indoor radio-map building for RSS based indoor locating by using hand-held laser mapping device for building floor plan and radio-map simultaneously. Geng et al. [4] proposed hybrid radio-map for indoor locating, which reduce the radio-map training efforts by the aid of sparsely deployed ultrasound ranging system. Molina-GarcÃŋa et al. [17] proposed to enhance in-building fingerprint by femtocell networks. Tian et al. [29] exploited illumination intensity at specific spots as internal landmarks. Harle et al. [6] used dead reckoning by mobile phones to track the inertial landmarks to conduct locating, so as to avoid the manually radio-map calibration.

Another major approach exploited automatic labeling Yang et al. [35] leveraged dead reckoning by motion sensors to construct a radio-map firstly in the radio space, then stretch and associate the radio-space geometry to physical space by geometrical matching using the path information from the floor-plan. Other approaches also exploited Expectation-Maximisation (EM) and Manifold methods to learn radio-map parameters by training parametric radio-map models [22][36].

2.2 Improve the locating accuracy

From RADAR [2], a series of works focused on improving the locating accuracy. Some approaches used enhanced online locating algorithms to tolerate random RSS noises in the locating process. Haque et.al. [5] proposed LEMON, which enhances K-nearest neighbor (KNN) approach be mining the oversampled neighborhoods. Wu et.al. [32] exploited Support Vector Machine (SVM) in radio-map. Horus [38] proposed a probabilistic radio-map model, in which the probability density of the RSS signatures are collected and stored as radio map. Compass [12] is also an probabilistic radio-map model, which also leverages the object orientation to improve location accuracy. Other major approaches used information fusion techniques, which exploited the motion continuity information, floor map information to design Bayesian filter [5], particle filter [8], and Markov Random Field models [33] to narrow down the search space to improve the locating accuracy against noises. Zampella et al. [39] proposed the use of a particle filter to fuse foot mounted inertial measurements with radio-map. A recent work by Herrera et al. [1] proposed the fusion of radio-map and IndoorOSM floor plan for accurate indoor locating. Geng et al. [4] proposed hybrid radio-map method to reduce calibration cost and to improve location accuracy. Shu et al. [26] first applied sequence-matching idea in radio-map to improve locating accuracy. But *scalability* is a key challenge for sequence-based localization. A comparative study of radio-map based location accuracy performance can be referred to [9].

2.3 Improve the radio map adaptivity

Even if the radio-map was offline calibrated, it maybe outdated due to the environment change. How to design adaptive radio-map to tolerate the environment impacts is an important problem. Ji et al. [10] investigated the impact of building environment on the performance of dynamic indoor location. Ni et al. [19] proposed to use landmark RFIDs to measure RSS signatures online to make the radio-map be adaptive to environments. Yin et al. [36, 37] proposed adaptive temporal radio-map model by learning algorithms. Pan et al. [22] proposed adaptive localization in dynamic environment using multi-view learning. Lo et al. [15] proposed adaptive radio maps for pattern-matching based localization via inter-beacon co-calibration. Yang et al. [34] proposed AdaMap, which use linear regression model to represent the radio-map and online adapts the model coefficients by online learning.

However, note all existing work mainly use point-type radio-map to represent the RSS signatures of specific locations. The spatial dependency among the RSS signatures are rarely utilized. In this paper, we explore the trace-type fingerprint to further improve the locating accuracy and robustness.

3 BACKGROUND AND PROBLEM MODEL

3.1 Point-type radio-map

In traditional point-type radio-map, let $\mathbf{L} = \{\mathbf{l}_1, \mathbf{l}_2, \cdots, \mathbf{l}_K\}$ denote K points of interest. The RSS signatures of these points are offline trained and stored as $\mathbf{M_p} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_K\}$, where $\mathbf{y}_i$ is the captured RSS vector at location $\mathbf{l}_i$; In online locating, the online collected RSS signature is $\mathbf{x}_t$ at time t. Point-type matching is to find a location whose RSS signature in $\mathbf{M_p}$ matches best with $\mathbf{x}_t$.

$$l_i^* = \underset{(i):\mathbf{y}_i \in \mathbf{M_p}}{\arg\min}\ \text{Dist}\left(\mathbf{x_t}, \mathbf{y}_i\right). \tag{1}$$

where $\mathrm{Dist}(\mathbf{x}_t, \mathbf{y}_i)$ measures the similarity between $\mathbf{x}_t$ and $\mathbf{y}_i$. Fig. 1(a)(b) illustrate the locating process of point-matching, which is indeed carried out by matching of RSS vectors. For clarity, the notations used in this paper are listed in Table 1.

3.2 Sequence-type radio-map

In sequence-type radio-map, the RSS signatures along indoor paths are offline calibrated. Let $\Gamma = \{\Gamma^1, \cdots, \Gamma^K\}$ be the set of calibrated paths; $\mathbf{M_q} = \{\mathbf{Y}^1, \cdots, \mathbf{Y}^K\}$ are the captured RSS sequences for these paths. The row length for $\mathbf{Y}^s$ (i.e, signatures for path Γ^s) is the number of detectable APs on the path, and its column length m_s is the number of sample points on path Γ^s. In online locating, a target measures RSS sequences within a moving time window $\{t-w+1, \cdots, t\}$. This forms a online measured RSS matrix $\mathbf{X} = (\mathbf{X}_{t-w+1}, \cdots, \mathbf{X}_t)$, whose column length is w, and t is the current time. Sequence matching is to find the best match between $\mathbf{X}$ and a subsequence $\mathbf{Y}^{s^*}_{[a^*, b^*]}$:

$$(s^*, a^*, b^*) = \mathop{\arg\min}_{(s,a,b):\mathbf{Y}^s \in \mathbf{M_q}, 1 \le a \le b \le m_s} \mathrm{Dist}\left(\mathbf{X}, \mathbf{Y}^s_{[a,b]}\right). \tag{2}$$

where $\mathbf{Y}^s_{[a,b]}$ is a subsequence of $\mathbf{Y}^s \in \mathbf{M_q}$. a and b are the start point and the end point of the subsequence.

$\mathbf{X}$ is matched to a subsequence of $\mathbf{Y}^s$ because the online moving window is generally shorter than the offline trained sequence. The real-time location of the target is given by $\Gamma^{s^*}_{b^*}$, which is the end point of the best matched sub-path. Fig. 1(c)(d) illustrate the locating principle of localization by sequence radio-map. It can be seen that

Table 1. List of notations and explanation

Notations	Explanation
Γ^s	a calibrated path i
Γ^s_b	the bth sample point of Γ^s
$\mathbf{Y}^s$	the RSS sequence collected on path Γ^i
$\mathbf{Y}^s_i$	the ith sample point in sequence $\mathbf{Y}^s$
$\mathbf{Y}^s_{[a,b]}$	subsequence of $\mathbf{Y}^s$ from $\mathbf{Y}^s_a$ to $\mathbf{Y}^s_b$
$\mathbf{Y}^s_{i,j}$	the RSS value of jth AP of ith sample point in $\mathbf{Y}^s$
$\mathbf{X}$	RSS sequence in collected online moving window
$\mathbf{X}_t$	the tth sample point in $\mathbf{X}$
m_s	length of RSS matrix $\mathbf{Y}^s$
w	length of moving window
$\mathbf{B}^x$	the efficient AP union set in the moving window
$\mathbf{B}^s$	the efficient AP union set of $\mathbf{Y}^s$
$\mathbf{G}$	the trace-graph
$\mathbf{V}$	the vertex set in $\mathbf{G}$
$\mathbf{E}$	the edge set in $\mathbf{G}$
$\mathbf{B}^i$	the efficient AP union set of vertex i
N	the number of path crosses in $\mathbf{G}$
N_V	the size of $\mathbf{V}$
N_E	the size of $\mathbf{E}$
M	the size of *candidate vertex set (CSV)* extracted from $\mathbf{V}$
$\mathcal{P}$	*candidate sequence set (CSS)* generated based on CVS
L	the size of CSS
p	the size of extracted principal components from $\mathbf{B}^x \cap \mathbf{B}^s$

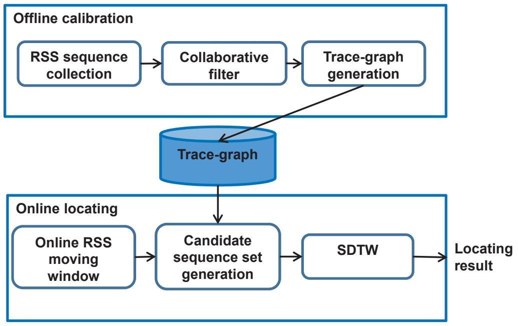
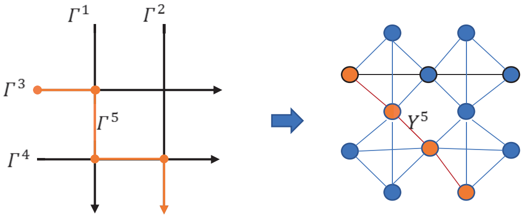


Fig. 2. System structure of WarpMap.



(a) A combined path extracted from original map.

(b) Corresponding path in sequence-type radio-map.

Fig. 3. Combinational explosion of sequence-type radio-map.

sequence matching is essentially to find the best match for the RSS matrix **X**. But note that the detectable APs in different locations maybe different, therefore, online locating needs to firstly search candidates paths by the list of detectable APs, and then to determine the target location by sequence matching. These will be detailed in Section 5.

It can be understood intuitively that sequence-type radio-map implicitly embeds the motion continuity and the path information into the radio-map, which is promising to provide better location accuracy and robustness. But key challenges prevent the wide adoption of sequence type radio-map.

(1) Combinational explosion of indoor paths causes the sequence-type radio-map hard to enumerate and calibrate, which is illustrated in Fig. 3. The red line shows a combined path extracted from sequence-type radio-map. We prove in Section 4 that the number of combinatorial paths extracted from a indoor map **G** is in the order of $O(2^N)$, where N is the number of crossing points.
(2) RSS miss-of-detection and RSS noises are remarkable during user movement, leading highly “dirty" radio-map by mobile calibration. One example is shown in Fig. 4. Fig. 4(a) shows the point-type radio-map captured for a room. Fig. 4(b) shows the sequence-type radio-map captured by a user walking recursively in the room. Compared with the point-type radio-map, RSS values are randomly miss-of-detection and are highly noisy in the sequence-type radio-map.
(3) The users’ moving speeds, moving directions may be different in training phase and online phase, causing RSS sequences hard to be aligned in sequence matching.

3.3 Working Flow of Warp-map

We propose WarpMap to address these difficulty, which includes efficient model and algorithms to address these difficulties. The working flow of WarpMap is shown in Fig. 2. It contains an offline phase and an online phase.

- In offline phase, the RSS signatures along a set of indoor paths are calibrated to construct a sequence-type radio-map. The RSS sequence is firstly smoothed by an proposed collaborative filter for noise removing and empty item filling. Then a trace-graph model is proposed to store the RSS signatures of the paths.
- In online phase, RSS signatures are collected into a moving window when a user is walking along indoor paths. A *candidate sequence set (CSS)* is extracted from the stored radio-map based on AP-list matching, which reduces the searching space of sequence matching. Finally, the location of the user is calculated by a proposed subsequence dynamic warping algorithm.

We assume the indoor map can be obtained in advance, e.g. by Google Map [16] or Baidu Map [3], which is increasingly available for public spots, such as shopping mall. The storage complexity in offline phase and the computation complexity in online matching are key problems for scalability, while the location accuracy and robustness are improved by subsequence dynamic time warping.

4 TRAINING: TRACE-GRAPH

An undirected graph model, i.e., *trace-graph* is proposed for calibrating sequence-type radio-map using little human efforts, while avoiding the path combinatorial explosion problem. We firstly introduce the main idea of trace-graph construction.

4.1 Trace-Graph Construction

For the innumerable indoor path combinations, the idea of sequence radio-map calibration is firstly to divide the indoor paths into path segments, i.e., segments of path between crossing points. Each path segment will be treated as a vertex in the trace-graph model. The RSS signatures of each path segment are calibrated to form vertex of *the trace-graph*. The advantages of trace-graph includes:

- The RSS sequences of any desired path can be generated from the trace-graph, even if the path is not calibrated.
- Trace-graph efficiently reduces the storage space, which does not repeatedly store the RSS signature of the same path segment.

More specifically, the trace-graph is constructed by the following three steps.

(1) RSS sequence collection is carried out by a user walking along indoor paths while taking a mobile phone running a calibration APP. The user clicks the start point and the end point for each direct path, and the calibration APP records and interprets the RSS signatures of this path. Section 6.1 gives the details of RSS sequence collection.

(2) Then the cross points of these paths are found according to the map in the APP. These crossing points divide the collected RSS sequences into a set of path segments.

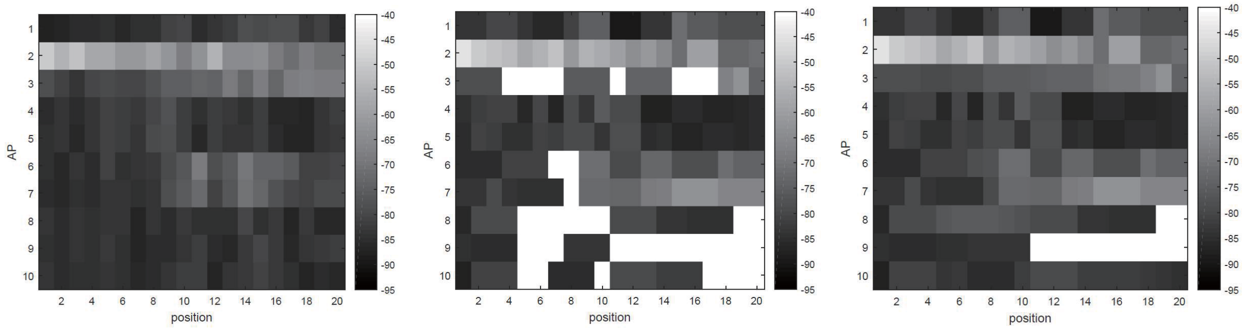
(3) Suppose there are N crossing points. We treat each path segment as a vertex, denoted by V_i, then an undirected trace-graph is generated as $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ where $\mathbf{V} = \{V_i, i = 1, 2, \cdots, N_V\}$ is the segment set; $\mathbf{E} = \{E_{i,j}\}$ represents the connections of these segments. $E_{i,j} = 1$ if segment V_i and V_j share a common crossing point. Let N_V and N_E denote the number of vertexes and edges. The graph is stored as sequence radio-map.

An example of trace-graph construction is shown in Fig. 3. Fig. 3(a) shows four direct paths with four cross points. The cross points divide the paths into 12 path segments. The RSS signatures of these path segments can be calibrated easily by a user just walking along the four directed path once. However, these path segments can combinatorially form a large set of possible indoor paths. Fig. 3(b) shows the formed trace-graph, which is an undirected graph. Each vertex is a path segment, and each edge indicates the adjacency of two path segments. The RSS signature of any uncalibrated path can be generated from this trace-graph. For example, the RSS signatures of path Γ^5, which is an uncalibrated path can be generated as $\mathbf{Y}^5$ by connecting four calibrated path segments.

Mathematically, the trace-graph in Fig. 3(b) can be modeled by an adjacency matrix A. Then the number of length-n paths between node i and j in the graph can be calculated by the (i, j) entry of the nth power of A, i.e., $A^n_{i,j}$. So the total number of length-n paths in the graph can be calculated by:

$$P(n) = \sum_{i=1}^{N} \sum_{j=i+1}^{N} A^n_{i,j} \tag{3}$$

In this particular example, it can be calculated that the number of paths containing $\{1, 2, 3, 4, 5\}$ path segments are $\{12, 24, 84, 413, 1764\}$ respectively.



(a) Point radio-map

(b) Sequence radio-map
with RSS miss-of-detection and RSS noises

(c) Sequence radio-map
cleaned by Collaborative Filter

Fig. 4. Comparison between point-type radio-map, sequence-type radio-map and sequence-type radio-map filtered by CF.

Theorem 4.1 (Storage Scalability). *The trace-graph is efficient to store. Suppose there are N cross points in the indoor paths, then N_V (the number of vertexes) of the trace-graph is in the order of $O(N)$; and N_E (the number of edges) is also $O(N)$ in* **G**.

Proof. In indoor paths, the cross points have low degrees, which are generally less than 4. Let a constant C be the maximum degree of a cross point, then the number of vertexes is at most $N_V = CN = O(N)$. Each cross point will generate at most $C(C-1)/2$ edges, so $N_E \leq C(C-1)/2 \cdot N = O(N)$.

□

Theorem 4.2 (Number of Combined Paths). *The number of combined paths extracted from* **G** *is at most $O(2^N)$, where N is the number of crossing points.*

Proof. The number of length-n paths in a complete graph is in the order of $O\left(\frac{N_V!}{(N_V-n)!}\right)$ where N_V is the number of vertexes and n is the path length. So the number of combined paths is $O\left(\sum_{n=1}^{N_V} \frac{N_V!}{(N_V-n)!}\right) = O(2^{N_V}) = O(2^N)$.

□

Note that the trace-graph radio-map model is also suitable when there is an open area without clear path. The RSS signatures can be calibrated just by walking in directed paths to calibrate a trace graph like that shown in Fig. 3(b).

In online locating, the moving window of online collected RSS sequence is actually short (the RSS sequence collected in the past one minute can be used as the online RSS sequential signature), so only a limited number of short paths need to be generated from the trace-graph for online sequence matching. Using AP list, we can also narrows down the searching space of graph vertexes. These avoid the complexity to generate many long paths in online phase, which is also a key for scalability in sequence-based radio-map. This problem will be detailed in Section IV.B.

4.2 RSS Sequence Cleaning

Although RSS sequence collection by walking along indoor paths can collect the RSS sequences quickly, the collected RSS values are highly noisy and are frequently missed during detection. So another key problem in constructing the trace-graph is to clean the collected RSS sequence.

As the detectable AP-list at different locations is different due to the limited coverage range of the APs and the random miss-of-detection of RSS values, the row length of RSS sequence $\mathbf{Y}^s$ should be regularized to be the same.

At first, we remove the fake APs by filtering their SSID and MAC addresses, e.g. some mobile hotspots are easily identified by their SSIDs. Then let $\mathbf{B}^s$ indicate the union set of effective APs detected on Γ^s, we regularize the row length of $\mathbf{Y}^s$ to the size of $\mathbf{B}^s$.

Algorithm 1 Collaborative filter

```
Require: $\mathbf{Y}^s$: raw RSS matrix on path $\Gamma^s$
Ensure: $\mathbb{Y}^s$: cleaned RSS matrix on path $\Gamma^s$
  for round = 1: $T$ do
    Update polynomial model of each row;
    for each empty value $\mathbf{Y}^s_{i,j}$ do
      $\hat{\mathbf{Y}}^s_{i,j,1} \leftarrow$ poly-fit by row $i$;
      $\hat{\mathbf{Y}}^s_{i,j,2} \leftarrow$ w eighted interpolation by column $j$;
      if $|\hat{\mathbf{Y}}^s_{i,j,1} - \hat{\mathbf{Y}}^s_{i,j,2}| <$ Threshold then
        $\mathbb{Y}^s_{i,j} = (\hat{\mathbf{Y}}^s_{i,j,1} + \hat{\mathbf{Y}}^s_{i,j,2})/2$
      end if
    end for
  end for
```

Algorithm 2 Weighted interpolation based on the APs similarity

```
Require: $\mathbf{Y}^s$: raw RSS matrix on path $\Gamma^s$, $i$: the row index of empty RSS value, $j$: the column index of empty RSS
  value, $h_s$: row length of $\mathbf{Y}^s$
Ensure: $\hat{\mathbf{Y}}^s_{i,j,2}$: one estimated value for $\mathbf{Y}^s_{i,j}$
  for $k = 1 : h_s$ and $k \neq i$ do
    $S(\mathbf{Y}^s_i, \mathbf{Y}^s_k) = (\mathbf{Y}^s_i \cdot \mathbf{Y}^s_k)/(\|\mathbf{Y}^s_i\| * \|\mathbf{Y}^s_k\|)$
  end for
  Select top-$K$ most similar rows of $\mathbf{Y}^s_i$. Let $\mathbf{K}$ denotes the set of their row indexes.
  $\hat{\mathbf{Y}}^s_{i,j,2} = (\sum_{k\in\mathbf{K}} S(\mathbf{Y}^s_i, \mathbf{Y}^s_k) * \mathbf{Y}^s_{k,j})/(\sum_{k\in\mathbf{K}} S(\mathbf{Y}^s_i, \mathbf{Y}^s_k))$
```

Regularized $\mathbf{Y}^s$ then need to be cleaned. There are many traditional data clean methods to fill the missing data in $\mathbf{Y}^s$, e.g., polynomial fitting [30], discrete fourier transform [7] and discrete wavelet transform [25]. However, there are practical challenges to apply these methods: 1) In some rows of $\mathbf{Y}^s$, only a small number of RSS values can be detected. 2) RSS values in the two ends of some rows are usually missed.

So *collaborative filter (CF)* [28] is exploited which takes time and APs correlation factors into account to filter the RSS sequence data. The routine of CF is given in Algorithm 1. For a missing value, one prediction value is calculated in row by polynomial fitting. Polynomial fitting can filter the noisy RSS data for user's movement on irregular paths. The path segments in offline training phase are generally direct path segments which can be designed by the trainer. So quadratic function is generally good enough for fitting the RSS signatures of a path segment without using higher degree functions. Usually when RSS sampling rate is higher than 0.5 Hz, CF works well. The second step in RSS data filling is to fill data in column by a weighted interpolation based on the APs' RSS sequence similarity, which is briefly introduced in Algorithm 2. If the two predicted values in column and row have difference smaller than a threshold, the average value of these two predictions will be filled into the empty value. This process repeats a fixed number of times. One example of CF results is shown in Fig. 4(c). We can see CF fill as many empty values as possible reasonably. In Section. VII-C, we investigate the effectiveness of different data clean methods. It can be proved that RSS sequence clearning can improve the locating accuracy and CF provides effective data cleaning.

5 ONLINE LOCATING

After trace-graph construction, the online locating is carried out by three steps: 1) online RSS sequence collection in a moving window; 2) candidate RSS sequence extraction from the trace-graph; 3) subsequence dynamic time warping for location determination.

5.1 Online RSS Sequence Collection using a Moving Window

As the target is moving, we capture RSS periodically and stores the measured RSS vectors into a moving window. Let $\mathbf{X} = (\mathbf{X}_{t-w+1}, \cdots, \mathbf{X}_t)$ denote the RSS sequence collected in the moving window and w be the window size. $\mathbf{X}_t$ is the latest collected RSS sample. Let $\mathbf{B}^x$ be the set of AP list in the moving window.The row length of $\mathbf{X}$ should be regularized to the length of $|\mathbf{B}^x|$ and CF is then applied to filter the online RSS matrix.

5.2 Candidate RSS sequence extraction from the trace-graph

5.2.1 Candidate Vertex Set Extraction. The first step is to extract all vertices who has similar AP list with $\mathbf{X}$. The preselection of vertices can greatly reduce the scale of CSS.

For every vertex in $\mathbf{G}$, let $\mathbf{B}^i$ denote the AP list of the vertex i, Jaccard similarity score [13] is calculated between $\mathbf{B}^i$ and $\mathbf{B}^x$:

$$J(\mathbf{B}^i, \mathbf{B}^x) = \frac{|\mathbf{B}^i \cap \mathbf{B}^x|}{|\mathbf{B}^x|} \tag{4}$$

Vertex i is added to the *candidate vertex set (CVS)* if $J(\mathbf{B}^i, \mathbf{B}^x) >$ Threshold, which means the AP list in $\mathbf{B}^i$ cover most of the APs in $\mathbf{B}^x$. In implementation, we set the Threshold to 0.7.

5.2.2 Candidate Sequence Set Generation. The CVS set selects a small subset of vertexes from trace-graph for reducing online matching complexity. Since the vertexes with similar AP list are generally neighboring to each other, the vertexes in CVS set are in a subgraph of the trace-graph. Then, the *candidate sequence set (CSS)* is generated based on the vertexes in CVS, which online builds the possible paths that the target may be moving on.

Since the moving window is short, it is unnecessary to generate long candidate paths. This heuristic can limit the size of CSS set. In implementation, we generate possible sequences containing at most c vertices, where $c = 3$ or 4. A direct way for CSS generation is to let each vertex in CVS to conduct Breadth-First-Search c steps to find paths originated at that vertex with length at most c, and then merge the repeated paths returned by different CVS. The remained distinct paths will form the CSS for online locating. Fig. 5 illustrates the process of CVS extraction and CSS extraction. We denote the returned CSS by $\{\mathbf{Y}^s : s = 1, \cdots, L\}$, where L is the number of candidate sequences.

Theorem 5.1 (Computational Complexity). The time complexity of Algorithm 3 to generate candidate sequence set is at most $O(M^c)$, where M is the size of CVS and c is a small user-defined constant.

Proof. Let $\mathcal{P}_k$ denote all combined paths of length k extracted from CVS. We always have $|\mathcal{P}_k| \leq M^k$ throughout the algorithm. Note that it takes $O(1)$ time to add a new path to the database since c is a constant. Therefore the time used by Algorithm 3 is at most $O(\sum_{k=1}^{c-1} M^k \cdot M) = O\left(\frac{M^2(M^{c-1}-1)}{M-1}\right) = O(M^c)$. □

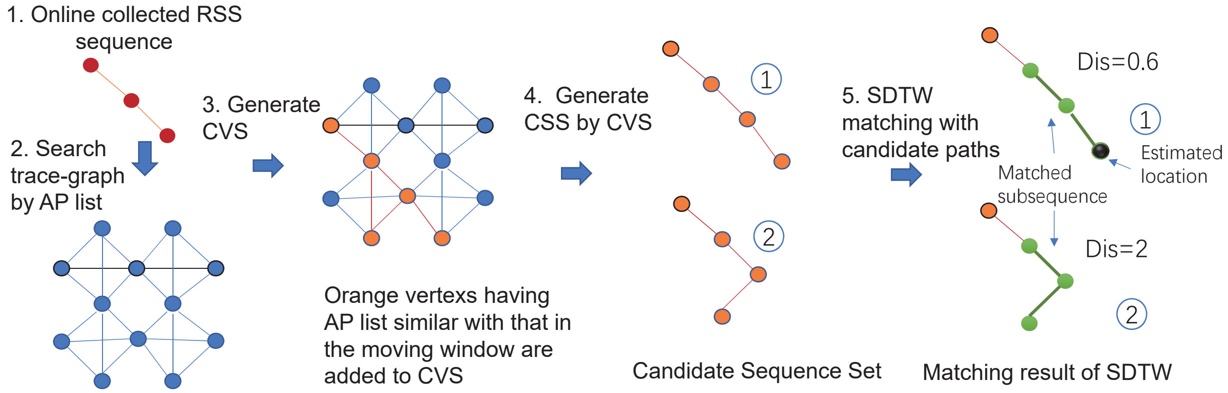

1. Online collected RSS sequence
2. Search trace-graph by AP list
3. Generate CVS
Orange vertexs having AP list similar with that in the moving window are added to CVS
4. Generate CSS by CVS
1
2
Candidate Sequence Set
5. SDTW matching with candidate paths
Dis=0.6
1
Estimated location
Matched subsequence
Dis=2
2
Matching result of SDTW

Fig. 5. The working flow of online matching. After RSS sequence is online collected into a moving window, the trace-graph is searched by the AP-list of the collected RSS. The trace-graph vertexes with similar AP list will be added into a CVS set. Then the CSS is generated from the CVS set. After that, for each candidate path in CSS, SDTW is applied to find the subsequence matched with the online collected RSS sequence in the moving window. The subsequence with the minimum warping distance will be calculated as the matched path (in the figure, the matched subsequences are in green, with warping distance 0.6 and 2 respectively, so the subsequence in CSS1 is the matched path), and the end point of the subsequence is the current location of the target (the dark point).

```
**Algorithm 3** CSS generation
**Require:** $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, $c$, CVS set
**Ensure:** CSS containing at most $c$ Vertices
  **for** each vertex $V_i$ in CVS **do**
    Breadth-First-Search $c$ steps to find paths originated at vertex $V_i$ with length no larger than $c$.
  **end for**
  Merge the returned paths of different CVS and delete the repeated paths.
  Return the RSS sequences on the remained distinct paths.
```

Theorem 5.2 (Matching Scalability). The number of generated candidate paths, i.e., $|\mathcal{P}|$, is at most $O(M^c)$, where M is the size of CVS.

Proof. Since $|\mathcal{P}_k| \leq M^k$, the number of generated candidate path set is at most $\sum_{k=1}^{c} M^k = \frac{M(M^c-1)}{M-1} = O(M^c)$. □

Theorem 5.2 indicates the scalability of online locating by sequence matching. The online collected RSS sequence needs only to be compared with a limited number of RSS sequences generated in the CSS set.

5.3 Subsequence Dynamic Time Warping

Recall $\mathbf{X}$ is the online collected RSS sequence in a moving window; $\{\mathbf{Y}^s : s = 1, \cdots, L\}$ is the candidate RSS sequence for comparison. Since $\mathbf{X}$ is short, the goal of matching is to find a particular sequence $\mathbf{Y}^{s^*}$ in CSS, whose subsequence $\mathbf{Y}^{s^*}_{[a^*,b^*]}$ satisfies the objective equation (2). The end point of the subsequence $\Gamma^{s^*}_{b^*}$ is the location of the target. This is carried out by a proposed *subsequence dynamic time warping* (SDTW) algorithm.

5.3.1 *Dynamic Time Warping (DTW).*

SDTW is an extension of *dynamic time warping* (DTW) [18][23], an algorithm for measuring similarity between two temporal sequences which may vary in time or speed. Consider two sequences $X = (x_1, \cdots, x_n)$ and $Y = (y_1, \cdots, y_m)$. DTW calculates a warping path $P = (p_1, \cdots, p_l)$ where $p_k = (i_k, j_k)$; $i_k \in [1 : n]$; $j_k \in [1 : m]$; and $k \in [1 : l]$ that satisfies the following alignment conditions:

$$\begin{cases} p_1 = (1,1), p_l = (n,m) \\ p_{k+1} - p_k \in \{(0,1),(1,0),(1,1)\}, \forall k \in [1 : l-1] \end{cases} \tag{5}$$

The first condition enforces the first elements of X and Y as well as the last elements of X and Y are aligned to each other. The second condition reflects no elements in X and Y can be omitted and there are no replications in alignment. An alignment satisfies above conditions is called a warping path.

The *warping distance* between X and Y is defined as the summation of local distances along the warping path.

$$\mathrm{DTW}(X, Y) = \sum_{k=1}^{l} d(x_{i_k}, y_{j_k}). \tag{6}$$

The optimal warping path, i.e., P^* that minimizes $\mathrm{DTW}(X, Y)$ can be calculated by dynamic programming[14].

5.3.2 *Subsequence Dynamic Time Warping (SDTW).*

SDTW relaxes the boundary condition in (5). It allows X match a subsequence $Y_{[a,b]} = (y_a, y_{a+1}, \cdots, y_b)$ $(1 \le a \le b \le m)$ of Y [14], such that

$$(a^*, b^*) = \mathop{\arg\min}_{(a,b):1 \le a \le b \le m} \mathrm{DTW}(X, Y_{[a,b]}). \tag{7}$$

But the locating problem has some differences from the traditional SDTW.

(1) It needs only to determine the end point of the matched subsequence, i.e., b^* without the need to determine the start point a^*. This feature is utilized to design an efficient subsequence matching algorithm which doesn't need the cost of back-trace by dynamic programming, as detailed in Algorithm 4.

(2) It needs to search over multiple candidate sequences $\{\mathbf{Y}^s : s = 1, \cdots, L\}$ to find the subsequence with the overall minimum warping distance.

(3) We need to consider the situation that the user may walk in the reversed direction of the training direction of the candidate sequences. So matching is also conducted reversely, as detailed in Algorithm 3.

5.3.3 *SDTW for sequence-based locating.*

The sequence-based locating method based on SDTW is summarized in Algorithm 4. Reducing the height of $\mathbf{X}$ and $\mathbf{Y}^s$ by *Principal component analysis (PCA)* [11] is the first step to reduce computation complexity of sequence matching. Then the optimal warping distance $D(n, b_s^*)$ and the ending point are calculated by (10) between $\mathbf{X}$ and each $\mathbf{Y}^s$. We also need to reverse the direction of $\mathbf{X}$ and recalculate the optimal warping distance. Finally, the target location is determined as the end point in the matched subsequence having the least warping distance.

Note that since the ending point and the optimal warping distance $D(n, b_s^*)$ have already been provided by (10), it is not necessary to calculate a_s^* by back-tracing. This saves the cost of dynamic programming than the traditional SDTW [14].

Fig. 6 further illustrates the warping distance matrix. The values represented by "+" is given as the boundary values. They produce the other entries of distance matrix by (9). After getting the matrix, the entry with the smallest distance in the nth row can be found, i.e., the star. Its column index is b_s^*, and its value is $D(n, b_s^*)$. In locating problem, we don't need to back-trace to find a_s^*, and the warping path (represented by the dashed line), although they have already been determined by the distance matrix. This saves computation cost than traditional SDTW [14]. It is easy to verify that the computation cost is $O(mn)$ in this step. The overall computation cost is $O(Lmn)$, which is efficient.

Algorithm 4 SDTW for sequence-based locating

Require: $\mathbf{X}$: online RSS sequence, $\{\mathbf{Y}^s, s=1,\cdots,L\}$: CSS
Ensure: s^*: the location of target

STEP1: AP Set Reduction

· Extract principal components from AP union set of $\mathbf{X}$ and $\mathbf{Y}^s$ by PCA and replace $\mathbf{X}$ and $\mathbf{Y}^s$ by the extracted represented sequences.

STEP2: SDTW between $\mathbf{X}$ and $\{\mathbf{Y}^s, s=1,\cdots,L\}$

· Set the boundary conditions of warping distance matrix:

$$\begin{aligned} D_s(i,1) &= \sum_{k=1}^{i} d(\mathbf{X}_k, \mathbf{Y}_1^s), \quad i \in [1:n] \\ D_s(1,j) &= d(\mathbf{X}_1, \mathbf{Y}_j^s), \quad j \in [1:m_s] \end{aligned} \tag{8}$$

· Then the warping distance matrix is produced as:

$$D_s(i,j) = \min\{D_s(i-1,j-1), D_s(i,j-1), D_s(i-1,j)\} + d(x_i, y_{j,s}) \tag{9}$$

· The matched end point between $\mathbf{X}$ and $\mathbf{Y}^s$ is determined by D_s:

$$b_s^* = \operatorname*{arg\,min}_{b_s \in [1:m_s]} D_s(n, b_s), \tag{10}$$

· The optimal warping distance with $\mathbf{Y}^s$ is given by:

$$D(\mathbf{X}, \mathbf{Y}^s) = D(n, b_s^*) \tag{11}$$

STEP3: Reverse sequence $\mathbf{X}$ to match with $\{\mathbf{Y}_s, s=1,\cdots,L\}$

· Reverse the direction of $\mathbf{X}_t$ and recalculate (8) - (11) to obtain $\overleftarrow{b_s^*}$ and $\overleftarrow{D}\left(n, \overleftarrow{b_s^*}\right)$ for $s=1,\cdots,L$.

STEP4: Determine the target location and the minimum warping distance

· For all the candidate sequences, the overall best matched sequence is the one with the overall least warping distance:

$$s^* = \operatorname*{arg\,min}_{s \in [1:L]} \min\left\{D(n, b_s^*), \overleftarrow{D}\left(n, \overleftarrow{b_s^*}\right)\right\} \tag{12}$$

· The location of the target is determined as the end matching point in the matched subsequence that has the overall least warping distance.

5.3.4 Using Different Distance Functions. We also investigate the impacts of using different distance functions, i.e., $d(x_{i_k}, y_{j_k})$ to check their impacts to the locating accuracy. Four different functions were investigated.

1) Euclidean distance: $d_{xy} = \sqrt{\sum_{k=1}^{n}(x_k - y_k)^2}$
2) Cityblock distance: $d_{xy} = \sum_{k=1}^{n} |x_k - y_k|$
3) Chebyshev distance: $d_{xy} = \max_k |x_k - y_k|$
4) Cosine distance: $d_{xy} = \frac{dot(x,y)}{\|x\| * \|y\|}$

In Section. 7.4, we will concretely analyze the effect of basic warping distance functions on locating accuracy for SDTW.

5.3.5 AP Set Reduction in SDTW. As SDTW distance calculation between $\mathbf{X}$ or $\mathbf{Y}^s$ needs high computation complexity when the RSS sequence has high dimension in the number of APs. So reducing the height of $\mathbf{X}$ or $\mathbf{Y}^s$

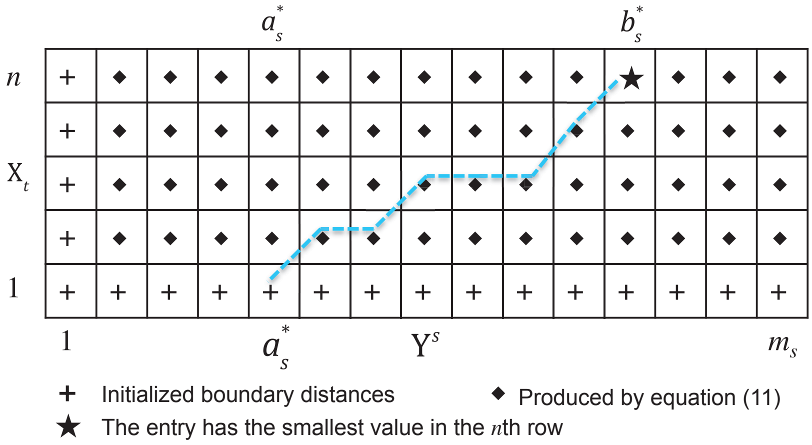

a_s^*
b_s^*
n
X_t
1
1
a_s^*
Y^s
m_s
+ Initialized boundary distances
◆ Produced by equation (11)
★ The entry has the smallest value in the nth row

Fig. 6. Illustration of the warping distance matrix in SDTW.

is also important for efficiency of sequence matching. We investigate PCA reduce the dimension of RSS sequence for matching efficiency.

Principal component analysis (PCA) [11] is an efficient method to extract a set of linearly uncorrelated variables from a set of correlated variables. Such uncorrelated variables represent main properties of original variable set. We use PCA to extract principal components from AP union set of $\mathbf{X}$ and $\mathbf{Y}^s$ and use extracted corresponding RSS sequences for SDTW distance calculation. Let $|\mathbf{B}^x \cup \mathbf{B}^s|$ denote the size of AP union of $\mathbf{X}$ and $\mathbf{Y}^s$ and p denote the size of extract principal components. We know $1 \leq p \leq |\mathbf{B}^x \cup \mathbf{B}^s|$. The height of $\mathbf{X}$ and $\mathbf{Y}^s$ should be regularized to p for subsequent SDTW distance calculation. In Section. VII-E, we investigate the effect of principal component size p on locating accuracy. It can be proved that reducing the height of $\mathbf{X}$ and $\mathbf{Y}^s$ can meet the requirements of high computational efficiency and high locating accuracy.

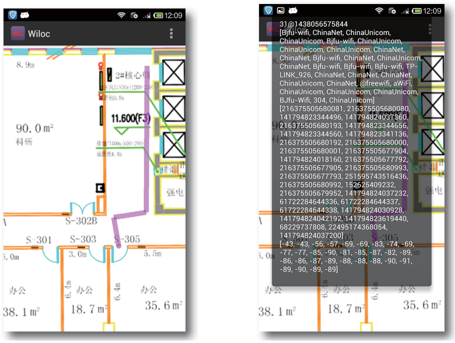
6 IMPLEMENTATION

We have developed an APP on Android platform to implement sequence-based indoor localization. The android APP carries out path calibration, RSS path collection, and data transmission to server and real-time position display. The location determination is carried out by a location server. The server constructs the trace-graph generation, CSS extraction based on real-time detected AP list, and SDTW-based online locating.

6.1 RSS sequence collection

RSS sequence collection can be carried out in many ways using interactive smart phones or indoor navigation systems by inertial sensors [20]. We use an interactive way by an APP developed on the mobile phone. The APP renders the floor-plan and gets the scaling factor of the floor-plan, so that each pixel on the screen maps to a physical location.In order to calibrate a path, a user clicks on the map to firstly mark the path. Several clicks on the screen will characterize a path. Then the user walks along the specified path to collect RSS sequence. The WiFi-scanner calls the RSS sampling library of Android to take RSS samples in approximately equal intervals, so the positions where RSS signal is measured on the path can be calculated by assuming the user is moving at constant speed.

But in practice, we found that RSS sampling rate of different phones differ greatly, which can be referred to Table. 2. The difference is mainly due to the power optimization policy of the phone. The locating performance of



```
31@1438056575844
[Bjfu-wifi, ChinaNet, ChinaUnicom,
ChinaUnicom, Bjfu-wifi, ChinaUnicom,
ChinaUnicom, ChinaUnicom, ChinaNet,
ChinaNet, Bjfu-wifi, ChinaNet, ChinaUnicom,
ChinaNet, Bjfu-wifi, Bjfu-wifi, Bjfu-wifi, TP-
LINK_926, ChinaNet, ChinaNet, ChinaNet,
ChinaUnicom, ChinaNet, @freewifi, aWiFi,
ChinaUnicom, ChinaUnicom, ChinaUnicom,
BJfu-Wifi, 304, ChinaUnicom]
[216375505680081, 216375505680080,
141794823344496, 141794824037360,
216375505680193, 141794823344656,
141794823344560, 141794823341136,
216375505680192, 216375505680000,
216375505680001, 216375505677904,
141794824018160, 216375505677792,
216375505677905, 216375505680993,
216375505677793, 251595743516436,
216375505680992, 152625409232,
216375505679952, 141794824037232,
61722284644336, 61722284644337,
61722284644338, 141794824030928,
141794824042192, 141794823619440,
68229737808, 224951743680054,
141794824037200]
[-43, -43, -56, -57, -69, -69, -83, -74, -69,
-77, -77, -85, -90, -81, -85, -87, -82, -89,
-86, -86, -87, -89, -88, -88, -88, -90, -91,
-89, -90, -89, -89]
```

Fig. 7. Snapshots of RSS sequence calibration.

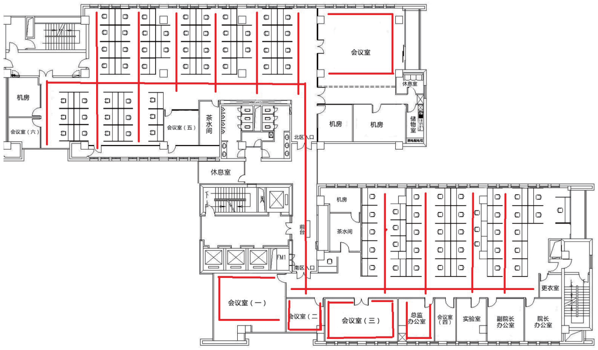


Fig. 8. Map of NEC Laboratories China.

SDTW may deteriorate greatly when phones collect the RSS sequences in different frequency during the off-line training and on-line locating process. For example, the Sumsang phones can collect RSS values in 38.03 Hz, while Meizu phones only collect RSS values in 0.52 Hz. There are duplicate values in the highly frequent RSS scans. So in practice, we regularize phones RSS sampling rate to 0.5 Hz in order to improve SDTW's robustness in different phones and to remove the duplicated RSS scans for energy efficiency. In case the RSS values are missed in a scan, the CF can effectively clean and fill the missed data. A RSS sequence on a path will be a series of <location, RSS vector> pairs, which is a mapping from Γ^s to Υ^s. Then these collected RSS sequences are transmitted to the server to compose the trace-graph. Snapshots of RSS sequence calibration are shown in Fig. 7.

6.2 Online locating

After RSS sequence calibration, an APP is developed for on-line locating, which collects RSS sequence in a moving window and sends the sequence to a localization server. The server calculates the target position through SDTW-based sequence matching, and sends the locating result back to the APP for location rendering. The client-server structure supports multiple user localization. The clients do not interfere with each other. When the number of users is very large, we may need to investigate the loading balancing problem of the server, which is not the focus of our paper.

7 PERFORMANCE EVALUATION

7.1 Experiment setup

7.1.1 Experiment area. We conducted experiments at three different locations.

(1) The first experiment was conducted in NEC Laboratories China, located in Innovation Plaza No.1 of Tsinghua Science Park. The office area is about $1100m^2$ and more than 90 APs can be detected in the area. We did the experiment for almost one month, including training trace-graph and conducting online locating. The map and collected RSS sequences are shown in Fig. 8.

(2) The second experiment was in Hang Lung Square, which is a shopping mall located covering $25000m^2$ in Liangxi Street, Wuxi, China. The experiments were conducted in summer 2017. The map of the experiment floor, i.e. first floor, is available from Baidu Maps, which is shown in Fig.15. We mark training RSS sequences marked by red lines and testing RSS sequences marked by green lines.

(3) The third experiment location was in Suning Square, which is another shopping mall covering $23000m^2$ located in Renmin Middle Street, Wuxi, China. Its forth floor was chosen as experiment floor whose map is

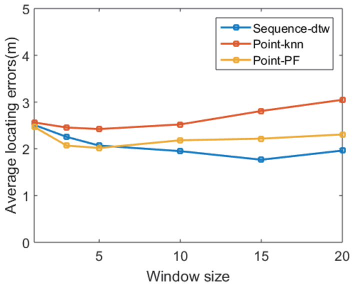


Fig. 9. Mean locating error as a function of moving window size w.

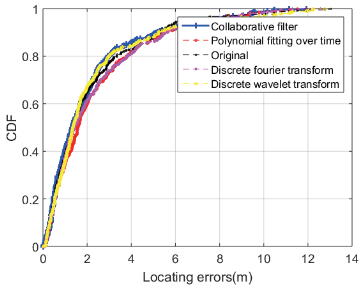


Fig. 10. CDF of locating error as a function of data clean methods.

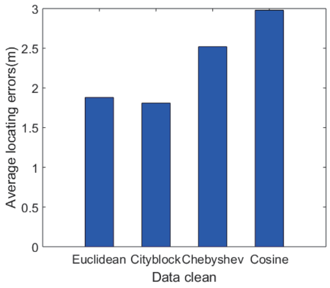


Fig. 11. Mean locating error as a function of distance functions.

achieved from Baidu Map. Fig. 16 shows collected training RSS sequences in red lines and testing RSS sequences in green lines. We conducted the experiment in summer 2017.

7.1.2 Experiment phone. 5 phones with different brands are chosen to evaluate SDTW's robustness to phone change, i.e. Mi, Sumsang C7, Sumsang C9, Leshi and Meizu. We used Mi to conducted experiment in NEC Laboratories China and other phones in Hang Lung Square and Suning Square. Their sampling rates are shown in Table. 2. Sampling rates of other common phone brands are also investigated, e.g. 0.58 Hz for Huawei, 0.52 Hz for Lenovo. It can be seen that RSS sampling rates of different phone brands differ greatly. So we regularized RSS sampling rate to be the same to improve SDTW's robustness to different phones, which is detailedly discussed in Section 7.8.

7.1.3 Comparing Algorithms. Two traditional point-type radio-map based locating methods were implemented for comparison: (1) Point-type radio-map using K-nearest Neighbor, abbreviated as *Point-KNN*. (2) Point-type radio-map using K-nearest Neighbor and particle filter, abbreviated by *Point-PF* [27]. The Point-PF uses signal similarity to generate M next-time candidate positions of particles and movement consistency is used to remain K best particles (or trajectories) from MK candidate particles. The end point of the best particle is treated as the estimated location.

7.1.4 Investigated Metrics. Experiments are carried out in two categories:

(1) The locating accuracy. We first analyzed four inner factors of SDTW, including the length of moving window w, different data clean methods, different distance functions, the reduced height p of $\mathbf{X}$ and $\mathbf{Y}^s$. Then locating accuracy of SDTW based on the best four factors was compared with two point-type radio-map methods.
(2) The locating robustness. We evaluated SDTW's robustness to the changes of environments, phones, dates, walking speed and walking pattern.

7.2 Accuracy vs. Length of Moving Window

We investigated the moving window length first. The results are from the experiments in NEC Labs. The experiments was conducted by varying the length of moving window from $\{1, 3, 5, 10, 15, 20\}$. The average locating error as a function of w is plotted in Fig. 9. When $w = 1$, SDTW is degraded to Point-KNN. Compared with Point-PF, SDTW works worse when $w \leq 5$. However, as w increases, SDTW shows better locating accuracy than both Point-KNN and Point-PF. When $w \geq 15$, the locating accuracy of SDTW tends to decrease a little bit.

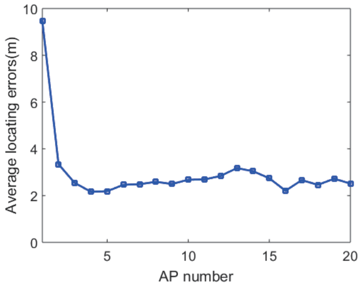


Fig. 12. Mean locating error as a function of the height of RSS sequences.

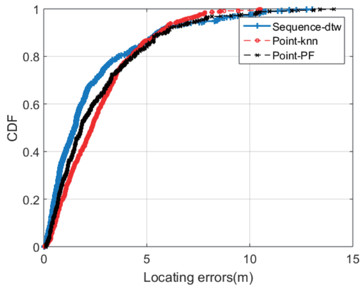


Fig. 13. CDF of locating errors for SDTW, Point-KNN, Point-PF.

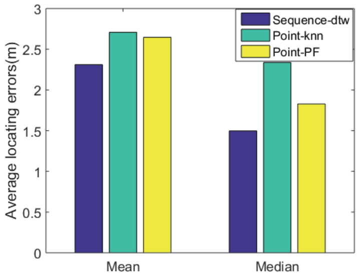


Fig. 14. Mean and median of locating errors for SDTW, Point-KNN, Point-PF.

The reason is when w is large, sequence matching accuracy decreases at some boundary locations, because it is hard to form a long sequence. So a window size between 10 and 15 is appropriate.

7.3 Accuracy vs. Data Clean Methods

Then the impact of data clean methods on SDTW was accessed, based on the experiment data of NEC labs. We compared the filtering performance of CF with that of polynomial fitting, discrete fourier transform and discrete wavelet transform, which is introduced in Section. IV-B. Their corresponding locating errors are shown in Fig. 10. It can be seen CF improves locating accuracy best.

7.4 Accuracy vs. Different Distance Functions

In Section. 5.3.4, we have introduced four different distance functions for $d(x_{i_k}, y_{j_k})$. We separately applied them in SDTW algorithm and calculated average locating errors. The results are shown in Fig. 11. Euclidean distance and cityblock distance work better than the other distances. It is reasonable as these two distance functions give larger values (all four distance functions give positive values) for the same difference between two vectors, which means they distinguish similar vectors better.

7.5 Accuracy vs. PCA Parameter

As discussed in Algorithm 4, PCA is used to extract principal components from AP union set of $\mathbf{X}$ and $\mathbf{Y}^s$ for online sequence matching. We calculated locating errors by varying the number of principal components p from 1 to 20. The average locating errors are shown in Fig. 12. We find SDTW works well when $p \geq 5$.

7.6 Accuracy vs. Point-type Radio-map Methods

In the subsection, we first set the best inner factors for SDTW and compared its locating accuracy with Point-KNN and Point-PF. The best factor setting includes (1) choosing the size of moving window $w = 15$, (2) using CF for data clean, (3) using euclidean distance as distance function and (4) setting the height of $\mathbf{X}$ and $\mathbf{Y}^s$ as 10. The cumulative distribution function (CDF) of locating errors is shown in Fig. 13 and mean and median errors are plotted in Fig. 14. We can see SDTW reduces the average locating error more than 20% than Point-KNN and Point-PF. And SDTW's locating accuracy can be further improved by collaborating with particle filter.

7.7 Robustness vs. Environments

For locating methods, the robustness to environment changes is a critical indicator for being widely used. So in the subsection, locating errors of SDTW in different indoor areas, i.e. NEC Laboratories China, Hang Lung

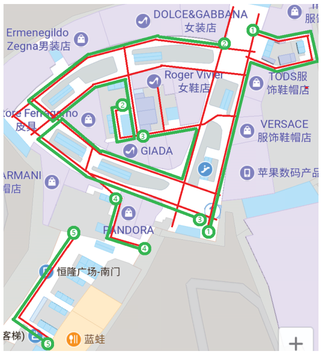


Fig. 15. Map of Hang Lung Square.

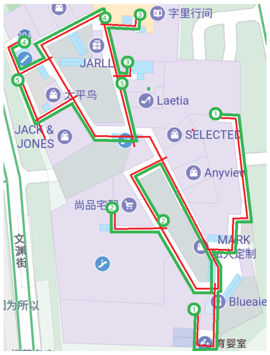


Fig. 16. Map of Suning Square.

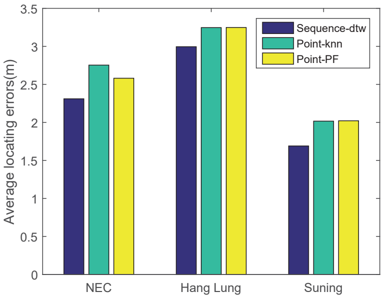


Fig. 17. Mean locating errors when positions change.

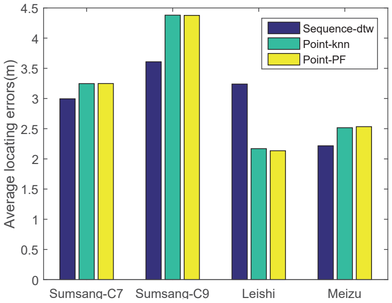


Fig. 18. Mean locating errors when phones change.

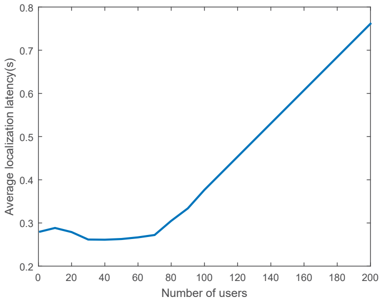


Fig. 19. Mean responding time as a function of the number of users.

Square and Suning Square, are calculated. Mean locating results in the three experiment areas are shown in Fig. 17. We can conclude no matter in which environment, SDTW can guarantee better locating accuracy than other two point-type radio-map methods, which reveals the robustness of SDTW to different environments.

7.8 Robustness vs. Phone

The calibration effort is the major cost of conducting radio-map location. If the locating performance can be robust to the type of phones and environment changes, frequent recalibration will not be needed. To test the impact of phone changes, Samsung C7, Samsung C9, Leshi and Meizu are used to collect RSS sequence. Their sampling rates are shown in Table. 2. We regularized RSS sampling rate at 0.5 Hz for a balance of energy efficiency

Table 2. Frequency statistics of different phone brands

Phone brand	Frequency (Hz)
Mi	1.43
Sumsang-C7	38.03
Sumsang-C9	34.53
Leshi	0.15
Meizu	0.52

and sampling fidelity and then calculated locating errors using different phones in on-line test phase. Samsung C7 was used to in both training and testing phases, while the remaining three phones were only used in the testing phase. The locating results by using different testing phones are shown in Fig. 18. We first notice SDTW works bad in Leshi phone because the RSS sampling rate of Leshi is too low to build comparable RSS sequence with the training data. It can also be found that SDTW works better than the other two point-type radio-map methods using other phones. It may be strange that particle filter cannot improve locating accuracy of Point-KNN. We give one possible reasons as that after regularizing the RSS sampling rates as 0.5 Hz, the movement consistency of successive samples is much reduced.

7.9 Robustness vs. Date

We then test the impact of time changes on SDTW. For the experiments in NEC labs., we trained the trace-graph on the morning of Oct. 31, 2015, when there were a few people. Then, we collected three sets of testing paths at three different times: (1) on the evening of Oct 31, 2015, when there were few people; (2) on the afternoon of Nov 2, 2015, when many people were worked in the offices; (3) at noon time of Nov 4 when many people worked in the offices and moved in corridors. The mean locating errors of SDTW, Point-KNN and Point-PF are shown in Fig. 20. We find all three locating methods keep reasonable robustness when the testing environment differs from the training environment.

SDTW has better locating accuracy than the other two point-type radio-map based methods on Oct 31 and Nov 2. Point-PF performs a little better than SDTW on Nov 4, but generally speaking, their accuracy are at the same level.

7.10 Latency vs. The number of users

The scalability performance of SDTW to support multiple users is evaluated. We used the RSS sequences collected in previous experiments to simulate simultaneous localization of multiple users. The number of users is denoted by u, which varies in $[1, 200]$. A Thinkpad X1 computer worked as the backend server, which has 2-core processors (Intel(R) Core(TM) i7-4600U CPU @2.10GHz 2.69 GHz) and 8.00G RAM. Multi-thread programming, i.e., u threads were generated to response the localization tasks. Each user is handled by one thread. The mean responding time as a function of the number of users was calculated and the result was shown in Fig. 19. It can be seen that, when $u < 70$, the mean responding time is around $0.2s$. This is because the multi-thread computing hasn't reached the computation capacity when $u < 70$. The user localization tasks are processed almost simultaneously without waiting in the queue. When $u > 70$, the mean responding time increases linearly with the number of users, because some tasks have to wait in the queue for the full usage of the processors. So it can be concluded that in the worst case, the mean responding time increases linearly with the number of users. But the responding time can be guaranteed to be small when the backend server has enough computation resources, since all users can be localized in parallel.

7.11 Robustness vs. Walking Speed and Walking Pattern

To evaluate SDTW's tolerance to the walking speed and walking pattern variance, two experiments were conducted in the subsection. The impact of different walking speeds on the locating accuracy of SDTW was first evaluated. The training paths were collected by walking in almost constant middle speed, about 1 m/s. Then three speeds were tested for online locating: (1) high speed about 3 m/s; (2) middle speed about 1 m/s; (3) low speed about 0.5 m/s. The locating errors for different speeds are shown in Fig. 21. We find when the training and testing paths are collected in the same speed, the locating error is the smallest. And when the speeds are different, the locating accuracy becomes worse. We also find middle-speed training is more robust with low-speed testing than high-speed testing.

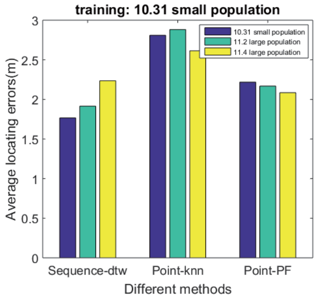


Fig. 20. Mean locating errors when environment changes.

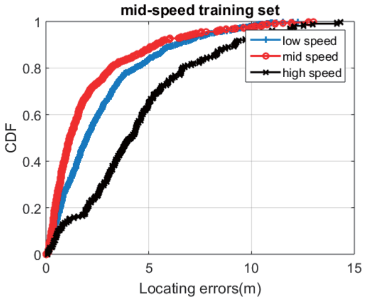


Fig. 21. CDF of locating error when locating speed changes.

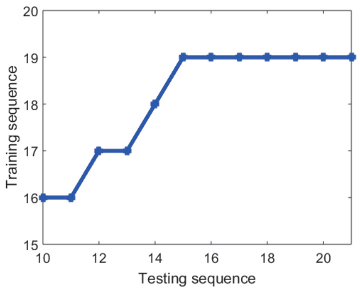


Fig. 22. Warping path of in an instance of walk-stop pattern.

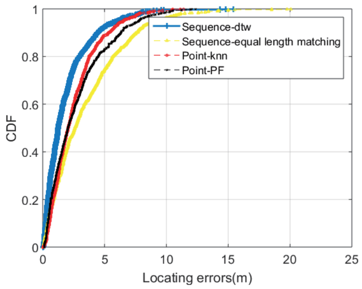


Fig. 23. CDF of locating error in walk-stop pattern.

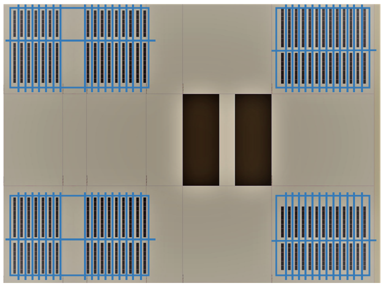


Fig. 24. Floor map of the library.

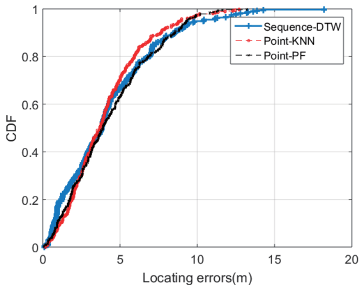


Fig. 25. CDF of locating error in the library.

Then we evaluated SDTW's robustness to different walking patterns. We collected training paths in almost constant middle speed and testing paths in a walk-stop pattern, i.e., walks, stops, and then walks again repeatedly. We compared SDTW with Point-KNN, Point-PF and an sequence-based equal-length matching algorithm in which the distance is calculated point by point between sequences without warping. One sequence warping result of SDTW is shown in Fig. 22. SDTW matches the moving window with indices 10 - 20 to a subsequence with indices 16 - 19 in CSS. The stopping sequence is efficiently warped. However, equal-length matching cannot warp the RSSs at the stopping time. Furthermore, the CDF of locating errors in the series of walk-stop experiments is plotted in Fig. 23. It shows that SDTW performs the best in the walk-stop pattern. The reason Point-PF works terribly is movement consistency is broken in the walk-stop pattern. Sequence-based equal-length matching algorithm works the worst and its bad adaption to walk patterns may be the main reason why sequence matching locating methods were not widely used before.

8 CONCLUSION

This paper investigates feasibility and performances of indoor locating by using sequence-type radio-map. In offline training phase, a trace-graph model is proposed to efficiently calibrate and store the RSS sequences of indoor paths, which can overcome the indoor path combinatorial explosion problem. Collaborative filter has been developed for filling the missed value in trace-graph. Then in online locating phase, RSS sequences are collected by a target using a short moving window. Based on the APs detected in the moving window, candidate sequences are generated from trace-graph to prepare possible routes of targets for location determination. An efficient subsequence dynamic time warping algorithm was then proposed to determine the location of the target, which can tolerate the differences of moving speeds, moving patterns, and moving directions between the training and locating phases. Experiments in office environment shows that Warpmap is efficient and easy to use.

However in some extremely severe circumstance, e.g. open space, the locating accuracy of WarpMap deteriorates. Take the library for instance, whose map is shown in Fig. 24. There are many short paths partitioned by short bookcases there. For SDTW, it needs to remain a constant-length moving window for online locating, so SDTW cannot shows its superiority in very short paths. Meanwhile, two paths partitioned by the bookcase shows great similarity in their RSS sequences. Fig. 25 gives the locating results of SDTW in the library. We can see SDTW doesn't outperform Point-KNN in this environment. Point-PF also works poorly since movement consistency is not obvious in short paths. Locating in open space is also a challenging problem for using WarpMap. But we can specify grid-type paths in the open space to generate a trace graph with higher path density. The framework of Warpmap can then be used in open space. But note that the locating accuracy of point-type radio-map in open space is also bad because the RSS signatures are similar in nearby locations in open space.

WarpMap works as a basic block for RSS radio-map matching, it can be widely collaborated with fusion algorithms and additional information. In future work, we will exploit the fusion of WarpMap with particle filter, dead reckoning algorithm using inertial sensors, and digital floor map information. It can be foreseen that by combining these information, sequence-based indoor location can achieve better locating accuracy. We will also apply WarpMap method to other locating systems, such as bluetooth and ultrasound based locating system etc.

REFERENCES

[1] J. C. Aguilar Herrera, P. G. Plöger, A. Hinkenjann, J. Maiero, M. Flores, and A. Ramos. Pedestrian Indoor Positioning Using Smartphone Multi-sensing, Radio Beacons, User Positions Probability Map and IndoorOSM Floor Plan Representation. 2014.

[2] P. Bahl and V. Padmanabhan. RADAR: an in-building RF-based user location and tracking system. In *IEEE INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings*, volume 2, pages 775–784 vol.2, 2000.

[3] D. U. Chuan-Ming. Application of baidu map api in small geographic information system. *Geomatics & Spatial Information Technology*, 2011.

[4] X. Geng, Y. Wang, H. Feng, and Z. Chen. Hybrid radio-map for noise tolerant wireless indoor localization. In *2014 IEEE 11th International Conference on Networking, Sensing and Control (ICNSC)*, pages 233–238, Apr. 2014.

[5] I. Haque and C. Assi. Profiling-Based Indoor Localization Schemes. *IEEE Systems Journal*, Early Access Online, 2013.

[6] R. Harle. A Survey of Indoor Inertial Positioning Systems for Pedestrians. *IEEE Communications Surveys Tutorials*, 15(3):1281–1293, 2013.

[7] F. J. Harris. On the use of harmonic analysis with the discrete fourier transform. *Proceedings of the IEEE*, 66(1):51–83, 1978.

[8] J. Hightower and G. Borriello. *UbiComp 2004: Ubiquitous Computing: 6th International Conference, Nottingham, UK, September 7-10, 2004. Proceedings*, chapter Particle Filters for Location Estimation in Ubiquitous Computing: A Case Study, pages 88–106. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[9] V. Honkavirta, T. Perälä, S. Ali-Löytty, and R. Piché. A comparative survey of wlan location fingerprinting methods. In *WPNC*, pages 243–251, 2009.

[10] Y. Ji, S. Biaz, S. Wu, and B. Qi. Impact of Building Environment on the Performance of Dynamic Indoor Localization. In *Wireless and Microwave Technology Conference, 2006. WAMICON '06. IEEE Annual*, pages 1–5, Dec. 2006.

[11] I. T. Jolliffe. *Principal Component Analysis.* Springer-Verlag, 1986.

[12] T. King, S. Kopf, T. Haenselmann, C. Lubberger, and W. Effelsberg. Compass: A probabilistic indoor positioning system based on 802.11 and digital compasses. In *Proceedings of the 1st International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*, WiNTECH '06, pages 34–40, New York, NY, USA, 2006. ACM.

[13] M. Levandowsky and D. Winter. Distance between sets. *Nature*, 234(5323):34–35, 1971.

[14] S. Z. Li and A. Jain, editors. *Encyclopedia of Biometrics*, chapter Dynamic Time Warping (DTW), pages 231–231. Springer US, Boston, MA, 2009.

[15] C.-C. Lo, L.-Y. Hsu, and Y.-C. Tseng. Adaptive radio maps for pattern-matching localization via inter-beacon co-calibration. *Pervasive Mob. Comput.*, 8(2):282–291, Apr. 2012.

[16] G. Map. http://www.google.com/maps. 2018.

[17] M. Molina-García, J. Calle-Sánchez, J. I. Alonso, A. Fernández-Durán, and F. B. Barba. Enhanced In-Building Fingerprint Positioning Using Femtocell Networks. *Bell Labs Technical Journal*, 18(2):195–211, Sept. 2013.

[18] M. Müller. *Information Retrieval for Music and Motion.* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[19] L. Ni, Y. Liu, Y. C. Lau, and A. Patil. LANDMARC: indoor location sensing using active RFID. In *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, 2003. (PerCom 2003)*, pages 407–415, 2003.

[20] J.-O. Nilsson, A. K. Gupta, and P. Händel. Foot-mounted inertial navigation made easy. In *International Conference on Indoor Positioning and Indoor Navigation*, volume 27, page 30th, 2014.

[21] J.-O. Nilsson, I. Skog, P. Handel, and K. Hari. Foot-mounted INS for everybody - an open-source embedded implementation. In *Position Location and Navigation Symposium (PLANS), 2012 IEEE/ION*, pages 140–145, Apr. 2012.

[22] S. J. Pan, J. T. Kwok, Q. Yang, and J. J. Pan. Adaptive Localization in a Dynamic WiFi Environment Through Multi-view Learning. In *Proceedings of the 22Nd National Conference on Artificial Intelligence - Volume 2*, AAAI'07, pages 1108–1113, Vancouver, British Columbia, Canada, 2007. AAAI Press.

[23] H. Sakoe. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26:43–49, 1978.

[24] P. Scholl, S. Kohlbrecher, V. Sachidananda, and K. Van Laerhoven. Fast indoor radio-map building for RSSI-based localization systems. In *2012 Ninth International Conference on Networked Sensing Systems (INSS)*, pages 1–2, 2012.

[25] M. J. Shensa. Discrete wavelet transform: Wedding the a trous and mallat algorithms. *IEEE Transactions on Signal Processing*, 40(10):2464–2482, 1992.

[26] Y. Shu, K. G. Shin, T. He, and J. Chen. Last-mile navigation using smartphones. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, MobiCom '15, pages 512–524, New York, NY, USA, 2015. ACM.

[27] L. Song and Y. Wang. Multiple target counting and tracking using binary proximity sensors: Bounds, coloring, and filter. In *Proceedings of the 15th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '14, pages 397–406, New York, NY, USA, 2014. ACM.

[28] X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009(12):4, 2009.

[29] Y. Tian, R. Gao, K. Bian, F. Ye, T. Wang, Y. Wang, and X. Li. Towards ubiquitous indoor localization service leveraging environmental physical features. In *2014 Proceedings IEEE INFOCOM*, pages 55–63, Apr. 2014.

[30] C. Vamos and M. Craciun. *Polynomial Fitting*. Springer Netherlands, 2012.

[31] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury. No need to war-drive: Unsupervised indoor localization. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*, MobiSys '12, pages 197–210, New York, NY, USA, 2012. ACM.

[32] Z. l. Wu, C. h. Li, J. K. y. Ng, and K. R. p. h. Leung. Location Estimation via Support Vector Regression. *IEEE Transactions on Mobile Computing*, 6(3):311–321, Mar. 2007.

[33] Z. Xiao, H. Wen, A. Markham, and N. Trigoni. Lightweight map matching for indoor localisation using conditional random fields. In *IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*, pages 131–142, Apr. 2014.

[34] Z. Yang and Y. W. and. Adamap: Adaptive radiomap for indoor localization. In *Ad-hoc, Mobile, and Wireless Networks - 14th International Conference, ADHOC-NOW 2015, Athens, Greece, June 29 - July 1, 2015, Proceedings*, pages 134–147, 2015.

[35] Z. Yang, C. Wu, and Y. Liu. Locating in fingerprint space: Wireless indoor localization with little human intervention. In *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking*, Mobicom '12, pages 269–280, New York, NY, USA, 2012. ACM.

[36] J. Yin, Q. Yang, and L. Ni. Learning Adaptive Temporal Radio Maps for Signal-Strength-Based Location Estimation. *IEEE Transactions on Mobile Computing*, 7(7):869–883, 2008.

[37] J. Yin, Q. Yang, and L. M. Ni. Adaptive temporal radio maps for indoor location estimation. In *3rd IEEE International Conference on Pervasive Computing and Communications (PerCom 2005), 8-12 March 2005, Kauai Island, HI, USA*, pages 85–94, 2005.

[38] M. Youssef and A. Agrawala. The horus wlan location determination system. In *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services*, MobiSys '05, pages 205–218, New York, NY, USA, 2005. ACM.

[39] F. Zampella, A. Jimenez R, and F. Seco. Robust indoor positioning fusing PDR and RF technologies: The RFID and UWB case. In *2013 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–10, Oct. 2013.

Received July, 2017; revised November 2017; accepted March 2018