

## Fingerprint positioning based on piecewise filtering of received signal strength indices and space-scene constraints

Yong Shi\*, Xianjian Xiao, Fuqiang Lu, Xiaofei Yang

School of Computer Information and Engineering, Changzhou Institute of Technology, Changzhou 213032, China

Corresponding Author Email: [shiy@czu.cn](mailto:shiy@czu.cn)

<https://doi.org/10.18280/rces.050203>

### ABSTRACT

**Received:** November 17 2017

**Accepted:** June 29 2018

**Keywords:**

*fingerprint positioning, piecewise filter, space-scene, received signal strength indices (RSSIs)*

This paper aims to reduce the dimensionality in fingerprint algorithm and achieve the optimal positioning accuracy at the minimal cost. For these purposes, the piecewise feature of iBeacon signal transmission was taken as the filtering factor of fingerprint positioning and adopted to filter the received signal strength indices (RSSIs) collected in real time. Then, the related fingerprints were filtered into fragments for subsequent online matching. After that, the indoor space-scene was divided into passage and hall, and the relevant constraint factor and data structure were discussed for fingerprint indexing. On this basis, the author proposed a novel method to optimize fingerprint positioning considering RSSI filtering and space-scene constraints. The experiments on an office space-scene reveal that the proposed method achieved the same result as the traditional one using 88% shorter matching time. This research provides an efficient and accuracy way of fingerprint positioning.

## 1. INTRODUCTION

Fingerprint positioning generally consists of such three steps as grid partitioning, offline collection and online matching [1]. Specifically, the grid partitioning can be expressed as  $grid_i = \{(x, y)\}$  (1-1) with  $i$  being the grid number and the tuple  $(x, y)$  being the coordinates of each grid; the offline collection can be expressed as a chain of received signal strength indices (RSSI) of the acquired fingerprints:  $fingerprint = (RSSI_1, RSSI_2, \dots, RSSI_n)$  (1-2). Among various online matching methods, the uncertainty matching is a relatively simple approach [1]. By this method, the Euclidean distance between real-time RSSI and fingerprint is calculated, and then the fingerprint corresponding to the minimum distance is selected according to  $\min(D), D = \sqrt{\sum_{i=1}^n (RSSI_i - FP_i)^2}$  (1-3) with  $FP_i$  being fingerprints. Below is a brief review of relevant studies on fingerprint positioning.

Reference [2] suggests that a high accuracy can be achieved with only a few grids, but the limited number of grids may increase the workload of fingerprint collection, sorting and maintenance. Reference [3] proposes a partitioning strategy considering pedestrian features and applies it to divide an indoor space into reachable and unreachable areas with a few grids. Reference [4] develops a time-saving online matching method by indexing the known sample points and restricting the search area. Reference [5] discovers the length inconsistency between fingerprints collected in real-time and those in the library and attributes the disparity to the interference of indoor environment and the power attenuation or damage of sensors. To overcome the problem, References [6,7] filter the fingerprint length via piecewise fitting and prove the validity of the filtering method.

This paper aims to reduce the dimensionality in fingerprint algorithm and achieve the optimal positioning accuracy at the

minimal cost. For these purposes, the fingerprint positioning method was discussed in details considering piecewise filtering and space-scene constraints [8] and subjected to accuracy and complexity verification through the case study.

## 2. RSSI PIECEWISE FILTER

### 2.1 RSSI piecewise filtering factor

Out of iBeacon signal transmission features, the piecewise feature was taken as the filtering factor of fingerprint positioning : (1) the factor is an RSSI at a division point; (2) the factor is used as a filter when it is necessary to select an RSS (beacon) as the parameter of the positioning algorithm; (3) the RSSIs between zero and the factor value can be trusted. [7]

### 2.2 Role of the filtering factor

**Table 1.** RSSI filtering algorithm

Input: $\langle M_i, RSSI_i \rangle$ // $M_i$ represents a beacon and its RSSI at time $t$ .
list_RSSI // list of $\langle M_i, RSSI_i \rangle$
Output: list-M // list of the selected beacons
1: For each $M_i \in \text{list\_RSSI}$
If $M_i, RSSI_i > -79\text{dBm}$
list_M.add ( $M_i$ ) // store in list_M
list_RSSI.remove( $M_i$ ) // Remove $M_i$ form list1_RSSI
2: If (list_M.size()=0) break.
3: For each(list_M)
Sort RSSI value by max value.

The RSSI piecewise filter performs differently from traditional RSSI selection, which is usually based on the

minimum RSSI variation. The RSSI filtering algorithm, the RSSIs after mean filtering, and the selected RSSIs are presented in Table 1, Table 2 and Table 3, respectively.

**Table 2.** RSSI values after mean filtering

beacon location \	b1	b2	b3	b4	b5	b6
11	-74	-54	-79	-73	-75	-60
12	-69	-72	-76	-65	-70	-65
13	-64	-73	-74	-66	-69	-60
14	-63	-79	-66	-71	-73	-70

**Table 3.** Selected RSSIs by piecewise filter

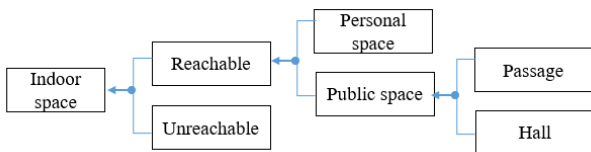
beacon location \	b1	b2	b3	b4	b5	b6
11	-74	54	79	-73	75	65
12			o	*		*
13		o	o	o	o	*
14		o*				

Note: \*piecewise filtering ominimum change filtering

### 3. SPACE-SCENE CONSTRAINTS

#### 3.1 Space-scene classification

The indoor space-scene was divided into passage and hall (Figure 1) by the classification standard for the reachable and the unreachable, after analysing the pedestrians and positioning requirements in offices, shopping malls, museums, etc.



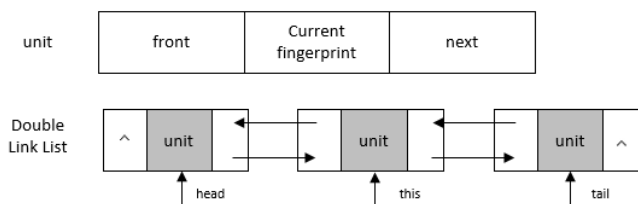
**Figure 1.** Space-scene classification

#### 3.2 Passage constraint

The passage space-scene was not constrained by the commonly used central line. Firstly, the author proposed a data structure called double-link-list (DLL) to index fingerprints (Figure 2). The DLL can be expressed as:

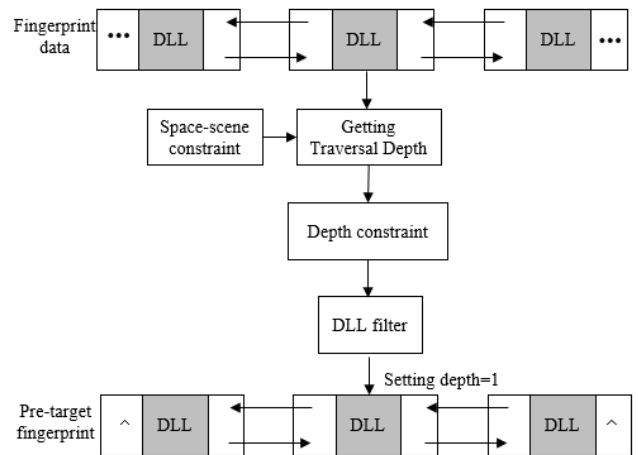
$DLL = (cur, FP, front, next, grid)$  (3-1).

where cur is the current fingerprint; FP is the value of the current fingerprint; front is the previous fingerprint; next is the next fingerprint; grid are the coordinates of the current fingerprint.



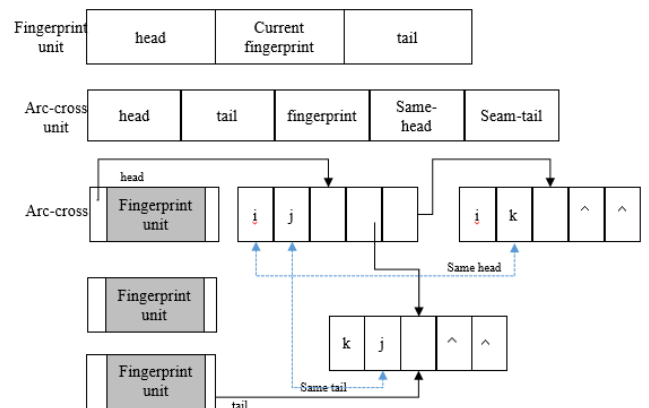
**Figure 2.** Structure of the DLL

The traversal depth of the DLL represents the constraint on the reduction of matching time. If the pedestrian speed is 2m/s, the traversal depth will be 2 with the grid radius of 1m. This principle of the DLL is illustrated in Figure 3.

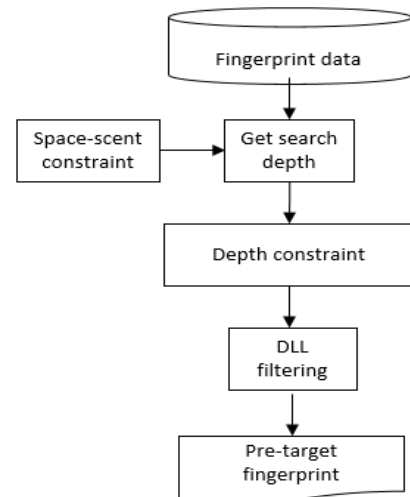


**Figure 3.** Principle of the DLL

#### 3.3 Hall constraint



**Figure 4.** Structure of the ALL



**Figure 5.** Principle of the ALL

An arc-cross link list (ALL) (Figure 4) was defined for the indexing of fingerprints in the hall space-scene:

$ALL=(cur, FP, front, next, left, right, grid) (3-2)$

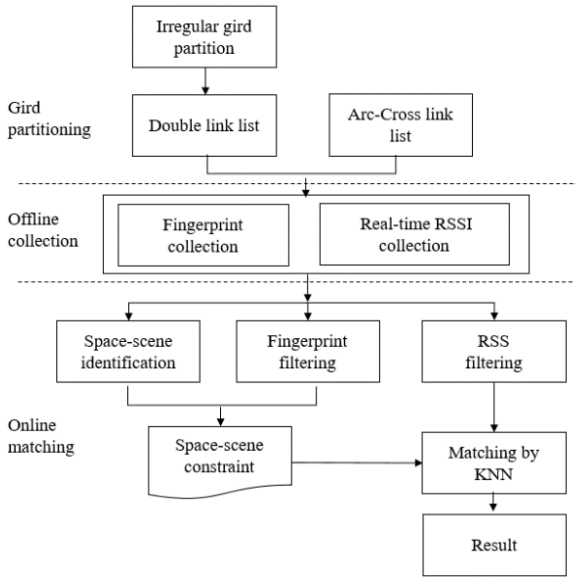
where  $cur$  is the current fingerprint;  $FP$  is the value of the current fingerprint;  $front$  is the previous fingerprint;  $next$  is the next fingerprint;  $left$  is the left fingerprint;  $right$  is the right fingerprint;  $grid$  are the coordinates of the current fingerprint.

Inspired by Reference [3], the ALL algorithm was constructed as Figure 5.

#### 4. FINGERPRINT POSITIONING CONSIDERING RSSI PIECEWISE FILTERING AND SPACE-SCENE

##### 4.1 Structure of the proposed positioning method

According to the steps of fingerprint positioning, the proposed positioning method was divided into three parts, namely, grid partitioning, offline collection and online matching.



**Figure 6.** Structure of the proposed fingerprint positioning method

According to Figure 6, the proposed method involves the following steps: (1) The real-time RSSI was filtered by algorithm 2-1 using the three optimal beacon nodes, because it has been proved repeatedly that the fingerprint fragments obtained with 3 beacons are the same with those obtained with all beacons in the k-nearest neighbours (KNN) matching. (2) The target space-scene was segmented through feature analysis on special motion constraints like direction and speed at the current position and the assessment of the pre-position. (3) The fingerprint was filtered by piecewise factor into fragments as step (1). (4) The optimal coordinates were outputted, and the error between the real-time RSSI in step (1) and the fingerprint in steps (2, 3) was computed by the KNN.

##### 4.2 Filtering of fingerprint fragments

The nature of filtering is to reduce the length of each fingerprint. When the real-time RSSI was filtered by piecewise factor, the fingerprint should also be filtered by the algorithm in Table 4.1 for further use.

**Table 4.** Fingerprint filter algorithm

---

Input:  $V//speed, t//time, ARC, L//cur\ fingerprint, L//length\ of\ L$   
Output:  $fragment//each\ fingerprint\ selected, pre-List//pre-selected\ of\ fingerprint$

Based on table 2.

```

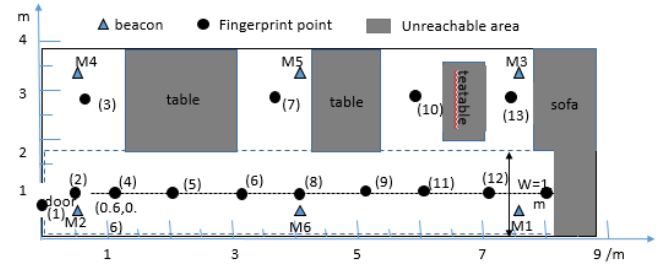
1: foreach beacon rss_M
  foreach (fingerprint )
    if(beaconfingerprint)
      segment.add(beacon)
    end for
  pre-List.add(segment)
end for
2: foreach (segment pre-list)
  if segment.length<3
    remove(segment)
  3: if segment_real.length>3
Select(max(3)); return;
  
```

---

#### 5. CASE STUDY

##### 5.1 Overview

As shown in Figure 7, the study area is an 8.9m-long, 3.8m-wide hall space-scene. The industrial iBeacon was selected as the reference beacon. The sensor deployment is shown in Table 5.



**Figure 7.** Study area

**Table 5.** Sensor deployment

Beacon minor	Coordinate x(m)	Coordinate y(m)	Hight(m)
M 1	7.7	0.6	2.8
M 2	0.6	0.6	2.8
M 3	7.7	3.2	2.8
M 4	0.6	3.2	2.8
M 5	3.6	3.2	2.8
M 6	3.6	0.6	2.8

##### 5.2 Experimental design

The experiment was carried out five times at points 2, 7, 9 and 10 in Figure 7. In each time, the RSSI data were collected in real time for 3min at the frequency of 300ms. In this way, over ten thousand RSSI values were acquired originally, which could reduce the effect of singular value on the positioning result.

And according to Figure 6, the experimental procedure is as follows: (1) Select a test point from Figure 7 and determine its coordinates; carry out five positioning experiments; in each experiment, collect the RSSI data in real time for 3min at the frequency of 300ms and conduct mean filtering of the data. (2) Execute algorithm 2-2 and filter the RSSI data by the filtering

Factor. (3) Determine the optimal transversal depth according to the constraint of the space-scene (hall). (4) Output the optimal coordinates and compute the error by the KNN.

### 5.3 Experimental results

The collected fingerprints are listed in Table 6, and the real-time RSSIs are recorded in Table 7.

**Table 6.** Collected fingerprints

beacons fingerprints	iBeacon1 (dBm)	iBeacon 2 (dBm)	iBeacon 3 (dBm)	iBeacon 4 (dBm)	iBeacon 5 (dBm)	iBeacon 6 (dBm)
1	-87.1613	-68.5781	-91.2619	-73.0635	-73.0952	-68.8793
2	-84.2231	-59.2529	-88.4928	-70.1717	-77.4435	-65.1331
3	-84.7087	-72.5739	-89.691	-61.2825	-73.0271	-73.2463
4	-80.7486	-69.7258	-88.2798	-86.3377	-77.2811	-66.6485
5	-82.5902	-61.9753	-86.842	-83.698	-73.7031	-74.687
6	-79.6654	-73.3355	-88.0583	-80.2636	-73.1264	-64.2287
7	-74.2987	-68.556	-84.5	-70.2772	-65.4437	-66.3141
8	-76.2927	-67.0545	-87.1484	-79.6852	-71.2748	-58.0402
9	-66.4031	-76.1891	-82.6061	-76.7274	-72.2813	-61.7605
10	-69.955	-77.2222	-77.5612	-78.3473	-69.133	-71.5012
11	-67.1282	-84.4528	-75.3544	-84.1654	-76.7576	-69.0885
12	-62.1652	-81.6415	-73.1179	-91.4777	-81.2	-73.903
13	-63.2005	-78.4128	-72.6258	-82.91	-81.9084	-79.9232

**Table 7.** Real-time RSSIs

Test Point	Positioning Target	iBeacon1 (dBm)	iBeacon 2 (dBm)	iBeacon 3 (dBm)	iBeacon 4 (dBm)	iBeacon 5 (dBm)	iBeacon 6 (dBm)
1	fingerprint 2	-74	-54	-79	-73	-75	-65
2	fingerprint 7	-69	-72	-76	-65	-70	-65
3	fingerprint 9	-64	-73	-74	-66	-69	-60
4	fingerprint 10	-63	-79	-66	-71	-73	-70

Table 8 shows the filtering results of real-time RSSIs by algorithm 2-2. It can be seen that the RSSIs no smaller than the filtering factor (-77dBm) were taken as the positioning

parameters. Here, the k value of the KNN is set to 3. The KNN-filtered fingerprint corresponding to the first test point is also given in Table 8.

**Table 8.** Filtered real-time RSSIs and fingerprint corresponding to test point 1

Test Point	Positioning Target	iBeacon1(dBm)	iBeacon 2(dBm)	iBeacon 3(dBm)	iBeacon 4(dBm)	iBeacon 5(dBm)	iBeacon 6(dBm)
1	fingerprint 2		-54		-73		-65
2	fingerprint 7	-69			-65		-65
3	fingerprint 9	-64			-66	-	-60
4	fingerprint 10	-63		-66			-70

**Table 9.** Pre-selected fingerprint and fingerprint corresponding to test point 1

beacons fingerprints	iBeacon1 (dBm)	iBeacon 2 (dBm)	iBeacon 3 (dBm)	iBeacon 4 (dBm)	iBeacon 5 (dBm)	iBeacon 6 (dBm)
1		-68.5781		-73.0635		-68.8793
2		-59.2529		-70.1717		-65.1331
3		-72.5739		-61.2825		-73.2463
4		-69.7258		-86.3377		-66.6485
5		-61.9753		-83.698		-74.687
6		-73.3355		-80.2636		-64.2287
7		-68.556		-70.2772		-66.3141
8		-67.0545		-79.6852		-58.0402
9		-76.1891		-76.7274		-61.7605
10		-77.2222		-78.3473		-71.5012
11		-84.4528		-84.1654		-69.0885
12		-81.6415		-91.4777		-73.903
13		-78.4128		-82.91		-79.9232

Table 9 presents the fingerprint pre-selected by algorithm 3-2 under the ALL constraint and the traversal depth of 1, and the KNN-filtered fingerprint corresponding to the first test point.

The fingerprint derived from the minimum values in Table 9 is shown in Table 10.

**Table 10.** Positioning result

Test point fingerprint	p1 (m)	p2 (m)	p3 (m)	p4 (m)
1	15.08556			
2	5.967408			
3	23.45828			
4	20.68605			
5				
6		18.63661		
7		7.592886		
8		17.81228	18.49962	
9			11.13334	18.23737
10			17.89401	15.8654
11			20.5516	17.47013
12				
13				

### 5.4 Results analysis

The positioning result of the traditional fingerprint positioning algorithm is displayed in Table 11 and compared with that of the proposed method in Table 12.

**Table 11.** Result of the traditional fingerprint positioning algorithm

Test point	Result	Pre-target	Result
1		Finger print 2	Finger print 2
2		Finger print 7	Finger print 7
3		Finger print 9	Finger print 9
4		Finger print 10	Finger print 10

**Table 12.** Comparison between the results of the traditional and proposed algorithms

Method	Length of each fingerprint segment	Mount of fingerprint segment	Time mount of matching
Traditional	6	13	6*13*13
This paper	3	3	3*3*13

The results in the above tables demonstrate that the proposed algorithm achieved the same result as the traditional algorithm but saved 88% of matching time. Thus, the proposed algorithm is an efficient and accuracy way of fingerprint positioning.

## 6. CONCLUSIONS

This paper proposes a fingerprint positioning algorithm with piecewise filtering factor and pedestrian scene features as

the optimization parameters. The proposed algorithm was validated through a case study on an office scene, revealing that it could reduce the matching time by 88%. Suffice it to say that the proposed algorithm is an efficient and accuracy way of fingerprint positioning. In future research, the space-scene will be further categorized to yield more accurate positioning constraints, and the proposed method will be applied to more positioning cases.

## ACKNOWLEDGMENT

This Research is partially supported by National Natural Science Foundation of China (Grant No.41571382), Supported by the Natural Science Foundation of the Jiangsu Higher Education Institutions of China (Grant No. 15KJB170006, No.16KJB520003). Supported by Taizhou Science and technology support program of China (Grant No. TS201621), Supported by Changzhou Science and technology support program of China (Grant No. CE20172023). Supported by Collaborative Innovation Center of Changzhou Institute of Technology for Digital Information Technology, and supported by Excellent Scientific and Technological Innovation Team of Changzhou Institute of Technology.

## REFERENCES

- [1] Chen L. (2014). Key technologies research on fingerprinting positioning based on WLAN. Huadong Normal University.
- [2] Hang G. (2014). Fingerprint database optimization algorithm based on zigbee indoor positioning system. *Computer Engineering*, 40(2): 193-198.
- [3] Xu Y, Shi Y, Zheng X, Long Y. (2016). An Indoor space partition method and its fingerprint positioning optimization considering pedestrian accessibility. *ISPRS Archive*, XLI-B4 347-350. <https://doi.org/10.5194/isprs-archives-XLI-B4-347-2016>
- [4] Zhou Y, Cao H, Li JX. (2007). A shortest route-planning algorithm within a restricted area. *Microelectronics and Computer* 24(8): 110-112.
- [5] Xie DJ, Kong FZ, Hu HY. (2014). Research on robustness of location fingerprint under terminal heterogeneity. *Computer Engineering* 40(5): 81-85.
- [6] Lin FX, Zhu MH. (2015). Adaptive piecewise curve fitting indoor localization algorithm based on RSSI. *Transducer and Microsystem Technologies* 34(10): 151-153.
- [7] Shi Y, Long Y, Xu Z. (2017) Indoor RSSI trilateral algorithm considering piecewise and space-scene. *IEEE International Conference on Smart Cloud* 278-282.
- [8] Shi Y. (2016). Indoor positioning algorithm considering space-scene. Nanjing Normal University 12.