Removing Useless APs and Fingerprints from WiFi Indoor Positioning Radio Maps

Samih Eisa, João Peixoto, Filipe Meneses, Adriano Moreira

Mobile and Ubiquitous Systems research group,
Centro Algoritmi, University of Minho
Guimarães, Portugal

Abstract—**Maintaining consistent radio maps for WiFi fingerprinting-based indoor positioning systems is an essential step to improve the performance of the positioning engines. The radio maps consist of WiFi fingerprints collected at a predefined set of positions/places within a positioning area. Each fingerprint consists of the identification and radio signal level of the surrounding Access Points (APs). Due to the wide proliferation of WiFi networks, it is very common to observe 10 to 20 APs at a single position and more than 50 APs across a single building. However, in practical, not all of the detected APs are useful for the position estimation process. Some of them might have weak signals at certain positions or might have less significance for a position's fingerprint. Thus, those useless APs will add additional computational overheads during the position estimation, and consequently they will reduce the overall performance of the positioning engines. A similar phenomenon also occurs with some of the collected fingerprints. While it is widely accepted that the larger and more detailed the radio map is, the better is the accuracy of the positioning system, we found that some of the fingerprint samples on the radio maps do not contribute significantly to the estimation process. In this paper, we propose two methods for filtering the positioning radio maps: APs filtering and Fingerprints filtering. Then we report on the results of a set of experiments that have been done to evaluate the performance of a WiFi positioning radio map before and after applying the filtering approaches. The results show that there is possibility to simplify the radio maps of the positioning engines without significant degradation on the positioning precision and accuracy, and therefore to reduce the processing time for estimating the position of a tracked WiFi tag. This result has an important impact on increasing the number of tags a single instance of a WiFi positioning engine can handle at a time.**

Keywords- WiFi indoor location; positioning engine performance; filtering positioning radio maps.

I. INTRODUCTION

Indoor positioning systems using the IEEE 802.11 (WiFi) technology has a distinct advantage over other indoor positioning technologies in offering low cost, easily deployable, and highly available positioning systems. This is due to the fact that WiFi networks nowadays become quite common in many places and as a result, services/applications based on WiFi technology are getting popular. Most of the WiFi Indoor positioning systems make use of the Fingerprinting technique [1-2]. The Fingerprinting technique uses the method of matching WiFi signals measured at unknown positions with pre-measured signals collected at known positions. Hence, two main phases can be distinguished for the Fingerprinting technique: Calibration phase (offline) and positioning phase (online). In the calibration phase, a fingerprinting radio map for the positioning area is constructed by collecting WiFi signals from nearby Access Points (APs) and associating those WiFi measurements with physical coordinates/places creating what we called calibration points. The constructed radio map will be used later on during the positioning phase to approximate the position of the tracked person/object. The challenge of this method is how to maintain consistent radio maps that help in providing accurate position estimates with better performance in terms of estimation response time [3]. In fact, each fingerprint sample on the positioning radio map consists of the Received Signal Strength Indicator (RSSI) of the surrounding WiFi Access Points (APs) and due to the wide proliferation of WiFi networks it is very common to observe a large number of WiFi APs at a single position. However, not all of the detected APs are useful for the positioning radio maps. Some of the detected APs might have weak signals at certain positions leading to non-trivial inconsistency on the radio maps. This inconsistency will add additional computational overheads during the position estimation and, as a results, it will reduce the overall performance of the positioning engines. Similarly, not all of the collected fingerprint samples contribute significantly to the estimation process. Only part of the fingerprint samples has considerable impact during the positions matching. Taking into account these points and based on some experimental tests using Weka toolkits [4], we found that we could remove those useless APs and fingerprints samples from the positioning radio maps without significant degradation on the positioning precision.

In this paper, we propose two approaches for filtering the WiFi positioning radio maps: APs filtering and Fingerprints filtering. The APs filtering is defined to simplify the positioning radio maps by removing the useless APs, while the Fingerprints filtering deals with removing the "bad" fingerprint samples from the positioning radio maps by applying some classification rules. In this paper we present the results of a set of experiments that have been done, using a developed java tool with the support of Weka APIs, to evaluate the identified filtering approaches using positioning datasets collected from a WiFi indoor positioning prototype developed in our lab. Furthermore, we applied the identified filtering approaches to evaluate the performance of our positioning engine in terms of computational load, positioning precision and positioning error rate. The results of the performance evaluation, before and after applying the filtering approaches, also are presented.

The reminder of this paper is organized as follows. In section II we give a brief description about the positioning radio maps in general. In section III we present some of the related work, describing briefly some of the prior work for filtering the positioning radio maps. Section IV describes the proposed filtering approaches. Section V gives an overview about the experiments details, describing the data collection and the preprocessing methods. Section VI reports on the results of the experiments and section VII concludes the paper.

II. POSITIONING RADIO MAPS

The first step in performing indoor positioning using the Fingerprinting technique is the construction of a radio map for the positioning area. The radio map contains a large number of entries that represent the WiFi coverage within the positioning area. Each entry is associated with physical coordinates/place within the positioning area creating what we called calibration points. To approximate the current position of a tracked WiFi tag/terminal, a matching process has to be made to compare the radio signals measured by the WiFi tag (test sample) against the stored entries of the radio map. As many entries the radio map may contain, the matching time will be longer, reducing the overall performance of the positioning engine. In this research work the goal is to propose methods for reducing the size of the radio maps by examining the entries of the radio maps in order to identify useless entries that do not contribute significantly to the position estimation process. Those entries might be collected from weak or far distant APs that are not deployed within the positioning area. By eliminating those useless entries we can actually reduce the size of the radio maps and also reduce the proceeding time during the matching phase.

III. RELATED WORK

Despite the fact that the WiFi Fingerprinting technique is becoming much more attractive for indoor localization, the effort needed for performing the site survey (calibration) is still an issue. Most of the prior work focused on how to construct the positioning radio maps by examining new ways for data collection including participatory sensing [5-6]. However, on the other hand, few works has been done on the data management of the positioning radio maps. In [3], Kim et al, proposed a new scheme for filtering the redundant fingerprint samples based on clustering techniques. The paper focused on two management aspects: scalability and consistency. The proposed scheme clusters the fingerprints collected at different time periods with different types of devices and then filters the redundant fingerprints while keeping only representative fingerprints. Another approach also has been proposed in [7]. The authors proposed a method for reducing the calibration effort for a positioning system that used for estimating the location of a tracked user in a multi-floor environment. The proposed method aims at reducing the effort by collecting and labeling fingerprint data only on one floor and then collecting unlabeled data on the other floors and then co-embed these two different data in one common manifold. Even though, the authors showed a promising effort-reduction method, further investigation in this method is still required. Furthermore, in [8-10] the problem of real-time APs selection for WLAN positioning has been addressed. The authors in [8] proposed an

algorithm that uses parametric and nonparametric divergence measures for selecting only subset of the available APs, within the positioning area, to be used for position estimation. Also the authors in [9] incorporated the RSSI measurements into the estimation method and couple that with a methodology that indicates which of the detected APs can be ignored during positioning without degradation on the overall position accuracy. Both methods were proposed to reduce the impact on the computational complexity of the positioning algorithm. In [10], Chen et al, proposed a power-efficient location estimation algorithm that, intelligently, selects the number of APs used for location estimation. The algorithm uses machine learning techniques (Clustering and Decision Tree) to choose only subset of the detected APs for positioning with less computational power and high location accuracy.

IV. PROPOSED FILTERING APPROACHES

Two approaches are proposed in this paper for filtering the WiFi positioning radio maps. The first filtering approach is defined to exclude the "useless" APs from the positioning radio maps while the other approach is defined to reduce the number of fingerprint samples in the positioning radio maps by removing the "bad" fingerprint samples. This reduction will be an essential step to reduce the computation overheads that might occur during the position estimation and thus, enhance the overall performance of the positioning engines afterwards.

A. Filtering APs

The useless APs are those APs that do not contribute significantly during the position estimation. By identifying some statistical rules based on the received signal level of the detected WiFi APs within the positioning area, those useless APs can be clearly identified and removed from the positioning radio map. The following are the statistical (heuristics) rules that we have defined for APs filtering. The identified thresholds for each of the three heuristics rules have been optimized by performing multiple tests using different values for each rule. The chosen thresholds scheme showed the best results during the experiments.

1) Number of Distinct RSSI values

The number of distinct RSSI values measured for an AP (RSSI values are negative integer numbers). A broad range of RSSI values tends to reflect good distribution on the positioning radio maps. Thus, as a rule the APs that have less than or equal to 10 distinct RSSI values should be discarded.

2) Percentage of missing RSSI values

Not all of the available APs within the positioning area are observed at every calibration point. A missing value for AP_i is defined as a fingerprint where AP_i is not observed. Thus the APs with higher percentage of missing values should be discarded as it might be an indicator of unreliable APs. The APs with percentage of missing values, across the entire radio map, greater than 80% should be discarded.

3) Overall Standard Deviation (STDEV)

The overall STDEV of the RSSI values measured for an AP. Higher STDEV means that the differences between the RSSI values will vary leading to more diverse values across different positions and that means relatively good distribution.

The small STDEV tends to be more confusing to differentiate between the RSSI values as the differences will be smaller leading to similar distribution at most of the calibration points even if the points are not closed to each other. This criterion has to be combined with the previous rules to discard the APs with lower STDEV. The APs with STDEV less than or equal to 5 should be discarded.

Table I lists the results of applying different thresholds schemes for APs filtering. In the table we presented the corresponding number of useful APs after applying each of the listed threshold schemes (the total number of APs before filtering is 50).

TABLE I.   DIFFERENT THRESHOLDS SCHEMES FOR APs FILTERING

Scheme	# Distinct values	% missing values	STDEV	Useful APs
A	5	95	2	21
B	10	80	5	16
C	15	90	3	20

As shown in Table I, the identified thresholds scheme (scheme B) was the best among the others. Only 16 out of 50 APs were selected as useful APs. Nevertheless, these results do not mean that the identified thresholds will be applicable for each positioning map. It is based on the statistics of the RSSI measurements of the positioning map. These values have been defined based on our positioning radio map. We actually looked at the statistics of the datasets on Weka GUI tools and then we defined those values according to the distribution of the RSSI values measured for each AP in the dataset. The precision and accuracy results before and after APs filtering are presented in section VI-A.

B. Filtering Fingerprints

The aim of this filtering approach is to reduce the number of fingerprint samples involved in the positioning radio map by removing the "bad" fingerprint samples. The idea behind this approach is to use machine learning classification algorithms to classify the fingerprint samples in the positioning radio map. The role of the classification algorithms is to classify the samples into as many classes as the number of distinct rooms in the positioning area. The samples that are not classified on the correct class (room) will be identified as outliers or "bad" samples, and will be removed from the positioning radio map. For performing this filtering approach we used the J48 algorithm for classification, because it showed the best performance compared to other algorithms [11]. We configured the examined classifier (J48) to use the option "Use training set" in Weka to perform the classification. This option allows the classifier to use all of the samples in the positioning radio map as training set and then classify each individual instance. Based on the classification results, only the correctly classified instances should be considered for position estimation. Using this approach allows the positioning engines to use only the most probable correct portion of the positioning radio map during the position estimation and as a result, reduce computational overheads, as the number of samples will be reduced. The results of this filtering approach are presented in section VI-A.

V.   EXPERIMENTS

Several experiments have been done to evaluate the performance of the positioning radio maps before and after applying the filtering approaches described in the previous section. The main objective of the experiments is to assess the impact of eliminating useless APs and "bad" fingerprint samples (outliers) from the WiFi positioning radio maps on the precision of the position estimation. We used the J48 algorithm here as well to evaluate the estimation precision before and after applying the filtering approaches.

Another set of experiments have also been done to evaluate the performance of our WiFi positioning engine, a positioning engine that has been develop at our lab [12]. The performance evaluation is defined in terms of positioning accuracy, positioning mean error and processing time. Further details will be described in section VI-B.

A. Data Collection

The data for the experiments were collected from a WiFi indoor positioning prototype developed in our lab. The prototype consists of two main tools: Calibration tool and positioning engine. The prototype uses the Ultra-low Power WiFi module from GainSpan [13] to represent the WiFi tag sensor. Each WiFi tag sensor consists of a GS1011M (802.11b) RF Module and On-board PCB Trace Antenna type. The tag was mounted on a calibration car and connected to a laptop computer through a serial port. For the data acquisition, a developed calibration tool running on the laptop computer was used to identify the calibration points on a floor plan map representing the positioning area. The points specify the positions where the WiFi samples were to be collected. The experiments were carried out in a building that consists of two floors with multiple and different size rooms in each floor. Figure 1 shows the calibration points on the map. During the calibration phase, the developed calibration tool was configured to collect three samples of WiFi data at each point with a 2 second time interval between each two samples. Each sample consists of: a unique identifier (SampleID); a timestamp; the coordinates of the calibration point (x,y); the floor; the room; a list with the identification (BSSID) and corresponding signal level (RSSI) values of the surrounding WiFi APs. Not all of the rooms were involved in the calibration phase, as shown in Figure1.

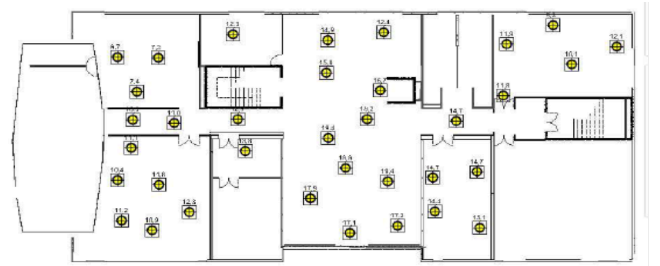

Figure 1.   Calibration Points on the floor map.

These data is herein described as the "Calibration Set". Additional data, with the same format, were collected on different days for testing purposes, namely for offline

assessment of the position estimation. These data is herein described as the "Testing Set". The points that were selected as testing points were not coincided with the calibration points. The testing points were distributed in a random fashion throughout the positioning area and also not all of the rooms were involved.

B. Data Preprocessing

Before applying the proposed filtering approaches we developed a preprocessing application to put the collected Calibration Set and the Testing Set in a format suitable for the filtering process. First of all, we gathered the data in columns with MAC addresses as headers and additional column for the location of the measurements (i.e. the rooms). Each MAC address column contains the corresponding RSSI values collected from a single AP at each calibration point. This data format is required for the filtering process. Also as part of the preprocessing phase, we normalized the samples of the datasets to handle the missing values. We applied multiple normalization values in order to evaluate the impact of each value on the obtained results. For each value we generated separate Calibration and Testing Sets. We have chosen these values based on the results presented in [9]. The authors presented experimental results to illustrate the variation on the RSSI values of a single AP for two different collecting setups over a period of 10 hours. Based on the results we have got an idea about the applicable range for the RSSI values and therefore, we identified different normalization values to evaluate their impact on the experimental results. We defined values {-85,-95,-100,-110, NaN} to be used for normalization. The results showed that using different values for normalization has significant impact on the position estimation results. Further discussion will be presented in section VI.

VI. RESULTS

We categorized the final results into two main categories, described in the following subsections.

A. Classification results

As mentioned before, the J48 classifier was used to evaluate the performance of the positioning radio maps before and after applying the filtering rules. The classification results were collected for each normalization value, meaning that we evaluated the estimation precision for each of the previously mentioned normalization values. However, in this section we will present the flow of the experiments in a diagram in figure 3, showing only the results of the -85 datasets, because it showed the best results among the other values. Further results of the experiments are presented in the Appendix.

Figure 2 shows the result before applying the filtering. This result will be used as reference to compare with the other obtained results after filtering. The original Calibration Set consists of 2376 fingerprint sample whereas the Testing Set consists of 630 fingerprint samples. The number of distinct APs in both datasets is 50 APs.
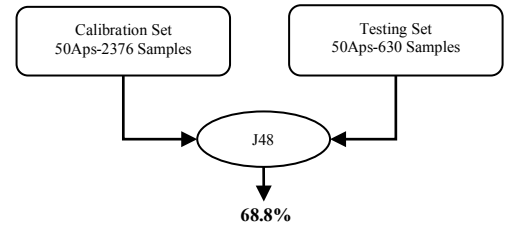

Figure 2. Experiment flow-results before filtering

As shown in the figure, we obtained 68.8 % precision in estimating the correct room before filtering. This precision will be used as reference to compare the results after applying the filtering approaches.

1) Removing "useless" APs

To remove the useless APs from the positioning radio map (Calibration Set) we applied each of the previously mentioned APs filtering schemes (Table I) to the original Calibration Set before normalization. This is because the APs filtering approach uses the original statistics of the RSSI measurement to compute the filtering rules. The original Calibration Set includes 50 distinct APs, representing the detected APs within the positioning area. After applying the identified APs filtering schemes, the 50 APs have been reduced to a smaller number of APs according to each filtering scheme. As shown in Table I the filtering scheme B was the best among the other filtering schemes. The original 50 APs have been reduced to only 16 APs which is considered a remarkable reduction in the number of APs. The same filtering scheme also has been applied to the Testing Set and the number of APs also has been reduced to 16 APs.

Then the list of useful APs (i.e. 16 APs) was used together with the normalized Calibration and Testing sets to evaluate the precision of the position estimation using the J48 classifier. The filtered Calibration Set was used to train the J48 classifier and the Testing Set was used for testing. Table II shows the precision results after applying APs filtering scheme B.

TABLE II. APS FILTERING-PRECISION AND GAIN COMPARISON

Dataset	Before Filtering	After APs Filtering	Gain
-85	68.8%	72.5%	3.7
-95	70.9%	70.3%	-0.6
-100	70.9%	70.3%	-0.6
-110	70.9%	70.3%	-0.6
NaN	66.3%	65.8	-0.5

As shown in Table II the results illustrate the precision gain corresponding to each normalization value. We observed a precision gain of about 3.7% with the -85 datasets. Further filtering results using the other filtering schemes and datasets are presented in the Appendix.

2) Removing "bad" Fingerprints

To remove the bad fingerprints samples independently from the APs filtering, we applied the fingerprint filtering approach, described in section IV-B, to the normalized Calibration Set in order to evaluate the impact of each normalization value on the filtering results.

TABLE III.   FINGERPRINT FILTERING-PRECISION AND GAIN COMPARISON

Dataset	Before Filtering	After Fingerprints Filtering	Gain
-85	68.8%	70.1%	1.3
-95	70.9%	70.9%	0
-100	70.9%	70.9%	0
-110	70.9%	70.9%	0
NaN	66.3%	66.1%	-0.2

Table III presents the obtained precision gain after fingerprint filtering. The datasets with -85 showed the best precision results with 78 samples have been removed. Further results are presented in the Appendix.

3) Combined Filtering

Figure 2 illustrates the experiment flow when combining the two filtering approaches. First of all we applied the APs filtering to remove the useless APs and then we used the resultant Calibration Set as input for fingerprints filtering.
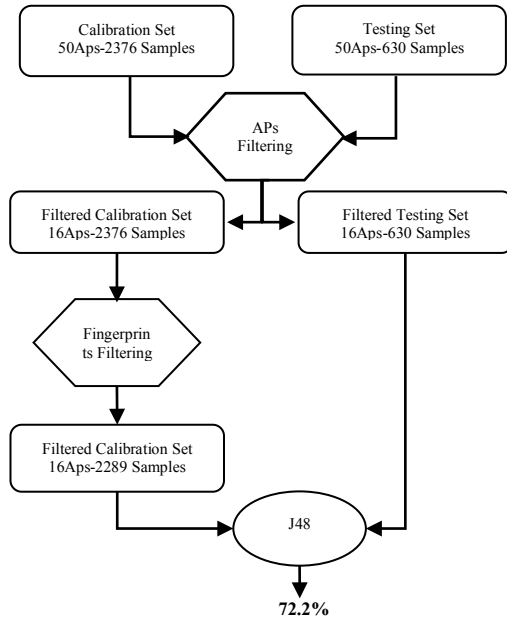

Figure 3.   Experiment flow-Combined filtering

As shown in the figure the number of fingerprint samples in the Calibration Set before filtering was 2376 samples and after applying the filtering rules this number has been reduced to 2289 samples. This result was obtained when using -85 datasets. Table IV shows the precision gain results of the combined filtering.

TABLE IV.   PRECISION AND GAIN COMPARISON

Dataset	Before Filtering	After Combined Filtering	Gain
-85	68.8%	72.2%	3.4
-95	70.9%	70.4%	-0.5
-100	70.9%	70.4%	-0.5
-110	70.9%	70.4%	-0.5
NaN	66.3%	66.0%	-0.3

As shown in the table the precision has been improved after applying the filtering rules with the -85 datasets in contrast to the other datasets. We observed precision gain about 3.4% after applying the combined filtering. Even though the precision results of the datasets with -85 were not the best before applying the filters, the precision has been improved after filtering. This results shows that there is possibility to reduce the size of the positioning maps while having the same/better precision. This result has considerable impact on the positioning processing time and the overall performance of the positioning engines. Further experimental results are provided in the Appendix.

B. Positioning engine performance results

This section presents the obtained performance results when applying the identified filtering approaches to evaluate our positioning engine. Four metrics are used to characterize the performance results:

- Right Floor Rate (RFR): refers to the frequency in correctly estimating the floor.

- Right Room Rate (RRR): refers to the frequency in correctly estimating the room.

- Average Error: refers to the average geometric error, in meters, as the average of the Euclidean distance between the correct position and the estimated position.

- Processing Time: refers to the average time required to estimate the position.

Table V illustrates the performance results of the positioning engine before applying the filtering approaches with respect to the normalization values. These results were used as reference to compare the obtained results after applying the filtering approaches. In this performance results we excluded datasets with NaN values, as the developed positioning engine accepts only numerical values.

TABLE V.   POSITIONING ENGINE PERFORMANCE RESULTS-BEFORE FILTERING

Dataset	Correct Floor	Correct Room	Mean Error (meter)	Processing time/sample(ms)
-85	98.7%	85.7%	2.77	10.82
-95	98.2%	76.6%	3.55	10.84
-100	98.2%	73.0%	3.98	10.85
-110	98.2%	69.0%	4.38	10.85

To obtain the processing time for a test sample, we computed the average processing time for 5 consecutive runs of the positioning engine on the same datasets. We divided the average value by the number of samples in the Testing Set. The result is the estimated processing time for each test sample. The processing time is presented in milliseconds.

Table VI illustrates the performance results after applying the APs filtering approach only. The presented results are based on scheme B of the APs filtering. Further results are presented in the Appendix.

TABLE VI.   POSITIONING ENGINE PERFORMANCE RESULTS-AFTER APS
FILTERING

Dataset	Correct Floor	Correct Room	Mean Error(meter)	Processing time/sample(ms)
-85	98.5%	81.5%	3.14	4.34
-95	98.4%	73.9%	3.81	4.3
-100	98.2%	71.4%	4.07	4.32
-110	96.6%	66.9%	4.40	4.34

As shown in the table, there is a considerable reduction in the processing time for estimating the position of a test sample. The processing time has been reduced to less than half of the original processing time when there was no filtering rule applied. However, on the other hand, there is a tiny reduction on the precision of estimating the correct floor when using the datasets with -85. The datasets with -85 were the best among the other datasets. Even though we experienced little degradation on the precision, it is still acceptable if we consider the significant reduction on the number of APs after applying the filtering approach. Table VII shows the performance results when using -85 datasets with respect to all of the APs filtering schemes. Further results for the other datasets are presented in the Appendix.

TABLE VII.   PERFORMANCE RESULTS-AFTER APS FILTERING FOR -85
DATASET AND ALL FILTERING SCHEMES

	A	B	C
%Correct Floor	98.4	98.5	98.5
%Correct Room	83.9	81.5	84.1
%Mean Error	2.80	3.14	2.84
Processing Time(ms)	4.82	4.34	4.67

As illustrated in Table VII scheme B does not always have the best performance. However, the remarkable reduction on the number of APs makes it an attractive option for APs filtering.

Table VIII illustrates the performance results after applying the combined filtering approaches (i.e. both: APs and Fingerprints filtering). As shown in the table, the datasets with -85 were the best among the other, as well. However, there is degradation on the precision of estimating the correct room when compared to the reference estimates before applying the filtering rules.

TABLE VIII.   POSITIONING ENGINE PERFORMANCE RESULTS-AFTER
COMBINED FILTERING

Dataset	Correct Floor	Correct Room	Mean Error(meter)	Processing time/sample(ms)
-85	98.5%	80.6%	3.19	4.19
-95	98.2%	73.4%	3.82	4.21
-100	98.2%	71.1%	4.08	4.14
-110	96.8%	66.8%	4.41	4.16

VII.   CONCLUSION

This paper reports on a set of experiments that have been done to evaluate the opportunity to simplify WiFi indoor positioning radio maps aiming to reduce the time required to estimate the position of a mobile device. The positioning radio maps usually consist of a large number of fingerprint samples collected within an operational area. WiFi network become quite common nowadays and accordingly, the radio maps might include fingerprint samples including a large number of APs. However, not all of the detected APs are useful for creating radio maps for positioning and also not all of the collected fingerprint samples are relevant for the position estimation process. In this paper we proposed two filtering approaches to reduce the amount of WiFi fingerprint samples used in the positioning radio maps. We have defined the filtering approaches based on some statistical and classification rules. The aim of the two approaches is to reduce the positioning radio maps to smaller portions that reduce the computational overheads, and consequently improve the overall performance of the positioning engine. Both filtering approaches have been used in a set of experiments using WiFi positioning datasets collected from a developed WiFi indoor positioning prototype. The results of the experiments showed that there is possibility to reduce the size of the positioning radio maps while keeping the same or less significant degradation on the positioning precision. The results showed a considerable reduction in the number of APs and fingerprint samples after applying the identified filtering approaches. Even though the precision results need to be improved, these results were enough to help us drawing some conclusions about the value of our approaches. We believe that this kind of experiments and results has an important impact on the performance of the positioning engines because it will lead, afterwards, to reduce the computational overheads generated during the online position estimation, and therefore leads to considerable improvement in the overall performance of the positioning engines. However, further research still is needed to evaluate multiple positioning maps for different WiFi positioning engines in order to verify the obtained results.

REFERENCES

[1] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in Proc. IEEE INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. , Tel Aviv, Israel, 2000, pp. 775-784.

[2] J. Xiao, K. Wu, Y. Yi, and L. M. Ni, "FIFS: Fine-Grained Indoor Fingerprinting System," in Computer Communications and Networks (ICCCN), 2012 21st International Conference on, 2012, pp. 1-7.

[3] Y. Kim, Y. Chon, H. Cha, M. Ji, and S. Park, "Scalable and Consistent Radio Map Management Scheme for Participatory Sensing-based Wi-Fi Fingerprinting."

[4] Weka. (27/7/2013). Weka 3: Data Mining Software in Java. Available: http://www.cs.waikato.ac.nz/ml/weka/

[5] Y. Kim, Y. Chon, and H. Cha, "Smartphone-based collaborative and autonomous radio fingerprinting," Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, vol. 42, pp. 112-122, 2012.

[6] J.-g. Park, D. Curtis, S. Teller, and J. Ledlie, "Implications of device diversity for organic localization," in INFOCOM, 2011 Proceedings IEEE, 2011, pp. 3182-3190.

[7] H.-Y. Wang, V. W. Zheng, J. Zhao, and Q. Yang, "Indoor localization in multi-floor environments with reduced effort," in Pervasive Computing and Communications (PerCom), 2010 IEEE International Conference on, 2010, pp. 244-252.

[8] A. Kushki, K. N. Plataniotis, and A. N. Venetsanopoulos, "Kernel-based positioning in wireless local area networks," *Mobile Computing, IEEE Transactions on,* vol. 6, no. 6, pp. 689-705, 2007.

[9] C. Laoudias, C. G. Panayiotou, and P. Kemppi, "On the RBF-based positioning using WLAN signal strength fingerprints," in *Positioning*

Navigation and Communication (WPNC), 2010 7th Workshop on, 2010, pp. 93-98.

[10] Y. Chen, Q. Yang, J. Yin, and X. Chai, "Power-efficient access-point selection for indoor location estimation," *Knowledge and Data Engineering, IEEE Transactions on,* vol. 18, no. 7, pp. 877-888, 2006.

[11] S. Drazin and M. Montag, "Decision Tree Analysis using Weka," Machine Learning-Project II, University of Miami, pp. 1-3, 2012.

[12] N. Marques, F. Meneses, and A. Moreira, "Combining similarity functions and majority rules for multi-building, multi-floor, WiFi positioning," in *2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Sydney, Australia 2012, pp. 1-9.

[13] GainSpan. (26/7/2013), Ultra low Power WiFi module Available: http://www.gainspan.com/

APPENDIX

The appendix lists the filtering and the positioning engine performance results of the experiments in more details.

TABLE IX.   APS FILTERING SCHEMES

Threshold Scheme	Distinct Values	% missing values	STDEV	Useful APs	Useless APs
A	< 5	> 95	<2	21	29
B	< 10	> 80	<5	16	34
C	< 15	> 90	<3	20	30

TABLE X.   J48 CLASSIFICATION RESULTS- APS FILTERING RESULTS

Dataset	A			B			C		
	Before any Filtering -50 APs	After APs Filtering - 21 APs	Gain	Before any Filtering -50 APs	After APs Filtering - 16 APs	Gain	Before any Filtering -50 APs	After APs Filtering - 20 APs	Gain
-85	68.8%	69.5%	0.7	68.8%	72.5%	3.7	68.8%	69.6%	0.8
-95	70.9%	72.6%	1.7	70.9%	70.3%	-0.6	70.9%	72.6%	1.7
-100	70.9%	72.6%	1.7	70.9%	70.3%	-0.6	70.9%	72.6%	1.7
-110	70.9%	72.6%	1.7	70.9%	70.3%	-0.6	70.9%	72.6%	1.7
NaN	66.3%	66.0%	0.3	66.3%	65.8%	-0.5	66.3%	66.0%	0.3

TABLE XI.   J48 CLASSIFICATION RESULTS WITH SCHEME B- IN DETAILS

Dataset	Before any Filtering -50 APs	After APs Filtering - 16 APs	Gain	After Fingerprints Filtering	# samples removed with 50APs	Gain	After Combined Filtering (APs First)	# samples removed with 16 APs	Gain	After Combined Filtering (Fingerprints First)	Gain
-85	68.8%	72.5%	3.7	70.1%	78	1.3	72.2%	87	3.4	71.5%	2.7
-95	70.9%	70.3%	-0.6	70.9%	73	0	70.4%	81	-0.5	70 %	-0.9
-100	70.9%	70.3%	-0.6	70.9%	73	0	70.4%	81	-0.5	70 %	-0.9
-110	70.9%	70.3%	-0.6	70.9%	73	0	70.4%	81	-0.5	70%	-0.9
NaN	66.3%	65.8 %	-0.5	66.1%	215	-0.2	66.0%	207	-0.3	66.1%	-0.2

TABLE XII.   POSITIONING ENGINE PERFORMANCE RESULTS USING THE THREE APS FILTERING SCHEMES

Filtering Scheme	A				B				C			
Dataset	-85	-95	-100	-110	-85	-95	-100	-110	-85	-95	-100	-110
%Correct Floor	98.4	98.2	98.2	97.6	98.5	98.4	98.2	96.6	98.5	98.2	98.0	97.6
%Correct Room	83.9	76.9	74.9	70.4	81.5	73.9	71.4	66.9	84.1	76.9	73.9	70.1
%Mean Error	2.80	3.43	3.72	4.06	3.14	3.81	4.07	4.40	2.84	3.43	3.76	4.06
Processing Time(ms)	4.82	4.93	4.94	4.94	4.34	4.3	4.32	4.34	4.67	4.64	4.65	4.55