MTAPS: Indoor Localization Algorithm Based on Multiple Times AP

Pengyu Huang[1], Haojie Zhao[1(✉)], Wei Liu[1], and Dingde Jiang[2]

[1] State Key Labs of ISN, Xidian University,
Xi'an 710071, Shaanxi, People's Republic of China
{pyhuang,liuweixd}@mail.xidian.edu.cn, hjzhao@stu.xidian.edu.cn
[2] School of Astronautics and Aeronautic,
University of Electronic Science and Technology of China,
Chengdu 611731, People's Republic of China
jiangdd@uestc.edu.cn

Abstract. In recent years, indoor localization base on fingerprint has become more and more common. In many fingerprint-based indoor positioning algorithms, it's very popular to use WiFi signal characteristics to represent the location fingerprint. However, with the great improvement of IEEE 802.11 protocols, WiFi has been broadly used. So there are numbers of WiFi access points (APs) have been deployed everywhere which can be used for localization purpose. The large amount of AP can greatly increase the dimension of the fingerprint and localization complexity. In this paper, we propose a novel indoor positioning algorithm MTAPS (indoor localization algorithm based on multiple times access point selection). MTAPS can effectively reduce the complexity of localization computation, and improve the performance of localization with an efficient access point selection algorithm. This indoor localization algorithm can get a better subset of APs through multiple times AP selection method. These selected APs will be more stable and can provide a better discriminative capability to reference locations. In addition, MTAPS uses k-means algorithm to cluster reference locations, and makes up a decision tree for every location cluster. After location clustering, MTAPS re-selects a suitable AP subset for every cluster. This method can further improve localization performance. Experimental results show that MTAPS has better localization performance than the indoor localization algorithm which is based on classical AP selection algorithm. And MTAPS can achieve the accuracy of over 90% within 2 m localization error.

Keywords: Location fingerprint · Multiple access point selections · K-means · Location clusters · Decision tree

Supported by the program of Key Industry Innovation Chain of Shaanxi Province, China (2017ZDCXL-GY-04-02), of the program of Xi'an Science and Technology Plan (201805029YD7CG13(5)), Shaanxi, China, of Key R&D Program – The Industry Project of Shaanxi (Grant No. 2018GY-017), of Key R&D Program – The Industry Project of Shaanxi (Grant No. 2017GY-191) and of Education Department of Shaanxi Province Natural Science Foundation, China (15JK1742).

H. Song et al. (Eds.): SIMUtools 2019, LNICST 295, pp. 13–24, 2019.
https://doi.org/10.1007/978-3-030-32216-8_2

1 Introduction

In recent years, location-based services (LBS) are more and more popular, and people have higher demands for localization and navigation. The GPS has high accuracy in open environment, but in indoor environment GPS signal can't pass through the wall. And GPS also experiences severe multipath effects, which seriously weaken the signal strength of GPS. These problems make GPS difficult to provide accurate indoor localization service. Recently, WiFi is becoming ubiquitous, and we can connect to WiFi in most common place, such as supermarkets, campuses or airports, In addition, our smart phones can easily connect WiFi and get WiFi signal RSSI (Received Signal Strength Indication). Therefore, there are a lot of indoor localization algorithms proposed based on WiFi, among which indoor localization algorithm based-fingerprint is popular.

Indoor localization algorithm based on fingerprint can be divided into two phases: offline phase and online phase. In the offline phase, the localization environment is divided into equal-sized grids, and the center of the each grid is used as a reference location. Then, collecting WiFi information in these reference locations, such as RSSI or CSI (Channel State Information). Using these WiFi information to represent fingerprint for each reference location, and make up location fingerprint database. In the online phase, fingerprint-based localization algorithm matches real-time localization data with all reference location's fingerprints in the fingerprint database. Choosing the reference location as target location, which has the highest similarity with real-time localization data.

Now WiFi is ubiquitous, when collecting WiFi data on offline phase, we can detect big numbers of WiFi access points in each reference location. If we directly use all detected access points to represent the reference location fingerprint. It means that each reference location fingerprint is a vector with big dimensions, and it also greatly increases the dimensions of the fingerprint database. Furthermore, the study in [1] found that when the number of the APs is large, the increase of AP will no longer results in any significant improvement of the location precision. Therefore, it's very important to select a suitable set of access points to represent the location fingerprint.

There are numerous AP selection algorithm proposed in recent years. Which could be divided into two main categories, as Highest Signal algorithm [2,3] and Information Gain-based AP selection [4]. In [2,3], the authors used the AP's RSSI to represent the importance of AP. The higher the signal strength, the more important the access point is. This AP selection algorithm is very easy, but WiFi RSSI changes frequently, and it is very sensitive in the indoor environment. So signal strength is not suitable to represent the importance of the AP. In [4], the author proposed an intelligent AP selection algorithm InfoGain (Information Gain-based AP selection). InfoGain algorithm uses position information entropy and conditional entropy to indicate the localization capabilities of different AP. In [5], AP selection was based on the principle of minimizing redundancy, using the correlation of APs to define redundancy. The correlation is got by computing two AP's divergence measure. Paper [6] proposed a real-time AP selection algorithm. Like [5], it was also focus on how to minimize redundancy

between APs. Paper [6] proposed two algorithms to get Ideal AP subset. In [7], the AP selection algorithm combines information gain with mutual information entropy, and uses mutual information entropy to express the similarity of APs. If the mutual information entropy of two APs is big, this paper just chooses the one with higher information gain. Paper [8] proposed RBF-based location algorithm, in which the covariance matrix of RSSI is used to select AP. This paper combines RSSI covariance matrix with weight matrix, and uses scaling parameters represents the importance of AP. Then, rank APs in terms of their scaling parameters, and pick out the APs with the highest scaling parameters to form an AP set.

The statistical distribution of RSSI of APs is always been required in the above AP selection algorithm [4–8]. Because they select AP or make up the decision model based on statistical distribution of RSSI of AP. However, in the process of collecting AP data, we often can find that, some APs only could be detect for a few time. This means for some APs, we only can get a few data of them, as show in Fig. 1. Figure 1 is a histogram to show the number of APs that could be detected for different times at a reference location in our experimental environment. We sampled fifty times at this reference location, and over one hundred APs were detected. In Fig. 1, the horizontal axis is the times of AP detected during the sampling period. And the vertical axis indicates the number of APs whose detection times is within a certain range. For example, in Fig. 1, the first column represents the number of APs, who are detected one to five times during the sampling period.

From Fig. 1, we could find that nearly half of the access points are collected less than five times, and more than 70% of the access points are collected less than twenty-five times during sampling period. If an access point appears too few times during the sampling period, we get few AP data of this AP, and using those small amounts of data cannot correctly describe AP's RSSI probability distribution. However, those algorithms [4–8] all depend on the AP's RSSI probability distribution to some extent, so those algorithm is not always useful.

In this paper, we propose a novel indoor localization algorithm MTAPS. This indoor localization algorithm is based on multiple access point selection method. MTAPS can effectively solves the above problems. And MTAPS can get a reliable APs subset by multiple access point selection method, which make the signal to be more stable and to obtain higher location accuracy. At the same time, MTAPS can effectively reduce complexity of localization computation, and improve the performance of localization in the meantime. In addition, MTAPS also uses k-means algorithm to cluster reference locations, and make up a decision tree for every location cluster. After clustering reference locations, MTAPS re-selects APs subset for every cluster, obtain a suitable APs subset for every cluster. This can further improve localization performance.

The rest of the paper is organized as follows: Sect. 2 describes MTAPS in detail. Section 3 is about the experimental results and the analysis of the results. Section 4 is a conclusion of MTAPS.

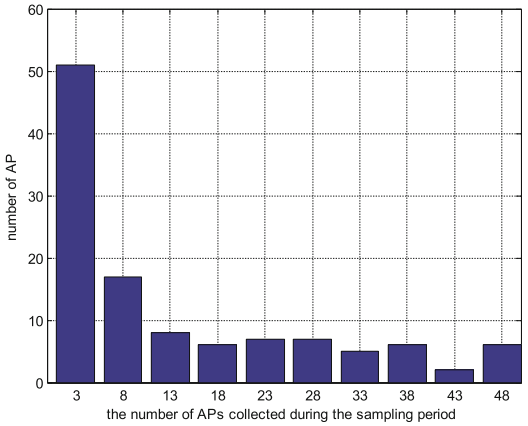

Fig. 1. A histogram of all APs detected at a reference location

2 Algorithm Description

MTAPS can be also divided into two phases: offline phase and online phase. In the offline phase, the MTAPS contains five steps, collecting AP data, AP selection, location clustering, AP re-selection and building decision tree.

2.1 Collecting AP Data

Collecting access point data is the basis of MTAPS. This step includes the following process: First, the localization environment is divided into equal-sized grids, and the center of the grid is used as a reference location. Then, collecting AP data for a period of time at each reference location.

Select Pre-selected AP Set. In this sub-step, we aim to delete access points that appear less frequently during the collecting process of AP data, and select a stable AP subset. The details of the process are as follows:

(1) Obtain the primary pre-selected AP subset.

Using $\overline{AP_i}$ represents AP set detected at reference location i,
$\overline{AP_i} = \left\{ AP_i^1, AP_i^2, AP_i^3, \ldots, AP_i^f \right\}$ AP_i^j denotes access point j detected at reference location i, and f is AP's number detected at reference location i. Calculating AP_i^j's number n_i^j collected at reference location i, so we can get AP's number set $\overline{n_i}$, $\overline{n_i} = \left\{ n_i^1, n_i^2, n_i^3, \ldots, n_i^f \right\}$. Such as AP_i^j is collected 30 times at location i during the data collecting period, so n_i^j is equal to 30.

Then, we set a threshold ***th*1**. If n_i^j less than ***th*1**, we remove AP_i^j from AP set of reference location i. Finally, we use the rest set of AP f to form primary

pre-selected AP subset $PRAP$. $PRAP = \{AP_1, AP_2, AP_3, \ldots, AP_g\}$, and g is the number of APs that satisfy the above-mentioned condition in localization environment.

(2) Obtain final pre-selected AP subset.

Calculating the sum times of AP_l at all reference locations, marked as N_l. Such as, suppose AP_l only appears at reference location i and j, the number of AP_l collected above 2 locations are num_i and num_j respectively. So N_l is equal to the sum of num_i and num_j. Therefore, we can get the set of corresponding values $\overline{N}$, $\overline{N} = \{N_1, N_2, N_3, \ldots, N_g\}$.

Then, we set another threshold ***th2***. If N_l is less than ***th2***, we delete AP_l from $PRAP$. Those remaining APs make up the final pre-selected AP subset $FPAP$, $FPAP = \{AP_1, AP_2, AP_3, \ldots, AP_h\}$, and h is the size of $FPAP$.

Obtain Final Target AP Set. In paper [4], the author proposed InfoGain algorithm to select AP set. InfoGain algorithm uses AP's information gain to represent the AP's discriminative power to location. The more discriminative power, the more important AP is. Such as, 's Information gain is calculated as follow:

$$InfoGain\,(AP_i) = H\,(G) - H\,(G|AP_i) \tag{1}$$

where $InfoGain\,(AP_i)$ is AP_i's discriminative power. $H\,(G)$ is the entropy of all reference locations without know AP_i's RSSI information.

$$H\,(G) = -\sum_{i=1}^{p} P\,(G_i) \log P\,(G_i) \tag{2}$$

where G_i is reference location, and p is the number of reference point. $H\,(G|AP_i)$ is the conditional entropy of location given AP's RSSI information.

$$H\,(G|AP_i) = -\sum_{v} \sum_{j=1}^{p} P\,(G_j, AP_i = \mathrm{v}) \log P\,(G_j|AP_i = \mathrm{v}) \tag{3}$$

where v is RSSI value of AP_i. $P\,(G_j, AP_i = \mathrm{v})$ and $P\,(G_j|AP_i = \mathrm{v})$ can be obtained based on collected AP data.

In this step, we calculate every AP's information gain of $FPAP$. Choosing the top k APs with the largest information gain to form fingerprint AP set $FingerAP$, $FingerAP = \{AP_1, AP_2, AP_3, \ldots, AP_k\}$.

Then, we get every reference location's fingerprint based on AP data in $FingerAP$, and obtain fingerprint database formed by all reference location's fingerprint. Set F_j is the location fingerprint of location j,

$F_j = \left\{RSSI_1^j, RSSI_2^j, RSSI_3^j, \ldots, RSSI_k^j\right\}$, where $RSSI_i^j$ is RSSI of AP_j in reference location i.

From formulas (2) and (3), we can know that we need to know the RSSI's probability distribution of every AP, when calculating $H\,(G|AP_i)$. However,

when collecting AP's data, we find some APs only are detected occasionally, so there is a few those AP's data, as Fig. 1. Therefore, for those access points detected occasionally, we cannot get reliable AP's RSSI probability distribution. Paper [1] does not consider this problem when the author proposes InfoGain algorithm. So only using information gain to select AP subset not always work well. In our algorithm, AP selection contains two step. The first step removes those APs, which are detected occasionally during AP data collecting period, and obtains pre-selected AP set. The second step gets final target AP set based on InfoGain algorithm. Those APs are stable in the pre-selected AP set, and can get their good probability distribution through their collected RSSI data. Therefore, our algorithm removes unstable APs. This method can better play the advantages of information gain algorithm.

2.2 Location Clustering

In the online phase, Indoor localization algorithm based on fingerprint match real-time position data with all fingerprints in the fingerprint database. Therefore, the elapsed time of real-time location is the linear relationship with the number of the reference locations in localization environment. If we divide all reference locations into some clusters, real-time positioning need just match with all fingerprints in a cluster that is most similar to it. So clustering location can effectively reduce the locating time. In our algorithm, we use a classical cluster algorithm, k-means algorithm [9], to cluster reference locations based on location fingerprint. Suppose there are M reference locations in location environment, and L clusters, and C_j is cluster center of cluster j, $C_j = \left\{c_1^j, c_2^j, c_3^j, \ldots, c_k^j\right\}$, where k is cardinality of $FingerAP$.

The process of location clustering as followed:

(1) Randomly selecting L locations as clusters'center $C_j = F_j$, so $C_j = F_j = \left\{RSSI_1^j, RSSI_2^j, RSSI_3^j, \ldots, RSSI_k^j\right\}$.

(2) Divided all reference locations into L clusters base on Euclidean distance between reference locations and all cluster centers. Such as, when deciding which cluster reference locations i belongs to, calculating Euclidean distance of reference location i to each cluster center. Then dividing reference location i to the cluster whose Euclidean distance to reference location i is minimal in L clusters. Euclidean distance is as followed:

$$Dis\left(F_i, C_j\right) = \sum_{h=1}^{k} \left(c_h^j - RSSI_h^i\right)^2 \tag{4}$$

where $Dis\left(F_i, C_j\right)$ is the distance of reference location i and cluster j.

(3) Updating the center of each cluster. When all reference locations are divided into clusters, calculating the average value of location fingerprints that the cluster contains as the new center of the cluster. Such as, cluster j has w

reference locations, so new center can be calculated as followed:

$$C_{jnew} = \frac{1}{w} \times \left\{ \sum_{h=1}^{w} RSSI_1^h, \sum_{h=1}^{w} RSSI_2^h, \ldots, \sum_{h=1}^{w} RSSI_k^h \right\} \tag{5}$$

(4) Determining whether to stop location cluster iteration, and updating each cluster's center with those average values. Calculating Euclidean distance between the new clusters' center and the old clusters' center. If the Euclidean distance less than a certain threshold, stopping iteration, and let C_j equal to C_{jnew}; else let C_j equal to C_{jnew} and back to step (2), continuing to iteration.

2.3 AP Reselection and Making up Decision Tree

After clustering, our algorithm re-select APs for each cluster again. Before clustering reference location, we select $FingerAP$ based on our AP selection algorithm. MTAPS aims to obtain a good AP set, which has high discriminative power. So $FingerAP$ is a reliable AP set for the indoor localization environment. However, our localization algorithm divides all reference locations into some clusters. Each cluster has its characteristics and there are some differences between clusters. Therefore, using a same AP set cannot well represent the characteristics and differences of each cluster. For this problem, our algorithm re-select AP set for each cluster. Therefore, through our re-select algorithm, one cluster's AP set can have some differences with the other cluster's AP set. The process of APs re-select in cluster as followed:

Suppose $Cluster_i$ is the set, which is made up by reference locations that divided into cluster i. So $Cluster_i = \left\{Loc_1^i, Loc_2^i, Loc_3^i, \ldots, Loc_K^i\right\}$, where Loc_j^i reference location j, and K is the number of reference locations in cluster i. According to those locations AP data collection in $Cluster_i$, calculating each AP's information gain in $FPAP$, and obtaining the re-selection AP set of cluster i, $ClusterAP_i = \left\{AP_1^i, AP_2^i, AP_3^i, \ldots, AP_c^i\right\}$, where c is the number of AP selected from $FPAP$.

After AP re-selection, we make up the decision tree for each cluster according to C4.5 algorithm [10] based on $ClusterAP$. The C4.5 algorithm is a classical algorithm for machine learning, and belongs to supervised learning algorithm. Using the C4.5 algorithm, we get the final decision tree of each cluster.

So let's do an example, as show in Fig. 2, there is a decision tree of a certain cluster. From Fig. 2, we can see that in this decision tree, the leaf nodes are the reference positions and the child nodes are APs from the $ClusterAP$. The range values on the decision tree's branches are the judgment condition to real-time data for location. For example, for the root node AP_4, it has three branches, and each branch has a value range. It has three branches, and each branch has a value range. If there is a test data, the value of AP_4 is 53, it satisfies the judgment condition on the second branch obviously. So it will go down to the second branch of the decision tree, and reach the next node AP_5.

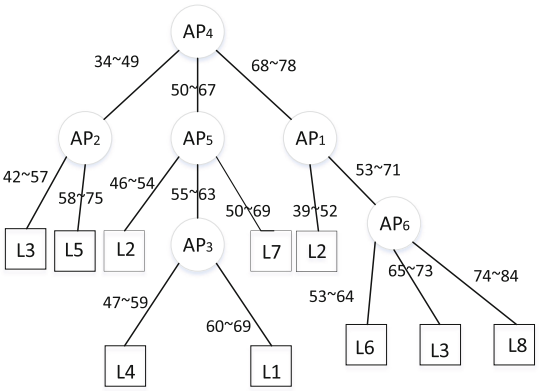

Fig. 2. Decision tree model

2.4 Online Phase

When the decision tree is established for each cluster, this means the offline phase has finished, and this localization algorithm can be used to identify the location of the user. The process of localization as followed:

Suppose the user's real-time localization data is T,

(1) Decide which cluster T belongs to. Getting localization fingerprint on the basis of $ClusterAP$, $Tfinger = \left\{\widetilde{AP_1}, \widetilde{AP_2}, \widetilde{AP_3}, \ldots, \widetilde{AP_c}\right\}$. Calculating Euclidean distance from T to each cluster, and choosing the cluster with the smallest Euclidean distance to T as T's target cluster. The location of T can be got from target cluster.

(2) Localization based on decision tree. In the previous sub-step, we obtain the target cluster that T belongs to. Then we use the target cluster's decision tree to determine T's precise location. Suppose the target cluster is cluster i, Table 1 is a user's localization data which is obtained based on from T. Decision tree of cluster $ClusterAP$ showed as Fig. 3.

From Fig. 3, we notice that the root node is AP_4. From Table 1, the value of AP_4 is 70 right between the range from 68 to 78. Obviously, next node is AP_1. From Table 1 the AP_1's value is 63 in the range from 53 to 71, so next node is AP_6, and so on. As the red line shows in the Fig. 3 the target location is L6.

Table 1. User's localization data

AP_1	AP_2	AP_3
63	56	45
AP_4	AP_5	AP_6
70	61	57

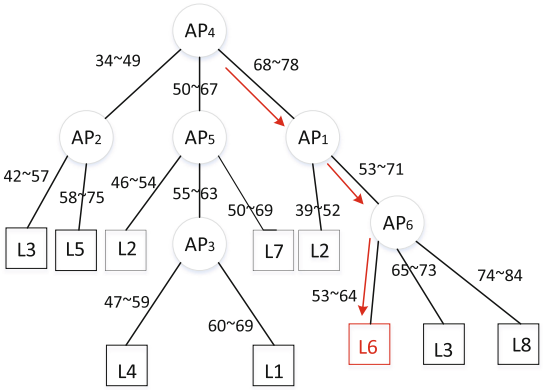

Fig. 3. The process of localization by using decision tree

3 Experimental Evaluation

In this section, we describe our experimental testbed, and assess the performance of the indoor localization algorithm based on multiple times access point selection.

Our experiment is carried out in the fourth-floor corridor of main building of Xidian University, as Fig. 4 showed. The experimental testbed include 177 reference locations, and every reference location is a 0.8 m * 0.8 m grid. In the phase of collecting AP data, we collect 50 samples at every reference location. Each sample lasts six seconds. And we can scan more than one hundred APs at every reference location in our localization environment.

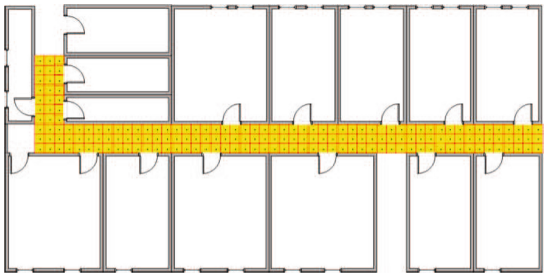

Fig. 4. Experimental testbed

Under the condition that the number of location clusters is 5, Fig. 5 indicates the localization accuracy of our localization algorithm changes with the number of APs within different positioning errors. From Fig. 5, we can see that our indoor localization algorithm has better localization accuracy. The algorithm

in this paper can achieve the best accuracy of over ninety percent within 2 m localization error. And within 1.6 m localization error, MTAPS also almost has the best accuracy of eighty percent.

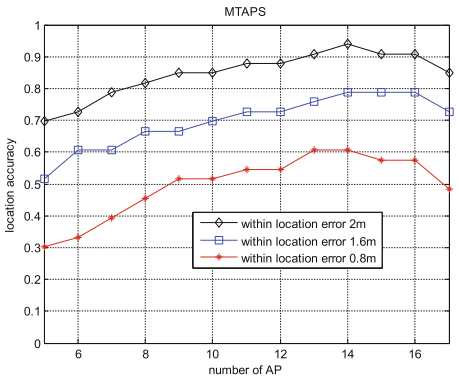

Fig. 5. The performance of MTAPS within different localization error under different AP number

Under the condition that the number of location clusters is 5, we compare MTAPS with InfoGain algorithm [1] within 2 m location error, as Fig. 6 shows. Figure 6 illustrates that the performance of our algorithm far exceeds that of the information gain algorithm. Therefore in the same condition, our algorithm always has better performance.

Under the condition that the number of location clusters is 5, we also did several experiments to observe the performance of InfoGain algorithm within different localization error. As Fig. 7 shows, the performance of InfoGain algorithm is very bad in our experimental environment. The main reason for this

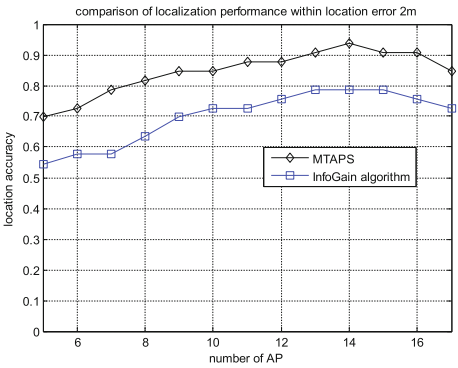

Fig. 6. Comparison of localization performance for MTAPS and InfaGain under different AP number

result is that in paper [1], their testbed is relatively simple, and a total of 25 access points can be detected in the environment. Duo to the number of APs is so small in their experimental environment, so there is less interference in their experimental environment. Therefore, information gain algorithm gets a good performance in their experiment.

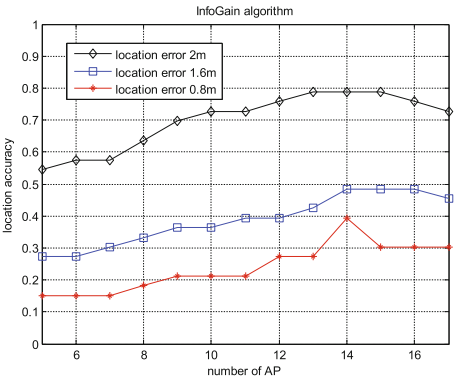

Fig. 7. The performance of InfoGain algorithm within different localization error under different AP number

However, today WiFi is everywhere, and we can detect hundreds of WiFi in my university or in a mall. Therefore, there is serious interference between APs, and at the same time the state of APs is more complex. Especially in the sampling process, some access points are merely detected. As a result, we can't correctly get those APs' probability distribution base on collected AP data under serious conditions of interference. But the performance of InfoGain algorithm is heavily dependent on probability distribution. So only using information gain algorithm to choose APs can't obtain satisfactory results. Our algorithm can remove unstable APs, and use reliable APs represent fingerprint. Therefore, our algorithm gets better performance.

4 Conclusion

In this paper, we propose an indoor localization algorithm MTAPS. This algorithm can get a reliable AP subset to represent fingerprint, and effectively remove unstable APs based on multiple access point selection method. In addition, our algorithm re-select AP subset for each location cluster, to get a special AP subset for location cluster. This localization algorithm divides all reference locations into some clusters by k-means algorithm, which can effectively decrease the cost-time of localization and improve efficiency. The results of the experiments show that our algorithm has the better performance. On the other side, we also analyze the causes of the bad performance for information gain algorithm. Our algorithm can effectively solve the defect of information gain algorithm, and obtains satisfactory positioning performance.

References

1. Kaemarungsi, K., Krishnamurthy, P.: Modeling of indoor positioning systems based on location fingerprinting. IEEE INFOCOM **2**, 1012–1022 (2004)
2. Youssef, M.A., Agrawala, A., Shankar, A.U.: WLAN location determination via clustering and probability distributions. In: IEEE International Conference on Pervasive Computing and Communications, pp. 143–150 (2003)
3. Du, L., Bai, Y., Chen, L.: Access point selection strategy for large-scale wireless local area networks. In: IEEE Wireless Communications and Networking Conference, pp. 2161–2166 (2007)
4. Chen, Y., Yang, Q., Yin, J., et al.: Power-efficient access-point selection for indoor location estimation. IEEE Trans. Knowl. Data Eng. **18**(7), 877–888 (2006)
5. Zhao, Q., et al.: An effective preprocessing scheme for WLAN-based fingerprint positioning systems. In: IEEE International Conference on Communication Technology, pp. 592–595 (2010)
6. Kushki, A., Plataniotis, K.N., Venetsanopoulos, A.N.: Kernel-based positioning in wireless local area networks. IEEE Trans. Mob. Comput. **6**(6), 689–705 (2007)
7. Zou, G., et al.: An indoor positioning algorithm using joint information entropy based on WLAN fingerprint. In: IEEE International Conference on Computing, pp. 1–6 (2014)
8. Laoudias, C., Panayiotou, C.G., Kemppi, P.: On the RBF-based positioning using WLAN signal strength fingerprints. In: 7th Workshop on Positioning, Navigation and Communication, Dresden, pp. 93–98 (2010)
9. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. Wiley, Hoboken (2012)
10. Quinlan, J.R.: C4.5: Programs for Machine Learning. Elsevier, Amsterdam (2014)