J. Shanghai Jiaotong Univ. (Sci.), 2015, 20(5): 548-557
DOI: 10.1007/s12204-015-1662-3

Indoor Localization with a Crowdsourcing Based Fingerprints Collecting

HUANG Zheng-yong[∗] (黄正勇), *YU Hui* (俞　晖), *GUAN Yun-feng* (管云峰), *CHEN Kun* (陈　坤)
(Institute of Wireless Communication Technology, Shanghai Jiaotong University, Shanghai 200240, China)

Abstract: Fingerprint matching is adopted by a large family of indoor localization schemes, where collecting fingerprints is inevitable but all consuming. While the increasingly popular crowdsourcing based approach provides an opportunity to relieve the burden of fingerprints collecting, a number of formidable challenges for such an approach have yet been studied. For instance, querying in a large fingerprints database for matching process takes a lot of time and calculation; fingerprints collected by crowdsourcing lacks of robustness because of heterogeneous devices problem. Those are important challenges which impede practical deployment of the fingerprint matching indoor localization system. In this study, targeting on effectively utilizing and mining large amount fingerprint data, enhancing the robustness of fingerprints under heterogeneous devices' collection and realizing the real time localization response, we propose a crowdsourcing based fingerprints collecting mechanism for indoor localization systems. With the proposed approach, massive raw fingerprints will be divided into small clusters while diverse devices' uploaded fingerprints will be merged for overcoming device heterogeneity, both of which will contribute to reduce response time. We also build a mobile cloud testbed to verify the proposed scheme. Comprehensive real world experiment results indicate that the scheme can provide comparable localization accuracy.
Key words: indoor localization, crowdsourcing, cluster, device diversity, fingerprint extraction
CLC number: TN 929.5 **Document code:** A

0 Introduction

A rapid proliferation of intelligent mobile devices such as smart phones and tablets stimulates applications that depend on determining users' indoor locations, such as route finding, photo and video geotagging, friend finding, targeted advertising, and suggesting local points of interests. Indoor localization scheme mainly leverages the wireless local area network access points to derive the location, and the proposed techniques can be categorized into the fingerprint-based[1-2] and the model-based scheme[3-4]. While the localization accuracy of the model-based systems is not admirable, the fingerprint-based schemes have been attracting much attention in recent years.

Fingerprints-based localization technique can be divided into the training and localizing phase. In the training phase, the fingerprints at positions of interests by measuring the received signal strength of wireless local area network (WLAN) access points (APs) around are collected into a database. In the localizing phase, the system will search for the user's current fingerprint in the fingerprints database and return the optimally matched location[5-6]. The fingerprints collection is generally accomplished by well-trained experts, which is expensive and laborious. This is because the jobs such as building map scanning and fingerprints database maintenance must be done with professional equipment. However, today's intelligent mobile devices such as smart phones and tablets embedded a rich set of sensors provide an opportunity to collect fingerprints with a crowdsourcing based approach, the basis of which is to utilize normal wandering users to collect fingerprints with their mobile devices[7-9].

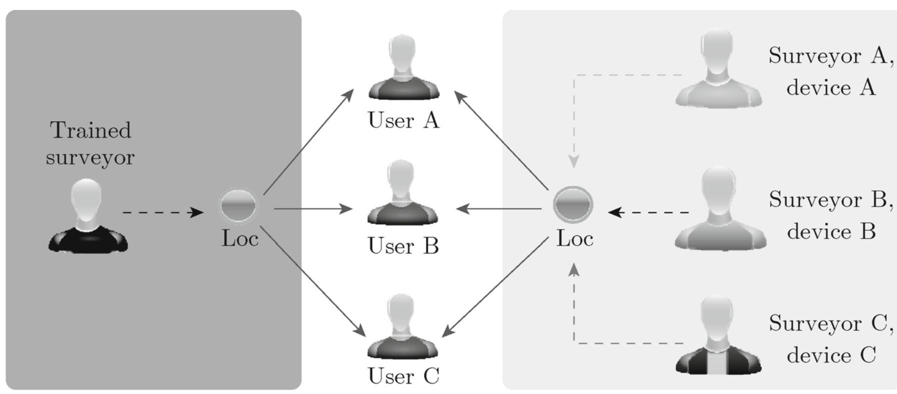
While the crowdsourcing based approach improves the efficiency of fingerprints collection, some formidable challenges need to be resolved. Firstly, the problem of massive raw fingerprint data caused by large amounts of normal users participating in fingerprints collection, which leads matching process during the localization phase to become time consuming. Secondly, the device heterogeneity problem, which is essential due to the normal users collect fingerprints using personal smart phones rather than professional equipment. To the best of knowledge, we are the first one to highlight and take response time into consideration of crowdsourcing

Received date: 2014-11-04
Foundation item: the National Science and Technology Major Project of China (No. 2013ZX03001007-004) and the Shanghai Basic Research Key Project (No. 11DZ1500206)
∗**E-mail:** zhengyonghuang@sjtu.edu.cn

indoor location system design, which in fact is extremely important for user experience of applications based on indoor positions. For the problems proposed and exist above, we introduce clustering method to reduce fingerprints matching calculation complexity and merging diverse devices' fingerprints to maintain only one fingerprint for one position, further we build an indoor localization testbed using CloudFoundry[10] to verify proposed fingerprints collecting mechanism and localization algorithm covering over 200 m^2 in one building of Shanghai Jiao Tong University. Extensive real-world experiments show that our indoor localization system with crowdsourcing based fingerprints collecting approach provides not only comparable location accuracy to previous approaches but also real time response.

1 System Model

In normal indoor location systems, the case is one surveyor-multiple users, in which one trained expert acting as the surveyor to construct the radio map and common users will exploit this constructed radio map to locate their positions. One significant shortage of this model is its high cost for employing and training expert surveyors, which leads this model hard to be deployed in large scale. However, crowdsourcing-based fingerprints collection approach takes advantage of multiple surveyors-multiple users architecture as shown in Fig. 1, in which common users will act as surveyors rather than the trained experts. The scenario shown in Fig. 1 hints us that the most obvious character distinct crowdsourcing model from normal model is that multiple common users will do the survey work with diverse devices. However after importing multiple surveyors and diverse devices, location accuracy should be guaranteed, which is the essential problem we are trying to give an answer in this paper.



(a) Non-crowdsourced model and one surveyor-multiple users

(b) Crowdsourced model and multiple surveyor-multiple users

Fig. 1 Comparison between crowdsourcing with non-crowdsourcing models

2 Fingerprints Extractions Based Crowdsourcing Samples

In crowdsourcing indoor location system, common users will act as surveyor roles instead of trained experts with diverse devices, which bring the problem of heterogeneous devices and massive raw fingerprint data. Base on such a fingerprint database constructed by massive raw data from heterogeneous devices, we need to maintain one representative fingerprint for one location and guarantee competitive location accuracy of the localization system. Hence we propose one fingerprints extraction method based on one kind of probability fingerprint, which we call it as Optimum Reception Fingerprint. This extraction method mainly depends on received signal strength indicator distribution estimation, which benefits from large sample data collected by crowdsourcing surveyors. Further we apply Kernel density estimation to work as received signal strength indicator distribution estimator, which will be proved fit for overcoming diverse devices problem in following sections.

2.1 Optimum Reception Fingerprint

Let's first view the data structure of upload sample received signal strength indicator series as Fig. 2(a) shown. In Fig. 2(a), for each location point, smart end device will detect a series of APs: $\{\mathrm{AP}_i, i = 1, 2, \cdots, M\}$. Further, for each detected AP, according to sample time span, we obtain sample received signal strength (RSS) value series: $\{\mathrm{AP}_i : \mathrm{RSS}_1, \mathrm{RSS}_2, \cdots, \mathrm{RSS}_n, \text{for } t_1, t_2, \cdots, t_n\}$, where $t_j\ (j = 1, 2, \cdots, n)$ is time. Based on this data structure, one common way[11] to obtain fingerprint value is to do the geometric mean calculation and use this mean value to fill fingerprint database. However, this mean value method compresses information provided by the sample received signal strength indicator series, which will further limits location accuracy. Hence we hope to extract more information from uploaded sample received signal strength indicator series and estimate which data point should be chosen to present as the fingerprint.

Given a set of fingerprinted locations and uploaded received signal strength indicator observation series, let F and L denote the random variable set of fingerprinted locations and correspondent real fingerprint values then denote O for the observation received signal strength indicator values. For location $l \in L$, its real fingerprint value is assumed to be $f \in F$. Once the sample received signal strength indicator series obtained, for each observation $o \in O$, we can estimate its probability of being the real fingerprint value f for location l, and transform this probability calculation according to Bayes' rules:

$$P_{F|O}(f|o) = \frac{P_{O|F}(o|f)P_F(f)}{P_O(o)} = \frac{P_{O|L}(o|l)P_L(l)}{P_O(o)}. \quad (1)$$

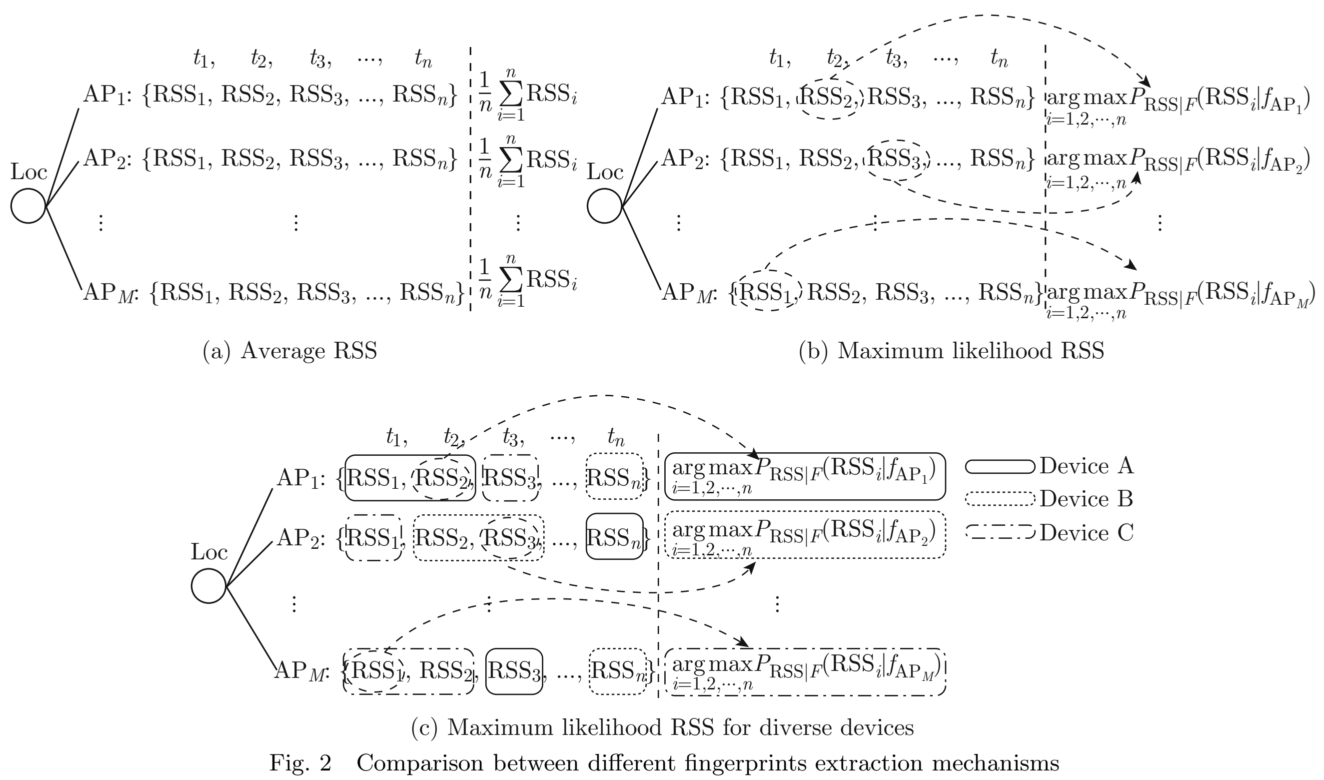
In Eq. (1), $P_{F|O}(f|o)$ can be replaced by $P_{O|L}(o|l)$ for that the real fingerprint value f presents radio information of location l. Note that observation likelihood $P_O(o)$ is fixed and can be ignored, while the prior probability $P_F(f)$ has uniform distribution, thus the problem to choose the maximum likelihood observation o to present f is to choose the maximum posteriori probability as follows:

$$f = \arg\max_{o\in O}\{P_{O|F}(o|f)\} = \arg\max_{o\in O}\{P_{O|L}(o|l)\}. \quad (2)$$

Equation (2) has the similar rule as optimum reception theory in wireless communication, thus we call this fingerprint as optimum reception fingerprint. This optimum reception fingerprint indicates that we should choose maximum likelihood received signal strength indicator value to present as fingerprint for location l, rather than just using mean calculation as Fig. 2(a) shown. In order to get optimum reception fingerprint, the key problem is to estimate received signal strength indicator probability distribution for any AP at any location. This will be discussed in the next section.

2.2 Reduction of Statistical Distribution Difference

In order to get optimum reception fingerprint which is assumed to be more closely with the real fingerprint value for each location, we should first estimate received signal strength indicator probability distribution mostly closing to reality, then choose the one who has the maximum probability. Once obtaining this estimated distribution, fingerprints extraction procedure can be shown as Fig. 2(b). The class-conditional probability of received signal strength indicator value presenting as fingerprint value can be estimated from uploaded training data in different ways, most previous work[12] treat this probability distribution as a Gaussian distribution. However, this estimation method is not appropriate in crowdsourcing-based models[13] for the reason of heterogeneous devices existing. Probability distribution of different devices is not only different at mean value and variance, but also distribution shapes. Mean value and variance can be transformed across diverse devices through linear transformation, but distribution shape cannot be transformed. So we adopt kernel



(a) Average RSS

(b) Maximum likelihood RSS

(c) Maximum likelihood RSS for diverse devices

Fig. 2 Comparison between different fingerprints extraction mechanisms density estimation[14] method to find this class-conditional probability distribution as

$$P_X^k(x) = \frac{1}{nh}\sum_{i=1}^{n} K\Big(\frac{x - x_i}{h}\Big), \tag{3}$$

where Kernel estimate function $K(\cdot)$ uses the standard Gaussian function, k is the index of kernel function, h represents kernel bandwidth, x and x_i are the RSS values.

In Fig. 3 we show three different types of devices' received signal strength indicator probability distribution using Gaussian density estimation and Kernel density estimation respectively, where P is the probability density. Figure 3 clearly classifies a fact that using Kernel density estimation will eliminate distribution shape difference across diverse devices, which hints us to use Kernel density estimation method to estimate received signal strength indicator probability distribution in crowdsourcing model and further based on this estimated probability distribution to extract fingerprint value across diverse devices. Table 1 lists the details of three different types of devices' parameters, and these devices will be used to verify our proposed algorithms' performance in the real-time testbed in Section 4.

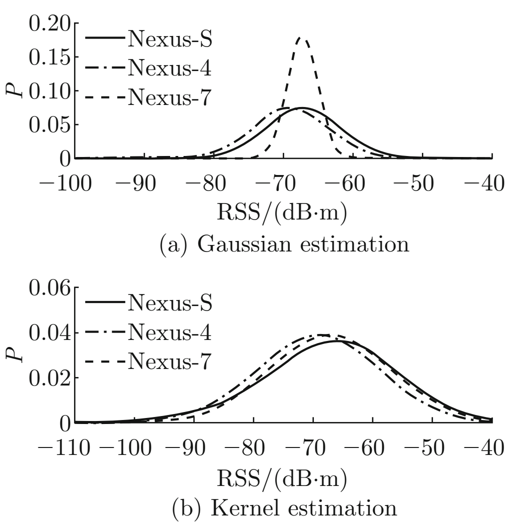


Fig. 3 Comparison between Gaussian and Kernel density estimation

Table 1 Different types of devices' parameters

Device	Wifi chipset	Operation system	Manufacturer
Nexus-S	Broadcom BCM4329	Android 4.1.2	Samsung
Nexus-4	Murata SS2908001	Android 4.2.2	LG
Nexus-7	Broadcom BCM4751	Android 4.2.0	Asus

2.3 Extraction Algorithm Adapting for Crowdsourcing Model

Further, let's consider crowdsourcing indoor location model in fingerprint extraction process again. As different users will upload received signal strength indicator sample series using diverse devices, and different devices' RSS probability distributions will achieve to be consistent by applying with Kernel density estimation, therefore we can merge different uploaded received signal strength indicator values from heterogeneous devices and perform Kernel density estimation based on merged statistic samples. Finally we can get optimum reception fingerprint for crowdsourcing model cross diverse devices. Figure 2(c) explains fingerprints extraction algorithm for crowdsourcing model, in which different rectangles present RSS values from different types of devices.

3 Localization Algorithm

3.1 Clustering Using Affinity Propagation

Since there are massive fingerprints in the database, searching for a matched fingerprint is time-consuming and lead to a slow response. Meanwhile, indoor propagation channel varies over time, however this variance is continuous within a small geographic region[15]. Affinity propagation[16] is exploited here to do clustering process after surveyors upload fingerprints into the database. Affinity propagation algorithm regards all data points as a full-connection network and passes messages between each data point, then it will find out each cluster's exemplar and calculate out which cluster each data point belongs to. It takes an input measures of similarity between pairs of fingerprints: $s(i, j)$, which indicates how well fingerprint j is suitable to act as the exemplar of fingerprint i. Euclidean distance is used to calculate this similarity as:

$$\begin{aligned} s(i,j) = -\sum_{t\in\Theta}\|\psi(i,t)-\psi(j,t)\|^2 - \\ \sum_{t\in\Lambda}\|\psi(i,t)+120\|^2 - \\ \sum_{t\in\Delta}\|-120-\psi(j,t)\|^2, \end{aligned} \tag{4}$$

where $\psi(i, t)$ presents fingerprint value i for AP_i; Θ is the set for AP indexes that exist both in fingerprint i and fingerprint j; Λ is the set for AP indexes that exist only in fingerprint i but not in fingerprint j; Δ is for the set that AP indexes exist in fingerprint j but not in fingerprint i.

In affinity propagation iteration process, two messages are passed: responsibility message $r(i, j)$ and availability message $a(i, j)$. $r(i, j)$ sending from fingerprint i to fingerprint j reflects the accumulated evidence for how well-suited fingerprint j is to serve as the exemplar for fingerprint i, taking into account of other potential exemplars j' for fingerprint i as:

$$r(i,j) = s(i,j) - \max_{j'\neq j}\{a(i,j') + s(i,j')\}, \tag{5}$$

where $a(i, j)$ sending from candidate exemplar fingerprint j to fingerprint i reflects the accumulated evidence for how appropriate it would be for fingerprint i to choose fingerprint j as its exemplar.

Taking into account the support from other fingerprints that fingerprint j should be an exemplar, as:

$$a(i, j) = \min\{0, r(j, j) + \sum_{i' \neq i, j} \max\{0, r(i', j)\}\}. \quad (6)$$

After recursively passing responsibilities and availabilities between each data point pairs, fingerprint database will be divided into small clusters and also we can obtain each cluster's exemplar. Figure 4 shows clustering results of Dongzhong Building in Shanghai Jiao Tong University, from which we can conclude that different rooms (Room101-1, Room101-2, Room101-3) and corridor are clustered into different regions, which keeps consistent with geographical distribution. Thus, we confine localization problem into small regions, eliminating large errors and reducing response time by matching small datasets of a few clusters instead the whole database.

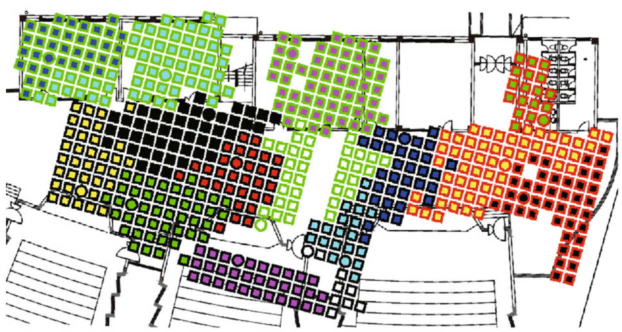


Fig. 4 Fingerprints cluster results in Dongzhong Building of Shanghai Jiao Tong University

3.2 Most Matched Cluster k Nearest Neighbor (MMC-KNN) Algorithm

After clustering results obtained, further we need to find a set of most matched clusters, Ref. [15] utilized a method based on similarities between observation point with each cluster's exemplar according to Eq. (4) and find such clusters whose exemplar has greater similarity with observation than a pre-defined threshold as:

$$S = \{j : s(o, j) > \delta, j \in H\}, \quad (7)$$

where H is the exemplar indexes set of each cluster and δ is the similarity threshold.

However, in Eq. (7), cluster matching mechanism use a fixed constant threshold to get most matched clusters, which will not always find out the most matched clusters in practical experiments. Thus considering uploaded observation's connections and similarities with all exemplars, we propose a novel cluster matching algorithm using affinity propagation again. We regard observation point as a new data point added into fingerprints' self-organization data network, and then affinity propagation principles should be performed again to find out which cluster this observation data point belongs to. Thus a fixed value threshold is abandoned, as Fig. 5 shown. The responsibility $r(i, j)$ passed in cluster iterations as Eq. (5) shown reflects the probability for how well-suited fingerprint j is to serve as the exemplar for fingerprint i, taking into account the probability of other potential exemplars j' for i. Thus for the new observation o, through enough iterations, a set of its responsibility for each exemplar will be obtained: $\{o : r(o, e(1)), r(o, e(2)), \cdots, r(o, e(n))\}$, e is the exemplar node. Just sorting this responsibility vector descending and we can get M most matched clusters.

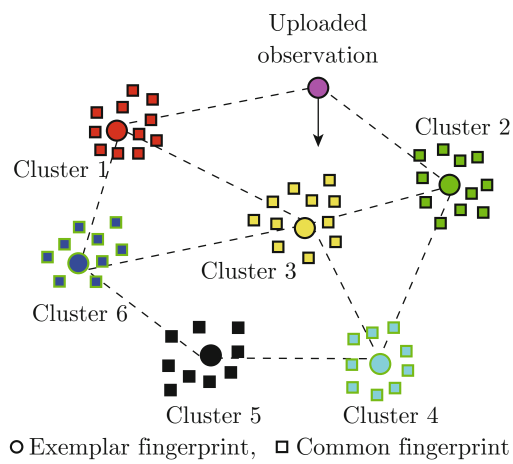


Fig. 5 Illustration of proposed algorithm for uploaded observation's cluster matching

In order to distinguish the most matched cluster from the second one and the followings, fingerprints belonging to different matched clusters are endowed with a weight factor as:

$$w(f) = \frac{e^c}{D(f, o)}, \quad (8)$$

where $c = M + 1 - n$, M is the number of matched clusters, n is the index for the cluster in the sorted responsibility set; $D(f, o)$ is the Euclidean distance for observation o with fingerprint f.

The M most matched clusters and all fingerprints endowed with weight factor construct the dataset for observation RSS value to match:

$$C = \bigcup_{j=1}^{M} C_j, \quad (9)$$

where C_j is the jth fingerprint.

Then the location algorithm k nearest neighbor is used as:

$$(x, y) = \frac{1}{k} \sum_{i=1}^{k} (x_i, y_i), \quad (10)$$

where x_i, y_i belongs to the k fingerprints who has nearest distance with observation.

3.3 Restrict Large Error Using Grid Window Filter

Using MMC-KNN algorithm will find k nearest fingerprints within M most matched clusters, while in fact some nearest fingerprints may be far away from observation's real position, such as in corridor environment where clustering region may cover a long and narrow region. In this case, using such "far away" matched fingerprints to estimate the position will decrease location algorithm's performance. Hence we propose to use a grid window filter to restrict matched fingerprints into a small bursting region. As each fingerprint's weight is defined in Eq. (8), we use an $a \times a$ window to scan through fingerprints, and calculate each window's sum of fingerprint weight factor as:

$$W = \sum_{i=1}^{\alpha} \frac{\mathrm{e}^{C_i}}{D_i}, \tag{11}$$

where α is the number presenting how many matched fingerprints within current window's region, D_i is the Eucli-distance of observation and fingerpring.

Then we can obtain a window which has the maximum sum weight and filter the matched fingerprints out. Finally by applying KNN with these filtered matched fingerprints instead of all k nearest fingerprints, we can further improve location accuracy, which will be verified in our real-time testbed.

4 System Architecture and Real-Time Experimental Testbed

This section gives a detailed description of our proposed system architecture. Experiments' results and analysis based on a real-time testbed are also presented in this section.

4.1 Proposed System Architecture

Facing the formidable challenges for crowdsourcing model, this paper aims at proposing a system-level solution covering all. As shown by the diagrams in Fig. 6, crowdsourcing indoor location system's architecture can be divided into two phases: crowdsourcing-based fingerprints collection phase and localization phase.

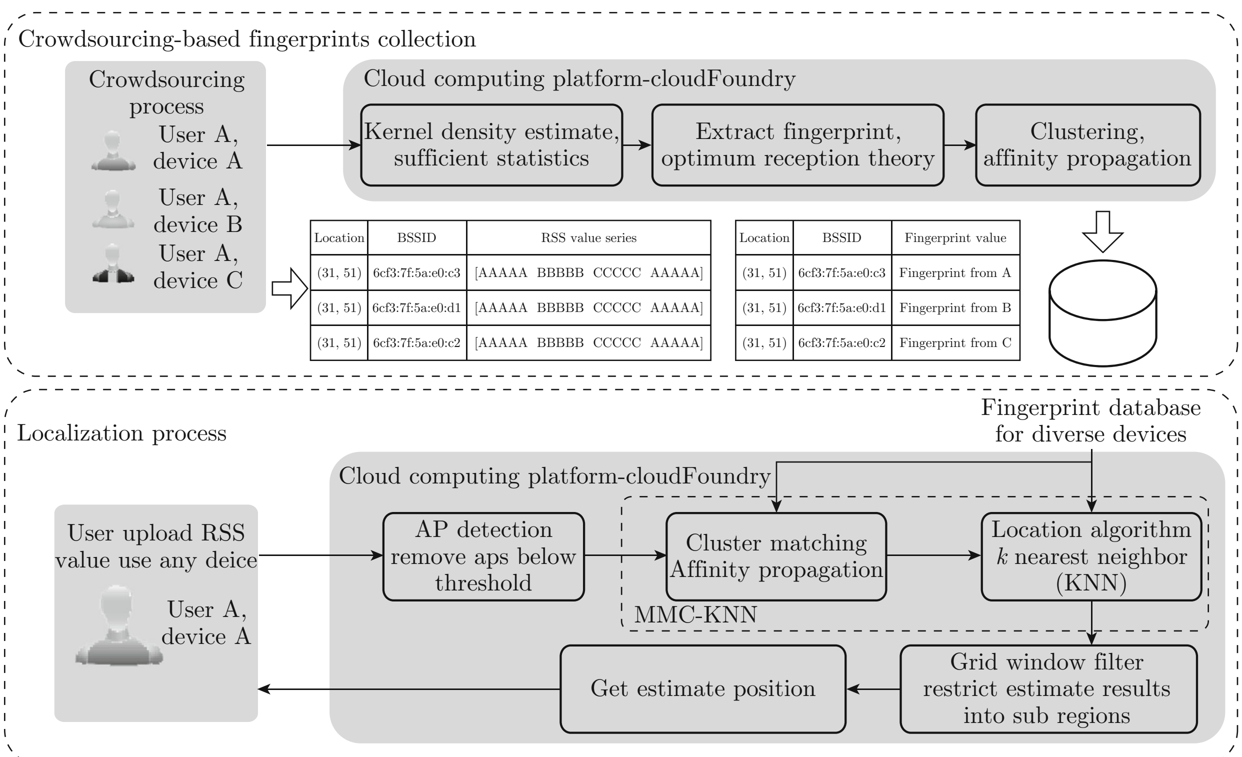


Fig. 6 Block diagram of the proposed crowdsourcing indoor localization system

In fingerprints collection process, multiple surveyors will contribute to construct the fingerprint database. For each AP at each location, diverse devices' uploaded received signal strength indicator series will be used to get universal probability distribution estimation by kernel density estimator based on sufficient statistic theory. Then a novel fingerprint extraction method across diverse devices based on optimum reception theory will be executed. Once obtaining fingerprints for each AP at each location and getting the fully-built radio map, an unsupervised cluster algorithm based on affinity propagation is performed to cut all fingerprints into small datasets with returning each cluster's exemplar data point. The clustering algorithm is introduced for the purpose of reducing large amounts of raw fingerprints data and degrading response time, while at the same time maintaining competitive location accuracy for crowdsourcing based fingerprints collection.

In location phase, any normal user will upload sample received signal strength indicator series by any device. Then our proposed localization algorithm MMC-KNN: k nearest neighbors combined with M matched clusters is performed. Before executing this localization algorithm, an AP detection module will be applied to filter those APs, whose received signal strength indicator values are rather low and has less significance for localization. M most matched clusters here refers to find M most matched cluster within fingerprints database according to similarity degree between the observation and each cluster's exemplars. Finally a grid window filter is introduced to restrict estimate results into subregions to avoid raising large errors.

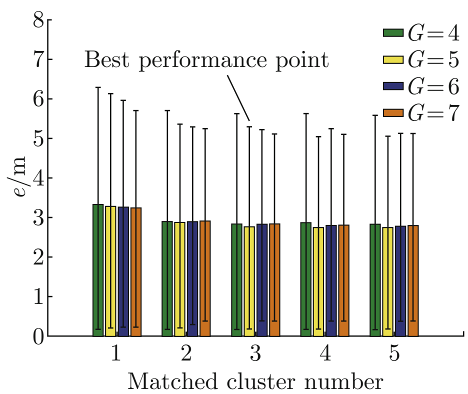
All procedures of fingerprints collection and localization phase are implemented in a cloud computing platform — CloudFoundry with the purpose of simulating a test environment close to reality and commercial products, experimental results will be shown as follows.

4.2 Experiment Setup

We set our test environment at three teaching rooms and one corridor in Dongzhong Building of Shanghai Jiao Tong University covering over 200 m^2 which is complex indoor environments with lots of obstacles and rather abundant visitors flow, whose ichnography is shown in Fig. 5. For this test region, we collect about 550 position's fingerprints and 220 positions' test observations using about 20 smart devices of three different devices. All fingerprints are collected during 6 different time periods at two different days: 8:00—11:30 am, 2:00—5:30 pm and 20:00—23:30 pm, while the two days have an interval of one week. Each time period's collection work is done by different volunteers who are not trained experts but common persons only taught with how to use our Android application to collect fingerprints. We are trying to build a test environment close to real crowdsourcing indoor location model, therefore experimental results are sufficient to verify proposed algorithms' performance.

4.3 Evaluation for Localization Accuracy

In this part of evaluation, for each kind of device, we construct the fingerprint database use the same type of device. During the whole evaluation process, we get the statistic result of each test observation's error distance from the real position. Figure 7 shows average error distance and error distance's range of fluctuation under different matched cluster number and grid window size for Nexus-S, where e represents the location error and G is the grid window size. From Fig. 7, we are clearly classified: increasing matched cluster number will improve location performance. However when matched cluster number is 3 and increasing it further will not significantly decrease the average error distance, instead this will add calculation complexity and extend response time. Considering location accuracy and calculation complexity, the best parameter for matched cluster number should be 3. And for parameter of grid window size, it can be easily reached that increasing grid window size will weaken location performance as greater window size means greater location region. Hence as experimental results shown in Fig. 7, the best parameter for grid window size should be 5. Figure 8



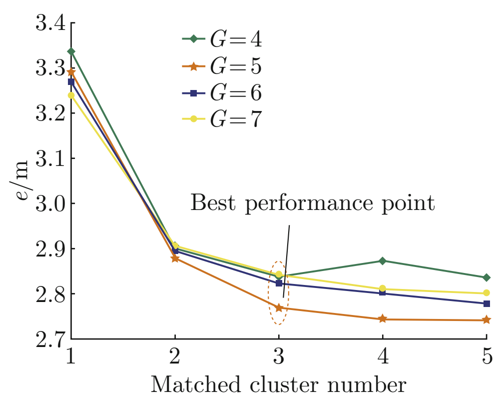


Fig. 7 Average error distance with different matched cluster number and grid window size for Nexus-S

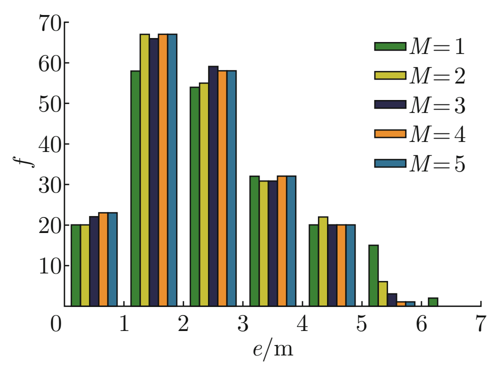


Fig. 8 220 observation's error distance statistic with best performance parameters for Nexus-S shows error distance distribution of 220 observations for Nexus-S under the best performance parameter, where f is frequency number.

Based on the optimum parameters of matched cluster number and grid window size, we further compare different location algorithm's performance: MMC-KNN together with grid window filter, MMC-KNN only and common KNN, as shown in Fig. 9. From Fig. 9 we can conclude that our proposed location algorithm MMC-KNN has a better performance than common KNN while grid window filter will further reduce mean error distance by getting rid of large error distance.

4.4 Evaluation for Response Time

Further we verify our MMC-KNN algorithm's performance on reducing response time. MMC-KNN algorithm divided large fingerprint database into small datasets, which will limit matching scope in localization phase. Thus only fingerprints belong to matched cluster will be taken into calculation during real-time localization request happen, while clustering process will always be run in background no matter localization request raise or not. Assume $M = 3$, and all cluster number is 10 in our experiments, and then calculation quantity will be brought down to 30%. Effects of MMC-KNN on reducing response time are illustrated by the response time column of Table 2, which also provides more performance details of each location algorithm for Nexus-S. Compared with common KNN, MMC-KNN increases response time about 75% reduced from 10.2 s to 2.49 s. Though MMC-KNN with grid window will increase response time about 0.5 s, it restricts more errors within 2 m, thus this response time's degradation is acceptable. Figure 9 also presents location error distance CDF for Nexus-4 and Nexus-7 to verify MMC-KNN algorithm performs well across diverse devices. However, Nexus-4 has a mean location error distance of 2.9 m while Nexus-7's mean location error distance is 2.31 m. Nexus-7 has a better location accuracy as it is a kind of tablet with larger appearance, which leads it to have a more stable received signal strength indicator probability distribution.

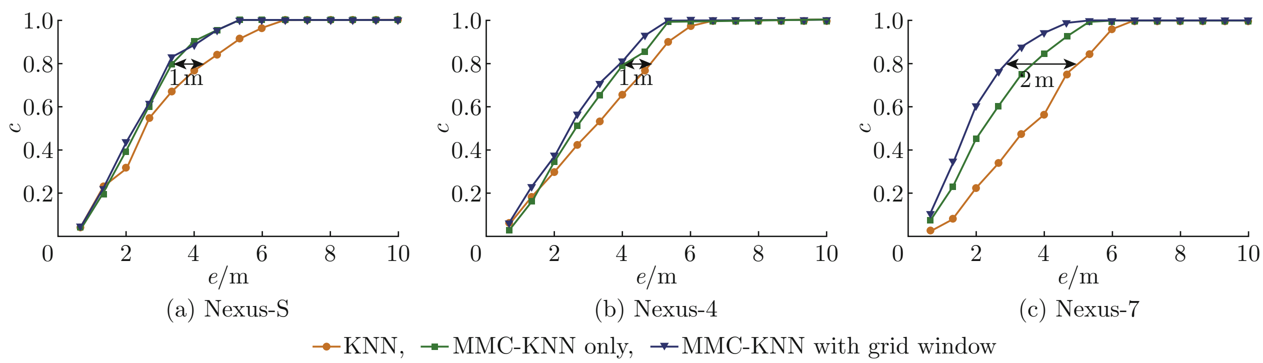


Fig. 9 CDF of location error distance for different algorithms

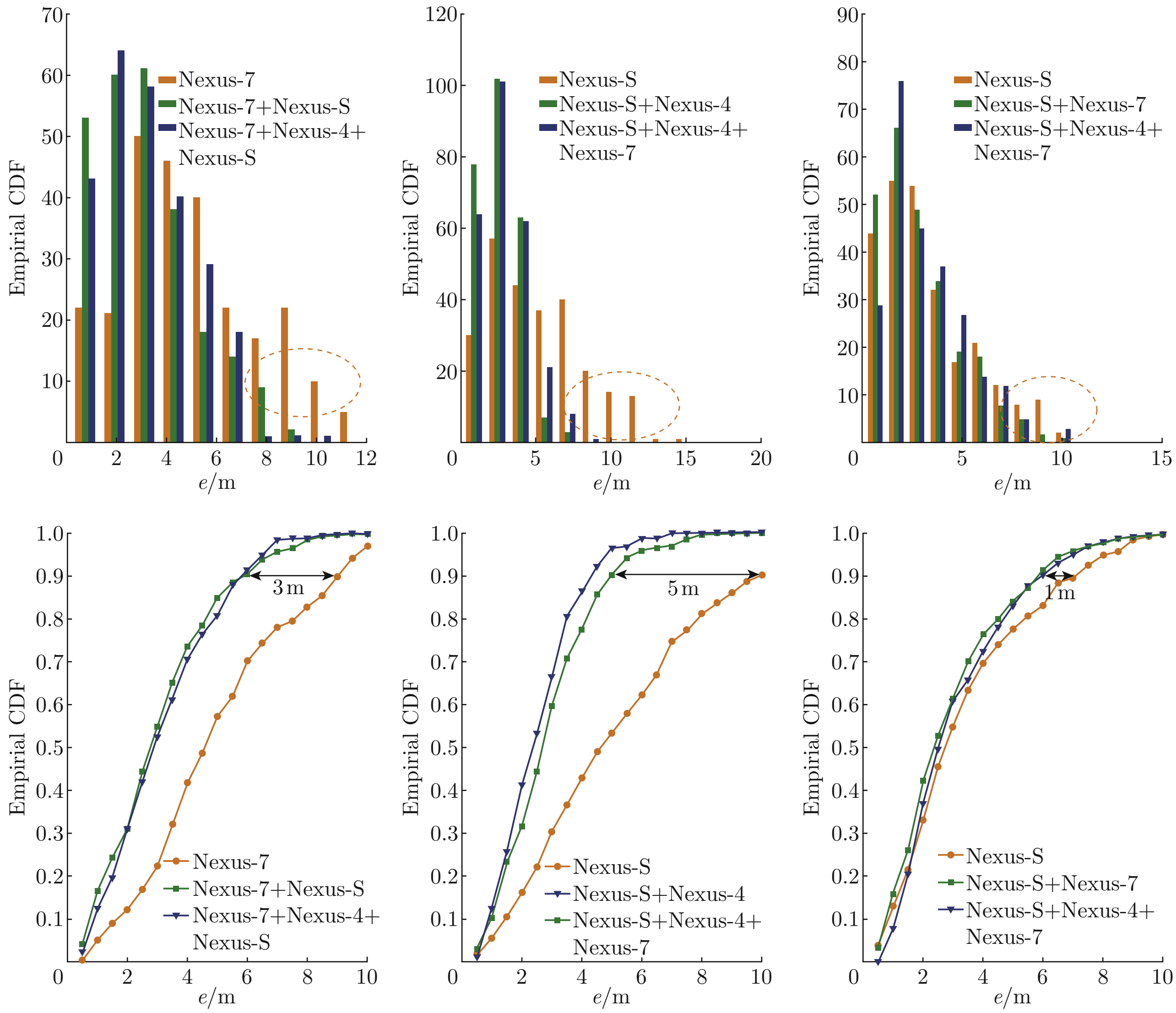
Table 2 Performance and response time comparison between different algorithms

Algorithm	Maximum error/m	Mean error/m	Localization error/%		Response time/s
			Error < 5 m	Error < 2 m	
MMC-KNN with Grid Window	5.091 4	2.343 6	98.51	43.78	3.26
MMC-KNN only	5.192 8	2.408 8	98.01	39.30	2.49
KNN	6.311 6	2.794 5	85.07	31.84	10.20

4.5 Evaluation for Solving Heterogeneous Devices Problem

In this part of evaluation, we address to identify our proposed algorithms for heterogeneous devices problem in crowdsourcing indoor location model. In Section 2, we propose to employ optimum reception fingerprint and merge received signal strength indicator series from diverse devices using Kernel density estimator rather than the Gaussian estimator. For each AP at every location, we only maintain one fingerprint value though new fingerprints will continually be uploaded into database. Thus this evaluation is to illustrate that our proposed fingerprint extraction algorithm works well and guarantee relative location accuracy cross heterogeneous devices for crowdsourcing indoor location model.

For three different types of devices: Nexus-S, Nexus-4 and Nexus-7, we first construct fingerprint database using another type of device different from localization device type. For example, when locating device is Nexus-S, then the fingerprint database will be Nexus-4 or Nexus-7. We construct a merged fingerprint database using two devices and further three devices to simulate a real Crowdsourcing environment, as shown in Fig. 10.



(a) Nexus-S (b) Nexus-4 (c) Nexus-7

Fig. 10 Comparison of different types devices' location performance under diverse fingerprint database

Figure 10 shows different types of devices' location performance under diverse fingerprint databases, more details of average error distance are listed in Table 3. From Fig. 10 and Table 3, it can be clearly concluded that using only one significantly different device to construct fingerprint database will decrease localization performance, while merging fingerprints of the same type device with another different type will improve localization accuracy: 3 m gain for Nexus-S, 5 m gain for Nexus-4 and 1 m gain for Nexus-7 if judging by 90% error distance point. Finally when the fingerprint database is constructed by merging all the devices, performance is not obviously weaken as the CDF curves almost overlap. Thus our proposed fingerprint extraction and merging algorithm's performance across heterogeneous devices can be verified.

Table 3 Average error distance with diverse fingerprint databases

Device	One different device error/m	One different and one same devices error/m	Three devices error/m
Nexus-S	4.983 3	3.054 7	3.161 0
Nexus-4	5.189 2	2.492 5	2.861 3
Nexus-7	3.344 2	2.892 3	3.101 0

5 Conclusion

This paper focuses on solving the challenges in crowdsourcing indoor location model which is potential to be used as a basic infrastructure in future indoor location based services (LBSs): the large raw fingerprint data, device diversity, response time and location accuracy. We propose a novel fingerprint extraction mechanism: optimum reception fingerprint based on Kernel Density Estimation to overcome heterogeneous devices problem and maintain only one fingerprint value for each AP at each location. Further, based on affinity propagation theory, we propose MMC-KNN algorithm with grid window filter to improve location accuracy and solve the problem of searching large scale raw fingerprint data, contributing to reduce response time. We test our proposed algorithms in a real-time testbed built with cloud computing platform-CloudFoundry, for the purpose of simulating a crowdsourcing indoor location system which is close to reality and commercial product, therefore the proposed algorithms' performance has been verified by real-world experiments. As experimental results shown, three different types of devices all achieve to limit mean error distance below 3 m, and location performance will not degrade when fingerprint database is merged by diverse types of devices as crowdsourcing indoor location model does. To be summarized, our proposed fingerprint extraction and localization algorithms overcome heterogeneous devices problem while achieving a competitive location accuracy and significantly reducing response time in crowdsourcing indoor location model.

References

[1] Kuo S P, Tseng Y C. A scrambling method for fingerprint positioning based on temporal diversity and spatial de endency [J]. *Knowledgeand Data Engineering*, 2008, **20**(5): 678-684.

[2] Jin Y, Soh W S, Wong W C. Indoor localization with channel impulse response based fingerprint and nonparametric regression [J]. *Wireless Communications*, 2010, **9**(3): 1120-1127.

[3] Xiang Z, Song S, Chen J, et al. A wireless lan-based indoor positioning technology [J]. *IBM Journal of Research and Development*, 2004, **48**(5/6): 617-626.

[4] Kao K F, Liao I E, Lyu J S. An indoor location-based service using access points as signal strength data collectors [C]//*2010 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. Banff, Canada: IEEE, 2010: 1-6.

[5] Haeberlen A, Flannery E, Ladd A M, et al. Practical robust localization over large-scale 802. 11 wireless networks [C]//*Proceedings of the 10th Annual International Conference on Mobile Computing and Networking*. PA, USA: ACM, 2004: 70-84.

[6] Krumm J, Horvitz E. Locadio: Inferring motion and location from wifi signal strengths [C]//*2th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*. Boston, USA: IEEE, 2004: 4-13.

[7] Bhasker E S, Brown S W, Griswold W G. Employing user feedback for fast, accurate, low-maintenance geolocationing [C]//*Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications*. Orlando, USA: IEEE, 2004: 111-120.

[8] Bolliger P. Redpin-adaptive, zero-configuration indoor localization through user collaboration [C]//*Proceedings of the First ACM International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments*. San Francisco, USA: ACM, 2008: 55-60.

[9] Park J G, Charrow B, Curtis D, et al. Growing an organic indoor location system [C]//*Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*. San Francisco, USA: ACM, 2010: 271-284.

[10] Dudin E, Smetanin Y G. A review of cloud computing [J]. *Scientific and Technical Information Processing*, 2011, **38**(4): 280-284.

[11] Zaruba G V, Huber M, Kamangar F, et al. Indoor location tracking using RSSI readings from a single wifi access point [J]. *Wireless Networks*, 2007, **13**(2): 221-235.

[12] Seco F, Plagemann C, Jimenez A R, et al. Improving RFID-based indoor positioning accuracy using Gaussian processes [C]//*2010 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. Banff, Canada: IEEE, 2010: 7-15.

[13] Park J G, Curtis D, Teller S, et al. Implications of device diversity for organic localization [C]//*The 29th Conference on Computer Communications*. Shanghai, China: IEEE, 2011: 3182-3190.

[14] Botev Z, Grotowski J, Kroese D. Kernel density estimation via diffusion [J]. *The Annals of Statistics*, 2010, **38**(5): 2916-2957.

[15] Feng C, Au W S A, Valaee S, et al. Compressive sensing based positioning using RSS of WLAN access points [C]//*The 29th Conference on Computer Communications*. San Diego, USA: IEEE, 2010: 1-9.

[16] Frey B J, Dueck D. Clustering by passing messages between data points [J]. *Science*, 2007, **315**(5814): 972-976.