A Device-Clustering Algorithm for Device Heterogeneity in Crowdsourcing-based Localization

Huang Cheng[1], Haiyong Luo[2], Fang Zhao[1]

1 Software School, Beijing University of Posts and Telecommunications, Beijing 100876, China
2 Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100876, China
chenghuang1990@gmail.com, yhluo@ict.ac.cn, zhaofanglhy@yahoo.com.cn

Abstract

Device heterogeneity significantly degrades the localization performance of fingerprinting-based localization, especially in the crowdsourcing-based positioning system. Although manual calibration can reduce positional error, the adjustment overhead is extremely heavy and to maintain ever-increasing device types is overly laborious. In this paper, we propose a novel Device-Clustering algorithm to operate the positioning system based on macro Device-Cluster (DC) rather than natural device. In this way, the system maintains less device types and the localization accuracy is improved obviously. The experimental result of different combination indicates the optimal operating flow is to combine DC and kernel density estimator when the tracking device is known and add the linear transformation phase when device is unknown.

Keywords: crowdsourcing; fingerprinting-based positioning; clustering-based algorithm; device heterogeneity

1 Introduction

The importance of space location has made positioning system an integral part of modern mobile applications. User's location can enhance a variety of LBS applications, including calendars, reminders, navigation assistants, and communication tools [1]. Moreover, with the remarkable advancement of assorted need in wide-area environment, metropolitan-scale location-based services (LBS) have found their ways into people's daily life.

However, conventional satellite-based positioning, such as the GPS, fails in both indoor scenes and complex urban environment. Compared with other Radio Frequency (RF) techniques, Wi-Fi-based localization benefits from the widespread deployment of APs and stable signal strength. Therefore, fingerprint-based Wi-Fi Location Systems has become a research hotspot in the world for its high accuracy and low cost of computation [2,3].

Fingerprint-based localization is simple apply. During the calibration phase, system keeps a radio map containing a location flag and a set of scans, or a fingerprint collected by training device. And during the tracking phase, to obtain the corresponding location flag, tracking device will find the closest match in the fingerprint database with its observing fingerprint [4]. Nevertheless, the accuracy of fingerprinting-based localization greatly depends on the fingerprint density and sampling coverage. Thus to build an abounded fingerprint database, traditional expert sampling is time-consuming, expensive and intrusive [5]. As an alternative, a Wikipedia-style crowdsourcing model [6] has been introduced to encourage users to contribute fingerprints unnoticed when tagging places.

Although crowdsourcing model could effectively reduce the sampling overhead, a new issue comes into being -- Device Heterogeneity. General user's involvement brings assorted WLAN-enabled mobile devices, which results in diverse Received Signal Strength (RSS). Hence when training device is different from the tracking device in positioning phase, such varying RSS can remarkably degrade positional accuracy. Besides, this device heterogeneity problem is not only caused by distinct Wi-Fi chipsets, but also related with distinctive driver properties, various operating system, complicated encapsulation materials and so forth [7]. Therefore, to maintain different Wi-Fi training and tracking devices is overly laborious and impractical for real-world deployment, especially in calibration phase before positioning.

This paper proposes a novel Device-Clustering algorithm. First in the training phase, distinct devices sharing the same RSS pattern are classed into same macro Device-Cluster (DC). Then in the tracking phase, all the operations are based on the DC. The known devices can directly exploit the fingerprints from corresponding DCs, while unknown devices requires being linearly transferred into available RSS pattern. Finally through user's feedback, new tracking fingerprint can be absorbed into database to augment radio map. Experiments prove the validity and practicality of this new positioning framework.

2 Related work

There is much research work done in solving device heterogeneity while employing crowdsourcing model. And these researches usually fall into three main categories.

The first strategy is to add a brief calibration period before location estimation. Haeberlen et al. has acknowledged the pairwise linear transformation between diverse devices gives good results in most cases [8]. Afterward, experiments carry out the major characteristics of device heterogeneity lie not only in the linear difference, but also in the signal strength distributions [5]. And then it's proved linear transformation works well only with uniformed combination of hardware and software [9]. Thus linear calibration alone does not solve the problem and assorted devices make the manual calibration unpractical.

Instead of extra adjustment, the second method is a calibration-free localization. M. B. Kjaergaard presents hyperbolic location fingerprinting, which replaces the original fingerprint with the ratio of signal strength between pairs of APs [10,11]. Then in [12], the signal strength differences between pairs of APs are introduced to reduce the overhead of adjustment. Both the two methods are based on the hypothesis that although devices are different, their reflections to different APs should be the same comparing with themselves. But such hypothesis usually fails in complex urban environment because of multipath effect and noises.

The third approach is based on the transfer learning algorithm. According to [13], the problem of mapping RSS signal patterns over devices could be treated as multiple learning tasks. By transferring fingerprint values into a latent feature space, the RSS pattern tends to be uniformed so that the device diversity could be solved. Likewise, in [14] a corresponding relationship is learnt from both tracking devices and training devices in a low-dimensional space with Manifold Alignment. Then the relationship is used to transfer knowledge from train domain to help with the classification in tracking domain. However, both of these solutions are limited in theoretical research presently, and the complexity of algorithm obstructs their promotion and deployment.

Although clustering-based solution has been exploited in solving the spatial diversity problem [15], its application in device diversity is still not developed. Given ample device types offered by crowdsourcing-based training and laborious sampling in total space to each device, in this paper, we plan to replace single device with macro DC. By comparing their similarity in fingerprints, we combine the strength of both hierarchical clustering and density-based clustering to class distinct devices into DC, so that devices in same cluster could share fingerprints. And the mix of linear transformation and Expectation Maximization (EM) is executed to unknown tracking devices to offer approximate positioning result.

3 Algorithm

Our algorithm framework is shown in Figure 1. This flow can be divided into three parts.

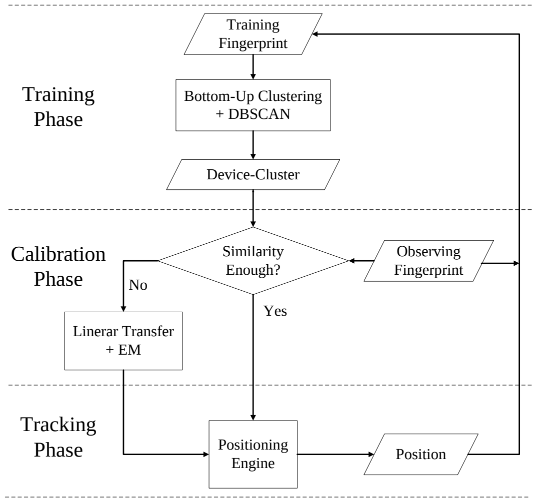

Fig. 1. Algorithm Framework

In the training phase system maintains a DC database rather than single device. Whenever an observe fingerprint is input in the calibration phase, system will decide whether this device belongs to one of the DCs. If the tacking device is totally new, EM algorithm will be applied to linearly transfer the observe fingerprint. Finally in the tracking phase, the estimated position will be worked out with certain positioning algorithm.

3.1 Training Phase

A. Fingerprint similarity measurement

When RSS fingerprints of a device are collected, they not only provide information at these points, but also imply the characters of specific type of devices. Thus by comparing these fingerprints with some measurements, similarities between pairwise devices can be obtained.

Here we choose the Pearson Correlation Coefficient [16] to measure similarities in fingerprints between pair devices.

$$r(\mathbf{p},\mathbf{q}) = \frac{\sum_{i=1}^{d} (\mathbf{p}_i - \bar{\mathbf{p}})(\mathbf{q}_i - \bar{\mathbf{q}})}{\sqrt{\sum_{i=1}^{d} (\mathbf{p}_i - \bar{\mathbf{p}})^2 - \sum_{i=1}^{d} (\mathbf{q}_i - \bar{\mathbf{q}})^2}} \quad (1)$$

$\mathbf{p}$, $\mathbf{q}$ represent fingerprints of two devices in radio map, and d is their dimension, or the number of APs. The Pearson correlation ratio r represents the linear dependency between two fingerprint vectors. r ranges from 0 to 1, where 0 indicates the least similarity while 1 indicates the greatest similarity.

B. Device similarity measurement

After defining the fingerprint similarity measurement between fingerprints, the device similarity measurement is also deduced.

Here we exploit the DBSCAN algorithm [17]. DBSCAN is a density-based clustering algorithm with two parameters: ε (radius of cluster) and $minPts$ (minimum points of cluster). It starts with an arbitrary core point, and then absorbs all the neighbor points within distance ε as the member of the cluster based on the distance measure. When the number of neighbors reaches the minimum requirement -- $minPts$, a cluster is formed.

Assuming $\mathbf{F} \subseteq R^d$ is a RSS fingerprint database, a **dist** matrix can be calculated as follows, in which fingerprint similarity measurement is treated as the distance measure.

$$\mathbf{dist}(\mathbf{F}) = \begin{pmatrix} r_{11} & \cdots & r_{1n} \\ \vdots & \ddots & \vdots \\ r_{n1} & \cdots & r_{nn} \end{pmatrix} \tag{2}$$

n is the total number of the fingerprint in $\mathbf{F}$. Then we choose the fingerprint $\mathbf{p}$, who owns the greatest similarity values, as the starting point. Thus in the d-dimensional hypersphere, we obtain an initial the ε-neighborhood N with the radius ε, minimum points $minPts$ and core point $\mathbf{p}$.

$$N_{\varepsilon}(\mathbf{p}) = \{\mathbf{q} \in \mathbf{F}, \varepsilon \in \mathbf{dist} \mid \mathbf{dist}(\mathbf{p},\mathbf{q}) \le \varepsilon\} \\ s.t. \left|N_{\varepsilon}(\vec{\mathbf{p}})\right| \ge minPts \tag{3}$$

In the ε-neighborhood N, except the core point $\mathbf{p}$, any fingerprint $\mathbf{q}$ is density-reachable to $\mathbf{p}$ under the condition $(\varepsilon, minPts)$.

Therefore, if $\mathbf{F}_i$, $\mathbf{F}_j$ are RSS fingerprint databases of two distinct devices, by mixing the two databases we can get the $(|\mathbf{F}_i| + |\mathbf{F}_j|)$ -dimensional **dist** matrix.

$$\begin{cases} \varepsilon = \alpha(\mu(|\mathbf{F}_i|) + \mu(|\mathbf{F}_j|)) \\ minPts = \beta(|\mathbf{F}_i| + |\mathbf{F}_j|) \end{cases} \tag{4}$$

$|\mathbf{F}_i|$, $|\mathbf{F}_j|$ are number of fingerprints in $\mathbf{F}_i$ and $\mathbf{F}_j$. α defines the radius of new cluster, while β is the slack variable in the range of $0 \sim 1$. The greater the β is, the harder $\mathbf{F}_i$ and $\mathbf{F}_j$ could be classed as one cluster.

C. Hierarchical clustering framework

To effectively class diverse mobile devices, we introduce the hierarchical clustering framework [18] which is shown in Figure 2.

In this framework, every natural device will be compared with the rest $n-1$ devices under the device similarity measurement, and the successful mixed cluster will enter the next level. After k-level's calculation, n devices will be classified into m DCs.

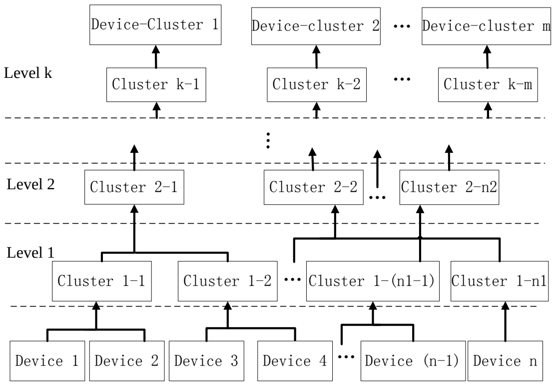

Fig. 2. Bottom-Up Clustering

3.2 Calibration Phase

In the calibration phase, whenever a new observing fingerprint enters, system will decide whether the tracking device has already been trained. If the tracking device is known, observing fingerprint will simply advance into positioning phase. But if the device is totally unknown, a calibration is necessary.

If $\mathbf{F}_{train}$ and $\mathbf{F}_{track}$ are fingerprint databases each from training devices and unknown tracking devices, then there should be a pair coefficient (a,b) which satisfies $\mathbf{F}_{train} = a \times \mathbf{F}_{track} + b$ [5,7,9]. In this paper, Expectation Maximization algorithm (EM) is employed to refine the linear parameter over and over by repeatedly computing expectation and maximization, until the fluctuation of linear transfer convergent [8]. In this way, the fingerprints pattern from tracking devices is transformed to training fingerprint pattern.

In the parameter-learning process, f^0 indicates a DC p 's training fingerprint database, $\Theta = (a,b)$ denotes the linear coefficients and f_{qt} represents the tracking device q 's RSS readings at time t. EM starts with an initial guess of the parameters (a_0,b_0) and an initial observing fingerprint f_{q0}. Then it seeks to find the optimal linear coefficient by iteratively applying the following two steps:

E-step: Treat Θ_t as a constant, then work out the $f_{q(t+1)}$ with transformation function $T(\Theta_t, f_{qt})$ until the expected log-likelihood of the following formula is greatest:

$$SIM = \{dist(f_{qt}, f_p) \mid f_p \in f^0\} \tag{5}$$

M-step: Treat $f_{q(t+1)}$ as a constant, find Θ_{t+1} to satisfy the greatest $E(\Theta_t, f_{qt})$.

The optimization problem is as follows:

$$\underset{\Theta}{argmax}\, E(\Theta_t, f_{qt}) = \underset{\Theta}{argmax} \left\{ \frac{E(\Theta_{t+1}, f_{q(t+1)})}{E(\Theta_t, f_{qt})} \right\} \tag{6}$$

When the fluctuation of linear transformation function stays within a fixed range, for example 0~1, we deem the function convergent and the final linear coefficients (a, b) could be used to map observing fingerprints.

3.3 Tracking Phase

Tracking phase is the final positioning phase, in which the transformed observing fingerprint and corresponding DC serve as the input to the positioning algorithm. Here we introduce the Bayesian localization framework [19], which is very suitable for crowdsourcing-based localization for its simplicity and high accuracy.

Bayesian localization method computes the posterior probabilities over locations to find the most possible position. Given an observing fingerprint o and a random location $l_k \in L^n$ where L indicates a set of n locations, the posterior probability that o belongs to l is shown by the Bayes' rule:

$$P(l_k \mid o) = \frac{P(o \mid l_k) P(l_k)}{\sum_{i=1}^{n} P(o \mid l_i) P(l_i)} \tag{7}$$

$P(l_k \mid o)$ indicates the prior possibility of the occur of observing fingerprint. $P(l_k)$ is usually set as the uniform distribution so in practical use people often ignore it. Therefore the estimated location l_e is the one obtaining the maximum value of the posterior probability.

$$l_e = \arg\max_{l_k} P(o \mid l_k) \tag{8}$$

Supposing each $o = (v_1, v_2, ..., v_m)$ and M is the index of m APs, then the $P(o \mid l_k)$ in Equation 8 becomes:

$$P(o \mid l_k) = \prod_{i \in M} P_{v_i|L}(v_i \mid l_k) \tag{9}$$

The $P_{v_i|L}$ here is often modeled as some kind of distribution. In this paper we choose two kinds of distributions: Gaussian distribution (maximum-likelihood parameter estimator) [20] and a kernel density distribution (Parzen window estimator) [21].

If in location l_k, there are totally n training fingerprints in one DC and each fingerprint scans m APs, we denote $\mathbf{T}_i = (s_1, s_2, .., s_n)$ as the RSS set of all the fingerprint value in AP i.

If $\mathbf{T}_i$ is treated as Gaussian distribution, then the probability function of v_i is as follows:

$$P_O(v_i \mid l_k) = \frac{1}{\sqrt{2\pi\delta}} e^{-\frac{(vi-u)^2}{2\delta^2}} \tag{10}$$

where μ and δ are mean value and variance of $\mathbf{T}_i$.

Likewise, if $\mathbf{T}_i$ is treated as a kernel density distribution, then the probability density function is as follows:

$$P_O(v_i \mid l_k) = \frac{1}{nk} \sum_{j=1}^{n} K\left(\frac{v_i - s_j}{h}\right) \tag{11}$$

where h is the kernel width, and $K(\square)$ is a kernel function. In this paper, we choose the Gaussian kernel as the kernel function.

4 Experiment

4.1 System Setup

In order to evaluate the performance of our Device-Clustering algorithm, we deploy a simulation environment on the 7th floor of Institute of Computing Technology (ICT) in the Chinese Academy of Sciences. The sampling and positioning area is shown in Figure 3. This area covers 14 stations as is shown in figure and the red points shown in Figure 3 represent the sampling locations.

We employ eight types of mobile devices, which are shown in Table 1. We carry each cellphone to these red points for 2 minutes to collect RSS data and corresponding location flags. Finally we choose 100 samples for every mobile device in each grid.

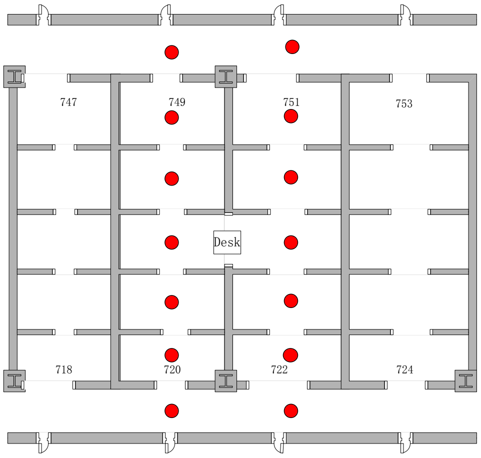

Fig. 3. Sampling area on 7th floor of ICT

Table 1 Devices used for data collection

Device	Platform	Wi-Fi Module	OS
1	HTC Magic G2	Texas Instruments WL1251B	Android 1.5
2	HTC G11	Broadcom BCM4329	Android 2.3.3
3	HTC G14	AVAGO ACPM-7868	Android 2.3.4
4	Samsung Galaxy S	Samsung SWB-B23	Android 4.0
5	Samsung Nexus S	Broadcom BCM4330	Android 4.0
6	HTC Desire z	Broadcom BCM4329	Android 2.3.7
7	M1 MIUI	AzureWave AW-NH6 11	Android 2.3.5
8	Huawei U8860	Unknown	Android 4.0

4.2 Performance Evaluation

A. Clustering Measurement Analysis

As mentioned above, argument ε and *minPts* decide the device similarity measurement, but in practical use of DBSCAN people care much more of the radius of new cluster rather than minimum points in cluster, because the concept of reachability density helps find the best *minPts* in operation. Thus to find the optimal clustering measure, we compare the trend of sampling cost and localization error as the argument ε 's factor α changes. The situation is shown in Figure 4.

Here the sampling cost is calculated with the formula: $T = lmn$, where l denotes sampling locations, m indicates the corresponding number of DC when α changes and n represents the sampling number every device needs in each location. The performances of different clustering result are denoted as error distance, which is calculated by the Euclidean distance between a predicted location and true location.

It's clear from Figure 4 that as α becomes greater, localization error rises correspondingly while the sampling cost decreases. When $\alpha = 0.1$, each device is self-styled to be a cluster, so that the sampling overhead every device assumes is overly laborious despite the best accuracy. And when $\alpha = 1$, it means all the fingerprints are mixed together to become a cluster, which not solves the device heterogeneity problem.

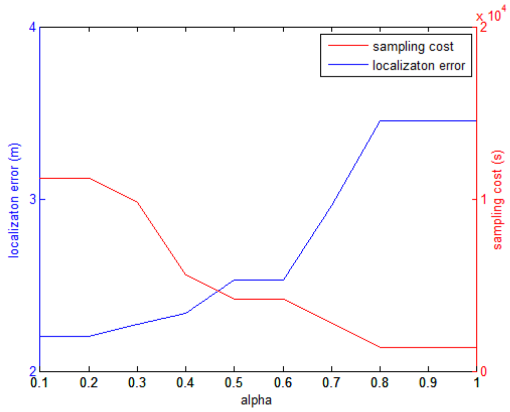

Fig. 4. Localization error vs. clustering number

Therefore, the key issue is to find the best balance between sampling cost and localization error. And in our situation, the joint point in Figure 4 shows the optimal α where all the devices are divided into four categories and the corresponding similarities between pairwise devices are shown in Table 2. The clustering result is listed in Table 3.

Table 2 Similarities between devices

Device	1	2	3	4	5	6	7	8
1	1	0.82	0.82	0.56	0.57	0.80	0.84	0.69
2	0.82	1	0.78	0.55	0.57	0.82	0.81	0.72
3	0.82	0.78	1	0.53	0.57	0.80	0.84	0.70
4	0.56	0.55	0.53	1	0.33	0.51	0.72	0.50
5	0.57	0.57	0.57	0.33	1	0.47	0.69	0.39
6	0.80	0.82	0.80	0.51	0.47	1	0.76	0.48
7	0.84	0.81	0.84	0.72	0.69	0.76	1	0.69
8	0.69	0.72	0.70	0.50	0.39	0.48	0.69	1

Table 3 Clustering result

Device	1	2	3	4	5	6	7	8
DC	1	1	1	1	2	1	3	4

B. Phase Combination Analysis

To test the performance of our Device-Clustering algorithm over diverse devices, we divide the tracking devices into two categories: known devices and unknown devices. We also design different algorithm combination by mixing different phases, to verify the effect of clustering-based method, linear transformation and Kernel density-estimation. Table 4 shows the abbreviation of different phases.

Table 4 Terms of each phase

Abbreviation	Description
C	Clustering-based method on DC
M	Based on original mixed database
K	Kernel density estimation
G	Gaussian estimation
L	Linear transformation

Figure 5 represents the localization error of different combination. In the figure we find the kernel estimation based on DCs represents the best accuracy when the tacking device is already known to the system, while to the unknown device the linear transformation method combing the kernel estimation and DC shows least localization error, which verify the validity and high degree of accuracy of our Device-Clustering algorithm.

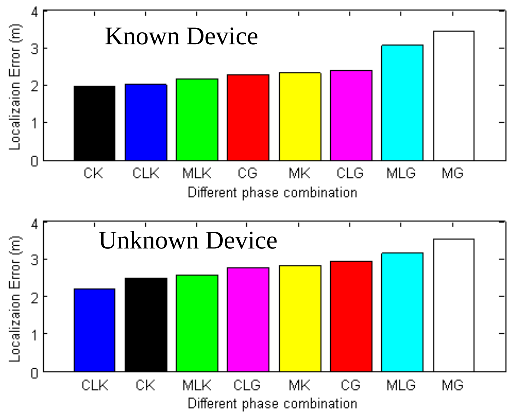

Fig. 5. Localization error of different combination

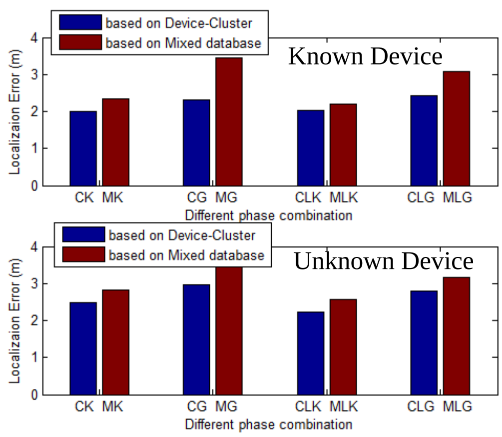

Fig. 6. Device-Cluster vs. Mixed database

Figure 6 indicates the comparison between flows with and without clustering phase. We can find the clustering-based method increase the positioning accuracy to both known tracking devices and unknown tracking devices.

The same result can also be seen in the Figure 7. No matter the tacking devices are known or unknown, the localization errors happen in flow with kernel density estimator always represent better accuracy.

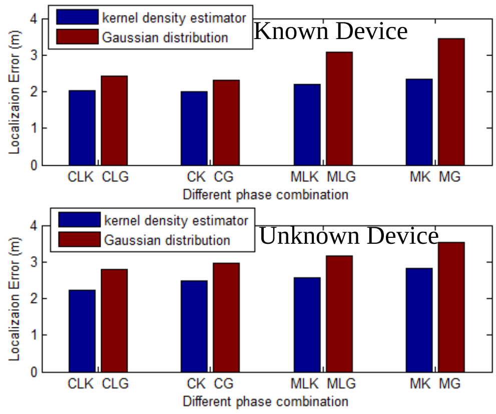

Fig. 7. Kernel density vs. Gaussian estimation

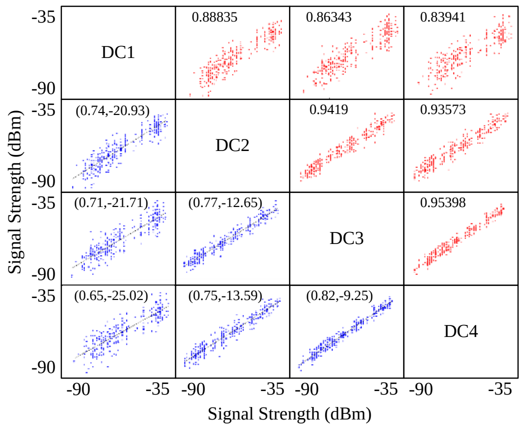

Fig. 8. Linear relations between Device-Clusters

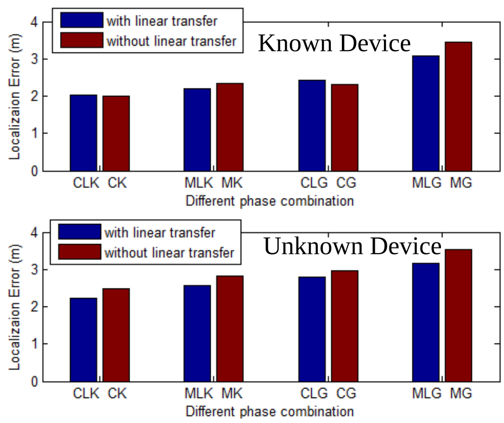

Fig. 9. With transfer vs. without transfer

In Figure 8, we represent the linear relations between each pairwise device. The lower part of Figure 8 shows the linear coefficient and the top half denotes the Pearson Correlation Coefficients. This Figure exhibits the linear relationship is unchanging after single devices are classed into several DCs.

However, the effect of linear transformation is different to known and unknown devices as shown in Figure 9. If the devices are known, CLK and CLG are not better than the combination only with CK and CG. But when it comes to the unknown devices, linear transformation phase proves an advantage in diminishing localization errors.

C. Computational Cost Analysis

In Bayesian positioning framework, one of the key issues to reduce localization error is to increase the sampling number. However, as the fingerprints per device samples increases, the cost of computation also rises. To better analyze our Device-Clustering algorithm, the change of computational cost is drawn in Figure 10 and Figure 11 as sampling scale grows.

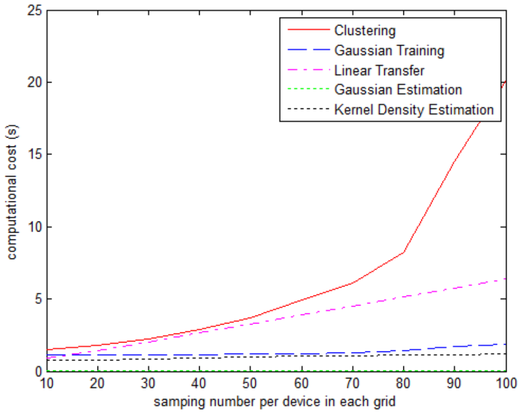

Fig. 10. Computational cost of each phase

From Figure 10 we find both the two kinds of positioning algorithm, Gaussian Estimation and Kernel Density Estimation, will not be affected greatly as sampling number increases. However, the computational cost of clustering, Gaussian Training and Linear Transfer step mushrooms. Therefore, although increasing number of sampling benefits positional accuracy, it promotes the training cost inevitably.

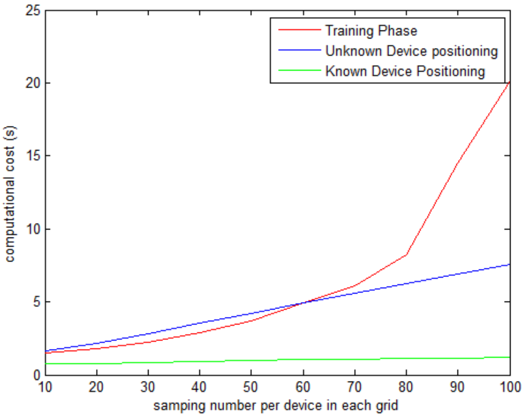

Fig. 11. Computational cost of our algorithm

The same result is also shown in Figure 11. The known device shows slight influence from sampling scale, while the time overhead of unknown devices grows remarkably.

5 Conclusion and Future Work

In this paper, we propose a novel Device-Clustering algorithm to provide better positioning accuracy in solving the device heterogeneity problem. Our algorithm is based on the macro Device-Cluster rather than original devices, so that the number of device types the system maintains is reduced. Besides, the sampling overhead to each device in total space is decreased. More importantly, with DC's entry into algorithm flow, the positioning accuracy is obviously increased. In the experiment, we employ eight different devices and sample enough fingerprints in a real academic building. The comparison of different combination also shows the best flow when executing our algorithm.

In the future, we will consider reducing the computational overhead of training phase. Because in the real-time positioning environment, the time-consuming clustering will also affect the localization effect. At the same time, we will try to reduce the computing cost in the positioning phase. If the number of training fingerprint is M and the number of observing fingerprint is N, the computing cost of kernel density estimator could be $O(MN)$. And with the increase of sampling scale, the time overhead grows remarkably. Besides, we will apply our algorithm in complex metropolitan environment and employ more general users to expand our fingerprint database.

Acknowledgement

This work is supported in part by the National Natural Science Foundation of China under Grant No.60973110, No.61003307, the Natural Science Foundation of Beijing City of China under Grant No.4102059 and the Major Projects of Ministry of Industry and Information Technology under Grant No.2010ZX03006-002-03, 2011ZX03005-005.

References

[1] Jun-geun Park, Ben Charrow, Jonathan Battat, Dorothy Curtis, Einat Minkov, Jamey Hicks, Seth Teller, Jonathan Ledlie, Growing an Organic Indoor Location System, Proc. 8th International Conference on Mobile Systems, Applications, and Services (MobiSys '10), pp.271 - 284, 2010.

[2] V. Bychkovsky, B. Hull, A. Miu, H. Balakrishnan, and S. Madden, “A Measurement Study of Vehicular Internet Access Using In-Situ 802.11 Networks,” in Proc. Annual International Conference on Mobile Computing and Networking (MobiCom), Los Angeles, CA, Sep. 2006.

[3] Arvin Wen Tsui, Wei-Cheng Lin, Wei-Ju Chen, Polly Huang, Hao-Hua Chu, Accuracy Performance Analysis between War Driving and War Walking in Metropolitan WiFi Localization, IEEE Transactions on Mobile Computing, vol. 9, no.11, pp. 1551-1562, November 2010.

[4] M. Azizyan, I. Constandache, and R. R. Choudhury, “SurroundSense: Mobile Phone Localization via Ambience Fingerprinting,” in Proc. Annual International Conference on Mobile Computing and Networking (MobiCom), Beijing, China, Sep. 2009, pp. 261–272.

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

[6] Minkyu Lee, Hyunil Yang, Dongsoo Han, Chansu Yu: Crowdsourced radiomap for room-level place recognition in urban environment. PerCom Workshops 2010: 648-653.

[7] Arvin Wen Tsui, Yu-Hsiang Chuang, Hao-Hua Chu: Unsupervised Learning for Solving RSS Hardware Variance Problem in WiFi Localization. MONET 14(5): 677-691 (2009).

[8] A. Haeberlen, E. Flannery, A. M. Ladd, A. Rudys, D. S. Wallach, and L. E. Kavraki, “Practical Robust Localization over Large-Scale 802.11 Wireless Networks,” in Proc. Annual International Conference on Mobile Computing and Networking (MobiCom), Philadelphia, PA, Sep. 2004, pp. 70–84.

[9] M. B. Kjærgaard, "Automatic Mitigation of Sensor Variations for Signal Strength Based Location Systems," in Proceedings of the Second International Workshop on Location- and Context-Awareness (LoCA 2006), 2006, pp. 30-47.

[10] Mikkel Baun Kjærgaard: Indoor location fingerprinting with heterogeneous clients. Pervasive and Mobile Computing 7(1): 31-43 (2011)

[11] Kjærgaard, M.B., Munk, C.V.: Hyperbolic location fingerprinting: A calibration-free solution for handling differences in signal strength (concise contribution). In: Sixth Annual IEEE International Conference on Pervasive Computing and Communications, pp. 110–116 (2008).

[12] Fangfang Dong, Yiqiang Chen, Junfa Liu, Qiong Ning, Songmei Piao: A Calibration-Free Localization Solution for Handling Signal Strength Variance. MELT 2009: 79-90.

[13] Zheng, V.W., Pan, S.J., Yang, Q., Pan, J.J.: Transferring multi-device localization models using latent multi-task learning. In: Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, pp. 1427–1432 (2008).

[14] Sun, Z., Yiqiang Chen, J.Q., Liu, J.: Adaptive localization through transfer learning in indoor wi-fi environment. In: Proceedings of the 2008 Seventh International Conference on Machine Learning and Applications, pp. 331–336 (2008).

[15] Xing-chuan Liu, Sheng Zhang, Qing-yuan Zhao, Xiao-kang Lin, "A real-time algorithm for fingerprint localization based on clustering and spatial diversity", Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2010 International Congress on, On page(s): 74 – 81.

[16] Pearson K (1900) Mathematical contribution to the theory of evolution VII: on the correlation of characters not quantitatively measurable. Philos. Trans. R. Soc. Lond. Ser. A: Math. Phys. Sci. 195:1–47, doi:10.1098/rsta.1900.0022.

[17] Sander, Jörg; Ester, Martin; Kriegel, Hans-Peter; Xu, Xiaowei (1998). "Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications". Data Mining and Knowledge Discovery (Berlin: Springer-Verlag) 2 (2): 169–194. doi:10.1023/A:1009745219419.

[18] Li Zheng, Tao Li, and Chris Ding. Hierarchical Ensemble Clustering. In Proceedings of 2010 IEEE International Conference on Data Mining (ICDM 2010).

[19] Seshadri V, Zaruba GV, Huber M (2005) A Bayesian sampling approach to in-door localization of wireless devices using received signal strength indication, in Proceedings of IEEE International Conference on Pervasive Computing (PERCOM 2005), pp. 75–84.

[20] T. Roos, P. Myllymaki, H. Tirri, P. Misikangas, and J. Sievanan. A probabilistic approach to WLAN user location estimation. International Journal of Wireless Information Networks, 9(3), July 2002.

[21] R. O. Duda, P. E. Hart, and D. G. Stork, Pattern Classfication, 2nd ed. Wiley-Interscience, 2000.