An Empirical Study of Indoor Localization Algorithms with Densely Deployed APs

Xin Chen, Junjun Kong, Yao Guo, Xiangqun Chen
Key Laboratory of High-Confidence Software Technologies (Ministry of Education),
School of Electronics Engineering and Computer Science, Peking University, Beijing, China
{chenxin12, kongjj07, yaoguo, cherry}@sei.pku.edu.cn

***Abstract*—Many indoor positioning algorithms have been proposed in the last decade, most of which are based on WiFi RSS fingerprints. However, the environment has changed dramatically since the original algorithms using only a few Access Points (APs). A typical building with densely deployed APs might contain hundreds of APs. The explosive growth of the number of APs introduces new challenges to these WiFi-based localization algorithms. This paper presents an empirical study of WiFi fingerprint-based indoor localization algorithms in a real-world environment with hundreds of APs. Our study aims to answer several important research questions regarding the influence of the number of APs, time variance and device variance. The study implements four existing algorithms and also proposes a new algorithm called LCS that is designed specifically for an AP-intensive environment. We compare the localization accuracy of different algorithms with different variances in the experimental results, which shows that the proposed LCS algorithm is able to efficiently resist diverse variances in an AP-intensive setup.**

I. Introduction

Location is one of the most important contexts for computing devices, especially mobile devices such as smartphones. Location information can be used to provide various location-based services (LBS) [10], including navigation, emergency rescue, traffic monitoring, etc. With the rapid development of mobile Internet, getting accurate location information becomes more and more important.

Global positioning system (GPS) is now widely used for localization, but it only works outdoors. Hence, many researchers have studied how to conduct indoor localization when GPS is unavailable. Many indoor localization techniques are proposed in the last decade. These techniques depend on various hardware devices, such as infrared [9], ultrasound [4], Bluetooth [3], radio-frequency identification (RFID) [11]. Since these techniques require the deployment of specific devices, it is expensive and difficult to deploy on a large scale. In comparison, indoor localization using wireless local area network (WLAN) signals is low-cost and easy to deploy.

Fingerprint-based localization technique [1] is one of the most common solutions to RSS-based indoor localization. It is used to improve indoor localization accuracy by collecting location related data (fingerprint). It takes RSS samples as fingerprints before positioning. When one positioning request is received, it matches the current RSS fingerprint with the samples. The location of the closest sample is regarded as the location of the current position.

It has been more than a dozen years since the first fingerprint-based technique was introduced [1]. Many algorithms have been proposed to improve the positioning accuracy. However, the environment has changed dramatically. Many previous studies are carried out with a few WiFi access points (APs). Existing studies before 2009 use no more than 26 APs [2], while more recent work observe no more than 100 APs [7], [8].

As a result of the popularization of WLAN, hundreds of APs may be deployed in a building. During our experiments in a university office building, 299 APs are detected on a single floor. The explosion of APs makes this environment more complex and uncontrollable, which brings several challenges.

- *There may exist different types of APs.* The number of APs are much higher because everyone can set up his/her own AP. The APs can be deployed using network adapters, PCs or even mobile phones. Since many of these APs are not public facilities, they are not promised to be stable. The ones set up by PCs and smartphones can even move around. It is hard to use them in localization algorithms if the environment contains many such APs.
- *APs come and APs go.* As time passes by, some APs disappear and new APs may emerge. Some APs might be temporarily unavailable due to various reasons. Some may be replaced by new ones. All these cases lead to the variation of the number of APs. When the environment is crowded with hundreds of APs, the variation may become more significant, which might increase the positioning errors.
- *Different devices might increase the difficulty of positioning algorithms.* Different devices may receive different RSS readings from the same AP, even at the same position and at the same time. In our experiment, the set of APs each device can detect are also different. It will be difficult to compare and match the fingerprints collected by other devices. Recent researches have attempted to use crowdsourcing [13], [15], [12] to save the labor of sampling, which will also introduce device diversity to indoor localization algorithms.

As many previous algorithms are designed for positioning in the environment with a few APs, we are curious whether they can overcome all the challenges in an environment with hundreds of APs. Our empirical study in this paper implements five fingerprint-based algorithms and analyzes their performance to answer the following research questions:

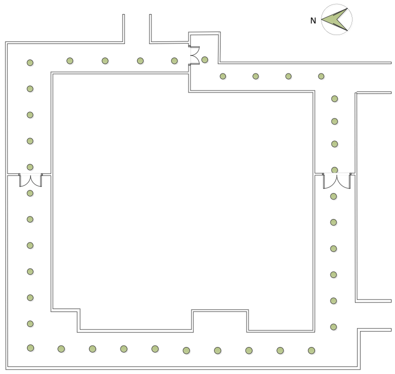


Fig. 1. The floor plan of the experiment environment.

TABLE I. SAMPLING CONFIGURATIONS

Sample #	Time	Devices
1	Reference time	Google Nexus S
2	3 days later	Google Nexus S
3	20 days later	Google Nexus S
4	2 months later	Google Nexus S
5	14 months later	Google Nexus S, ZTE N986, Sumsung Galaxy W I8150

- RQ1: How do different algorithms perform with densely deployed APs?
- RQ2: How do different algorithms perform as time goes by?
- RQ3: How do different algorithms perform with different devices?

The existing algorithms we studied include KNN Euclidean, KNN Tanimoto, N-gram and FreeLoc. All of them are fingerprint-based algorithms. This paper also proposes a simple algorithm called LCS (Longest Common Subsequence), which is designed for an AP-intensive environment. We use these five algorithms to answer the above three research questions by conducting the experiments and comparing their performances.

II. STUDY DESIGN

A. Experiment Setup

The experiment environment was the corridor of the fourth floor of an office building in Peking University. The map is shown in Fig. 1, which is roughly 50 m by 50 m. In each round of the experiments, we walked along the corridor and took samples every 4 meters. At each position, we collected 60 fingerprints for training and testing.

The sampling configuration is listed in Table I. In order to investigate RSS fluctuations as time passes by, we conducted five rounds of samplings within 14 months. The second to fourth samplings are, respectively, 3 days, 20 days, 2 months and 14 months after the initial sampling. Furthermore, we collected the data from three different smartphones in sampling #5, which is used for studying the influence of different devices on positioning accuracy.

B. Algorithms Studied

WiFi fingerprint positioning has been studied for many years. Although many algorithms have been proposed, only a few algorithms are commonly employed due to implementation difficulty and efficiency.

In order to maximize the value of this study, we choose algorithms based on two principles: (1) easy to implement, (2) efficient in localization. As a result, we choose four existing algorithms from the previous studies, which are KNN Euclidean, KNN Tanimoto, N-gram, FreeLoc. The first two KNN algorithms are the classical classification algorithms using the absolute signal strength, while the last two algorithms are based on the assumption that regardless of the RSS variance, the relative strength sequence is stable. We also propose a new LCS algorithm, which is designed specifically for the AP-intensive environment.

1) KNN: The KNN Algorithms [1] calculate the distances between fingerprints, and classifies the position of the query by assigning the position that is the most frequent among the K nearest fingerprints. We use two difference distance metrics in our experiment: Euclidean distance and Tanimoto coefficient.

- Euclidean distance between fingerprints X and Y:

$$D_E(X,Y) = \sqrt{\sum_i (RSS_{iX} - RSS_{iY})^2}$$

- Tanimoto Coefficient [6] between fingerprints X and Y:

$$C_T(X,Y) = \frac{\|RSS_X \cdot RSS_Y\|}{\|RSS_X\| + \|RSS_Y\| - \|RSS_X \cdot RSS_Y\|}$$

The larger $C_T(X,Y)$ is, the closer X and Y are.

2) N-gram: The N-gram algorithm [5] is a probabilistic method frequently used in computational linguistics. It is used to calculate the likelihood of two BSSID sequences. We sort the BSSIDs by the corresponding RSSs and calculate the probability of every Y's contiguous n-BSSID subsequence in X.

$$D_{n-gram}(X,Y) = \prod_i p_x(S_{yi})$$

where $p_x(S_{yi})$ is the probability that the Y's *i*th subsequence $p_x(S_{yi})$ appears in X. The distance from fingerprint X to Y is the product of all $p_x(S_{yi})$.

3) FreeLoc: FreeLoc[14] is proposed to solve problems in crowdsourcing-based systems. It uses only relative relationship information among RSS values. FreeLoc builds datasets for each AP. A dataset for an AP contains the APs with lower RSS than it. The likelihood of Fingerprint *X* and *Y* is the accumulating number of the common subset size of every AP.

As a result, FreeLoc does not depend on the actual RSS readings, thus making it robust to different devices, as long as the relative RSS readings keep unchanged on different devices.

4) LCS: Although FreeLoc solves the problem of device variance, it does not deal with time variances specifically when the set of APs changes after a period of time.

We propose an algorithm based on the idea of *longest common subsequences (LCS)* to deal with the AP changes in an AP-intensive environment. In order to tolerate the disappeared APs and newly-emerged APs, we define the similarity between two signatures X and Y as the length of the longest common subsequence prof their BSSID sequences ordered by strengths.

Compared to N-gram, the subsequence in LCS is non-continuous. This is because the RSS fluctuations of different APs are asynchronous, which may cause two adjacent APs in the sequence exchange orders. The similarity of two fingerprints can be determined as the length of their longest common subsequence.

Compared to FreeLoc, LCS is robust to AP changes because even a few APs disappear or emerge, it will not affect the length of their longest common subsequence. We will show later in the experiments that LCS performs well in a real-world environment with hundreds of APs.

III. Study Methodology and Results

We attempt to answer three research questions when applying existing localization algorithms in an environment with hundreds of APs. In this section, we investigate each question by discussing our motivation, approach and results analysis.

A. RQ1: How do different algorithms perform with densely deployed APs?

Motivation: Previous works have studied various positioning algorithms. However, most early studies are conducted in the environment with a limited number of APs. With the rapid development of wireless networks, more and more APs are deployed either in public or in private areas. This change in the environment raises the following concern: *under such conditions, can an algorithm perform as well in an environment with hundreds of APs as when there were only several APs?* We believe the answer to this question is important, because it will directly affect the validity of previous algorithms and the future research directions of indoor localization as well.

Approach: In order to study the performance of different algorithms with densely deployed APs, we conduct experiments by predicting using sample #2 with sample #1 as the training data. The results can also serve as the reference performance of each algorithm in the next two research questions.

Based on our observation, almost 300 APs coexist in the test environment. Some are public APs, and some are deployed by individuals, which makes this environment more complex and uncontrollable. This overwhelming number demonstrates the importance of conducting such kind of experiments.

Results: Fig. 2 shows the localization errors when predicting sample #2 based on sample #1.

The results show that, KNN Tanimoto, LCS and FreeLoc achieve better performances compared to the other two algorithms. When the tolerance error is 0 meter, KNN Tanimoto outperforms LCS and FreeLoc about 3 percent, but when the errors are more than 4 meters, FreeLoc becomes the best algorithm. KNN Euclidean performs the worst, whose precision is roughly 15% and 5% lower than KNN Tanimoto with tolerance error at 0 and 4 meters respectively.

We can see that LCS performs as well as the best algorithms in this set of results.

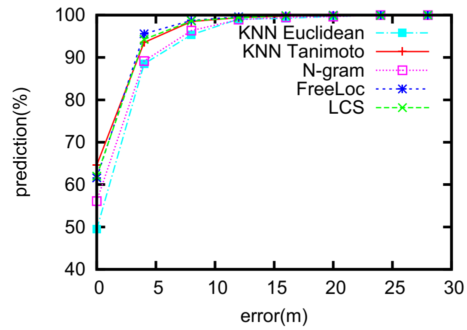


Fig. 2. Localization errors (CDF) with different algorithms.

TABLE II. AP variation through time compared to sample #1.

	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5
Total # of APs	278	280	272	288	299
# of Same APs	-	251	243	192	175
# of Disappeared APs	-	27	35	86	103
# of New APs	-	29	29	96	124

B. RQ2: How do different algorithms perform as time goes by?

Motivation: As time passes by, many factors will change. In this experiment, we observe two variations: (1) The signal strengths of each AP can change significantly. (2) Some APs may disappear, while new APs will be detected.

Fig. 3 shows the typical RSS distribution of one particular AP at the same position with the same device as time passes by. The median of the RSS decreased from -51dBm (sample #1) to -66dBm (sample #5). Besides some outliers, the RSS distributions of sample #1 and #5 does not even overlap. The observed tendencies of RSS variations are very different, which do not exhibit any general patterns.

Apart from the variance of AP RSSs over time, the APs that can be detected are not stable either. During the process, some APs might be damaged or out of function, while some may be replaced with new ones. There are also some mobile APs, which are set up with a PC or USB wireless routers. In our experiment, although the total number of APs does not change a lot, a significant portion of APs are replaced with new ones.

Table II presents the variation of APs at different time points. The APs are detected with the same smartphone (Google Nexus S) at a different time. We can see that even after only three days (Sample #2), about 10% of all the APs have disappeared, while 10% of new APs have emerged. The number of APs disappeared and emerged has grown gradually

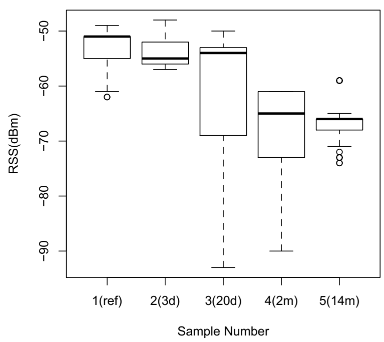


Fig. 3. RSS distribution of a typical AP at one position over time.

with time. After 14 months, pore than 1/3 of all the APs are changed, with 103 old APs replaced by 124 new ones.

When the RSSs fluctuate and the detected APs change and disappear, samples in the databases might be out of date. *Are the localization techniques robust enough to tolerate such variances?* In this research question, we will attempt to investigate this issue.

Approach: In order to answer this question, we use sample #1 as the training data to predict the positions in sample #2, #3, #4 and #5. We will compare the positioning accuracy of each algorithm when making predictions 3 days, 20 days, 3 months and 14 months after the reference sampling.

Result: Fig. 4 shows the experiment results. It is obvious that the positioning performance of every algorithm is somewhat affected by time. Overall, the accuracy of most algorithms becomes worse with time passing by.

Among all the algorithms, N-gram was affected most by time. The percentage of correct predictions (0m error) goes down from 56.1% to around 36.1%, while the prediction with errors less than 4 meters also goes down by more than 12%.

When comparing the two KNN algorithms, KNN Tanimoto degrades much faster than KNN Euclidean. Although KNN Tanimoto can correctly predict the positions at 64.6% after three days, which is much better than the 49.5% of KNN Euclidean. However, their performances in predicting the correct positions become very close after 14 months (both at 39%).

On the other hand, the performance of FreeLoc remains stable in a short period of time, for example, after 20 days and 2 months. However, after more than one year, its performance has shown a significant degradation. The correct predictions with errors of 0 meter and 4 meters have degraded by 16.3% and 9.2% respectively.

As we expect, because LCS relies on the longest common subsequence of APs, it is more time-tolerant. Although the correct predictions with no errors has also come down as other algorithms, when the error tolerance expands to 4 meters, it achieves nearly the same precision as three days later, which is much better than all four other algorithms. The results show that LCS is the most robust algorithm with time influence.

C. How do different algorithms perform with different devices?

Motivation: Although many early algorithms are conducting prediction based on training data collected on the same device, it becomes inevitable that more and more different devices will get involved in the localization process. Because different smartphones may have different WiFi adapter chips, it causes several problems, which include: (1) The signal strength readings of different devices might vary significantly. (2) The number of APs each device can detect might also differ. (3) The scanning speed of different devices could also be different.

Fig. 5 shows the RSS distributions of a typical AP with three different devices including Google Nexus S (Device 1), Samsung I8150 (Device 2) and ZTE N986 (Device 3). The RSSs are gathered at the same position and the same time. This guarantees that different devices reside in an environment with the same wireless signals, but the RSSs they receive turn out to be very different, not only in signal levels, but also in AP distributions. At this particular position, N986 receives higher RSSs on average with smaller variance while the RSSs Nexus S receives are lower on average with larger variance.

Table III compares the range of APs that different devices can detect. The data is collected from Sample #5, with the same three devices as above. The samples are taken at the same place and the same time, but the APs that can be detected differ greatly in quantity. Some APs can only be detected by one device. ZTE N986 observed 295 APs in total, which is close to the 299 APs detected by Google Nexus S. But 19 APs are new, and 23 APs can not be detected. One surprising finding is that the Samsung I8150 only detected 12.73 APs each time on average, while Nexus S and N986 could detect an average of 29.4 and 30.25 APs[1]. Because of the wide variance, the AP scanning ability may affect the positioning capabilities of each device.

Although the devices are installed with the same sampling app and are required to get scan results at the same frequency (every 0.5s), the executing time of Nexus S was nearly twice what N986 used. This can also affect localization accuracy and response speed.

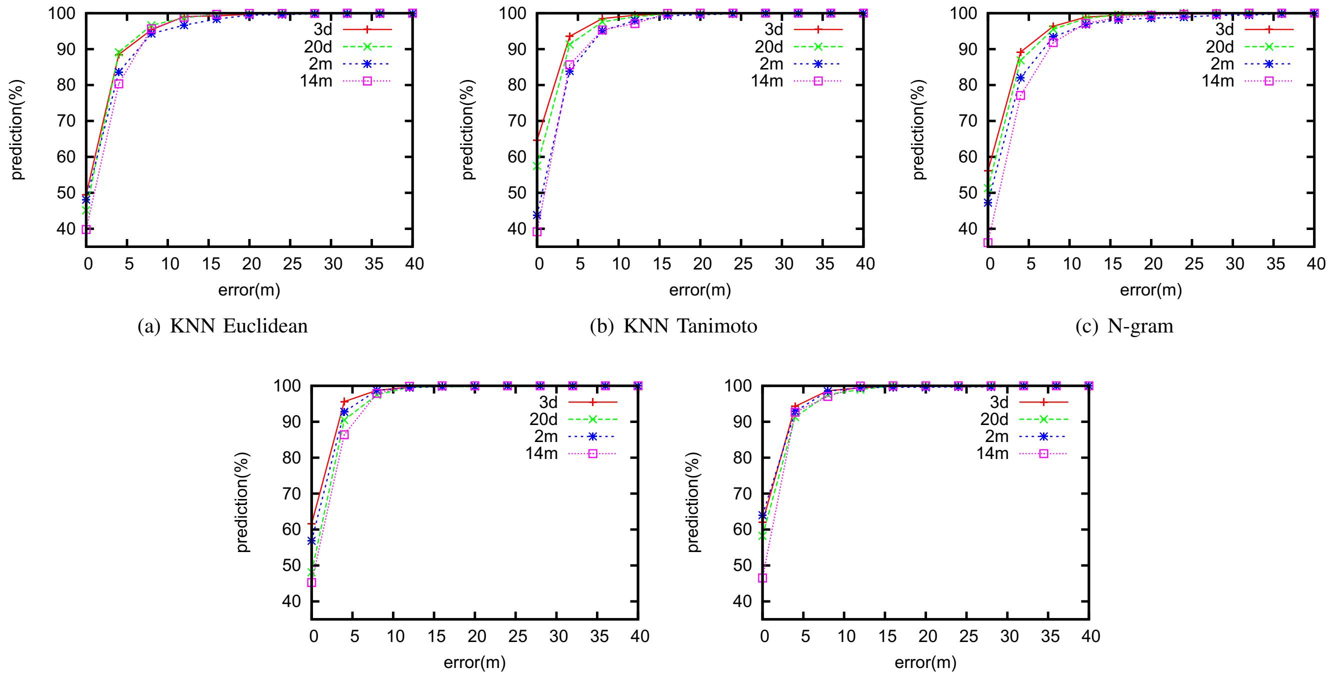
Since the algorithms are based on the same samples in database, if the sampling device is different from the positioning one, this could result in big difference in positioning accuracy. Besides sampling by one device, gaining samples from different phones can also lead to more serious problems.

Approach: We collected fingerprints from three different devices in Sample #5. We use sample #1, which is taken by device 1 (Google Nexus S), as the training data to predict fingerprints of different smartphones in sample #5 respectively.

Result: Fig. 6 presents the localization errors of an AP at the same position and the same time with different devices.

Among all five algorithms, KNN with Euclidean appears to be the most stable one. No matter which device is used, the prediction accuracy is almost the same. KNN with Tanimoto suffers most with different devices. To achieve over 80% precision, the error tolerance increases from 4 meters with Device 1 and 3 to 8 meters. In comparison, N-gram, FreeLoc

[1] Although Samsung I8150 and Nexus S are both produced by Samsung, they use different chipsets, which might explain the big gap in WiFi capabilities.



(a) KNN Euclidean (b) KNN Tanimoto (c) N-gram

(d) FreeLoc (e) LCS

Fig. 4. Localization Errors (CDF) as time passed by. (3d = 3 days, 20d = 20 days, 3m = 3 months, 14m = 14 months)

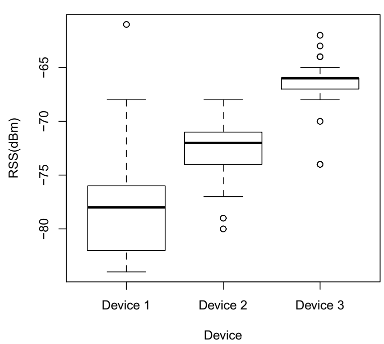


Fig. 5. RSS distribution with different devices of an AP at one place and the same time.

and LCS are slightly influenced. LCS and FreeLoc achieve the positioning precision close to KNN Euclidean with Device 2, while they perform much better than KNN Euclidean with the other two devices.

Since the number of APs that Device 2 can detect is much smaller than Device 1 and 3, the performance with Device 2 is generally worse than the others. although there is a big difference of RSS and executing time between Device 1 and 3, the positions of Device 3 can be predicted as precisely as Device 1 in general.

IV. Discussions

There are some limitations in our study. First of all, due to time and equipment restrictions, we conducted the experiments only in one environment. It will be helpful to perform extra experiments in a different place, such as a supermarket or shopping mall. However, we believe our study has revealed some interesting observations as discussed above. We will investigate it further in the near future.

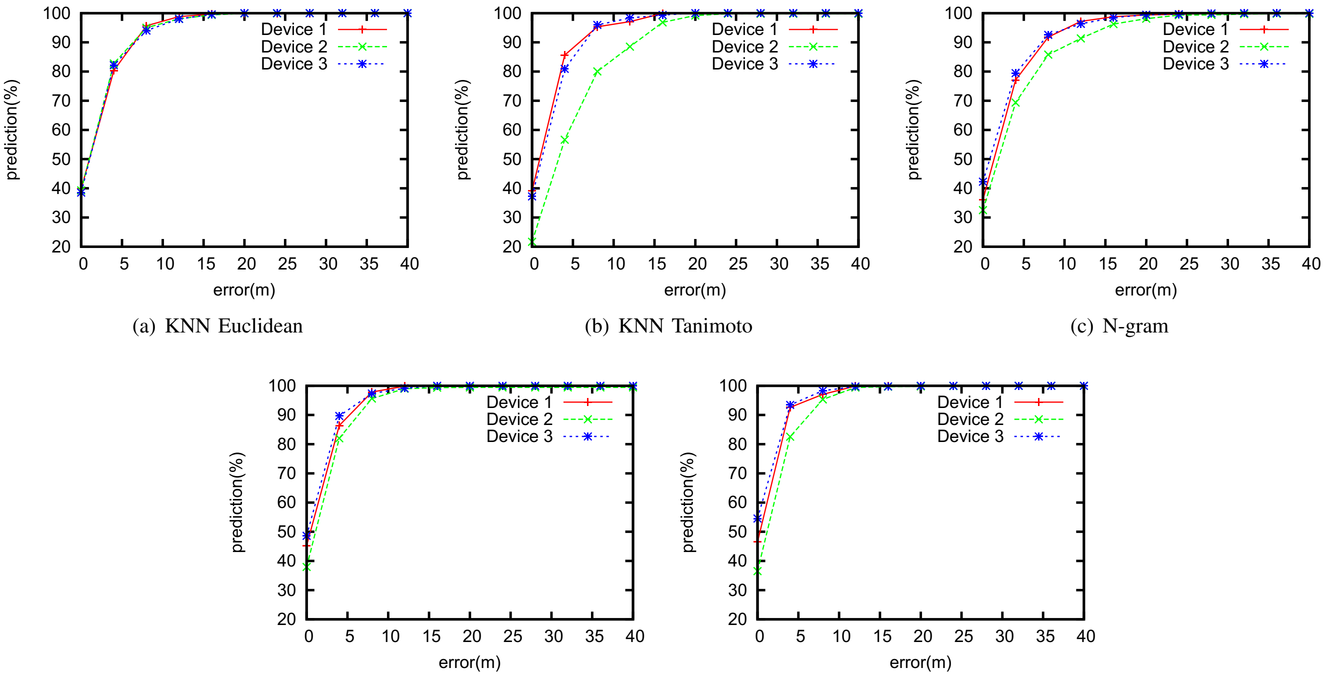
TABLE III. AP variations with different devices. *(The numbers shown in column 2-4 are in comparison with the APs in Device 1.)*

	Device 1	**Device 2**	**Device 3**
Total # of APs	299	222	295
# of same APs	-	206	276
# of Disappeared APs	-	93	23
# of New APs	-	16	19
Average # of APs in each scan	29.4	12.73	30.25

Another limitation is that the algorithms studied are mostly basic algorithms in order to test their efficiency due to different variances. More complicated algorithms will be able to tolerate all kinds of variances such as phone positions and RSS variation [7]. However, most of these algorithms have not been tested in an environment with hundreds of APs.

V. Conclusion

This paper presents an empirical study to compare the performances of different fingerprint-based indoor localization algorithms. We have shown that in a real-world environment with hundreds of APs, the performance of localization algorithms could vary significantly due to the number of APs, time variance and different devices. We have shown the number of APs could change significantly after a relatively long period (one year, for example).



(a) KNN Euclidean (b) KNN Tanimoto (c) N-gram

(d) FreeLoc (e) LCS

Fig. 6. Localization Errors (CDF) with different devices (Device 1 = Google Nexus S, Device 2 = Samsung I8150, Device 3 = ZTE N986)

We also propose a simple but effective algorithm based on longest common subsequences (LCS), which could tolerate the AP changes due to time. Experimental results show that LCS performs as well as the best existing algorithm in an AP-intensive environment.

Acknowledgment

This work is supported in part by the National Natural Science Foundation of China under Grant No.61103026, 61121063, the National Basic Research Program of China (973) under Grant No. 2011CB302604, and the High-Tech Research and Development Program of China (863) under Grant No. 2013AA01A605.

References

[1] P. Bahl and V. Padmanabhan. Radar: an in-building rf-based user location and tracking system. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 775–784 vol.2, 2000.

[2] G. Chandrasekaran, M. Ergin, J. Yang, S. Liu, Y. Chen, M. Gruteser, and R. Martin. Empirical evaluation of the limits on localization using signal strength. In *Sensor, Mesh and Ad Hoc Communications and Networks, 2009. SECON '09. 6th Annual IEEE Communications Society Conference on*, pages 1–9, June 2009.

[3] S. Feldmann, K. Kyamakya, A. Zapater, and Z. Lue. An indoor bluetooth-based positioning system: Concept, implementation and experimental evaluation. In W. Zhuang, C.-H. Yeh, O. Droegehorn, C.-T. Toh, and H. R. Arabnia, editors, *International Conference on Wireless Networks*, pages 109–113. CSREA Press, 2003.

[4] M. Hazas and A. Hopper. Broadband ultrasonic location systems for improved indoor positioning. *Mobile Computing, IEEE Transactions on*, 5(5):536–547, May 2006.

[5] Y. Jiang, X. Pan, K. Li, Q. Lv, R. P. Dick, M. Hannigan, and L. Shang. Ariel: Automatic wi-fi based room fingerprinting for indoor localization. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, UbiComp '12, pages 441–450, 2012.

[6] D. H. Kim, Y. Kim, D. Estrin, and M. B. Srivastava. Sensloc: Sensing everyday places and paths using less energy. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pages 43–56, 2010.

[7] Y. Kim, H. Shin, and H. Cha. Smartphone-based wi-fi pedestrian-tracking system tolerating the rss variance problem. In *Pervasive Computing and Communications (PerCom), 2012 IEEE International Conference on*, pages 11–19, March 2012.

[8] M. Kjaergaard, M. Wirz, D. Roggen, and G. Troster. Mobile sensing of pedestrian flocks in indoor environments using wifi signals. In *Pervasive Computing and Communications (PerCom), 2012 IEEE International Conference on*, pages 95–102, March 2012.

[9] S. Lee, K. N. Ha, and K.-C. Lee. A pyroelectric infrared sensor-based indoor location-aware system for the smart home. *Consumer Electronics, IEEE Transactions on*, 52(4):1311–1317, Nov 2006.

[10] D. Mohapatra and S. B. Suma. Survey of location based wireless services. In *Personal Wireless Communications, 2005. ICPWC 2005. 2005 IEEE International Conference on*, pages 358–362, Jan 2005.

[11] L. Ni, Y. Liu, Y. Lau, and A. Patil. Landmarc: Indoor location sensing using active rfid. *Wireless Networks*, 10(6):701–710, 2004.

[12] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen. Zee: Zero-effort crowdsourcing for indoor localization. In *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking*, Mobicom '12, pages 293–304, 2012.

[13] G. Shen, Z. Chen, P. Zhang, T. Moscibroda, and Y. Zhang. Walkie-markie: Indoor pathway mapping made easy. In *Proceedings of the 10th USENIX Conference on Networked Systems Design and Implementation*, nsdi'13, pages 85–98, 2013.

[14] S. Yang, P. Dessai, M. Verma, and M. Gerla. Freeloc: Calibration-free crowdsourced indoor localization. In *INFOCOM, 2013 Proceedings IEEE*, pages 2481–2489, April 2013.

[15] Z. Yang, C. Wu, and Y. Liu. Locating in fingerprint space: Wireless indoor localization with little human intervention. In *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking*, Mobicom '12, pages 269–280, 2012.