

Regular Article

A Scalable Wi-Fi Based Localization Algorithm

Truc D. Le¹, Nam T. Nguyen²

¹University of Science, Vietnam National University of Ho Chi Minh, Vietnam

²John von Neumann Institute, Vietnam National University of Ho Chi Minh, Vietnam

Correspondence: Truc D. Le, ldtruc@apcs.vn

Manuscript communication: received 16 September 2011, revised 8 November 2011

Abstract– This paper proposes an applicable and scalable approach which allows deploying the fingerprint Wi-Fi localization algorithm for different mobile devices. The original fingerprint localization algorithm performs accurately when the mobile device used in the deployment phase is the same as the mobile device used in the training phase. However, when a different mobile device is used in the deployment phase, a time-consuming re-training step (in the order of hours or days) is required to achieve the equivalent degree of accuracy. Our proposed approach replaces this re-training step by a short period of calibration (in the order of a few minutes), which can be done transparently to the user. To validate our approach, we did an analysis on collected data from a large scale experiment (14 laptops and 2 smartphones with 224-hour of collected data) and evaluated the performance on the real devices.

Keywords– access point; scanning point; signal strength; correlation; conversion, fingerprint.

1 INTRODUCTION

In ubiquitous computing, locating user's mobile device is one of the key issues to enable transparent and immediate services to user. Several localization techniques have been proposed that are based on GPS [1], cellular [2–4] or Wi-Fi [4–19] technologies. Although GPS-based technology has been improved to get the accuracy of up to a few meters [1], it can only be applied for outdoor locations since GPS signals from the satellites can be blocked easily by buildings, trees and other obstacles. Other localization approaches using Bluetooth devices are for short distances since Bluetooth signals cannot propagate very far. Approaches using cellular technology [2–4] can work for both outdoor and indoor locations (covered by cell towers) but offers low accuracy (several hundred meters). Recently, many approaches using Wi-Fi (802.11) signals [4–19] have been proposed for indoor locations thanks to their high accuracy rate and the increasingly popularity of the 802.11 access points. According to [5, 6], Wi-Fi based localization algorithms can be divided into four main categories: range-based, range-free (such as centroid [7, 8]), scene matching (such as fingerprint [5, 9]), aggregate and singular. Each of them has different strengths and weaknesses. Among those, fingerprint algorithms have been shown to offer very good results with high accuracy [5, 9] comparing to other approaches.

Techniques using fingerprint algorithm usually have two phases: training and testing (deployment). In the training phase, signal strengths are collected from all locations to infer the characteristics (fingerprint) of each location. After that each location is associated with the characteristics of OSSes (observed signal strengths) from the available APs. In the deployment phase, the

location of the mobile device is detected using the knowledge from the training phase. More specifically, the algorithm chooses k locations with the smallest distances (such as Euclidean distances) between the locations' fingerprints and the OSSes. The location of the device is then reported to be within those k locations. The choice of k depends on the accuracy of the applications.

Fingerprint approaches yield high accuracy compared to other approaches [5, 9]. However, the knowledge inferred in the training phase relies on a specific training mobile device, and thus the localization algorithm in the deployment phase may not work accurately for new mobile devices since different mobile devices, with different hardware and operating systems, may observe Wi-Fi signals differently. Our initial test showed that the accuracy may degrade severely (15 times) when different mobile devices are used in the training and deployment phases comparing to when the same mobile device is used in both phases. Hence, full re-training is normally required for each new mobile device to obtain the highest accuracy. This full training, however, is very expensive because it may require user to collect Wi-Fi signals from many positions for a long duration (in the order of hours or more depending on the accuracy and the size of the locations). For scenarios with many mobile devices from different users, full training for each new mobile device becomes impossibly practical.

In this paper, we described a method to allow deploying fingerprint algorithm for different mobile devices in the deployment phase without requiring the expensive re-training step. In this method, a sample mobile device is used in the training phase (as in the original fingerprint algorithm). However, in the deployment

phase, each new mobile device is calibrated through a partial training (in the order of a few minutes) to infer the conversion formulas. Once the formulas are inferred, the Wi-Fi signals observed by the new device can be converted to the sample mobile device and thus, the fingerprint algorithm can work on the new device immediately. The advantage of this method is the replacement of the expensive re-training step by a quick calibration step. Note that our approach does not change the fingerprint algorithms. Instead, it works together with any existing fingerprint algorithms to minimize the cost of training new mobile devices. For generality, we assume no knowledge of the source code of the underlying operating systems or wireless card's drivers of the mobile devices.

2 RELATED WORK

2.1 Localization Applications

Maps and navigation systems are the most pervasive application. It is useful for both civilian and army purposes. Based on the reported user's current location, other enhanced assistances such as identifying the nearest restaurant or bank can be applied. Location tracking can be used for security enhancement such as theft detection (e.g. software iLocalis [20] for iPhone); child monitoring for parents to supervise their children, (e.g. software SniperSpy [21]). Some locative media applications such as Smart Party [22] or Location-Driven Fiction [23] also need localization techniques. Additionally, location is an important factor to be considered for research on user's behaviors or data mining.

2.2 Localization Algorithms

Many localization techniques using 802.11 have been developed recently. For outdoor environment, GPS is likely the best choice because it gives very high accuracy, up to a few meters [1]. Nevertheless, it does not work in indoor environment (Section 1). Techniques using cellular technology can work both in indoor and outdoor but give low accuracy [2–4]. With the rapid increasing of 802.11 APs, Wi-Fi is a favorable technology nowadays [4–19].

There are several Wi-Fi based localization algorithms. Most of them contain two phases: off-line (training) phase and on-line (deployment) phase. In the off-line phase, they build a map which contains location indicators. In the deployment phase, the estimated location is obtained by searching the map for the most appropriate location corresponding to the OSSes. Based on the method used to build the location indicators and how to use them in deployment phase [5, 6], classifies some of localization algorithms as following.

Range-based family: These algorithms extract the *received signal strength index* (RSSI), *time of arrival* (TOA), *time difference of arrival* (TDOA), *angle of arrival* (AOA) from APs to estimate the distances. For instance, in the RSSI [10], the relationship between the signal loss and the propagation distance can be expressed as following:

$$I(r) = cr^{-a}, \quad (1)$$

where $I(r)$ represents OSS at distance r , $c > 0$ is a constant or environment dependent variation, and parameter $a > 0$ is not fixed and depends on situation. Since OSS is some interpretation of wave energy, for ideal case, a is 2 (i.e. wave energy is inversely proportional to the square of distance [11]). However, for indoor environment with many obstacles, these algorithms do not yield high accuracy because they do not take into account the effects of obstacles on the OSS. Furthermore, for short distance, it is extremely difficult to calculate the *time-of-flight* mentioned earlier.

Range-free family: Rather than use precise physical property measurements, range-free algorithms use coarser metric like connectivity or hop-counts to landmarks to place bounds on candidate positions. Centroid algorithm [7, 8] is an example. Techniques using the centroid algorithm are simple and very fast to compute and do not require wave propagation model. They build a lookup table: each entry represents an AP with its coordinates. In the on-line phase, the estimated location is the center of all coordinates from observable APs. Weighting by OSSes is an improved version of this algorithm. However, many experiments [7, 8] show that the algorithms in this family do not give high accuracy for indoor and even outdoor environment.

Scene matching: Many algorithms fall into this category [11–16]. A radio map of the environment is constructed, either by measuring actual samples, using signal propagation models, or some combination of the two. A node then measures a set of radio properties (often just the OSSes of a set of landmarks), the fingerprint, and attempts to match these to known locations on the radio map. These approaches are almost always used in indoor environments because signal propagation is extensively affected by reflection, diffraction and scattering, and thus ranging or simple distance bounds cannot be effectively employed. Matching fingerprints to locations can be cast in statistical terms, as a machine-learning classifier problem [13], or as a clustering problem [16]. Among these, fingerprint is a simple algorithm [5, 9] but achieves high accuracy [5, 9]. The fingerprint localization algorithm also includes two phases as mentioned earlier. In the training phase, each location is associated with average signal strengths observed by the sample device for each AP at this location. In other words, each location i can be represented as a vector \mathbf{x}_i containing average signal strengths of each AP. In the deployment phase, the location k of a deployed mobile device is estimated based on the observed \mathbf{x} of that device at that location. The estimated location of k is the location with minimum Euclidean distance to \mathbf{x} . The algorithm works best only when the sample device and the deployed device are the same. When the deployed device is different from the sample device, the accuracy degrades significantly (8.6 meters on average and more than 15 meters on the worst case in our experiment) because the capabilities of Wi-Fi observation of two mobile devices are different due to hardware, wireless card, operating system, drivers and other software. The current solution to this problem is to build a new training set for this new mobile device.

However, this re-training step is extremely expensive. For example, it may take up to more than three hours for training an area of 80 meter square. Thus, it is not scalable for deployment with different user devices. Therefore, in this paper, we proposed an approach to solve this problem by introducing the calibration step for each new device. The purpose of the calibration step is to infer the conversion formulas so that the knowledge from the training step can be used for the deployment step. This calibration step can be done quickly and transparently when the user first enters the site (at the ticket booths or the entrances).

Aggregate or singular: Aggregate approaches use collections of many nodes in the network in order to localize (often by flooding), while localization of a node in singular methods only requires it to communicate to a few landmarks. For example, algorithms using optimization [17] or multidimensional scaling [18] require many rounds of estimates between nodes.

3 PROPOSED APPROACH TO MAKE FINGERPRINT LOCALIZATION ALGORITHM ACCURATE & SCALABLE

3.1 Our Approach

According to 2.2, the main problem of fingerprint localization algorithm is the reduction of accuracy when the deployed device is different from the sample device. A full re-training step for new devices can increase the accuracy but also requires lots of time and efforts, which makes the algorithm not scalable. In order to solve this problem, we propose a new stage called calibration phase, which takes a few minutes. This phase occurs before the deployment phase (Figure 1).

In our approach, full training is performed only once for the sample device as in fingerprint approach. In the calibration phase, the newly deployed device is put at some well-known locations (usually a few) and observes signal strengths from APs. From these data together with the data of the sample device also at these well-known locations, we can infer a relationship between signal strengths observed by two mobile devices in a form of a conversion formula. From now on, the deployed mobile device can be localized with the training set of the sample mobile device. Indeed, in the deployment phase, the OSSes of the deployed

mobile device are converted through the conversion formula which makes them applicable to the OSSes of the sample mobile device. This calibration step is much less expensive than the full re-training as in the original fingerprint algorithm. This is an important factor to make it scalable. The next paragraph will discuss how to find the conversion formula.

According to [11], the signal strength is inversely proportional to the square of distance. In other words, if y is the signal strength value and x is the distance between the current location and the AP, we have $y = a / x^2 + b$. Different devices may have different coefficient values (a, b) due to the differences in hardware, device drivers, and OSs. This explains why different devices may observe different value of signal strength at the same location. If the OSSes follow the formula above, we expect the conversion formula between two different devices is linear (even when the values (a, b) vary for different mobile devices/APs). To verify this, we deploy Pearson product-moment correlation coefficient, or correlation for short [24]. Let X and Y be the random variables indicating the signal strengths from two devices respectively. Correlation coefficient is a quantitative scalar ranging from -1 to 1 inclusive, where values of ± 1 indicate a perfectly linear relationship between two variables ($+1$ indicates X increases as Y increases and -1 indicates X decreases as Y increases) and a value of 0 indicates no linear relationship. The correlation coefficient r is determined by the following formula.

$$r = \frac{n \sum xy - \sum x \sum y}{\sqrt{(n \sum x^2 - (\sum x)^2) (n \sum y^2 - (\sum y)^2)}}, \quad (2)$$

where n is the number of samples, i.e. the number of correspondence signal strengths between two mobile devices.

In the next section, we are going to present our data collection, which is needed for the calculation of these correlation coefficients between pairs of mobile devices.

3.2 Data collection

In this experiment, we collected an extensive set of OSSes from different mobile devices at different locations. The purpose of this large scale experiment is to analyze the behavior of the signal strengths which might affect the conversion algorithms in the calibration step. The data is also used to validate the performance of our approach described in Section 4. The data was collected at floor 5 and 7 in an 11-floor building on two different days (two trials). At each day (trial), we collected data from both floors 5 and 7. The two trials were one month apart to analyze the variance of signal strengths over time and its effects. Each floor is divided into 58 cells (representing 58 locations), where each cell is $1.2\text{m} \times 1.2\text{m}$. At each cell, each mobile device recorded the OSSes from all available APs. On average, there were at least 10 APs observed in all locations of the experimental area. 14 laptops and 2 smart phones (Table I) were used to collect the data. These devices,

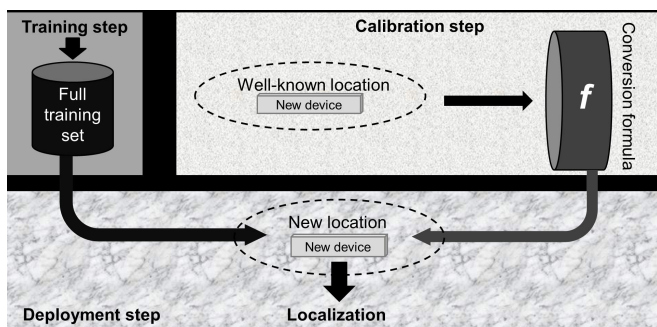


Figure 1. The scheme of our approach.

Table I
INFORMATION OF DEVICES USED TO COLLECT DATA

Device	Platform	OS	Wi-Fi card
1	Sony VPCEB	Win 7x64	Atheros AR9285
2	Asus X82 Q	Win 7	Atheros
3	Toshiba Satellite A105-s4064	Win 7	Intel(R) 3945ABG
4	Dell latitude d630	Win 7	Intel(R) 3945ABG
5	Asus	Win 7	Atheros AR5007EG
6	Dell Vostro 1400	Win XP	Intel(R) 3945ABG
7	Toshiba Satellite u305-s2806	Win 7	Intel(R) 3945ABG
8	HP Compaq 6730s	Win 7	Intel(R) 5100 AGN
9	Acer Aspire 4920G	Win 7	Intel(R) 3945ABG
10	Acer Aspire 5740	Win 7	Atheros AR5B93
11	Compaq Presario F577AU	Win 7	Broadcom 802.11b/g WLAN
12	Dell Inspiron 1545	Win 7	Dell 1397 WLAN Mini-Card
13	Toshiba Satellite L310	Win 7	Intel(R) 4965AGN
14	Dell	Win 7	Intel(R) 3945ABG
15	Phone Sony Ericsson X10	Android 1.6	Unknown
16	Phone HTC Desire	Android 2.1	Unknown

equipped with various wireless cards and operating systems, were provided from a group of volunteer students. The wireless signals were recorded at all cells using Vistumbler software [25]. At each cell, the device was put at center, aimed eastward and collected signals from all APs continuously for at least 200 scanning points (4 to 6 minutes). More than 60 students voluntarily helped collecting data for four full days (2 trials \times 2 floors). The total size of the collected data from all devices is equivalent to 224 hours of recording Wi-Fi signals. Figure 2 shows the position of the cells at floor 5. The same layout was applied at floor 7. At each location, different devices may observe wireless signals differently. This is because different devices may be equipped with different wireless cards from different manufacturers. Additionally, different OSs may interpret the raw data observed by the wireless card differently and may adjust them accordingly before reporting it back to our monitor software. Therefore, we assume no knowledge of the source code of the device driver or how the operating system re-adjusts the OSSes for each device.

3.3 Correlation Results

From the collected data above, we now can calculate the correlation coefficients (Section 3.1) for all pairs of devices.

Table II shows the correlation coefficients r of 13 pairs of devices. Because of the transitivity of linear relationship, it is sufficient to calculate the correlation coefficients of device 1 to the remaining devices. The result from Table II shows that these pairs have high values of r . In fact, the r values of all pairs are greater than 0.76. Among those, the r values of 13 pairs (87 percent of the pairs) are greater than 0.9. Since the values of r 's are very close to 1, there is strong evidence that signal strengths from a pair of devices have linear

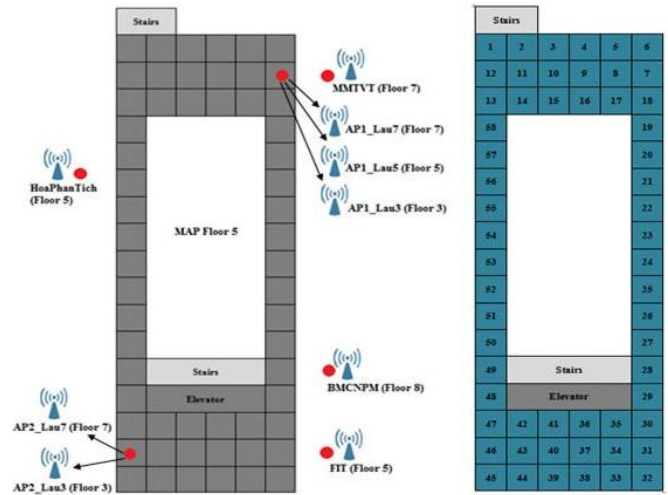


Figure 2. The map of 58 cells in floor 5 with positions of some APs.

Table II
CORRELATION COEFFICIENTS OF PAIRS OF DEVICES

Pair of devices	r value
1 – 2	0.948
1 – 3	0.878
1 – 4	0.921
1 – 5	0.759
1 – 6	0.946
1 – 7	0.933
1 – 8	0.949
1 – 9	0.927
1 – 10	0.906
1 – 11	0.916
1 – 12	0.928
1 – 13	0.950
1 – 14	0.841
1 – 15	0.940
1 – 16	0.927

relationship. Based on this observation, in Section 4, we proposed a conversion formula to allow the quick deployment of fingerprint localization algorithm for new devices without the expensive re-training step.

4 CALIBRATION ALGORITHM AND RESULTS

4.1 Calibration Algorithm

As discussed in Section 3, our goal is to design a calibration step which takes input as corresponding signal strengths from two mobile devices at some “well-known” locations and produces the conversion formula f . We will formulate the problem by calling the corresponding average mean of signal strengths from the deployed and sample devices are x_i and y_i respectively ($i = 1, \dots, n$), where n is the number of correspondence signal strengths between two mobile devices. We need to find coefficients b_1 and b_0 of the line ($y = b_1x + b_0$) such that E in (3) is minimized.

$$E = \sum_{i=1}^n (y_i - b_1x_i - b_0)^2. \quad (3)$$

This model is called *linear regression* [24]. We see that E is convex and the domains of b_1 and b_0 are convex sets. Hence, its only one critical point is the local minimum,

and also the global minimum. We can find it by taking derivatives of E with respect to b_1 and b_0 .

$$\begin{cases} \frac{\partial E}{\partial b_1} = \sum_{i=1}^n -2x_i(y_i - b_1x_i - b_0) \\ \frac{\partial E}{\partial b_0} = \sum_{i=1}^n -2(y_i - b_1x_i - b_0) \end{cases} \quad (4)$$

Setting these derivatives to 0 yields

$$\begin{cases} \sum_{i=1}^n x_i y_i = b_1 \sum_{i=1}^n x_i^2 + b_0 \sum_{i=1}^n x_i \\ \sum_{i=1}^n y_i = b_1 \sum_{i=1}^n x_i + b_0 n \end{cases} \quad (5)$$

This is a linear system of b_1 and b_0 . Solving this system, we obtain the solution in (6).

$$\begin{cases} b_1 = \frac{\sum_{i=1}^n x_i y_i - \frac{1}{n} \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{\sum_{i=1}^n x_i^2 - \frac{1}{n} (\sum_{i=1}^n x_i)^2} \\ b_0 = \frac{1}{n} \sum_{i=1}^n y_i - \frac{a}{n} \sum_{i=1}^n x_i \end{cases} \quad (6)$$

For computational simplicity (especially with some CAS like Matlab), we can use matrix representation for these calculations. Let view the original data as matrix representation as following:

$$\mathbf{y} = [y_1 \ y_2 \ \dots \ y_n]^T, \quad (7)$$

$$\mathbf{X} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_n \end{bmatrix}^T, \quad (8)$$

$$\mathbf{b} = [b_0 \ b_1]^T. \quad (9)$$

Thus E can be expressed as following:

$$\begin{aligned} E &= \|\mathbf{X}\mathbf{b} - \mathbf{y}\|^2 = (\mathbf{X}\mathbf{b} - \mathbf{y})^T (\mathbf{X}\mathbf{b} - \mathbf{y}) \\ &= \mathbf{b}^T \mathbf{X}^T \mathbf{X} \mathbf{b} - 2\mathbf{b}^T \mathbf{X}^T \mathbf{y} + \mathbf{y}^T \mathbf{y} \end{aligned} \quad (10)$$

Taking derivative of E with respect to vector \mathbf{b} , we obtain:

$$\frac{\partial E}{\partial \mathbf{b}} = 2\mathbf{X}^T \mathbf{X} \mathbf{b} - 2\mathbf{X}^T \mathbf{y}. \quad (11)$$

Setting (11) to 0 and solving for \mathbf{b} , we obtain:

$$\mathbf{b} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}. \quad (12)$$

The solution in (12) and (6) is the same and it is the unique solution of the original problem. In (11) to (12), we implicitly assume that $\mathbf{X}^T \mathbf{X}$ is invertible. Indeed, in most of the cases, this is true because we obtain \mathbf{X} by measurements (observations from deployed devices). All measurements cannot be identical. Thus, \mathbf{X} is full column rank, so $\mathbf{X}^T \mathbf{X}$ is full rank and it is invertible. In the case that the deployed device actually observes identical values, we need to go to another location for the calibration step.

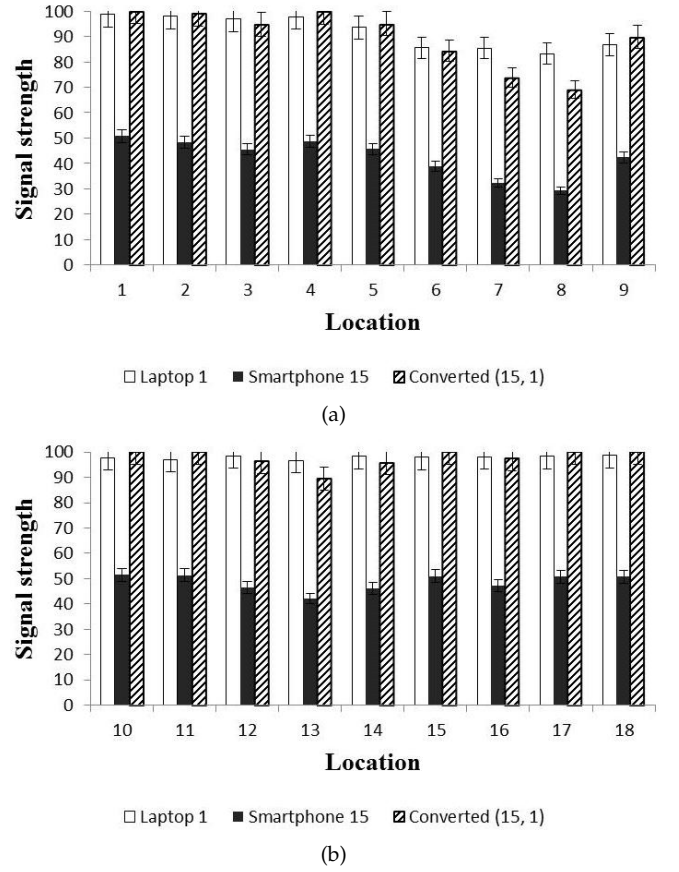


Figure 3. Linear conversion of AP AP1-Lau5 from phone 15 to laptop 1.

4.2 Results

Evaluate the conversion formula: This evaluation is only based on the two devices' signal strengths, and is independent with localization algorithm. We only show the conversions of two APs from two pairs of devices. Other pairs have similar pattern. Note that our calibration method can be applied for most Wi-Fi based localization algorithms with the same problem of the expensive re-training step (as in fingerprint approach). Figure 3 and Figure 4 show the performance of the linear regression approach when converting the signals observed by smartphone 15 to laptop 1 and laptop 6 to laptop 1 for 9 different locations. In Figure 3, the white and black bars are the AP1-Lau5 signals observed by device 1 and device 15 respectively. The crossed-bar is the converted signals from the observed ones by device 15. The closer between the white and the crossed bars are, the better it is. In this figure, the calibration is at cells 10 to 15. Figure 3 shows the signal strengths of device 1 (a laptop), device 15 (a smart phone), and converted signals from device 15 to device 1 using linear formula (confidence intervals are at 90%). We show only the first 18 cells (the other cells experience similar pattern). Similarly, Figure 4 shows the conversion of FIT signal strengths from laptop 6 to laptop 1. From Figures 3 and 4, the converted signals are much closer to the signals of sample device. We tried all pairs of devices but due to the space limit, we cannot display all results here.

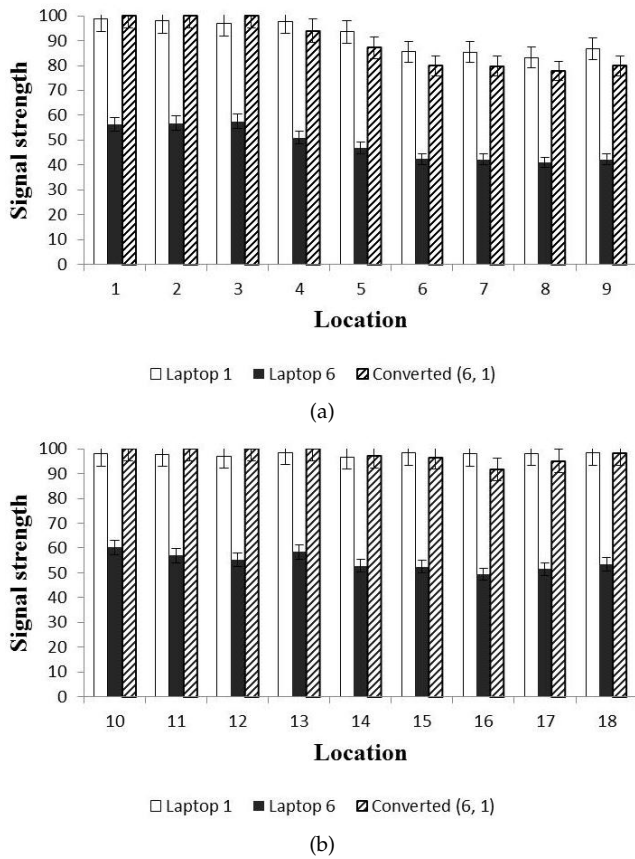


Figure 4. Linear conversion of AP FIT from laptop 6 to laptop 1.

Validate our calibration approach with the fingerprint localization algorithm: To validate our approach, we implemented the calibration step using the linear conversion method on a simple fingerprint localization algorithm. Note that our approach does not depend on a particular localization algorithm. It can be applied for other localization algorithm. We chose the fingerprint to deploy our calibration since it is simple and has been shown to provide good accuracy level [5, 9]. We used 60% of the data collected from each mobile device for the training set. The remaining 40% of the data was for the testing set. In the deployment phase, the location with the smallest Euclidean distance to the OSSes (from testing set) was reported. We evaluated the performance of our linear approach by comparing: *non-converted for the same device* (NCSD), *non-converted for different device* (NCDD), and *converted for different device* (CDD). The NCSD is the original fingerprint algorithm where the same device is used for both deployment and training phases. This is the ideal case but not very realistic on real deployment since new user can bring in any new device. The NCDD is the original fingerprint algorithm where the device used in the deployment phase is different from the one used in the training phase without calibration. The CDD is our calibration approach where the device used in the deployment phase is different from the one used in the training phase and calibrated.

Table III shows the average errors of the three approaches over 58 cells. In Table III, the cells marked with "x" are not applicable. For example, the pair laptop 1

Table III
AVERAGE ERRORS IN METERS OVER 58 CELLS

Sample-Deployed Devices	Average Errors in Meters			
	NCSD	NCDD	CDD	Gain
Laptop 1 – Laptop 1	0.71	x	x	x
Laptop 1 – Laptop 2	x	7.71	3.29	4.42
Laptop 1 – Laptop 3	x	2.72	1.92	0.80
Laptop 1 – Laptop 4	x	7.28	4.72	2.56
Laptop 1 – Laptop 5	x	3.91	3.04	0.87
Laptop 1 – Laptop 6	x	2.46	2.48	-0.02
Laptop 1 – Laptop 7	x	3.57	3.03	0.54
Laptop 1 – Laptop 8	x	4.79	2.68	2.11
Laptop 1 – Laptop 9	x	3.07	2.81	0.26
Laptop 1 – Laptop 10	x	2.92	2.86	0.06
Laptop 1 – Laptop 11	x	2.18	2.20	-0.02
Laptop 1 – Laptop 12	x	7.23	5.18	2.05
Laptop 1 – Laptop 13	x	8.60	6.48	2.12
Laptop 1 – Laptop 14	x	5.69	2.98	2.71
Laptop 1 – Smartphone 15	x	15.57	3.03	12.54
Laptop 1 – Smartphone 16	x	12.52	5.55	6.97

and 2 cannot have NCSD because they are two different devices. The gain column describes how much CDD is better than NCDD, which is actually the difference between these values. For NCSD (the ideal case), the average error is extremely low, only 0.71m. Without the conversion, different devices may suffer an error that is 15 times worse than when using the same device (Smartphone 15 and Laptop 1) (15.57m comparing to 0.71m). However, with our approach, the error is much better, only 3.03m. In Table III, there are two cases of laptop 6 and laptop 11 where we obtain negative gains (i.e. our approach is worse than the original fingerprint algorithm). This is because these devices happen to observe very similar AP signals and thus, do not need to be calibrated. Note that these negative gains are too small and can be negligible. In addition, we observe that the errors are severe when the deployed devices are very much different (laptop versus smartphone). It means that mobile devices from different categories are strongly recommended to use our calibration step before deployment.

5 CONCLUSIONS

We have proposed a scalable method to convert the signal strengths between two devices that can work together with the fingerprint algorithm. Our approach is validated by the analysis on correlation coefficients (Section 3) and a real deployment on a simple fingerprint localization algorithm (Section 4.2). Our approach only requires a few minutes of user time for the calibration step, which is more practical comparing to the original fingerprint algorithm, which requires a full re-training step for every new device. Through a large set of data collected from many mobile devices, our approach showed acceptable accuracy when working with fingerprint algorithm. In other words, there is strong evidence that our method is applicable for different mobile devices and makes Wi-Fi localization become more practical with high accuracy. Moreover, our approach can be used in other localization algorithms which require conversion of signal strengths.

6 FUTURE WORK

In our experiment, we have not focused on filtering APs. Different set of APs may yield different accuracy level. Thus, an AP weighted conversion may improve the accuracy of the calibration step. Besides, we will study more about the calibration strategies such as the time required for calibration, the number of locations required for calibration, and also the positions of calibrated locations. Picking the right device to use as the sample device may also affect the accuracy. Thus, it is interesting to discover some criteria for picking the sample device in the future work.

REFERENCES

- [1] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low-cost outdoor localization for very small devices," *IEEE Personal Communications*, vol. 7, no. 5, pp. 28–34, Oct. 2000.
- [2] M. Chen, T. Sohn, D. Chmelev, D. Haehnel, J. Hightower, J. Hughes, A. LaMarca, F. Potter, I. Smith, and A. Varshavsky, "Practical metropolitan-scale positioning for GSM phones," in *UbiComp 2006: Ubiquitous Computing*, ser. Lecture Notes in Computer Science, P. Dourish and A. Friday, Eds. Springer Berlin / Heidelberg, 2006, vol. 4206, pp. 225–242.
- [3] D. Gundlegard and J. M. Karlsson, "Handover location accuracy for travel time estimation in GSM and UMTS," *IET Intelligent Transport Systems*, vol. 3, no. 1, pp. 87–94, Mar. 2009.
- [4] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, J. Tabert, P. Powledge, G. Borriello, and B. Schilit, "Place lab: Device positioning using radio beacons in the wild," in *Pervasive Computing*, ser. Lecture Notes in Computer Science, H. Gellersen, R. Want, and A. Schmidt, Eds. Springer Berlin / Heidelberg, 2005, vol. 3468, pp. 301–306.
- [5] Y. Chen, K. Kleisouris, X. Li, W. Trappe, and R. Martin, "The robustness of localization algorithms to signal strength attacks: A comparative study," in *Distributed Computing in Sensor Systems*, ser. Lecture Notes in Computer Science, P. Gibbons, T. Abdelzaher, J. Aspnes, and R. Rao, Eds. Springer Berlin / Heidelberg, 2006, vol. 4026, pp. 546–563.
- [6] K. Kleisouris, "Improving the speed and accuracy of indoor localization," Ph.D. dissertation, Rutgers The State University Of New Jersey, 2009.
- [7] Y.-C. Cheng, Y. Chawathe, A. LaMarca, and J. Krumm, "Accuracy characterization for metropolitan-scale Wi-Fi localization," in *Proc. 3rd International Conf. Mobile Systems, Applications, and Services (MobiSys '05)*, New York, NY, USA, 2005, pp. 233–245.
- [8] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proc. 19th Annual Joint Conf. IEEE Computer and Communications Societies (INFOCOM 2000)*, vol. 2, Tel Aviv, Israel, 26 – 30 Mar. 2000, pp. 775–784.
- [9] A. Varshavsky, D. Pankratov, J. Krumm, and E. Lara, "Calibree: Calibration-free localization using relative distance estimations," in *Proceedings of the 6th International Conference on Pervasive Computing (Pervasive '08)*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 146–161.
- [10] H.-S. Ahn and W. Yu, "Indoor localization techniques based on wireless sensor networks," in *Mobile Robots - State of the Art in Land, Sea, Air, and Collaborative Missions*, X. Chen, Y. Chen, and J. Chase, Eds. InTech, 2009, ch. 13.
- [11] J. C. Ralston, C. O. Hargrave, and D. W. Hainsworth, "Localisation of mobile underground mining equipment using wireless Ethernet," in *Proc. Fourtieth IAS Annual Meeting Industry Applications Conf.*, vol. 1, 2 – 6 Oct. 2005, pp. 225–230.
- [12] A. Haeberlen, E. Flannery, A. M. Ladd, A. Rudys, D. S. Wallach, and L. E. Kavradi, "Practical robust localization over large-scale 802.11 wireless networks," in *Proc. 10th Annual International Conf. Mobile Computing and Networking (MobiCom '04)*, New York, NY, USA, 2004, pp. 70–84.
- [13] D. Madigan, E. Einahrawy, R. P. Martin, W.-H. Ju, P. Krishnan, and A. S. Krishnakumar, "Bayesian indoor positioning systems," in *Proc. IEEE 24th Annual Joint Conf. IEEE Computer and Communications Societies (INFOCOM 2005)*, vol. 2, 13 – 17 Mar. 2005, pp. 1217–1227.
- [14] J. Hightower and G. Borriello, "Particle filters for location estimation in ubiquitous computing: A case study," in *UbiComp 2004: Ubiquitous Computing*, ser. Lecture Notes in Computer Science, N. Davies, E. Mynatt, and I. Siio, Eds. Springer Berlin / Heidelberg, 2004, vol. 3205, pp. 88–106.
- [15] S. Saha, K. Chaudhuri, D. Sanghi, and P. Bhagwat, "Location determination of a mobile device using IEEE 802.11b access point signals," in *Proc. IEEE Wireless Communications and Networking Conf. (WCNC 2003)*, vol. 3, New Orleans, LA, 2003, pp. 1987–1992.
- [16] M. A. Youssef, A. Agrawala, and A. Udaya Shankar, "WLAN location determination via clustering and probability distributions," in *Proc. 1st IEEE Int. Conf. Pervasive Computing and Communications (PerCom 2003)*, 23–26 Mar. 2003, pp. 143–150.
- [17] L. Doherty, K. S. J. Pister, and L. El Ghaoui, "Convex position estimation in wireless sensor networks," in *Proc. 20th Annual Joint Conf. IEEE Computer and Communications Societies (INFOCOM 2001)*, vol. 3, Anchorage, AK, USA, 22–26 Apr. 2001, pp. 1655–1663.
- [18] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz, "Localization from mere connectivity," in *Proc. 4th ACM Int. Symp. Mobile Ad Hoc Networking & Computing (MobiHoc '03)*, New York, NY, USA, 2003, pp. 201–212.
- [19] K. Kaemarungsi and P. Krishnamurthy, "Properties of indoor received signal strength for WLAN location fingerprinting," in *Proc. First Annual Int. Conf. Mobile and Ubiquitous Systems: Networking and Services (MOBIQUITOUS 2004)*, 22–26 Aug. 2004, pp. 14–23.
- [20] ilocalis. [Online]. Available: <http://ilocalis.com>
- [21] Sniperspy. [Online]. Available: <http://www.sniperspy.com>
- [22] K. Eustice, V. Ramakrishna, N. Nguyen, and P. Reiher, "The smart party: A personalized location-aware multimedia experience," in *Proc. 5th IEEE Consumer Communications and Networking Conf. (CCNC 2008)*, Las Vegas, NV, 10–12 Jan. 2008, pp. 873–877.
- [23] K. Eustice, V. Ramakrishna, A. Walker, M. Schnaider, N. Nguyen, and P. Reiher, "nanosphere: Location-driven fiction for groups of users," in *Human-Computer Interaction. HCI Intelligent Multimodal Interaction Environments*, ser. Lecture Notes in Computer Science, J. Jacko, Ed. Beijing, China: Springer Berlin / Heidelberg, July 2007, vol. 4552, pp. 852–861.
- [24] D. C. Montgomery and G. C. Runger, "Simple linear regression and correlation," in *Applied Statistics and Probability for Engineers*. John Wiley & Sons, 2003, pp. 372–409.
- [25] Vistumbler. [Online]. Available: <http://www.vistumbler.net>



Truc D. Le was born in Ho Chi Minh, Vietnam, in 1989. He is currently a senior student of University of Science, Vietnam National University of Ho Chi Minh. He is going to graduate by the end of 2011. He is a member of ubiquitous research group mentored by Dr. Nam T. Nguyen. His research interests include machine learning, ubiquitous computing areas such as localization and its applications.



Nam T. Nguyen received his PhD degree in Computer Science from University of California, Los Angeles (UCLA), in 2009. He is currently the Head of Social Media Lab at Jon von Neumann Institute, Vietnam National University of Ho Chi Minh. He is also a faculty at University of Science, Vietnam National University of Ho Chi Minh. His research interests are in ubiquitous systems, wireless networks, network security.