People Help People: A Pattern-matching Localization with Inputs from User Community

Chi-Chung Lo*, Shih-Chin Lin*, Sheng-Po Kuo*, Yu-Chee Tseng*,
Shin-Yun Peng†, Shang-Ming Huang†, Yu-Neng Hung†, and Chin-Fu Hung†
*Department of Computer Science, National Chiao-Tung University, Hsin-Chu, 300, Taiwan
†Telecommunication Laboratories, Chunghwa Telecom Co., Ltd., Taoyuan, 326, Taiwan
E-mail:{ccluo, shihchinlin, spkuo, yctseng}@cs.nctu.edu.tw;
{sypeng, puregold_0719, jameshon, jeffhung}@cht.com.tw

***Abstract*—Location-based services are regarded as a killer application of mobile networks. Among all RF-based localization techniques, the pattern-matching scheme is probably the most widely accepted approach. A key factor to its success is the accuracy concern and the calibration efforts to collect its training data. In this paper, we propose a community-based approach to reduce the calibration effort. We show how to get some volunteers (called co-trainers) to help add more training data to our location database. We also show how to rate the credit level of a co-trainer and the trust level of a piece of training data contributed by a co-trainer. We believe that our framework can greatly reduce the calibration effort of the pattern-matching localization scheme.**

Keywords: localization, location-based service, pattern matching, pervasive computing, wireless positioning system.

I. Introduction

Recently, a lot of location-based services (LBS) [1], [2], such as navigation and tracking, have been proposed. At present, GPS is still the widest used technology for positioning. However, GPS is not applicable to indoor services. Therefore, much research has been dedicated to the wireless positioning system (WPS), which is based on Radio Frequency (RF) signals to locate a mobile user. A promising approach is the pattern-matching technique [3]–[6]. The pattern-matching technique consists of two phases: training phase and positioning phase. In the training phase, the service provider has to collect received signal strength indicator (RSSI) from beacons at all training locations and save them in a location database. During the positioning phase, a mobile device should collect its current RSSI and send it to the location server, which will compare the RSSI against those patterns in the location database. The best-matched location is regarded as the current location of the mobile user.

However, a main barrier for the pattern-matching approach to become widely acceptable is its high calibration efforts in the training phase. For example, in a wireless city, millions of training records may have to be collected and examined. Several works have been dedicated to reducing the calibration efforts [7]–[9]. In this paper, we propose to reduce the training efforts by involving the user to contribute some training data. We propose a framework based on the concept of web 2.0 to achieve this goal. This may greatly reduce the training effort of the pattern-matching approach.

Our framework works as follows. We define three roles in our system: *trainer*, *co-trainer*, and *beneficiary*. A trainer is a trusted person who will collect training data for the system. A co-trainer is a volunteer who may contribute training data to the system whenever he/she is willing to do so. A beneficiary is simply a user to receive the localization service, who may rate how accurate a location query is whenever he/she is willing to do so. Our framework will rate the credit level of a co-trainer and the trust level of a piece of training data. We show how to maintain a location database which includes training data from both trainers and co-trainers and how to respond to location queries based on such a database.

The rest of this paper is organized as follows. Section II gives some background knowledge. The framework of community-based training is presented in Section III. The design of trust model is presented in Section IV. Conclusions is drawn in Section V.

II. Background Knowledge

In recently years, due to the prospering of mobile devices, the focuses of localization demand has changed from measurement, navigation and field sports to personal location-based services such as local search, personal guidance and location-dependent multimedia services. Some critical disadvantages of GPS got more obvious in these application environments. For example, there must be no satellite signals inside a shopping mall, whereas a common location-based coupon service is desired. Even in an outdoor business section of a city, the accuracy of GPS is bad because there is no enough satellite in the sight or the multi-path fluctuation is serious.

For these reasons, many localization technologies have been proposed, such as infrared-based [10], ultrasonic-based [11], and wireless positioning systems (WPS) [3]. WPS utilized the wireless transmission characteristic to calculate

	Suburbs Accuracy	Urban Accuracy	Indoor Accuracy	Specified Device	Beacon Support	Computation Speed	Calibration Cost	Fluctuation Resistance
GPS	Good	Bad	Unavailable	Dedicated	Yes	Fast	No	Bad
Triangulating	Average	Average	Average	Common	Yes	Fastest	No	Bad
Pattern-matching	Average	Good	Good	Common	No	Slower	Yes	Good

Table I
COMPARISON BETWEEN GPS AND WPS

the position of mobile device. The WPS techniques can be mainly categorized into AoA-based [12], ToA-based [13], TDoA-based [14], and pattern-matching ones [3] [15] [16].

The ToA-based and TDoA-based localization calculates the location by time gaps or signal decay according to ideal sphere path loss model. The finer beacon coverage is guaranteed because the high density of base stations. However, the multi-path and signal fluctuation problems still existed.

The principle of pattern-matching based localization system is based on collecting the mapping of location labels and RSSI. While the mappings are collected, we can estimate the location from the nearest pattern. Signal fluctuation and multi-path problem will turn to the features of patterns. The more sophisticated circumstances are in the environment, the more outstanding characteristics store in the patterns. A pattern-matching localization system usually consists of two phases, *training phase* and *positioning phase*.

- In the training phase, we are given a set of beacon sources $\mathcal{B} = \{b_1, b_2, \ldots, b_m\}$ (from base stations or access points) and a set of training locations $\mathcal{L} = \{\ell_1, \ell_2, \ldots, \ell_n\}$. We then measure the RSSI of the beacons generated from $\mathcal{B}$ at each training location $\ell_i \in \mathcal{L}$ to create a *feature vector* $\boldsymbol{\upsilon}_i = [\upsilon_{(i,1)}, \upsilon_{(i,2)}, \ldots, \upsilon_{(i,m)}]$ for ℓ_i, where $\upsilon_{(i,j)} \in \mathbb{R}$ is the average RSSI of the beacon generated from b_j, $j = 1..m$. Then, the matrix $\mathcal{V} = [\boldsymbol{\upsilon}_1, \boldsymbol{\upsilon}_2, \ldots, \boldsymbol{\upsilon}_n]$ is called the *radio map* and used as the basis of positioning results.
- In the positioning phase, a user will measure its RSSI vector $\boldsymbol{s} = [s_1, s_2, \ldots, s_m]$ at the current location and compare s with each feature vector in $\mathcal{V}$. In particular, for each $\upsilon_{(i,j)} \in \mathcal{V}$, we define a distance function $h(\cdot)$ for the corresponding training location ℓ_i as [3]

$$h(\ell_i) = \|\boldsymbol{s}, \boldsymbol{\upsilon}_i\| = \sum_{j=1}^{n} \sqrt{(s_j - \upsilon_{i,j})^2}.$$

  Then, the user is considered at location ℓ_i if $h(\ell_i)$ returns the smallest value.

Table I compares the three localization methods. Obviously, the wireless pattern-matching based localization system is a good choice in large-scale environment. This paper is focusing on how to alleviate the main drawbacks in large-scale environment.

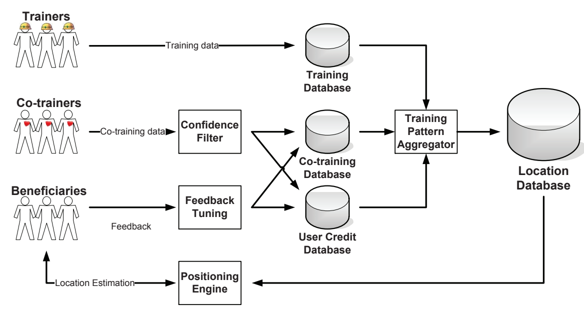

Figure 1. Members and the Data Flow

III. FRAMEWORK OF COMMUNITY-BASED TRAINING

Below, we propose a framework to allow volunteers to contribute training data to a location database. In Section IV, we will further discuss the trust model under this framework. We first define three types of equipments in our system.

- *Beacon*: A device that can transmit RF signals for positioning purpose. An example is Wi-Fi APs.
- *RSSI collector*: A device that can collect signals from nearby beacons and measure their signal strengths.
- *Positioning tool*: A tool that can identify the physical location of a portable device. A typical example is GPS receivers. If a user does not have a GPS receiver at hand or he/she can not receive GPS signals due to shading effect, he/she can still use a map interface to indicate his/her current location. For example, a user can chick on a Google Map screen to identify his/her current location.

For example, John is a co-trainer of our community. He carries an iPhone, which has 3G and Wi-Fi interfaces. When he walks into a department store with Wi-Fi beacons, he may help collect RSSI patterns by clicking on a Google Map interface to indicate where he is. In this way, our location database can collect more new data.

To achieve the goal of accepting volunteers' contribution, we define three roles in our system. Note that a user may play multiple roles.

- Beneficiary: A user who simply enjoys the localization services of our system.
- Trainer: A user who helps adding dependable training data into the location database. Incoming training data contributed by a trainer is always reliable.
- Co-trainer: A user who helps adding training data into the location database of which dependability is unknown.

Note that for trainers and co-trainers, we can provide some tools for them to contribute data.

Figure 1 demonstrates the relationships of these roles. Since our system accepts data from open community, it is important to build some trust model among users and the training data contributed by them. For each piece of training data, we will associate with it a trust level. The training data from a trainer always has a high trust level. However, the training data from a co-trainer needs to be evaluated by a *Confidence Filter (CF)*. It is affected by how this piece of training data is similar to the existing data in the current location database and by the past credit of the co-trainer. Training data that has a low trust level will be removed from the location database. On the other hand, for each co-trainer, we will associate with him/her a credit level. It is adjusted dynamically according to how dependable the training data contributed by him/her in the past was. In order to understand how dependable a piece of training data is and how dependable a co-trainer is, we will accept feedbacks from beneficiary. Whenever, a beneficiary queries his/her current location and gets a response from our system, he/she is allowed to input a feedback on how dependable the location estimation is. A positive response from the beneficiary will add some value to the trust level of the piece of training data used to position the user, as well as to the credit level of the co-trainer who contributed this piece of training data. Contrarily, a negative response from the beneficiary will lower down some value of the trust level of the piece of training data used to position the user, as well as some value of the credit level of the co-trainer who contributed this piece of training data. In the next section, we will show the details to realize these concepts.

IV. Implementation Details

Because the pattern-matching positioning service is based on the contribution of the community, the system should not believe that all the calibrations are reliable. There might be some malicious users pass misleading calibrations to our community database. Furthermore, even the calibrations which are already in our database, the system still have to keep eyes on whether the data is still applicable for any upcoming conditions.

To achieve the goal to calibrate the volunteers' contribution, the training data from a co-trainer needs to be evaluated by CF. Training data that has a low similarity will be removed from the location database. We define two data structures in our system.

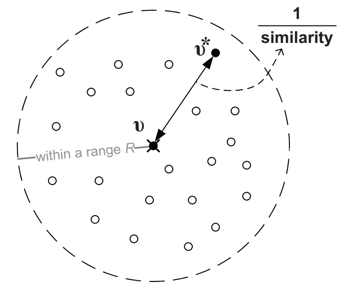

Figure 2. Calculate similarity for Confidence Filter.

- User profile: For each trainer and co-trainer j, we will associate with it a credit level $\boldsymbol{\mu_j}$ in $[0, 5]$. The $\boldsymbol{\mu_j}$ of trainers are always 5. The initial $\boldsymbol{\mu_j}$ of co-trainers are 3. The $\boldsymbol{\mu_j}$ of co-trainers will increase/decrease after *feedback tuning (FT)*.
- Training pattern: Trainers and co-trainers contribute the training data into the location database. When trainer/co-trainer j submits training data $\boldsymbol{v}_i$ to the location database, we will modify training data as a training pattern $\tilde{\boldsymbol{v}}_{i,j} = (\boldsymbol{v}_i, \boldsymbol{\mu_j})$, where $\boldsymbol{\mu_j}$ is the trust level equal to the credit level of the trainer/co-trainer j.

CF is coarse filtering process aims to remove dramatic outlier values from incoming calibrations. Basically, CF is a binary classification, decide which record should be regarded as abandoned, which is trustworthy. When a user submits a training pattern $\tilde{\boldsymbol{v}}_{i,j}$ to location server, CF makes a quick examination on the calibration through similarity calculation. The similarity is defined as the degree how much this calibration is close to our existing trusted calibration database. The calculation is done by following steps.

1) Consider the existing patterns at ℓ_i inside of the range R over RSSI space centered at $\boldsymbol{v}$, where $\boldsymbol{v}$ is the average pattern of that patterns submitted by the trainer, we must calculate the similarity by the distance from $\boldsymbol{v}$ to $\boldsymbol{v}^*$, where $\boldsymbol{v}^*$ is the incoming training pattern.
2) If the similarity is smaller than a threshold, the training pattern is regarded as *abandoned* pattern.
3) If the trust level of the incoming training pattern is lower than a threshold, the training pattern is also regarded as abandoned pattern. Note that the trust level of incoming training pattern comes from its contributor's credit level.

On the other hand, for each co-trainer, we will associate with him/her a credit level by FT. It is adjusted dynamically according to how dependable the training data contributed by him/her in the past was. In order to understand how dependable a piece of training data is and how dependable a co-trainer is, we will accept feedbacks from beneficiary. If a beneficiary enjoys our positioning service, he/she can make a recommendation for the system if he/she is unsatisfied with a positioning result. Whenever, a beneficiary queries his/her current location and gets a response from our system, he/she is allowed to input a feedback on how dependable the location estimation is. A recommendation is a trigger event which the database has to adjust the trust level of nearby existing patterns and its contributor. FT is designed by following steps.

1) A beneficiary can make a recommendation for the system if he/she is unsatisfied with a positioning result. The recommendation is positive response or negative response.
2) If the response is positive, we add some value to the trust level of the training patterns used to position the user, as well as to the credit level of the co-trainers who contributed this piece of training patterns.
3) Contrarily, a negative response from the beneficiary will lower down some value of the trust level of the piece of training patterns used to position the user, as well as some value of the credit level of the co-trainers who contributed this piece of training patterns.
4) If trust level of a training pattern is lower than a threshold, the training pattern will remove from the location database.
5) If credit level of a co-trainer is lower than a threshold, his/her incoming training patterns will never be trusted.

V. Conclusions

We propose community-based concept to deal with the calibration effort. We show how to get some volunteers to help add more training data to our location database. The idea basically leaves calibration problem from system developers to users and meet the concept of Web 2.0. Since the training data is not only maintained by the system developers, some data reliability problems must be introduced. We also show how to rate the credit level of a co-trainer and the trust level of a piece of training data contributed by a co-trainer. We believe that our framework can greatly reduce the calibration effort of the pattern-matching localization scheme.

References

[1] S.-P. Kuo, S.-C. Lin, B.-J. Wu, Y.-C. Tseng, and C.-C. Shen, "GeoAds: A Middleware Architecture for Music Service with Location-Aware Advertisement," in *IEEE Int'l Conf. on Mobile Ad-hoc and Sensor Systems (MASS)*, 2007.

[2] C.-C. A. Lo, T.-C. Lin, Y.-C. Wang, Y.-C. Tseng, L.-C. Ko, , and L.-C. Kuo, "Using intelligent mobile devices for indoor wireless location tracking, navigation, and mobile augmented reality," in *IEEE VTS Asia Pacific Wireless Commun. Symposium (APWCS)*, 2010.

[3] P. Bahl and V. N. Padmanabhan, "RADAR: An In-building RF-based User Location and Tracking System," in *IEEE INFOCOM*, vol. 2, 2000, pp. 775–784.

[4] T. Roos, P. Myllymäki, H. Tirri, P. Misikangas, and J. Sievänen, "A Probabilistic Approach to WLAN User Location Estimation," *Int'l Journal of Wireless Information Networks*, vol. 9, no. 3, pp. 155–164, 2002.

[5] S.-P. Kuo, B.-J. Wu, W.-C. Peng, and Y.-C. Tseng, "Cluster-Enhanced Techniques for Pattern-Matching Localization Systems," in *IEEE Int'l Conf. on Mobile Ad-hoc and Sensor Systems (MASS)*, 2007.

[6] J. Letchner, D. Fox, and A. LaMarca, "Large-Scale Localization from Wireless Signal Strength," in *Proc. of the Nat'l Conf. on Artificial Intelligence (AAAI)*, 2005, pp. 15–20.

[7] Y.-C. Cheng, Y. Chawathe, A. LaMarca, and J. Krumm, "Accuracy Characterization for Metropolitan-scale Wi-Fi Localization," in *ACM MOBISYS*, vol. 5, 2005, pp. 233–245.

[8] X. Chai and Q. Yang, "Reducing the Calibration Effort for Location Estimation Using Unlabeled Samples," in *IEEE PERCOM*, 2005, pp. 95–104.

[9] P. Krishnan, A. S. Krishnakumar, W.-H. Ju, C. Mallows, and S. Ganu, "A System for LEASE: Location Estimation Assisted by Stationary Emitters for Indoor RF Wireless Networks," in *IEEE INFOCOM*, vol. 2, 2004, pp. 1001–1011.

[10] E. Brassart, C. Pegard, and M. Mouaddib, "Localization using infrared beacons," *Robotica*, vol. 18, no. 2, pp. 153–161, 2000.

[11] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket Location-Support System," in *ACM/IEEE MOBICOM*, 2000, pp. 32–43.

[12] D. Niculescu and B. Nath, "Ad Hoc Positioning System (APS) Using AOA," in *IEEE INFOCOM*, vol. 3, 2003, pp. 1734–1743.

[13] M. Addlesee, R. Curwen, S. Hodges, J. Newman, P. Steggles, A. Ward, and A. Hopper, "Implementing a Sentient Computing System," *Computer*, vol. 34, no. 8, pp. 50–56, 2001.

[14] A. Savvides, C.-C. Han, and M. B. Strivastava, "Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors," in *ACM/IEEE MOBICOM*, 2001, pp. 166–179.

[15] S.-P. Kuo and Y.-C. Tseng, "A Scrambling Method for Fingerprint Positioning Based on Temporal Diversity and Spatial Dependency," *IEEE Trans. on Knowledge and Data Engineering*, vol. 20, no. 5, pp. 678–684, 2008.

[16] J. J. Pan, J. T. Kwok, Q. Yang, and Y. Chen, "Multidimensional Vector Regression for Accurate and Low-Cost Location Estimation in Pervasive Computing," *IEEE Trans. on Knowledge and Data Engineering*, vol. 18, no. 9, pp. 1181–1193, 2006.