Editorial Manager(tm) for Mobile Networks and Applications
Manuscript Draft

Manuscript Number:

Title: Unsupervised Learning for Solving RSS Hardware Variance Problem in WiFi Localization

Article Type: Multi-Hop Wireless Networking

Keywords: Localization systems, Wi-Fi network, unsupervised learning, Wi-Fi device variance.

Corresponding Author: Prof. Arvin Tsui,

Corresponding Author's Institution:

First Author: Arvin Tsui

Order of Authors: Arvin Tsui; Yu-Hsiang Chuang; Hao-hua Chu

Unsupervised Learning for Solving RSS Hardware Variance Problem in WiFi Localization

Arvin Wen Tsui • Yu-Hsiang Chuang • Hao-hua Chu

Arvin Wen Tsui

Graduate Institute of Networking and Multimedia
National Taiwan University
No.1, Roosevelt Road, Sec.4,
Taipei 106, Taiwan
Information and Communication Research Laboratories
Industrial Technology Research Institute
Rm. 513, Bldg. 51, 195, Sec. 4, Chung Hsing Rd.,
Chutung, Hsinchu 310, Taiwan.
arvin@itri.org.tw

Yu-Hsiang Chuang
Information and Communication Research Laboratories
Industrial Technology Research Institute
Rm. 513, Bldg. 51, 195, Sec. 4, Chung Hsing Rd.,
Chutung, Hsinchu 310, Taiwan.
wood@itri.org.tw

Hao-hua Chu
Graduate Institute of Networking and Multimedia
Dept. of Computer Science and Engineering
National Taiwan University
No.1, Roosevelt Road, Sec.4,
Taipei 106, Taiwan
hchu@csie.ntu.edu.tw

Arvin Wen Tsui is Ph. D. student at Graduate Institute of Networking and Multimedia, National Taiwan University and also works at Industrial Technology Research Institute, Taiwan as a technical deputy manager. He received his B.S. (1994) in Computer Science from Soochow University and M.S. (1996) in Computer Science and Information Engineering from National Taiwan University. From 1996-1998, he was a research assistant of Institute of Information Science, Academia Sinica. His current research interests include localization and ubiquitous computing.

Yu-Hsiang Chuang received the B.S. and M.S. degree in information management from National Chung Chen University, Chiayi, Taiwan in 2004 and 2006, respectively. He is currently an engineer in Information & Communications Research Laboratories of Industrial Technology Research Institute of Taiwan, R.O.C. His research interests include Localization over Large-Scale 802.11 Wireless Networks, wireless sensor network

Hao-Hua Chu is an associate professor at the Graduate Institute of Networking and Multimedia and Department of Computer Science and Information Engineering, National Taiwan University. He received his B.S. (1994) in Computer Science from Cornell University and PhD (1999) in Computer Science from University of Illinois at Urbana Champaign. From 1999-2000, he was a senior software engineer at Intel. From 2000-2003, he was a project manager at (NTT) DoCoMo USA Labs. His research areas are pervasive computing and sensor/wireless network.

Abstract Hardware variance can significantly degrade the positional accuracy of RSS-based WiFi localization systems. Although manual adjustment can reduce positional error, this solution is not scalable as the number of new WiFi devices increases. This paper proposes an unsupervised learning method of automatically solving the hardware variance problem in WiFi localization. The unsupervised learning method was designed and implemented in a working WiFi positioning system and evaluated using different WiFi devices with diverse RSS signal patterns. The experimental results demonstrated that the proposed learning method improves positional accuracy within 100 seconds of learning time.

Keywords: Localization systems, Wi-Fi network, unsupervised learning, Wi-Fi device variance.

1 Introduction

The WiFi location system is a promising technology aimed at complying with real-world location-based applications. In comparison with other location systems using GPS, cameras, RFID, ultrasound, *etc*., WiFi location systems have several advantages in deployment practicability. For examples, WiFi works in both indoor and outdoor environments; it also leverages existing and widely-deployed Wi-Fi networks. As a result, several companies such as Ekahau [1], Aeroscout [2], Innerwireless Pango [3] and Skyhook wireless [4] are actively developing location-based applications for using this technology in hospitals, warehouses, factories, amusement parks and other locations.

Despite its numerous advantages in deployment practicability, deployment of current WiFi location systems remains problematic. A major issue is the WiFi hardware variance problem: the WiFi device used for training the radio map during the calibration phase (training device) may differ from the WiFi devices used during the tracking phase (tracking devices). Varying Received Signal Strength (RSS) can degrade the signal-patterns between training and tracking devices as well as the positional accuracy of WiFi location systems. The experiments described in Section 4 indicated that the average positional error may increase by more than 100%. Further, this hardware variance problem is not limited to differences in the WiFi chipsets used by training and tracking devices (e.g., Intel Centrino vs. Lucent chipset) but also occurs when the same WiFi chipsets are connected to different antennas types and/or packaged in different encapsulation materials (e.g., Intel Centrino chipset in a Sony VAIO Laptop PC vs. the same chipset in a Panasonic Laptop PC). Signal-patterns are affected by both antennas and packaging materials.

Of the several proposed solutions for addressing this hardware variance problem, the most effective is manual adjustment, which was proposed by Haeberlen et al. [5]. For RSS mapping between training and tracking devices, RSS readings are collected from both devices at the same location during the training phase. Experiments conducted by the authors and others [5] [6] show that RSS mapping from a tracking device to a training device exhibits a linear relationship. Hence, RSS mapping can be modeled as a linear function. The difficulty lies is manually identifying the best linear coefficients in a linear function that maps the RSS signal-pattern of the target device to that of the training device. Although manual adjustment improves the positional accuracy of WiFi localization in hardware variance conditions, manually performing all possible combinations of pair-by-pair training for different WiFi training and tracking devices is overly laborious. Further, the ever increasing number of new WiFi chipsets, antennas and encapsulation materials make this manual adjustment approach impractical for real-world deployment.

This study analyzed this hardware variance problem and proposed an unsupervised learning approach for automatically determining a linear transformation function that can map RSS signal-patterns from any unknown tracking device to a training device rather than the manual procedure of exhaustive pair-by-pair training. Further, this work demonstrates that unsupervised learning accurately and efficiently determines these transformation functions.

Some notable contributions of this study are as follows: several unsupervised learning methods developed in this study resolve the WiFi variance problem by accurately and efficiently determining an RSS signal-pattern transformation function. These methods were implemented in a working WiFi positioning system, and performance was evaluated in an actual working environment. The performance of the proposed unsupervised learning method of RSS-based WiFi localization improved positional accuracy by as much as 46%. Additionally, when the tracking and training WiFi devices were identical, applying unsupervised learning to WiFi localization did not reduce positional accuracy.

The remainder of this paper is organized as follows. Section 2 presents the WiFi hardware variance problem by first demonstrating experimental results on the varying RSS signal patterns from different WiFi devices and then formulating the

WiFi hardware variance problem is formulated. Section 3 presents the design and implementation of the proposed unsupervised learning method to address this WiFi hardware variance problem. Section 4 describes the experimental setup and evaluation of the unsupervised learning methods in an actual working environment. Section 5 discusses related work. Section 6 draws conclusions and suggests future studies.

2 Rationale

Experiments were first conducted to identify varying RSS signal-patterns between different WiFi devices. The experimental observations revealed how RSS signal-pattern variations cause positional error in a WiFi localization system. The proposed approach is then formulated based on unsupervised learning to solve the RSS signal-pattern variation problem.

2.1 Signal-Pattern Variations of WiFi Client Devices

Experiments were performed to determine the variation in RSS signal-patterns from different WiFi devices. The test environment was the fifth floor of an office building. Fig. 1 shows the floor plan of the test environment, which was 25 meters by 47 meters and had sixteen WiFi access points (APs). While walking the path indicated by the blue dotted line in Fig. 1, RSS readings for four different WiFi devices were recorded. Table 1 shows the hardware profiles of the four WiFi devices.

Fig. 2 compares RSS readings of different training/tracking device pairs. Each point (RSS-x, RSS-y) on the plots represents RSS readings from two different devices at the same location and from the same WiFi AP. For example, if the RSS readings from the Compaq device are (x1, x2, x3) from three WiFi APs (AP1, AP2, AP3) while the HP device measures RSS of (y1, y2, y3) from the same three WiFi APs (AP1, AP2, AP3) at the same location, three points (x1, y1), (x2, y2), and (x3, y3) are plotted on the upper left graph of Fig. 2. Each of six plots in Fig. 2 is constructed from 500 sample points.

From the RSS trace data shown in Fig. 1, a general pattern of linear shift in RSS readings between two WiFi devices can be observed in all six tested pairs. For example, Fig. 2 shows the Compaq vs. HP RSS mapping relation, which can be approximated by a line with a slope of 0.93 at y-intercept (or “offset”) of 1.20.

This linear shift in RSS signal-patterns was also observed in experiments conducted by Kjaergaard et al. [7] and Haeberlen et al. [5].

Fig. 3 plots the RSS readings of a Sony VAIO SZ18 Laptop PC against Panasonic CFT5 Laptop PC at the same location. Both have the same Intel Centrino WiFi chipset but different antennas and packing materials. The RSS analysis shows that their RSS signal patterns differ with an approximate linear RSS mapping function of a slope (0.92) at an offset (-8.155 dbm).

2.2 Effect of Linear Signal-Pattern Shift on the Accuracy of a WiFi Positioning System

Before describing how this linear shift in RSS signal-patterns affects the positional accuracy of a WiFi positioning engine, the characteristics of a general RSS-based WiFi location system are examined. Such a system consists of two phases. In Phase 1, the offline training phase, a site survey is performed by using a training device to measure RSS signal-patterns from different APs at fixed sampled points in the environment. These RSS readings are encoded as fingerprints and recorded onto a radio map depicting the RSS of APs at different sample location points. Phase 2 is the online estimation phase, in which the location of tracking devices are calculated in real time by matching sample points on the radio map with the RSS fingerprint closest to the tracking device. By considering each RSS fingerprint a vector, the proximity of two RSS fingerprints can be measured by their Euclidean distance [8] [9] or a probabilistic model [11-14].

The example in Fig. 4 shows the effect of RSS signal-pattern variations on the positional accuracy of RSS-based WiFi localization. The blue line (▲) indicates the RSS fingerprint of a training device measured at location *x*. The green line (■) shows the RSS fingerprint of the same training device measured at location *y*. The red line (●) indicates the RSS fingerprint of the tracking device at location *x* tracking. Since the tracking device differs from the training device in RSS signal-patterns, the red (●) fingerprint exhibits a linear shift away from the blue (▲) fingerprint (Fig. 4). By computing their Euclidean distance, the red fingerprint vector is closer to the green fingerprint vector than to the blue fingerprint vector. Thus, the positioning system mistakenly estimates that the tracking device is located at *y* rather than at *x*.

2.3 Signal-Pattern Transformation Function

The proposed solution to the above problem is applying a transformation function on the RSS fingerprint of the tracking device such that the transformed fingerprint (grey dotted line in Fig. 4) are shifted closer to the RSS fingerprint of the training device. Thus, the positioning engine can estimate the tracking location of the tracking device.

This hardware variation problem is formally defined as follows. Denote the training device as Hc (c for calibration) and the tracking device as Ho (o for observed). Since Hc and Ho produce different RSS signal patterns, denoted as Sc and So, the probability of training RSS readings at location l using training device Hc, denoted as P (Sc | Location = l, Hardware = Hc), differs from the that of the tracking device, denoted as P (So | Location = l, Hardware = Ho) . The problem is finding an accurate transformation function F such that applying F to So shifts P (So | Location = l, Hardware = Ho) closer to P (Sc | Location = l, Hardware = Hc) for all li in the tracking space,

$$\underline{S_c} = F(\underline{S_o}) \tag{1}$$

A simple method of determining this transformation function is to survey the site by collecting RSS samples from devices Hc and Ho while manually holding them at each location. Using this training dataset as examples of RSS mappings between tracking and training devices, different learning algorithms are applied to learn a signal-pattern transformation function. From a learning perspective, this manual procedure of collecting training examples is analogous to manual labeling inputs in supervised learning. However, as mentioned in Section 1, this manual data collection is impractical given a large number of WiFi device pairs. Therefore, an unsupervised learning algorithm is needed.

2.4 Unsupervised Learning

The proposed unsupervised learning system automatically learns this signal-pattern transformation function at runtime for any unknown tracking device. The learning procedure consists of the following two general steps.

– The RSS readings from an unknown tracking device are first labeled with a rough location estimation using a correlation ratio computed from the Pearson product-moment correlation coefficient [10] defined below:

$$r = \frac{\sum_{i=1}^{k}\left(s_c^i - \mu(\underline{S_c})\right)(s_o^i - \mu(\underline{S_o}))}{\sqrt{\sum_{i=1}^{k}\left(s_c^i - \mu(\underline{S_c})\right)^2}\sqrt{\sum_{i=1}^{k}(s_o^i - \mu(\underline{S_o}))^2}} \tag{2}$$

The k is the number of APs (or the dimension of RSS fingerprint vectors), sc is the RSS fingerprints of the training device from the radio map, μ is the mean or expected value of a random variable, and so is the RSS fingerprint from the tracking device. The absolute value of Pearson correlation ratio has a value range of (0-1) where 1 indicates the best linear dependency (or greatest similarity between two fingerprint vectors), and 0 indicates complete linearly independency (or least similarity). The Pearson correlation ratio is therefore used to measure similarities in RSS fingerprints between tracking and training devices.

– After labeling the RSS data with rough location estimates, four learning algorithms, including linear regression, two versions of expectation maximization (EM) and neural network, are applied to train the transformation function. These four learning algorithms are explained in further detail in Section 3.

The proposed approach differs from that of Kjaergaard *et al.* [7], which computes an RSS ratio (*i.e.*, the RSS reading of one AP divided by the RSS reading from another AP) to reduce the effect of linear shift in RSS fingerprint matching. Since the Kjaergaard approach only approximates the ratio term but not the offset term in a linear shift function, approximation error is increased when the offset term is relatively large. In comparison, the proposed approach uses the Pearson formula in Equation (2), which captures both the ratio and the offset terms in a linear relationship.

3 Design and Implementation

Fig. 5 shows the system design. In step (1), RSS fingerprints measured from an unknown tracking device are sent to the system running on a location server. In step (2), the RSS fingerprints are transformed using the training-in-progress transformation function. Since the accuracy of the transformation function is iteratively refined, the transformed RSS fingerprints are likely to be inaccurate initially but progressively become more accurate through repetitive training.

In step (3), the WiFi positioning engine performs a lookup on the radio map to find the closest point to the transformed RSS fingerprints and obtains a "rough" location estimate. The positioning engine [18-21] employs a probabilistic method [11-14] with Bayesian inference. A motion model with particle filters [12] is also adopted. Instead of using absolute RSS readings, as stated in the previous section, Pearson's correlation ratio is used as the input.

In step (4), the unsupervised learning labels the rough location estimation as the position of untransformed RSS fingerprints. Finally, in step (5), the labeled data (rough position estimation, untransformed RSS fingerprints) are used as a dataset to iteratively train and refine the signal-pattern transformation function.

3.1 On-Line Regression Algorithm

The linear regression in the first learning algorithm assumes that the regression model for RSS mappings of a tracking device to a training device exhibits a linear relationship, as Fig. 2 shows:

$$RSS_c = b + a * (RSS_o) + \varepsilon \tag{3}$$

The RSS_c and RSS_o represent RSS fingerprints from the training and tracking devices, (a, b) are coefficients in this transformation function, and ε is the error term. The training dataset is determined by RSS readings of the tracking device and are labeled with rough position estimates by Pearson correlation ratio. After collecting sufficient data points, least squares analysis is applied to find the best (a, b) coefficients with the lowest sum of error squares.

3.2 EM Algorithm

An alternative approach to on-line regression is expectation-maximization (EM). Starting with initial estimates of (a, b) coefficients, EM runs an iterative procedure to refine (a, b) estimates by repeating the steps of computing expectation and maximization.

The true location is postulated by combining the original probabilistic location computation [11-14] with Pearson's correlation ratio. Particle filters are used for historical moving pattern adjustments with probability of correlation coefficient for each cell. At each iteration, estimated linear parameters (a, b) are used in the subsequent iteration to transform RSS fingerprints from the tracking device. This iterative procedure continues until a convergence criterion is met.

The output parameters at each iterative step are linear coefficients *(a, b)*.

The $\Theta = (a, b)$ and $\underline{S}_o(t)$ are denoted as RSS readings from the tracking device at time t. The optimization problem is as follows:

$$\arg\max_{\Theta} P(\Theta | \underline{S}_o(t)) = \arg\max_{\Theta} \left\{ \frac{P(\Theta, \underline{S}_o(t))}{P(\underline{S}_o(t))} \right\} \tag{4}$$

The EM is applied to solve the above problem by estimating $P(l_i | \underline{S}_o(t), \Theta(t))$ and maximizing $\Theta(t+1)$, where $l_i \in L$, (*L* for the set of all locations in space) is the latent variable.

3.3 Neural Network Algorithm

To capture any non-linear components of the signal-pattern transformation function, the third neural network learning method was implemented. Similar to online regression, the training dataset for neural network was obtained from RSS readings of the tracking device and labeled with rough position estimates from the Pearson correlation ratio. Radial basis functions [15] were implemented as the neural-network realization with the following form:

$$\varphi(\underline{S}) = \sum_{i=1}^{N} w_i \rho(\|\underline{S} - c_i\|) \tag{5}$$

where *N* is the number of neurons, and S is a single signal strength pattern value observed. The basic function $\rho(\|\underline{S} - c_i\|)$ is Gaussian.

$$\rho(\|\underline{S} - c_i\|) \propto \exp\left[-\beta \|\underline{S} - c_i\|\right] \tag{6}$$

The weights w_i are learned in a gradient descent training manner:

$$w_i(t+1) = w_i(t) + v\left[y(t) - \varphi(\underline{S}(t))\right]\rho(\|\underline{S}(t) - c_i\|) \tag{7}$$

, where *y(t)* is the signal pattern with highest correlation coefficient selected from the training data.

3.4 Extended EM Algorithm

Since both EM and neural networks are limited by local optimum, a fourth method was designed and implemented to extend EM as follows. First, six EM models with different initial seeds are executed in parallel. From the six EM models, the results with the highest probability are used as output.

4 Evaluation

This section describes the experimental procedure and analyzes the performance results of the unsupervised learning system in an actual WiFi localization system and environment.

4.1 Experimental Setup

As Fig. 1 shows, the test environment was the same as in the previous experiment for observing signal-pattern variations from different WiFi devices. The radio map was constructed as follows. First, the test environment was divided into 38*69 cells with each cell occupying a 0.69 meter by 0.69 meter space. The position of tracking devices was estimated by cell granularity. Additionally, to minimize manual calibration, 107 out of 2622 cells were uniformly selected as training cells. Thirty RSS samples were collected from each training cell. The RSS samples from other unselected cells were interpolated with those of the training cells. Interpolation was intended to reflect the common practice of minimizing manual calibration in deployments over a wide coverage area, even though interpolation reduces positional accuracy [17].

4.2 Positional Accuracy

Table 1 shows the four WiFi devices used in the experiments. The Compaq device was selected as the training device for calibrating the radio map. During the online estimation phase, a user carried each of these four WiFi devices while walking at a constant speed along a path indicated by the blue dotted line in Fig. 1. The RSS readings along this walking path were collected for use in the training dataset for each of the four learning algorithms (i.e., online regression, EM, neural network and extended EM) to train the signal-pattern transformation function.

Fig. 6 plots the mean positional error when applying each of four learning algorithms to each of four different WiFi devices tracking. Fig. 6 also plots the mean positional error for manual adjustment and non-adjustment. Manual adjustment re-implements the method developed by Misikangas et al. [22], which provides near-optimal performance. Non-adjustment means no transformation function is applied for WiFi localization; therefore, its performance is used as a baseline for measuring improvements in the four learning algorithms.

The results in Fig. 6 reveal the following findings: (1) the EM produced the least average positional error of the four tracking devices, and its performance approached that of the near-optimal manual adjustment; (2) except for the Orinoco device, which had an RSS signal-pattern similar to that of the Compaq training device, all four learning algorithms achieved less positional error than non-adjustment. This suggests that applying the unsupervised learning to RSS-based WiFi localization effectively reduced positional error due to hardware variance. (3) Under some training/tracking device pairs, EM outperformed the near-optimal manual adjustment. One explanation is that the unsupervised system adjusted dynamic changes in environmental factors (*e.g.,* humidity levels, open/closed doors, *etc*.) affecting signal patterns.

4.3 Training Time

In EM and neural networks, training time is important for determining a signal-pattern transformation function for an unknown tracking device. During the learning phase, positional estimates are unreliable.

The training process is complete when the changes in the linear parameters of a transformation function undergoing training fall within a limited range. However, the experiments revealed that function parameters rarely converge, but the output of transformation functions often do. The reason is that several solutions to function parameters may co-exist when the input training dataset are concentrated on a small segment of the function where the RSS readings fall between -90 *dbm* and -30 *dbm.* To address this problem, convergent criteria was determined by the change in function output rather than by the change in trained input parameters. The convergent criteria used the near-optimal output from manual adjustment (as determined during post-processing) as a baseline to determine the speed of the proposed runtime learning algorithms (EM or Neural Network) which then stabilized at the near-optimal output. Specifically, as Fig. 7 shows, the training curves were constructed using the near-optimal output differences and the following learning algorithm

$$|\, output_{manual\ adjustment} - output_{learning\_algorithm} \,| \qquad (8)$$

, where the tested learning algorithms were EM (dotted line) and neural network (solid line), and the training/tracking devices were Compaq/HP. The training times of both the EM and neural network were under 100 seconds. Although the

EM training time was slightly longer than that of the Neural network, EM training error was smaller (1.8 *dbm* vs. 4.2 *dbm*).
Whether or not the 100-second training time is sufficient depends on the specific application. For example, this training time may be acceptable for a location-based museum tour guide but may be too long for locating an emergency call.

4.4 Case study: the Orinoco as the training device

After using the Compaq computer as the training device in the above tests, the unsupervised learning system was tested using other training devices. Table 2 shows the amount and percentage of positional error reduction when the Orinoco was used as the training device.
The experiments indicated that the IBM device produced the strongest RSS readings followed by the Orinoco and Compaq devices, which had RSS readings similar in strength and finally the HP device, which recorded the weakest strength. Since the difference in RSS signal-patterns between Compaq (the tracking device) and Orinoco (the training device) was small, the 0.31 meter positional error reduction was also small. Conversely, the difference in RSS signal-patterns was large for Orinoco-IBM and Orinoco-HP pairs, and positional error reduction was large (1.66 meters and 1.96 meters, respectively). Table 2 shows that, given a larger RSS signal-pattern difference between tracking/training devices, the proposed system generally achieved higher positional error reduction. However, one exception is discussed in Section 4.7 below.

4.5 Case study: similarity between tracking/training devices

This case study investigated whether unsupervised learning on the same training/tracking device pairs degrades positional accuracy given that learning and applying transformation functions to the same device pairs is unnecessary. Table 3 shows the positional error for Orinoco and Compaq training and tracking devices, respectively, with and without using unsupervised learning. No degradation of positional accuracy was observed. Surprisingly, a slight improvement was observed (i.e., the positional error was reduced from 2.66 meters to 2.40 meters for the Orinoco devices and from 2.40 meters to 2.08 meters for the Compaq devices). A possible explanation is that, since training was performed one day before testing, the unsupervised system also adjusted to changes in environmental factors (e.g., humidity levels, open/closed doors, etc.) affecting WiFi signal-patterns.

4.6 Case study: variable-speed vs. constant-speed movement

Since variable-speed movement is common in real-world scenarios, this case study compared the effectiveness of unsupervised learning between constant-speed and variable-speed movements. The variable-speed trace was collected as follows. Fig. 8 shows several walking segments marked with arrowed lines. Each walking segment has a unique movement speed whereas the speed within each walking segment is constant.

Since most WiFi localization systems adopt a motion model that assumes constant-speed movement, variable-speed movement often increases positional error. Table 4 shows positional error after applying different learning methods to WiFi localization in two cases involving constant-speed movement and variable-speed movement. The analytical results in Table 4 show that the unsupervised learning method is effective for both constant-speed and variable-speed movement.

4.7 Relationship between RSS signal-pattern difference and positional error

Intuitively, since increased difference in RSS signal-patterns between tracking and training devices produces larger difference between the RSS fingerprint of the tracking device and the RSS fingerprint on the radio map constructed by training devices, the positional error in WiFi localization should be larger. Although this relationship between the RSS signal-pattern difference and the positional error seems reasonable, the experiments in this study proved otherwise.

Fig. 9 plots this relationship between the positional error and the RSS signal-pattern variance on several training/tracking device pairs. The x-axis measures the percentage of the average AP RSS readings between the training and tracking devices, which is computed as follows,

$$\frac{\left| avg(RSS_c - avg(RSS_o) \right|}{avg(RSS_o)} \tag{9}$$

The *y*-axis measures the average percentage of increased positional error due to device hardware variance, which is computed as the difference between the positional error without hardware variance (*i.e.*, using the same device for tracking and training) and the positional error with hardware variance. The analytical results in Fig. 9 show no correlation that greater RSS signal-pattern variance increases positioning error.

Several factors such as AP spatial distribution can enlarge or mitigate the effect of RSS signal-pattern variance on positional error. An interesting situation in which a large RSS signal-pattern variance produces negligible positional error is the following. This situation arises under two conditions: (1) the RSS readings between tracking/training devices differ only in offset but not in ratio and (2) the spatial distribution of APs is balanced or evenly distributed relative to the position of the tracking device. Fig. 11 shows an example of balanced distribution of APs, which are marked in red circles, relative to the location of a tracking device, which is moving within the blue dotted rectangular area. The distribution is balanced to the tracking device in that the APs to its right and left are more or less equal in number and distance. Appendix A describes an analytical model showing how this balanced AP distribution mitigates the effect of RSS signal-pattern variation on positional accuracy.

5 Related Works

Hardware variance can be minimized by not relying on RSS information to implement WiFi positioning systems. For example, Cheng et al. [23] developed a method of ranking (i.e., high to low) RSS values instead of relying on absolute RSS values. Ranking overcomes the hardware variance problem because a linear shift in RSS does not affect their ranking. However, the ranking approach sacrifices some positional accuracy because detailed data for signal strength levels is not used.

This hardware variance problem was addressed by Misikangas et al. [22], who proposed a manual approach based on pair-wise mapping. In this approach, different hardware devices are placed at the same position at the same time to differentiate their signal patterns; their mapping functions are then derived. Haeberlen et al. [5] proposed a similar manual solution. Their experiments demonstrated a linear RSS mapping function between different WiFi devices. They then used the least-squares fit method to determine the coefficients in the linear transformation function. However, they did not apply the signal-pattern transformation function in a positioning engine to demonstrate improved positional accuracy. Although manual adjustment can achieve good positional accuracy, its main problem is the required manual labor, which does not scale well with the size of the environment and the wide array of emerging WiFi devices.

Misikangas et al. [22] described an automatic approach for solving this hardware variance problem. Firstly, available hardware pairs were obtained manually. Misikangas then assumed the existence of an easily distinguishable location (i.e., a location with a unique RSS fingerprint) such as the entrance to a room. When an unknown WiFi device moves through this location, its signal patterns are collected, and a small set of calibrated-observed variant signal pairs are produced. Instead of computing the transformation from this limited information, the best-fit transformation identified in the manually compiled transformation database is assigned to this unknown device. However, an easily distinguishable location is required, which may only exist in certain environments. Additionally, a new WiFi device may never pass through these easily distinguishable locations or may not pass through them early enough to learn its transformation function. Haeberlen et al. [5] also explored fully automatic calibration. They suggested EM and particle filtering to learn coefficients. However, no implementation or experiments were mentioned. Kjærgaard et al. [7] proposed an automatic method using RSS ratios when matching closeness in the RSS fingerprints. Although using the RSS ratios reduces the linear shift effect in RSS fingerprints caused by hardware variance, it can still produce errors (as described in Section 2) when the offset component is in the linear shift is large.

6 Conclusion & Future work

This work presents an unsupervised learning method for solving the hardware variance problem in WiFi localization. At runtime, the unsupervised learning automatically learns a transformation function for mapping WiFi signal-patterns from an unknown tracking device to a training device under which the radio map is calibrated. Several learning algorithms, including online regression, EM, neural network and extended EM were designed, implemented and evaluated in a working WiFi localization system and environment. The experimental results demonstrated that, in WiFi localization, applying a transformation function learned from our unsupervised learning reduces position error caused by device hardware variance by an average of 4%. Additional case study experiments showed that (1) positional accuracy in the same training/tracking device pairs did not degrade, and (2) unsupervised learning was effective for both variable-speed and constant-speed movement. Finally, RSS signal-pattern variance revealed no correlation with positioning error.

Several questions remain for future studies in unsupervised learning. First, this technique could be applicable to other RSS-based localization systems such as Zigbee, WiMax and GSM, in reducing positional error caused by hardware variance. Second, this unsupervised learning technique could also be applicable in adapting WiFi localization to dynamic environment factors affecting WiFi signal-patterns such as humidity level, human presence, open/closed doors, etc.

References

1. "Ekahau Corporate Website." Internet: http://www.ekahau.com/ [Feb. 20, 2008]

2. "Aeroscout Corporate Website" Internet: http://www.aeroscout.com/ [Feb. 20, 2008]

3. "Pango Product Website." Internet: http://www.innerwireless.com/vision-over.asp [Feb. 20, 2008]

4. "Skyhook Wireless Corporate Website" Internet: http://www.skyhookwireless.com/ [Feb. 20, 2008]

5. A. Haeberlen, E. Flannery, A. M. Ladd, A. Rudys, D. S. Wallach, andL. E. Kavraki, "Practical robust localization over large-scale 802.11 wireless networks," in *Proceedings of ACM MOBICOM,* 2004.

6. P. Tao, A. Rudys, A. M. Ladd, and D. S. Wallach. "Wireless LAN location–sensing for security applications", In *Proceedings of the AAAI Thirteenth National Conference on Artificial Intelligence, Aug.* 1996

7. M. Kjaergaard, C. Munk, "Solving RSS Client Differences by Hyperbolic Location Fingerprinting," In *Proceedings of IEEE International Conference on Pervasive Computing (PERCOM 2008)*, 2008.

8. P. Bahl, V. N. Padmanabhan, "RADAR: An In-Building RF-based User Location and Tracking System," In *Proceedings of IEEE Infocom,* 2000

9. P. Bahl, V. N. Padmanabhan, and A. Balachandran, "Enhancements to the RADAR User Location and Tracking System," *Microsoft Research Technical Report: MSR-TR-00-12*, (February 2000)

10. Pearson, K. "Mathematical contribution to the theory of evolution VII: On the correlation of characters not quantitatively measurable." *Philosophical Transactions of the Royal Society, Series A*, 1900,195, 1–47.

11. V. Seshadri, G. V. Zaruba, and M. Huber. “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)*, 2005.
12. J. Hightower, and G. Borriello, “Particle Filters for Location Estimation in Ubiquitous Computing: A Case Study,” In *Proceedings of Ubiquitous Compuating,*” 2004.
13. D. Schulz, D. Fox, and J. Hightower, “People Tracking with Anonymous and ID-Sensors using Rao-Blackwellised Particle Filters”, In *Proceedings of IJCAI*, 2003, pp. 921-926.
14. D. Fox, J. Hightower, L. Liao, D. Schulz, and G. Borriello, “Bayesian Filtering for Location Estimation,” In *Procddings of IEEE Pervasive Computing*, 2003, pp. 24-33
15. T. Michell. *Machine Learning.* McGraw Hill, 1993, pp. 239-240.
16. El-Gallad A., El-Hawary M., and Sallam A., “Swarming of intelligent particles for solving nonlinear constrained optimization problem”, *Intemational Journal of Engineering intelligent systems,* Vol. 9, No. 3*,* pp.155-163, September 2001.
17. P. Krishnan, A. 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 *Proceedings of IEEE INFOCOM*, 2004.
18. Yi-chao Chen, Ji-rung Chiang, Hao-hua Chu, Polly Huang, Arvin Wen Tsui, "Sensor-assisted Wi-Fi indoor location system for adapting to environmental dynamics," in *Proceedings of ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM 2005)*, 2005, pages 118-125
19. Tsung-han Lin, Ju-peng Chen, Hsing-hau Chen, Polly Huang, Hao-hua Chu, "Enabling energy-efficient and quality localization services," in *Proceedings of the IEEE International Conference on Pervasive Computer and Communications (PERCOM 2006), Work In Progress Session*, 2006.
20. Li-wei Chan, Ji-rung Chiang, Yi-chao Chen, Chia-nan Ke, Jane Hsu, Hao-hua Chu, "Collaborative localization - enhancing WiFi-based position estimation with neighborhood links in clusters," in *Proceedings of the International conference on Pervasive Computing (PERVASIVE 2006)*, 2006, pages 50-66
21. Chuang-wen You, Yi-Chao Chen, Ji-Rung Chiang, Polly Huang, Hao-hua Chu, Seng-Yong Lau, "Sensor-enhanced mobility prediction for energy-efficient localization," in *Proceedings of Third Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON 2006)*, 2006.
22. P. Misikangas, L. Lekman, “Applicatioins of Signal Quality Observations,” Patent Cooperation Treaty (PCT) WO 2004/008796 A1, Jul. 8, 2003
23. Y.C. Cheng, Y. Chawathe, A. LaMarca, J. Krumm “Accuracy Characterization for Metropolitan-scale Wi-Fi Localization,” in *Proceedings of the Third International Conference on Mobile Systems, Applications and Services (MobiSys)*, 2005.
24. S. Ray, R. Ungrangsi, F. D. Pellegrini, A. Trachtenberg, and D. Starobinski, “Robust location detection in emergency sensor networks,” in *Proceedings of IEEE INFOCOM*, 2003.
25. J. Hightower, G. Borriello, “Location systems for ubiquitous computing,” *IEEE Computer*, vol. 34, pp. 57-66, August 2001

26. P. Castro, P. Chiu, T. Kremenek, and R. Muntz, “A probabilistic roomlocation service for wireless networked environments,” in *Proceedingsof Ubiquitous Computing*, 2001.

Table 1. WiFi client devices and hardware profiles

Abbreviations	**WiFi Client Devices**
IBM	IBM Notebook PC with Intel PRO/Wireless 2915 ABG
Compaq	Compaq WL110 Wireless LAN PC Card (Attacted to Fujitsu Tablet PC)
Orinoco	Orinoco Wireless LAN PC Card (Attached to Fujitsu Tablet PC)
HP	HP iPAQ RW6828 PDA with built-in WiFi

Table 2. positional error reduction when using orinico as the training device

Tracked device	**Positional error reduction (meters)**	**Positional error reduction (%)**
Compaq	0.31	13.13
IBM	1.66	46.51
HP	1.96	44.41

Table 3. positional error reduction in the same training/tracking device pair with/without Learning

	Positional error without learning (meters)	**Positional error with EM learning (meters)**
OriNoco	2.66	2.40
Compaq	2.40	2.08

Table 4. Positional error under Constant-Speed Movement and Variable-Speed Movement

	EM	**Neural network**	**Modified EM**	**Online regression**
Constant-speed motions	2.46	3.75	3.51	3.50
Variable-speed motions	2.37	3.82	2.48	3.11

Fig. 1. Sixteen WiFi APs (red circles), were distributed throughout the 25 meters x 47 meters test environment. The movement of the tracking device is marked with a blue dotted line.

Fig. 2. Correlation of Serveral hardware pairs

Fig. 3. RSS signal-patterns from Sony VAIO and Panasonic Laptop PCs with the same Intel Centrino WiFi Chip

Fig. 4. Example of error in positional estimation caused by RSS signal-pattern variation in RSS-based WiFi localization.

Fig. 5. System Design

Fig. 6. Resulting positional error (measured in meters) after applying four different learning algorithms (Online regression, EM, Neural network and Modified EM) to four different tracking devices (Orinoco, HP, IBM and Compaq). The Compaq was the training device.

Fig. 7. Covergent time Comparison

Fig. 8. Moving testing trace

Fig. 9. RSSI Difference vs. Decreased Positional Accuracy

Fig. 10. A balanced AP distribution that mitigates the effect of RSS signal-pattern variation on positional accuracy. The APs are marked in red circles. The tracking device is moving within the blue rectangular area.

Fig. 11. Highest Probability Distribution at Location U_0

Fig. 12. Highest Probability Distribution at Location U_0 Shifted by Hardware Difference with Linearity of $y=x+b$

APPENDIX A: analytical model for the balanced AP distribution

A typical location in an RSS fingerprint localization system is characterized by several (RSS, AP) pairs, where RSS is not a single value but rather a distribution of signals collected from the training phase and often modeled by Gaussian distribution. While tracking a device, the probability of a set of observed (RSS, AP) pairs against a certain location is then computed by multiplying all the probabilities acquired from the previously modeled Gaussian probability distribution function. The location with the highest joint probability is the output result.

The above localization system is assumed here. Further, without loss of generality, the following assumptions are made:

1. RSS decays linearly
2. The variances of all pairs are identical

Consider the one-dimensional example in Fig. 11. Two access points AP_a and AP_b are at either side of a tracking device. Suppose the tracking device is at an arbitrary position u_0 on the line from 0 (the leftmost position) to z (the rightmost position).

According to the first assumption above, if the RSS directly beneath an access point is s, the distributions of $(RSS=s/u_0, AP= AP_a)$ and $(RSS=s/(z-u_0), AP= AP_b)$ at position u_0 would be identical, and u_a and u_b would be located at u_0. If the above two RSS signal-patterns are entered into an RSS-based positioning engine, the estimated location would be u_0.

As Fig. 12 shows, if the tracking device differs from the training device with a linear RSS mapping function with slope = 1 as in the first assumption, the RSS distribution is simply shifted .

Although u_0 is no longer the [location with the highest probability OR SIMPLY most probable location] for both APs, the multiplied probability is still the highest. This outcome is demonstrated by comparing the multiplied probability at each position. Since the RSS variances are assumedly identical, in the p.d.f. of Gaussian distribution, $\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$, we need only compare the $(x - u)^2$ part. Restated, the smaller the value, the higher the probability.

At u_0, after multiplication, the next procedure would be

$$[u_0-(u_0-b)]^2+[u_0-(u_0+b)]^2=2b^2$$

Assume an arbitrary position denoted as (u_0+d) in Fig. 12. After substitution, $(x - u)^2$ becomes

$$[(u_0+d)-(u_0-b)]^2+[(u_0+d)-(u_0+b)]^2=2(d^2+b^2)$$

Since $2(d_2 + b_2) > 2b_2$ for any nonzero d, u_0 would be the location with maximum probability.

Fig. 1
Click here to download high resolution image

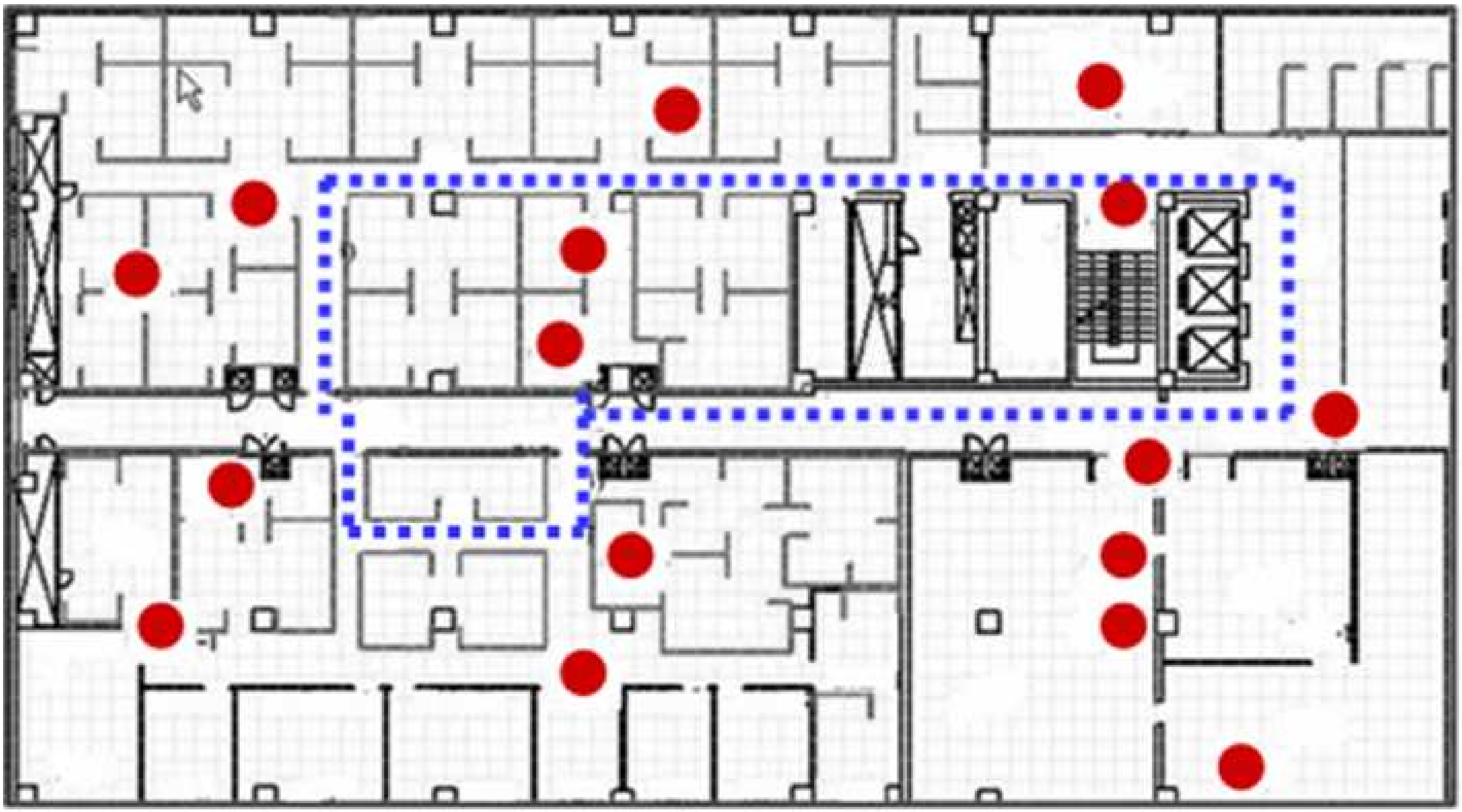

Fig. 2
Click here to download high resolution image

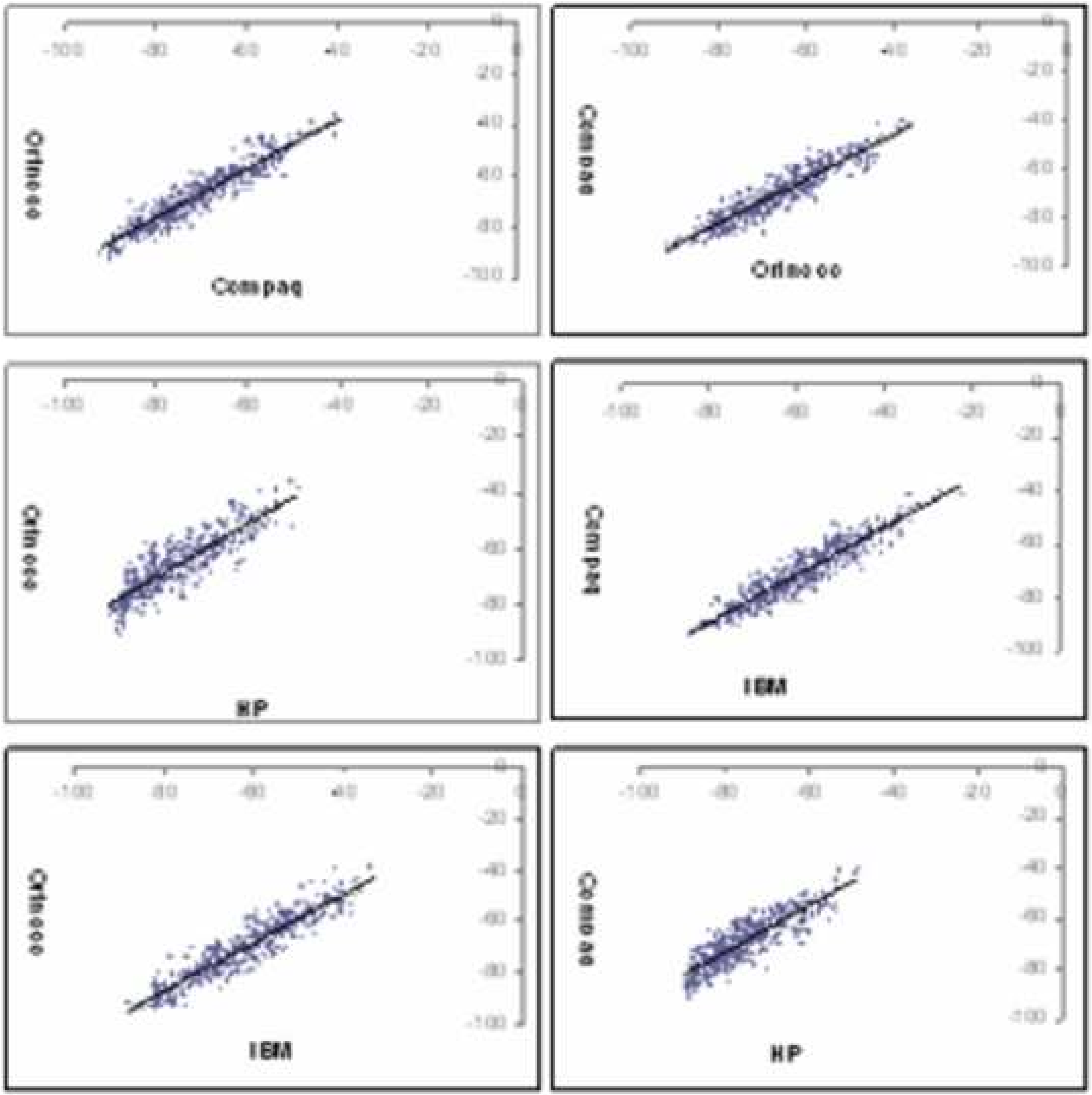

Fig. 3
Click here to download high resolution image

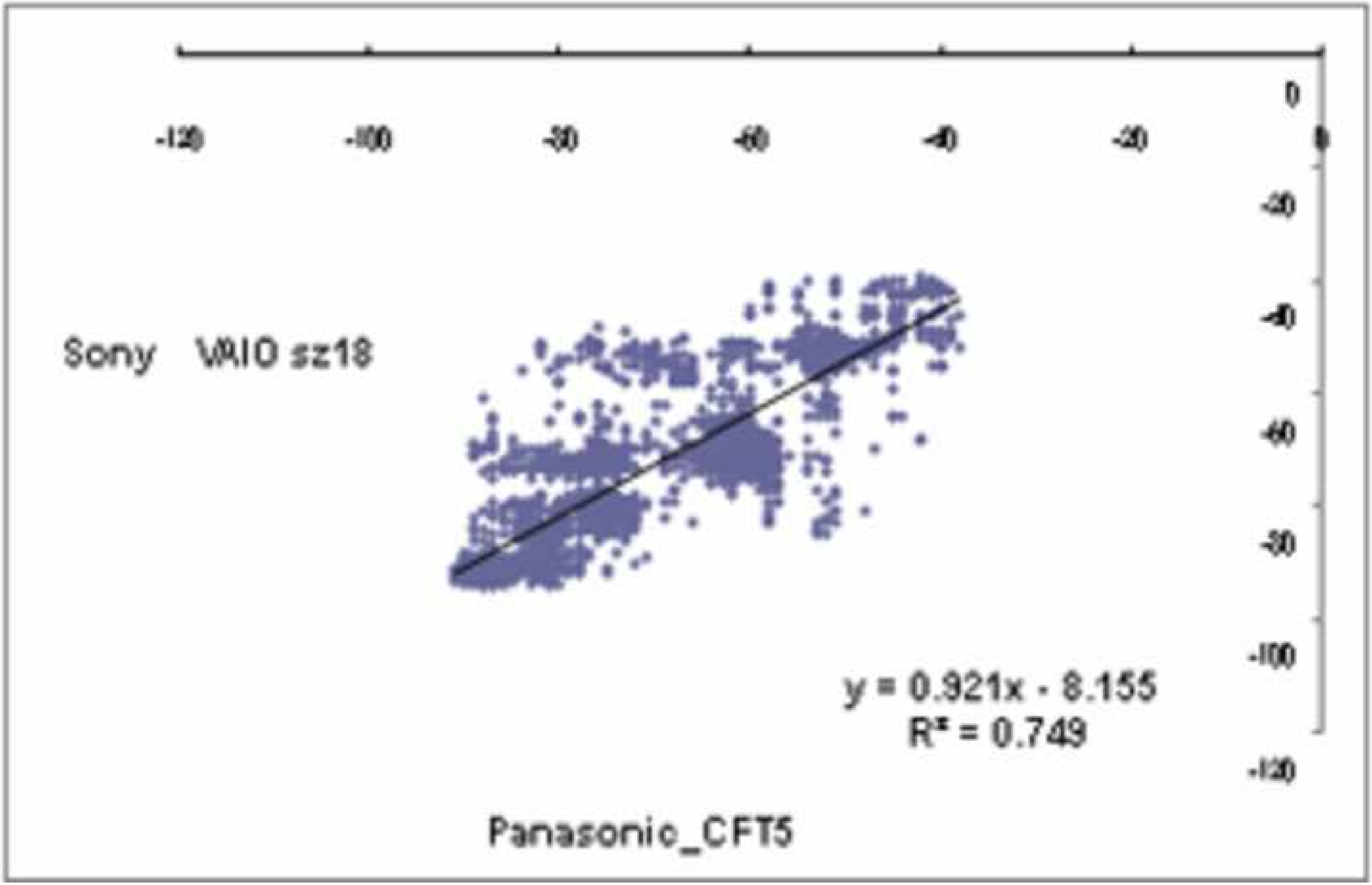

Fig. 4
Click here to download high resolution image

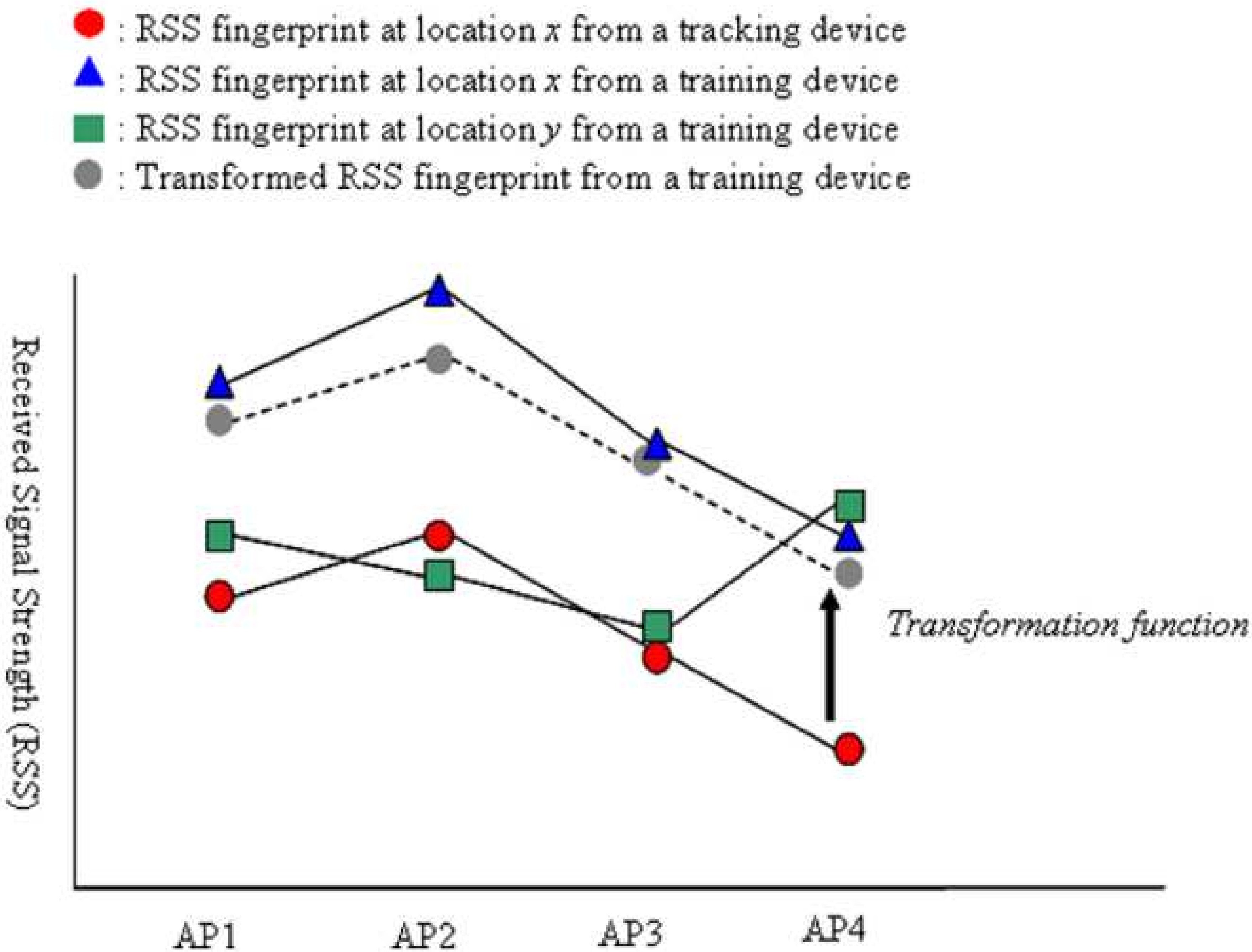

Fig. 5
Click here to download high resolution image

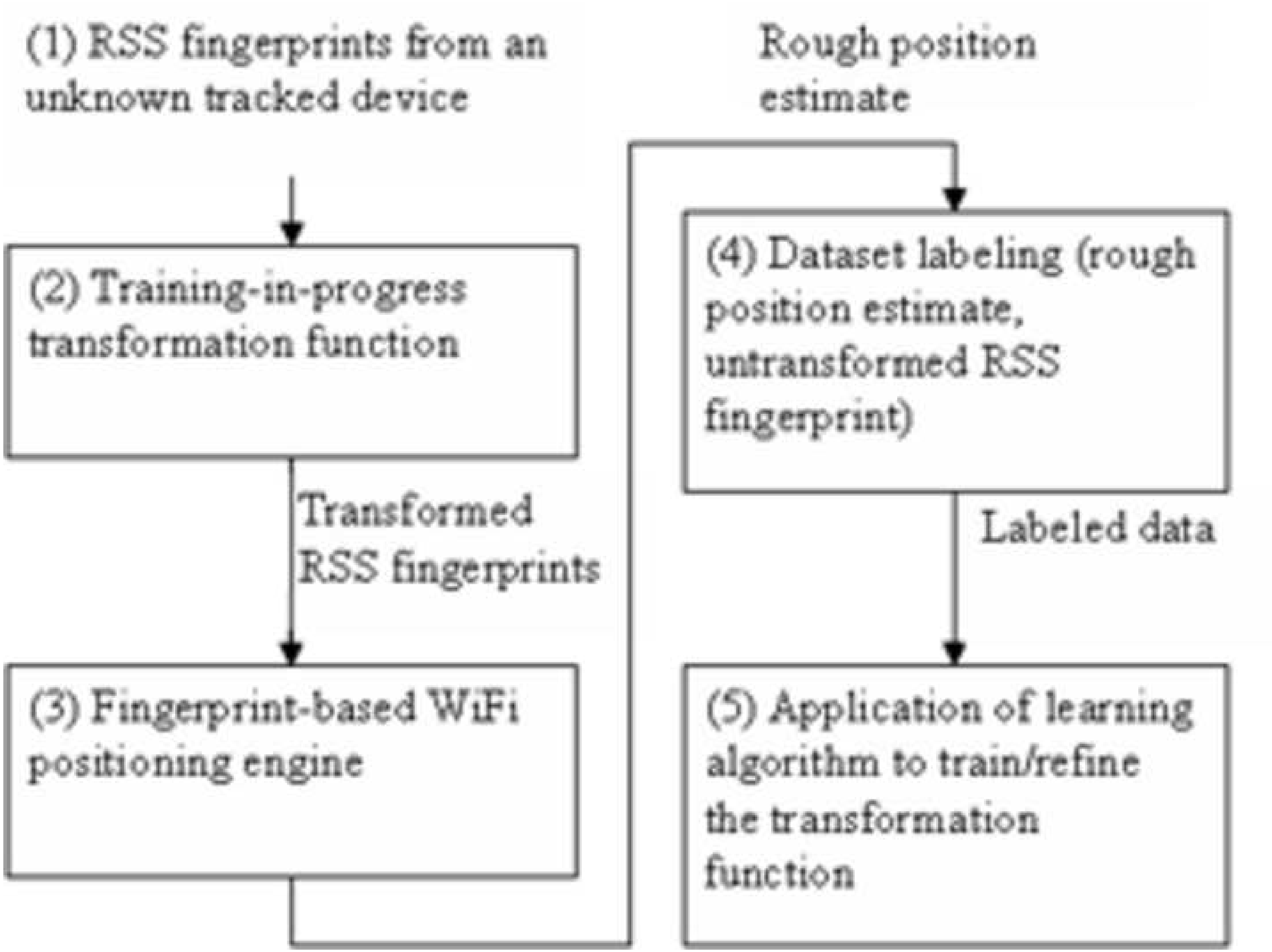

Fig. 6
Click here to download high resolution image

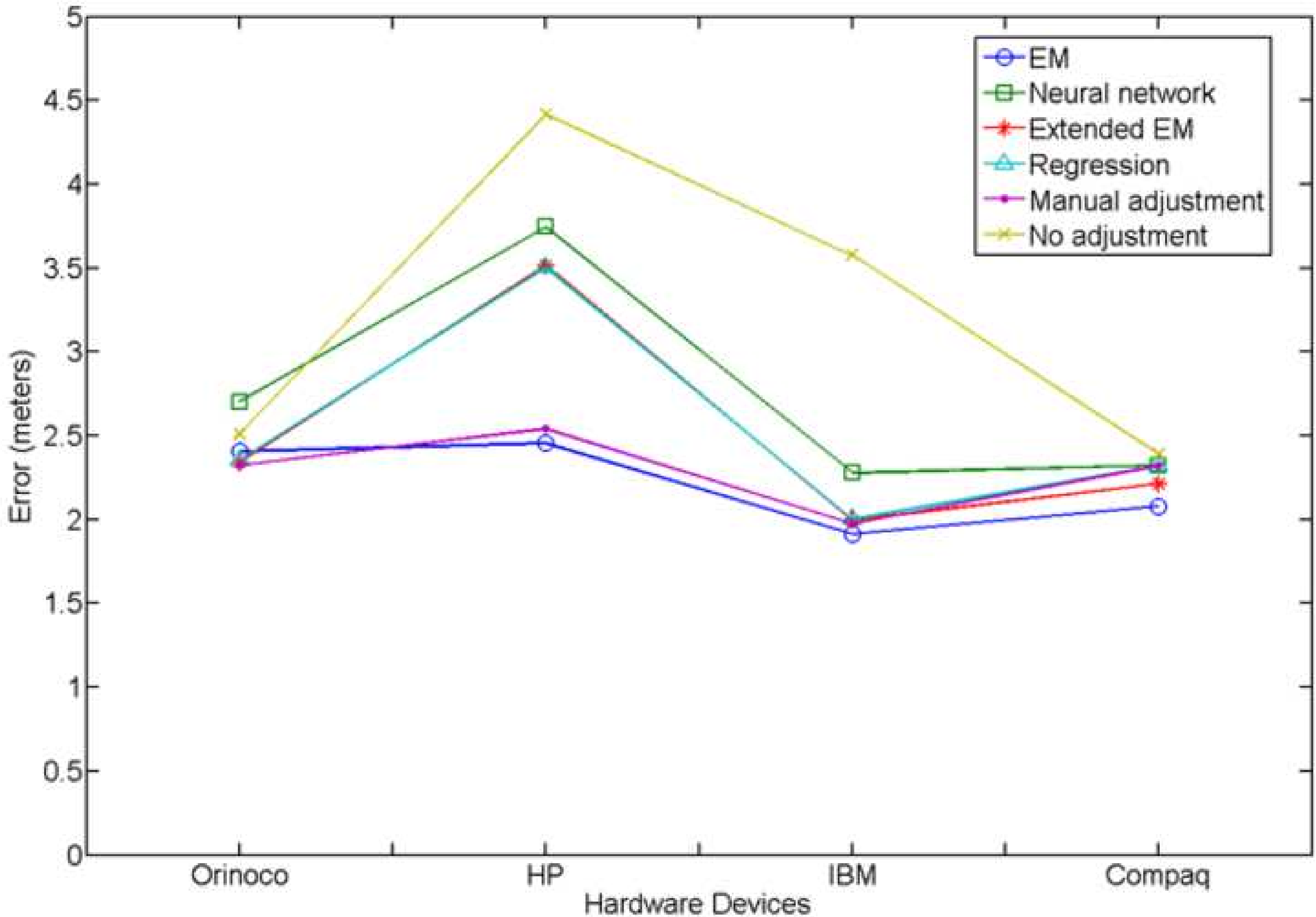

Fig. 7
Click here to download high resolution image

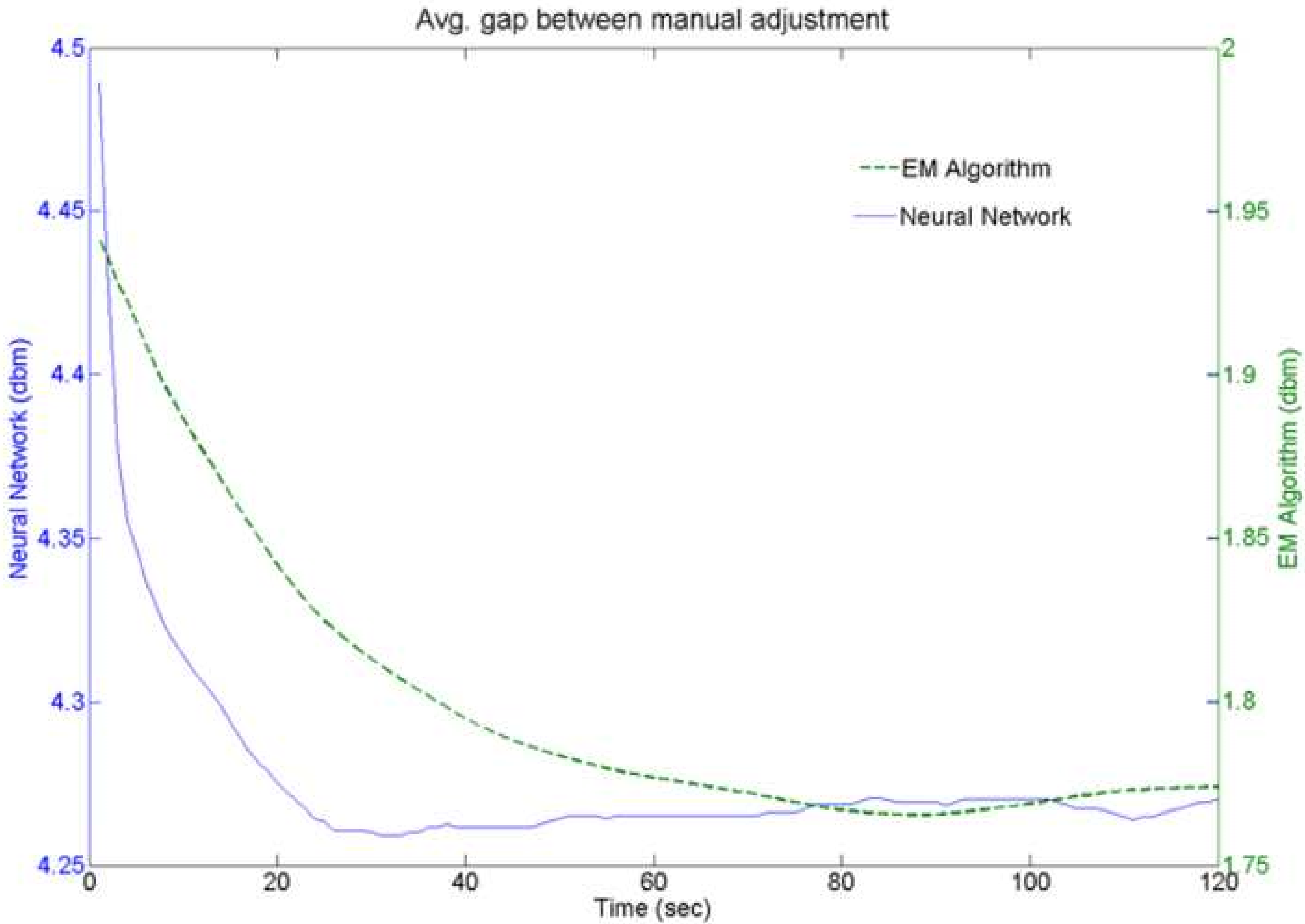

Fig. 8

Click here to download high resolution image

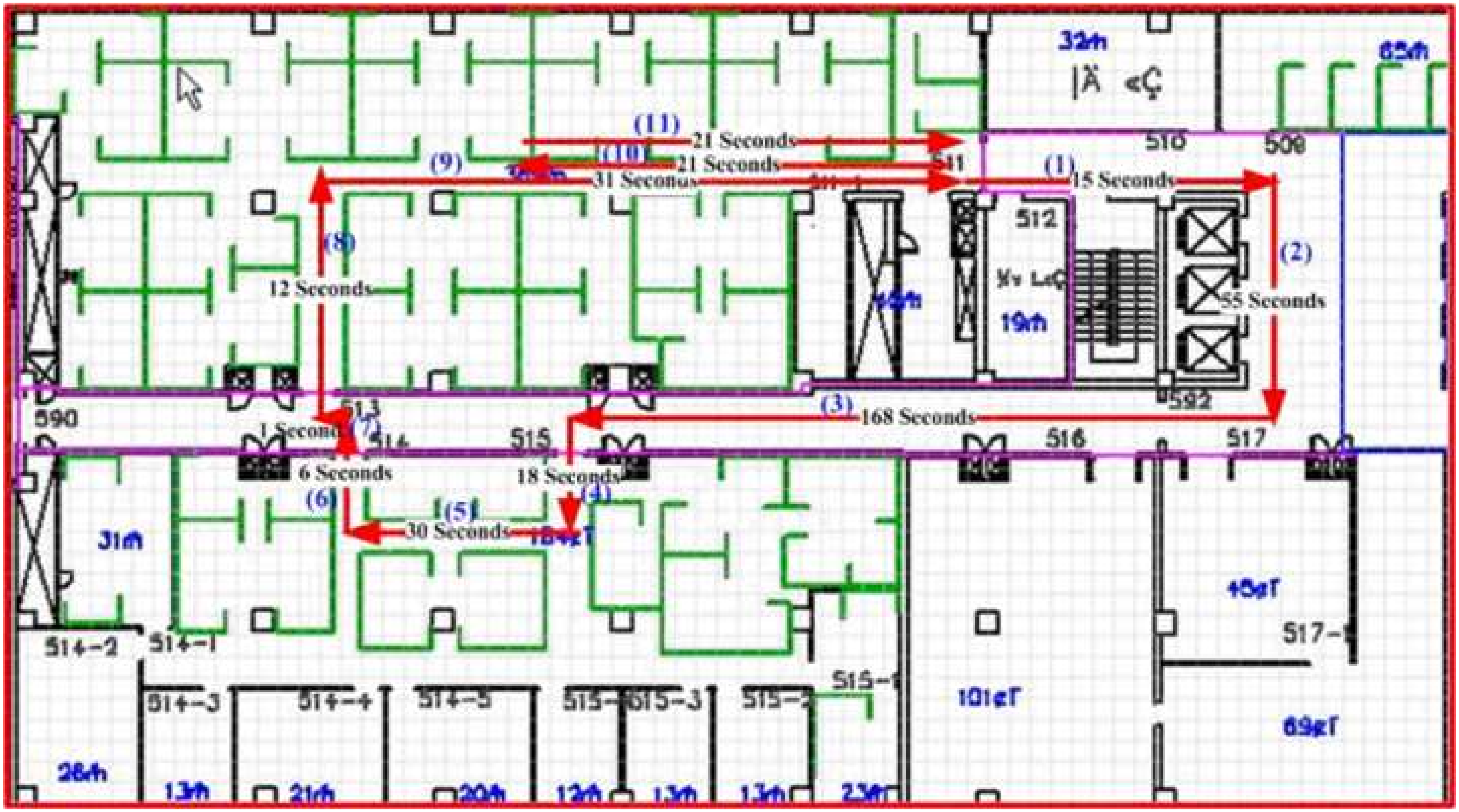

Fig. 9
Click here to download high resolution image

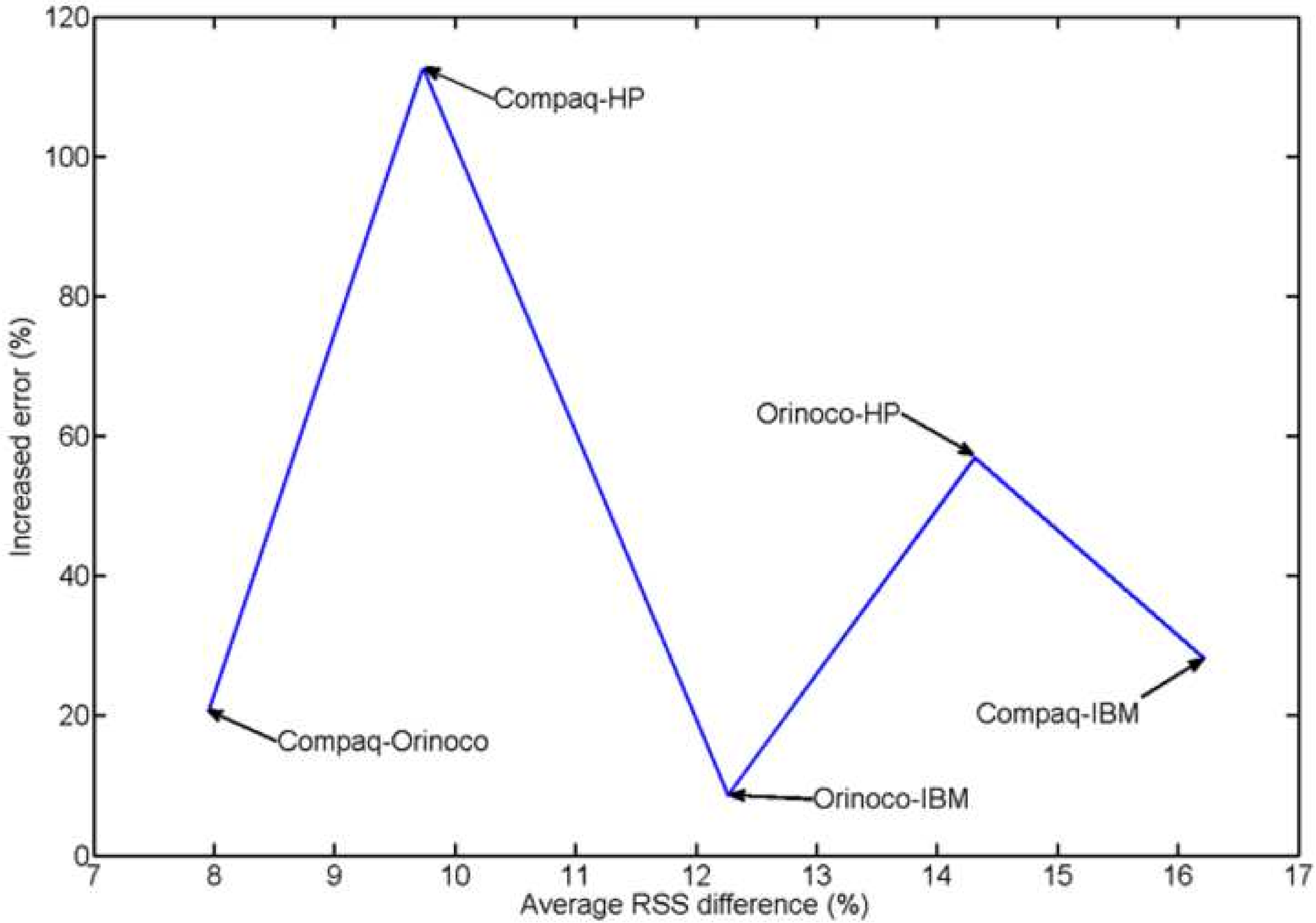

Fig. 10

Click here to download high resolution image

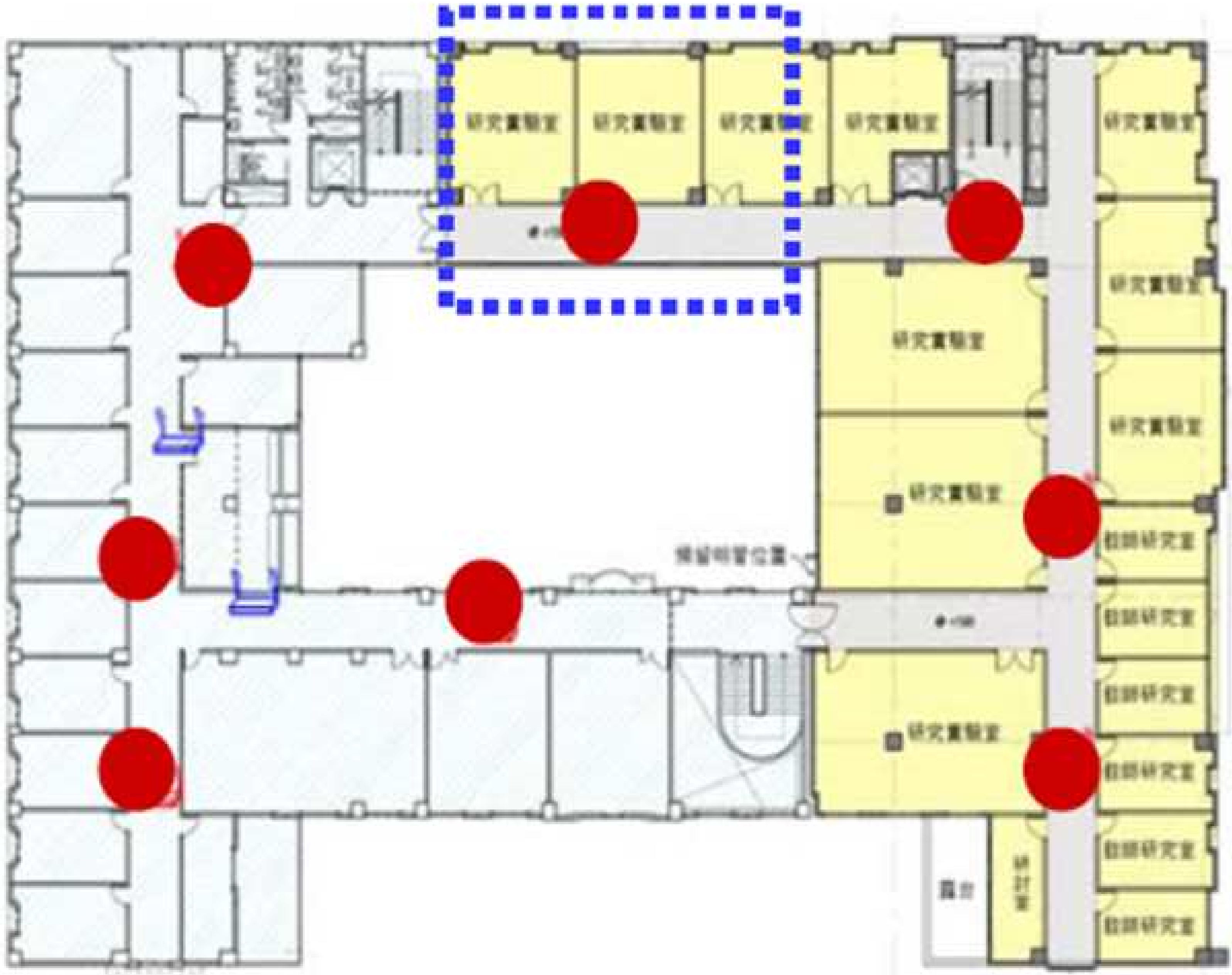

Fig. 11

Click here to download high resolution image

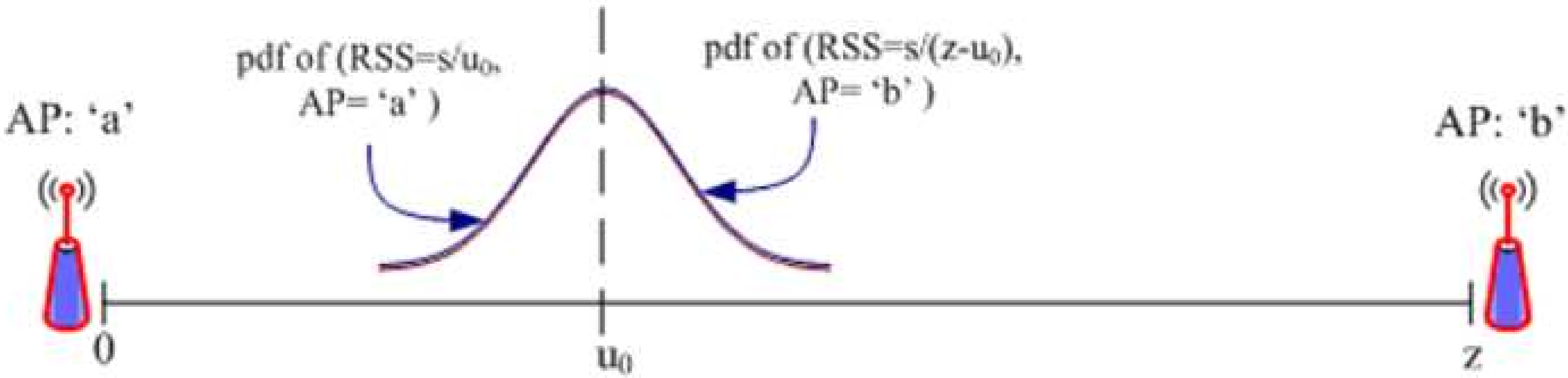

Fig. 12
Click here to download high resolution image