This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING



# A Scrambling Method for Fingerprint Positioning Based on Temporal Diversity and Spatial Dependency

Sheng-Po Kuo and Yu-Chee Tseng
Department of Computer Science,
National Chiao Tung University, Hsin-Chu, 30050, Taiwan

***Abstract*— Signal strength fluctuation is one of the major problems in a fingerprint-based localization system. To alleviate this problem, we propose a *scrambling method* to exploit *temporal diversity* and *spatial dependency* of collected signal samples. We present how to apply these properties to enhance the positioning accuracy of several existing schemes. Simulation studies and experimental results show that the scrambling method can greatly improve positioning accuracy, especially when the tracked object has some degree of mobility.**

***Index Terms*— Context Awareness, Location-Based Service, Pervasive Computing, Sensor Network, Indoor Positioning, Location Tracking.**

## I. INTRODUCTION

Recently, location-based applications are regarded as one of the most important services in wireless networks [16], [8]. Location tracking is critical to support location-based services. Although GPS [6] has been widely used, indoor localization is still a challenging problem. Localization models can be classified into *angle of arrival (AoA)* [10], *time of arrival (ToA)* [1], *time difference of arrival (TDoA)* [14], and *fingerprint* [2], [13], [15], [3], [4], [11], [9]. In this work, we are interested in fingerprint-based localization systems, such as RADAR [2]. Unlike other propagation-based localization methods, the fingerprint method does not rely on calculating signal fading in an environment but relies on a training phase to learn the signal strength patterns at a set of training locations from pre-deployed beacons in a sensing field. These beacons can be existing infrastructures, such as IEEE 802.11 access points. To position an object, we will compare its received signal strengths (RSSs) against those of the training locations.

However, because signal fluctuation is inherent to RF systems, fingerprint schemes have their limitation to positioning accuracy. To conquer this problem, [13] presents a probabilistic framework to handle uncertainty in signal strength measurement. Signal variations are modeled by probability distributions. Based on the similar concept, [15] uses recursive *Bayesian filters* for localization. Reference [3] adopts a *neural network* model, which is a multi-layer structure with a number of interconnected neurons, to implement its positioning algorithm. This model has a forward and backward propagation mechanism to adaptively assign suitable weights for the connections based on the training samples. Then, this network can take a number of real-valued numbers as inputs and generate a number of real-valued numbers for the neurons in the output layer. Finally, at the positioning stage, RSSs can be fed into the network and the location whose representative neuron has the highest output is the estimated location.

Although the fingerprint-based approach is very similar to the traditional classification problem [7], there are still some properties particular to the localization problem that have not been well exploited. In this paper, we point out that the observed RSSs may have some degrees of *temporal diversity* and *spatial dependency*. Thus, we propose a *scrambling method* to take advantage of these properties. Based on temporal diversity, its basic idea is to enlarge the comparison space by recombining observed samples in a short period of time. Through scrambling, samples with less interference are expected to appear. Then, spatial dependency will exploit the moving trajectories of objects to select a better location estimation. This scrambling method should be considered an enhancement to existing schemes, instead of a new scheme. We will show how to integrate it with two methods [2], [13]. The contributions of this work are three-fold. First, it is the first work exploiting the temporal diversity and spatial dependency simultaneously. Second, it handles signal fluctuation well in indoor environments. Third, it greatly improves positioning accuracy, especially when the tracked object has some degree of mobility. Our simulation and experimental results do support these claims.

## II. THE PROPOSED SIGNAL SCRAMBLING METHOD

A fingerprint-based localization system generally works as follows. We are given a set of *beacons* $B = \{b_1, ..., b_n\}$ in a field, which are capable of transmitting radio signals, and a set of training locations $\mathscr{L} = \{\ell_1, ..., \ell_m\}$. At each training location $\ell_i$, we measure the signal strengths from beacons for a period of time and create a *characteristic vector* $\boldsymbol{c_i} = \langle c^i_1, c^i_2, \ldots, c^i_n \rangle$ in a location database, where $c^i_j$ is derived from the received signal strengths of $b_j$, $j = 1, \ldots, n$. When an object moves into the field, it also measures its received signal strength vector $\boldsymbol{s} = \langle s_1, s_2, \ldots, s_n \rangle$ and compares $\boldsymbol{s}$ against the database to determine its location.

The challenge to the above localization problem is that signal fluctuation is unavoidable, which may be due to multipath fading and interference. For example, in Fig. 1, there are three beacons, $b_1$, $b_2$, and $b_3$, and three training locations, $\ell_1$, $\ell_2$, and $\ell_3$. Ideally, an object at location $\ell_1$ should observe a signal vector $\langle s_1, s_2, s_3 \rangle$. However, if the received signal strength from beacon $b_2$ is slightly degraded to $s^*_2$, the estimated location may become $\ell_2$. Similarly, if the signal quality of $b_3$ is degraded to $s^*_3$, the estimated location may become $\ell_3$.

To improve localization accuracy, our scheme will not be based on a single observation. Instead, it will be based on a sequence of vectors observed close to the current time $t$. Let

Digital Object Indentifier 10.1109/TKDE.2007.190730

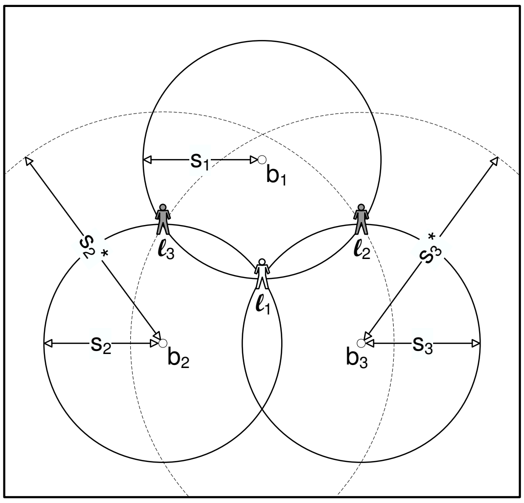



Fig. 1
AN EXAMPLE LOCALIZATION ERROR DUE TO SIGNAL FLUCTUATION. SOLID LINE CIRCLES STAND FOR CORRECT DISTANCE MEASUREMENT WHILE DOTTED LINE CIRCLE STAND FOR DEGRADED DISTANCE MEASUREMENT.

$\boldsymbol{s}^{(i)} = \langle s_1^{(i)}, s_2^{(i)}, \ldots, s_n^{(i)} \rangle$ be the signal strength vector that the object observes at time $i$. Our scheme will exploit temporal diversity by generating a set of scrambled vectors from the recently received $w$ signal vectors to enlarge the comparison space, and exploit spatial locality by examining the possible locations of the object in the previous $h$ time steps. Fig. 2 shows the system flowchart. The *vector buffer* is a shift register which can keep the most recent $w$ signal strength vectors. The *vector scramble module* takes these $w$ vectors and generates $w^\delta$ scrambled vectors, where $\delta$ is the *scramble degree*. These scrambled vectors are then sent to the *location estimation module* to generate $w^\delta$ predicted locations. Finally, the *location selection module* will choose one location based on spatial dependency.

### A. Vector Scramble Module

Given $w$ vectors $\boldsymbol{s}^{(i)}$, $i = t-w+1, \ldots, t$, the vector scramble module will generate $w^\delta$ vectors. Temporal diversity means that the RSSs from the same beacon over a short period of time, though maybe fluctuating due to multipath fading and interference, are expected to contain a correct value statistically. At any location, the distribution of the received signal strength $\boldsymbol{s}_j^{(i)}$ from $b_j$ can be modeled by a Gaussian normal distribution $G(\mu, \sigma)$ with a mean $\mu$ and a standard deviation $\sigma$. Let $I = [\mu - \Delta, \mu + \Delta]$ be the interval within which the signal fluctuation is tolerable for the positioning purpose at this location. Then the probability that we see at least one signal strength falling in the interval $I$ over $w$ continuous observations is $1 - \prod_{i=t-w+1}^{t} prob[\boldsymbol{s}_j^{(i)} \notin I]$, which should be high as long as $w$ is large enough. Note that the temporal diversity is based on the assumption that the samples of the same beacon observed from similar environments may fluctuate. However, over a short period of time, within a few samples, the mean or close-to-mean signal strength value as observed in the training phase of that beacon is very likely to appear. Therefore, with proper scrambling, the mean or close-to-mean signal strength values of all beacons are likely to appear in the scrambling results.

To tolerate fluctuation and noise, we will scramble the signal strengths of the $\delta$ beacons over the past $w$ steps, where $\delta \leq n$. Here we adopt the strategy of selecting the beacons with the strongest average RSSs for scrambling[1]. Specifically, let $V_j = \{s_j^{(i)}, i = t-w+1, \ldots, t\}$ denote the set of signal strengths from $b_j$, $avg(V_j)$ the average of $V_j$, and $\hat{V}$ the set of beacons whose $avg(V_j)$ are top $\delta$ among all beacons. Then the set of scrambled vectors is defined as $R = \{\langle r_1, r_2, \ldots, r_n \rangle | r_j \in V_j$ if $b_j \in \hat{V}$ and $r_j = avg(V_j)$ if $b_j \notin \hat{V}\}$. Intuitively, if the average signal strength $avg(V_j)$ of $b_j$ is low, we let $r_j = avg(V_j)$; otherwise, $r_j$ is selected from the set $V_j$. Therefore, there are at most $w^\delta$ vectors in $R$.

For example, in the scenario in Fig. 1, if we have $w = 2$ vectors $\boldsymbol{s}^{(t)} = \langle s_1, s_2^*, s_3 \rangle$ and $\boldsymbol{s}^{(t-1)} = \langle s_1, s_2, s_3^* \rangle$, using $\delta = 3$ (which means that the signal strengths of the 3 beacons with the strongest average RSSs will be scrambled), we can generate a set of scrambled vectors $R = \{\langle s_1, s_2, s_3 \rangle$, $\langle s_1, s_2^*, s_3 \rangle$, $\langle s_1, s_2, s_3^* \rangle$, $\langle s_1, s_2^*, s_3^* \rangle\}$. Therefore, with temporal diversity, the correct vector $\langle s_1, s_2, s_3 \rangle$ has appeared, thus helping us correctly position the object.

### B. Location Estimation Module

This module can be implemented by plugging in any fingerprint-based positioning algorithm, such as [2], [3], [4], [13], [15]. For each scrambled vector $\boldsymbol{r} \in R$, we can use the algorithm to determine a location $loc(\boldsymbol{r})$. Below, we show how to plug in algorithms [2], [13] into this module.

*1) Nearest Neighbor in Signal Space (NNSS) Algorithm [2]:* In NNSS, the Euclidean distance between signal vectors is used as the metric. Therefore, for each $\boldsymbol{r} = \langle r_1, r_2, \ldots, r_n \rangle \in R$ and each training location $\ell_i$'s characteristic vector $\boldsymbol{c}_i = \langle c_1^i, c_2^i, \ldots, c_n^i \rangle$, we will compute a distance $dist(\boldsymbol{r}, \ell_i) = \sqrt{\sum_{j=1}^{n} (r_j - c_j^i)^2}$. Then the training location with the minimum Euclidean distance will be chosen as the predicted location, i.e., $loc(\boldsymbol{r}) = \arg\min_{\ell_i \in \mathscr{L}} dist(\boldsymbol{r}, \ell_i)$.

*2) Probability-based Algorithm [13]:* To account for signal fluctuation, this algorithm regards the characteristic vector $\boldsymbol{c}_i = \langle c_1^i, c_2^i, \ldots, c_n^i \rangle$ of the training location $\ell_i$ as a vector of probability distributions, i.e., $c_j^i$ is a probability density function instead of a scalar. For each $\boldsymbol{r} \in R$, this algorithm computes a likelihood function $Pr(\boldsymbol{r}|\ell_i) = \prod_{j=1}^{n} c_j^i(r_j)$ to estimate the probability that the object is at $\ell_i$. The predicted location is set to the training location with the highest probability, i.e., $loc(\boldsymbol{r}) = \arg\max_{\ell_i \in \mathscr{L}} Pr(\boldsymbol{r}|\ell_i)$.

### C. Location Selection Module

From the above derivation, at each time $t$, we already generated a set of scrambled vectors. Now, let the set of scrambled vectors generated at time $t$ be $R^{(t)}$, and the corresponding set of locations $L^{(t)} = \{loc(\boldsymbol{r}) | \boldsymbol{r} \in R^{(t)}\}$. This module will pick one location $loc^*$ in $L^{(t)}$ as the final estimated location of the object. Spatial dependency implies that an object's movement trajectory should not have "jumping effect", i.e., the successive locations should be smooth. The location selection module is developed based on this assumption.

[1]There may exist other strategies to select beacons for scrambling. For example, this can be done according to the information gain in [5]. This can be directed to future research.

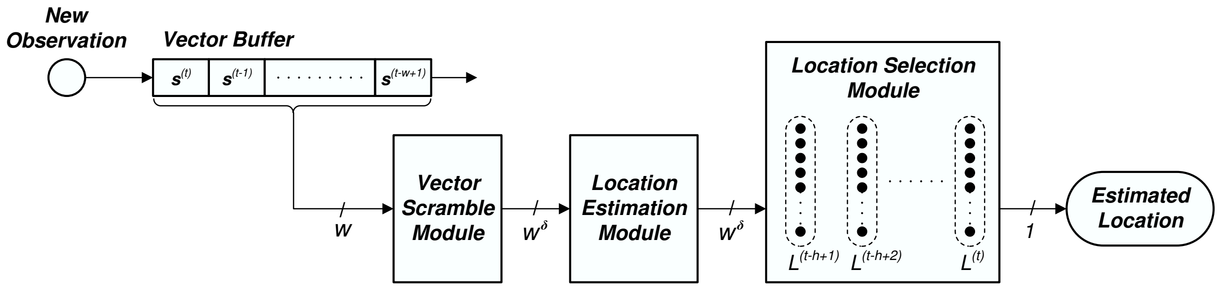



Fig. 2

THE SYSTEM FLOW OF THE SCRAMBLING METHOD.

At time $t$, this module will memorize the previous $h$ sets of locations, $L^{(t-h+1)}$, $L^{(t-h+2)}$, ..., and $L^{(t)}$, where integer $h$ is the spatial dependency factor. Then we compute a path $\ell_{t-h+1} \to \ell_{t-h+2} \to \ldots \to \ell_t$, such that $\ell_i \in L^{(i)}$ and the total path distance is minimum (here we define the distance of two locations to be their Euclidean distance). Intuitively, this path has the minimum jumping effect. Then the final location at time $t$ is predicated as $\ell_t$. (Note that the locations other than $\ell_t$ on the path are not necessarily the predicted locations at the previous $h-1$ steps.)

## III. Simulation and Experimental Results

### A. Simulation Environment

We have simulated a 50×50 square meters region with eight beacons placed at $(0,0)$, $(49,0)$, $(0,49)$, $(49,49)$, $(25,0)$, $(0,25)$, $(25,49)$, and $(49,25)$. In the training phase, we collect data at 624 grid points, each separated by 2 m, both horizontally and vertically, except for the locations with beacons. In our simulations, signal strengths in the training and positioning phases are both generated by the *log-distance path loss model* [12], which specifies the decay between a transmitter and a receiver by

$$PL(d) = PL(d_0) + 10\alpha log(\frac{d}{d_0}) + X_\sigma,$$

where $d_0$ is a reference distance close to the transmitter, $d$ is the distance between the transmitter and the receiver, $\alpha$ is a path loss exponent, and $X_\sigma$ is a zero-mean Gaussian random variable with a standard deviation $\sigma$. The received signal strengths are generated by $P_t - PL(d)$, where $P_t$ denotes the transmit power. In the positioning phase, objects move based on a random waypoint mobility model. An object will switch between *moving* and *pausing* states. In the moving state, a traveling speed and a destination location will be chosen randomly. After reaching the destination, the object enters the pausing state and stays there for a period of time. This process will be repeated until the simulation is terminated. Each simulation will last 2000 seconds. Unless stated otherwise, the other simulation parameters are set to transmit power $P_t = 15$ dBm, $PL(d_0 = 1\text{ m}) = 37.3$ dBm, $\alpha = 3.3$ dBm, $\sigma = 4$ dBm, moving speed = 1~10 m/sec, sampling period = 1 sec, and pause time = 10 sec. In the location estimation module, we adopt three fingerprint-based positioning algorithms: nearest neighbor in signal space (*NNSS*) [2], probability-based (*Prob*) [13], and neural network (*NeuNet*) with a 3-layer structure [4] algorithms. These schemes, when plugged into our scrambling method, are denoted as *scr-NNSS*, *scr-Prob*, and *scr-NeuNet*, respectively. We also simulate a simple method which uses the average of the signal strengths of the most recent $w$ samples for database matching, denoted as *avg-NNSS*, *avg-Prob*, and *avg-NeuNet*, respectively.

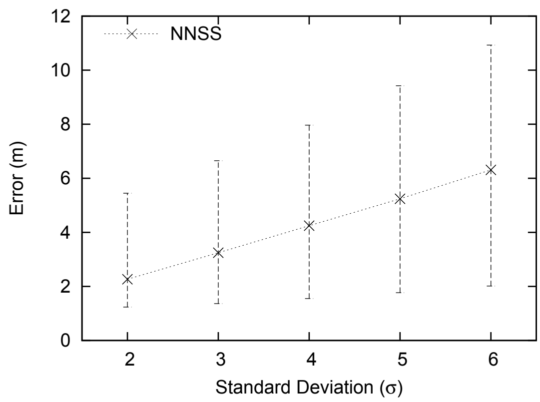



Fig. 3

THE RANGES OF AVERAGE POSITIONING ERRORS OF THE SCRAMBLED LOCATIONS GENERATED BY *scr-NNSS* ($w = 3,\ \delta = 3$).

Before we evaluate the performance of the scrambling method in details, we first show the impact of the increased search space using the scrambling method. For the scrambled locations, we estimate the Euclidean distance from each of them to the corresponding true location. In Fig. 3, we show the ranges of average maximum and minimum positioning errors of the scrambled locations generated by *scr-NNSS* ($w = 3,\ \delta = 3$) under different levels of noise. This reflects the increase of search space. These error ranges generally cover the average errors of NNSS without scrambling, denoted by crossed points. By incorporating with proper location selection module, the location errors are expected to be decreased, as shown in our simulation and experimental results below.

### B. System Parameters

First, we evaluate some system parameters of the scrambling method with different positioning algorithms, such as vector buffer size $w$ and spatial dependency factor $h$. [2]Fig. 4(a) shows

[2]We can observe that the performance of the *NeuNet*-based scheme is worse than the others. This is against the results in [4]. This is because we use a simpler structure to implement the *NeuNet* algorithm without any performance tuning. However, this does not violate our objective of showing the capability of our scrambling method in improving each specific localization scheme.

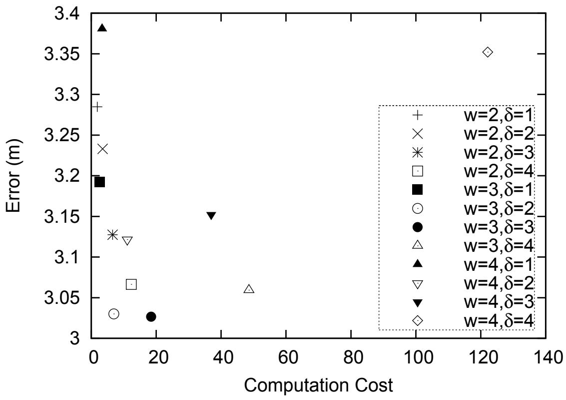



Fig. 5
COMPUTATION COST VS. POSITIONING ERROR UNDER DIFFERENT VECTOR BUFFER SIZES AND SCRAMBLE DEGREES.

the result when $w$ ranging from 2 to 5. A longer $w$ means more scrambled vectors being generated. However, when objects are moving, a too large $w$ may result in some unreal scrambled vectors. This is because the the observed samples' environment conditions may vary significantly due to mobility. Thus, a too large $w$ might be harmful to positioning accuracy. From Fig. 4(a), we can observe that a $w$ around 3 can achieve better performance. In Fig. 4(b), we examine the impact of the spatial dependency factor $h$. We can observe that a larger $h$ can result in higher accuracy. However, after $h \geq 4$, the amount of improvement becomes very insignificant.

Fig. 4(c) examines the effect of the scramble degree $\delta$. A larger $\delta$ means more scrambled vectors being generated. Note that when $\delta = 0$, our *scr-NNSS* is in fact the same as the *avg-NNSS* scheme. The result shows that the positioning accuracy is kept on being improved when $\delta \leq 4$. However, a larger $\delta$ also implies higher computation overhead. Also, we see that the positioning error will increase when $\delta \geq 5$. This is possibly due to choosing too weak beacons for scrambling. The samples contributed by these beacons do not help discriminate one location from the others, thus easily producing positioning errors. Hence, a scramble degree $\delta$ around 4 can achieve relatively satisfied performance.

With different vector buffer sizes and scramble degrees, the corresponding computation cost may vary. Generally, larger buffer sizes or/and scramble degrees will result in more scramble samples, thus increasing the computation cost. Fig. 5 shows the average computation cost versus positioning error under different settings. The results closer to the lower-left corner are better choices. In some cases, higher computation cost does not imply higher accuracy. The reason is that more scrambled locations will increase the complexity of location selection. Hence, after considering both positioning accuracy and computation cost, we choose $(w, \delta) = (2,4)$, $(3,2)$, and $(3,3)$ and $h = 5$ in the following simulation studies.

### C. Simulation Studies

Moving speed will affect accuracy because a higher speed will result in received samples observed at locations that are farther apart. This violates our assumption of temporal diversity. Fig. 6(a) shows that the positioning errors of the original *NNSS* algorithm, two average schemes *avg-NNSS ($w = 2$)* and *avg-NNSS ($w = 4$)*, and two scrambling schemes *scr-NNSS ($w = 2$, $\delta = 4$)* and *scr-NNSS ($w = 4$, $\delta = 2$)*. All schemes except *NNSS* are influenced by the moving speed, but *avg-NNSS* is more sensitive to the moving speed than *scr-NNSS*. A larger buffer size $w$ is harmful as the moving speed increases. This is because these schemes all rely on the assumption that samples in the buffer are collected under similar environment conditions. In particular, we see that *avg-NNSS ($w = 4$)* outperforms all other schemes at low speeds, but quickly deteriorates as the speed increases. But a larger scramble degree can reduce the negative effect caused by high moving speeds. Therefore, the proposed scrambling method can not only improve localization accuracy, but also reduce the effect of mobility. In Fig. 6(b) and (c), we see the similar effect when comparing to *Prob* and *NeuNet* algorithms. To summarize, when the object is fixed or has low mobility, the average method with a larger buffer size is a good choice to reduce the effect of signal fluctuation. However, if mobility is not negligible, then the scrambling method with a reasonable small buffer size can provide quite satisfiable and stable performance.

Fig. 7 shows the impact of sampling periods on different schemes. We simulate mobile objects with a normal walking speed of 1.5 m/sec. A longer sampling period will hurt schemes with buffering. Therefore, we see higher positioning errors when the sampling period increases. However, compared to the moving speed, the sampling period is more controllable. Hence, we suggest to decrease sampling periods under the hardware constraints.

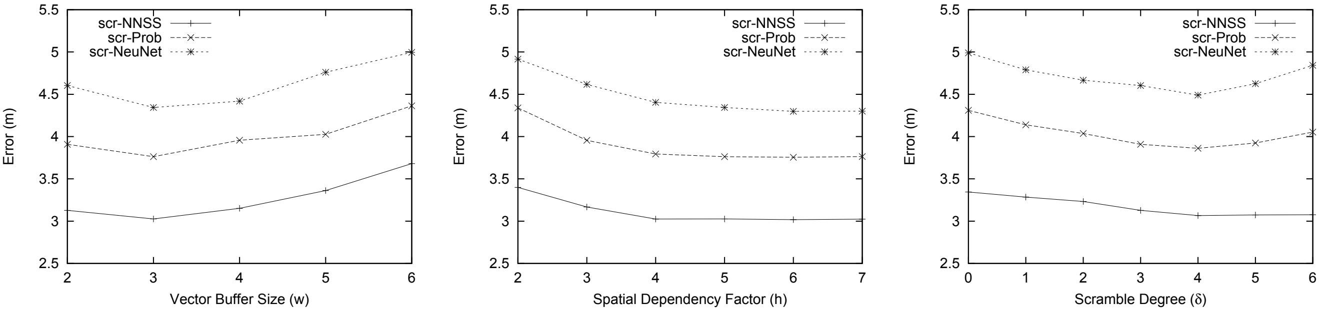
In our simulation model, we use a normal distribution to simulate the noise effect. Next, we observe the impact of the standard deviation $\sigma$ of the distribution (which reflects the level of noise). Fig. 8 shows the result. The original *NNSS*, *Prob*, and *NeuNet* schemes are also very sensitive to the enlargement of $\sigma$. By taking the average of several samples, the effect can be reduced. Using our scrambling methods, such as *scr-NNSS ($w = 3$, $\delta = 3$)*, *scr-Prob ($w = 3$, $\delta = 3$)*, and *scr-NeuNet ($w = 3$, $\delta = 3$)*, the error can be further reduced. Therefore, the proposed scrambling method can help to reduce the impact of environmental noise.

### D. Experimental Results

In order to verify our simulation results, we also conducted a median-size experiment with real signal strength data based on IEEE 802.11 WLANs. We will compare NNSS with and without scrambling. Fig. 9 shows our experimental environment at the National Chiao Tung University, Engineering Building III. There are over 20 access points in the environment. We collect 100 samples in each of east, south, west, and north directions at 39 distinct locations. This data collection procedure is performed twice, one for training purpose and the other for testing purpose. We simulate a user trace from location 1 to location 39 in that order at different sampling frequencies $f$. That is, at each location, we will randomly choose $f$ samples from the testing database. So there are $39 \times f$ samples for experimental evaluation. (The value of $f$ can somehow be interpreted as the roaming speed of a user.)

Fig. 10 shows the experimental results of different parameter settings. In Fig. 10(a), the ranges of average maximum and minimum positioning errors of the scrambled locations are presented. This result shows that with scrambling, larger search space can be generated, thus being able to provide potential locations that are closer to the real location at the cost of some computational overhead. Note that the crossed points in Fig. 10(a) are average

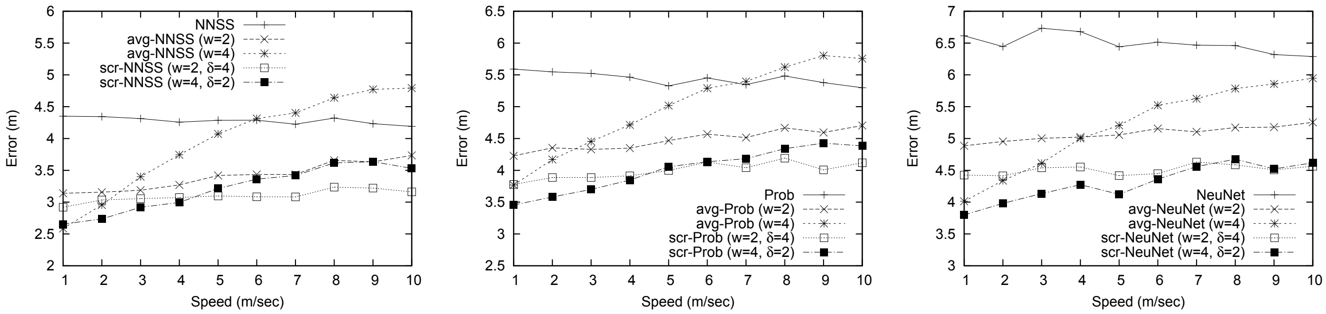
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.


(a) (b) (c)

Fig. 4

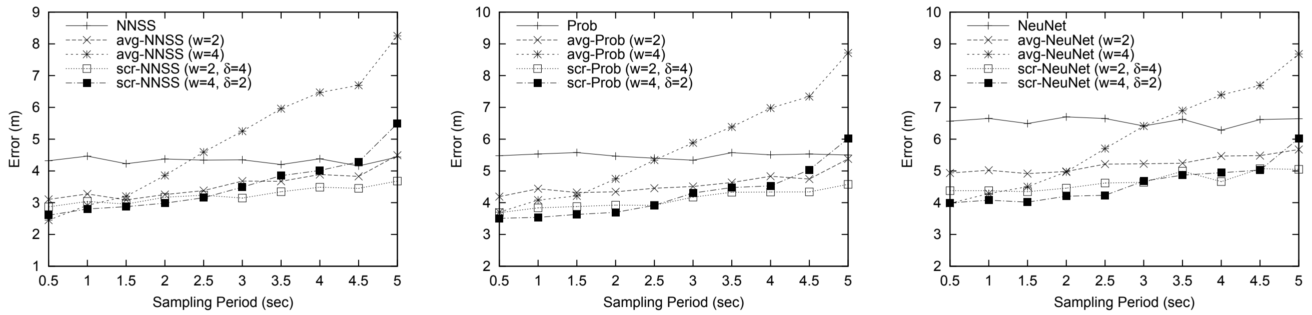
THE PERFORMANCE EVALUATION OF THE SCRAMBLING METHOD UNDER DIFFERENT (A) BUFFER SIZE $w$ ($\delta = 3$, $h = 5$), (B) SPATIAL DEPENDENCY FACTOR $h$ ($\delta = 3$, $w = 3$), AND (C) SCRAMBLE DEGREE $\delta$ ($w = 2$, $h = 5$).




(a) (b) (c)

Fig. 6

THE EFFECT OF MOVING SPEED ON THE SCRAMBLING METHOD WITH (A) *NNSS*, (B) *Prob*, AND (C) *NeuNet*.




(a) (b) (c)

Fig. 7

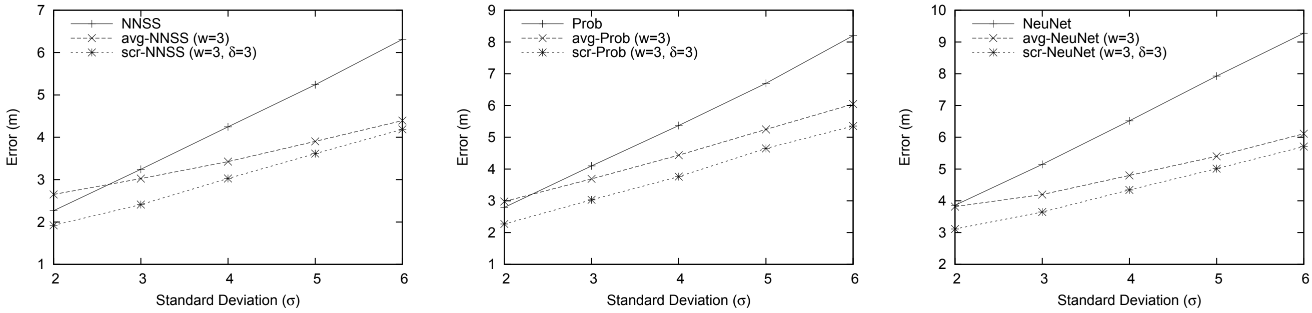
THE EFFECT OF SAMPLING PERIODS ON THE SCRAMBLING METHOD WITH (A) *NNSS*, (B) *Prob*, AND (C) *NeuNet* WHEN THE MOVING SPEED = 1.5 M/SEC.

errors incurred by NNSS without scrambling. Fig. 10(b) shows a similar comparison as Fig. 5, except that the results are obtained from experiments. Hence, to balance the trade-off between positioning accuracy and computation cost, the scrambling methods with $(w, \delta) = (3,2)$, $(3,3)$, $(3,4)$, $(4,2)$, and $(4,3)$ are better choices.

Based on the above observation, we select *scr-NNSS* ($w = 3$, $\delta = 3$) and *avg-NNSS* ($w = 3$) to evaluate the positioning errors under different $f$ in Fig. 11. We observe that when $w \geq f$, the impact of signal fluctuation can be effectively reduced. Therefore, *scr-NNSS* ($w = 3$, $\delta = 3$) performs better when $f \geq 3$. However, when $f \leq 2$, *scr-NNSS* ($w = 3$, $\delta = 3$) performs worse because we may usually scramble samples from different locations.

## IV. CONCLUSIONS

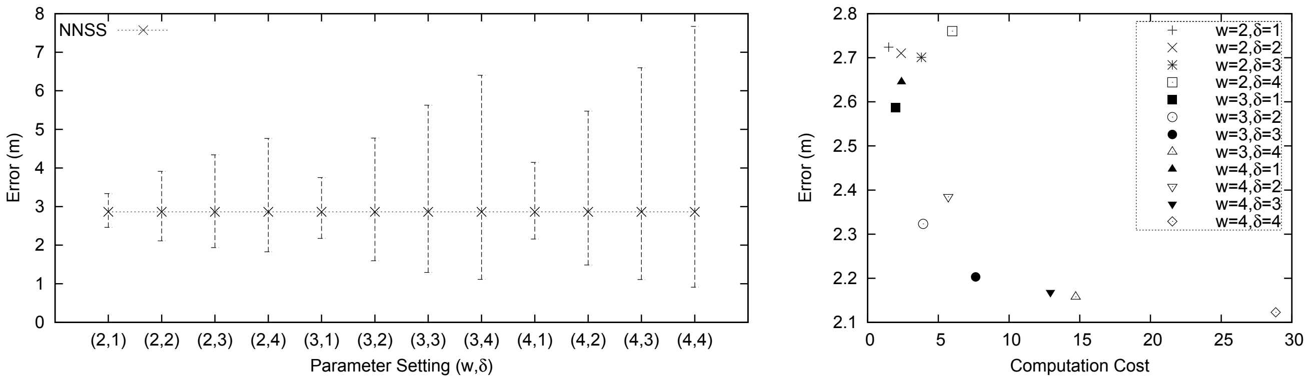
We have presented a novel scrambling method which considers both temporal diversity and spatial dependency for localization in noisy environments. By means of scrambling, we can enlarge the sample space. Through recombining the limited observed signal vectors, samples with less interference are expected to appear. Also, a trajectory mechanism is developed to select a final location from the scrambled results. This scrambling method




(a) (b) (c)

Fig. 8

THE EFFECT OF NOISE ON THE SCRAMBLING METHOD WITH (A) *NNSS*, (B) *Prob*, AND (C) *NeuNet*.




(a) (b)

Fig. 10

THE EXPERIMENTAL RESULTS OF (A) THE RANGES OF AVERAGE POSITIONING ERRORS OF THE SCRAMBLED LOCATIONS AND (B) THE COMPUTATION COST VS. POSITIONING ERRORS WHEN $f = 4$ UNDER DIFFERENT PARAMETER SETTINGS.

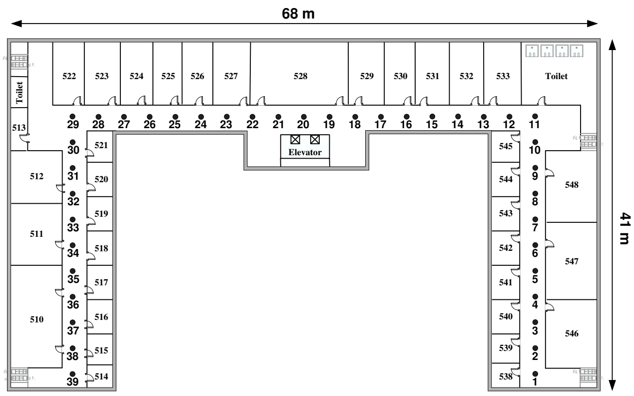



Fig. 9

THE EXPERIMENT ENVIRONMENT AT THE NATIONAL CHIAO TUNG UNIVERSITY, ENGINEERING BUILDING III.

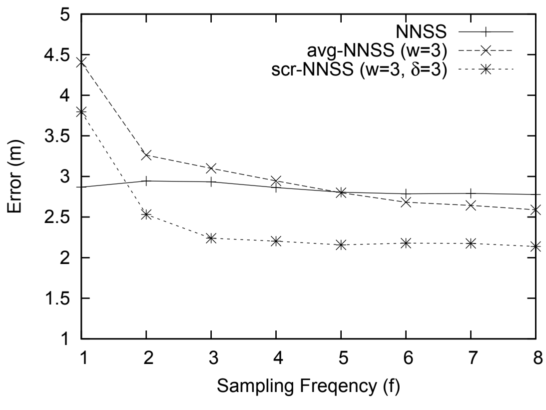



Fig. 11

THE EXPERIMENTAL RESULTS OF *scr-NNSS* ($w = 3,\ \delta = 3$) AND *avg-NNSS* ($w = 3$) UNDER DIFFERENT SAMPLING FREQUENCIES.

can be integrated with various fingerprint-based positioning algorithms to improve their positioning accuracy. Our simulation and experimental results show that with a reasonable increase in computation overhead, positioning accuracy can be significantly improved.

## ACKNOWLEDGMENT

Y. C. Tseng's research is co-sponsored by Taiwan MoE ATU Program, by NSC grants 93-2752-E-007-001-PAE, 96-2623-7-009-002-ET, 95-2221-E-009-058-MY3, 95-2221-E-009-060-MY3, 95-2219-E-009-007, 95-2218-E-009-209, and 94-

2219-E-007-009, by Realtek Semiconductor Corp., by MOEA under grant number 94-EC-17-A-04-S1-044, by ITRI, Taiwan, by Microsoft Corp., and by Intel Corp.

## REFERENCES

[1] M. Addlesee, R. Curwen, S. Hodges, J. Newman, P. Steggles, A. Ward, and A. Hopper. Implementing a Sentient Computing System. *Computer*, 34(8):50–56, 2001.

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

[3] R. Battiti, T. L. Nhat, and A. Villani. Location-aware Computing: A Neural Network Mmodel for Determining Location in Wireless LANs. Technical Report DIT-02-0083, Universita di Trento, Dipartimento di Informatica e Telecomunicazioni, 2002.

[4] M. Brunato and R. Battiti. Statistical Learning Theory for Location Fingerprinting in Wireless LANs. *Computer Networks*, 47(6):825–845, 2005.

[5] Y. Chen, Q. Yang, J. Yin, and X. Chai. Power-efficient Access-point Selection for Indoor Location Estimation. *IEEE Trans. on Knowledge and Data Engineering*, 18(7):877–888, 2006.

[6] P. Enge and P. Misra. Special Issue on GPS: The Global Positioning System. *Proc of the IEEE*, 87(1):3–15, 1999.

[7] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2000.

[8] 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.

[9] 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.

[10] D. Niculescu and B. Nath. Ad Hoc Positioning System (APS) Using AOA. In *IEEE INFOCOM*, volume 3, pages 1734–1743, 2003.

[11] 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*, 18(9):1181–1193, 2006.

[12] T. S. Rappaport. *Wireless Communications: Principles and Practice*. IEEE Press Piscataway, NJ, USA, 1996.

[13] 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*, 9(3):155–164, 2002.

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

[15] V. Seshadri, G. V. Záruba, and M. Huber. A Bayesian Sampling Approach to In-door Localization of Wireless Devices Using Received Signal Strength Indication. In *IEEE PERCOM*, pages 75–84, 2005.

[16] Y.-C. Tseng, S.-L. Wu, W.-H. Liao, and C.-M. Chao. Location Awareness in Ad Hoc Wireless Mobile Networks. *IEEE Computer*, 34(6):46–52, 2001.

**Sheng-Po Kuo** received his B.S. and M.S. degree in Computer Science and Information Engineering from National Chiao Tung University, Taiwan, in 2001 and 2003, respectively. Currently, he is pursuing his Ph.D. in Computer Science at National Chiao Tung University, Taiwan. His current research interests are primarily in pattern-matching localization algorithms, indoor location tracking systems, and location-based services.

**Yu-Chee Tseng** obtained his Ph.D. in Computer and Information Science from the Ohio State University in January of 1994. He is Professor (2000 preset), Chairman (2005-present), and Associate Dean (2007-present) at the Department of Computer Science, National Chiao-Tung University, Taiwan. From 2006 to present, he is Adjunct Chair Professor at the Chung Yuan Christian University. Dr. Tseng received the Outstanding Research Award, by National Science Council, ROC, in both 2001-2002 and 2003-2005, the Best Paper Award, by Int'l Conf. on Parallel Processing, in 2003, the Elite I. T. Award in 2004, and the Distinguished Alumnus Award, by the Ohio State University, in 2005. His research interests include mobile computing, wireless communication, network security, and parallel and distributed computing. Dr. Tseng served as an Associate Editor for *Telecommunication Systems* (2005 present), as an Associate Editor for *IEEE Trans. on Vehicular Technology* (2005 present), and as an Associate Editor for *IEEE Trans. on Mobile Computing* (2006 present).