

Inverting Systems of Embedded Sensors for Position Verification in Location-Aware Applications

Shu Chen, *Student Member, IEEE*, Yingying Chen, *Member, IEEE*, and Wade Trappe, *Member, IEEE*

Abstract—Wireless sensor networks are typically deployed to monitor phenomena that vary over the spatial region the sensor network covers. The sensor readings may also be dual-used for additional purposes. In this paper, we propose to use the inherent spatial variability in physical phenomena, such as temperature or ambient acoustic energy, to support localization and position verification. We first present the problem of localization using general spatial information fields, and then, propose a theory for exploiting this spatial variability for localization. Our Spatial Correlation Weighting Mechanism (SCWM) uses spatial correlation across different phenomena to isolate an appropriate subset of environmental parameters for better location accuracy. We then develop an array of algorithms employing environmental parameters using a two-level approach: first, we develop the strategies on how the subset of parameters should be chosen, and second, we derive mapping functions for position estimation. Our algorithms support our theoretical model for performing localization utilizing environmental properties. Finally, we provide an experimental evaluation of our approach by using a collection of physical phenomena measured across 100 locations inside a building. Our results provide strong evidence of the viability of using general sensor readings for location-aware applications.

Index Terms—Localization, sensor networks, wireless networks.

1 INTRODUCTION

ALTHOUGH the data associated with sensor readings in sensor networks might be intended to drive specific applications, e.g., the remote monitoring of temperature, this wealth of data may also be dual-used for additional purposes. In particular, since the purpose of a sensor network is to provide sampling of a physical phenomena across a wide geographic/spatial distance, the close link between sensor data and location may be used to assist in applications involving localization and position verification.

In this paper, we propose the use of spatially varying environmental properties to support localization, without requiring the deployment of a localization infrastructure and additional access points (i.e., landmarks, with known locations). We present the problem of localization using general spatial information fields. We examine the use of physical properties, such as temperature and ambient acoustic/RF energy, and explore whether the inherent spatial variability may be used to localize the position of a mobile entity.

Each physical parameter has its unique spatial characteristics relative to the environment. For computational savings, it is desirable to use the most efficient subset of environmental properties that captures spatial variability. We propose a scheme, Spatial Correlation Weighting Mechanism (SCWM), which can guide in parameter selection by determining the parameter combination with the strongest discriminative characteristics needed for localization.

In our localization model, an array of sensors has been initially deployed for environmental monitoring. These sensors are stationary and the deployment information, such as locations of the sensors, is known to the system. The data collected are used as a baseline database, and a user reports the physical readings at its location. We further developed a set of algorithms, employing environmental properties, to determine a user's position. A two-level approach is taken: first, we perform parameter selection, and then, we take measurements to localize. More specifically, 1) from the perspective of how the subset of parameters should be chosen, we present the *Flexibly choosing Environmental Parameters (Flex-EP)* algorithm that tries to find a globally-optimal parameter subset; further, the *Progressive Flexibly choosing Environmental Parameters (Prog-Flex-EP)* algorithm is presented, which is a sequential algorithm that locally customizes the best set of parameters for each user and 2) we implemented different schemes in mapping the selected set of parameters to a physical location.

To validate our approach, we collected an array of environmental parameters at 100 locations in a real building environment. Using these data for our experiments, we found that SCWM is highly effective in evaluating and selecting the parameter subset with the highest discriminative power. Further, we observed that environmental

- S. Chen is with the Department of Computer Science and Wireless Information Network Laboratory (WINLAB), Rutgers University, 671 US Route 1, North Brunswick, NJ 08902-3390. E-mail: shuchen@cs.rutgers.edu.
- Y. Chen is with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Castle Point on Hudson, Hoboken, NJ 07030. E-mail: yingying.chen@stevens.edu.
- W. Trappe is with the Wireless Information Network Laboratory (WINLAB) and Department of Electrical and Computer Engineering, Rutgers University, 671 US Route 1, North Brunswick, NJ 08902-3390. E-mail: trappe@winlab.rutgers.edu.

Manuscript received 3 July 2008; revised 7 Apr. 2009; accepted 19 June 2009; published online 26 June 2009.

Recommended for acceptance by Y. Pan.

For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2008-07-0249. Digital Object Identifier no. 10.1109/TPDS.2009.110.

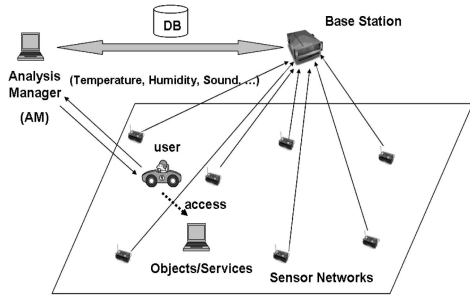


Fig. 1. Using environmental properties for spatial determination.

parameters have localizing capability. By using environmental readings plus the Received Signal Strength (RSS) from one access point, we can achieve qualitatively the same performance as traditional localization schemes employing RSS with at least four access points [1]. Moreover, under the assistance of additional environmental parameters, the localization performance can be refined and improved over traditional approaches.

In summary, our contributions in this work include:

- A localizing mechanism that uses existing sensor network readings and does not need additional localization infrastructure.
- An environmental parameter selection method that optimizes the subset of parameters for localization.
- An approach that uses these environmental readings to refine conventional localization results.

The remainder of this paper is organized as follows: In Section 2, we provide an overview of our problem. We then formulate the theoretical model of exploiting environmental properties for localization and propose SCWM in Section 3. Next, in Section 4, we present a spectrum of algorithms that we developed. Section 5 shows our experimental methodology and our evaluation results. In Section 6, we discuss in depth about two issues, given the result of our experiments. We then place our work in the context of the broader localization literature in Section 7. Finally, we conclude in Section 8.

2 PROBLEM OVERVIEW

Wireless sensor networks typically record environmental readings corresponding to underlying physical information fields for high-level applications to utilize. For instance, temperature, humidity, and ambient acoustic energy are common environmental parameters under constant monitoring. The fact that these data are measurements of the environment in a specific area and at a specific time interval suggests that they can be used for spatial localization, as well as their original purpose.

Our proposed model for utilizing the environmental readings for localization and position verification is built upon existing wireless sensor networks, as presented in Fig. 1. In the area of interest, there are sensors deployed to perform environmental monitoring. Sensors periodically report environmental readings back to Base Stations. The sensors are stationary and their locations are known to the Base Station. The reported environmental information is stored in a database, associated to the sensors' location information, in

real time for retrieval by the upper level applications. A management entity containing data processing and analysis capabilities, the *Analysis Manager (AM)*, calculates a user's location. The *AM* can be combined with the base station or operate alone in a centralized manner or run with multiple distributed instances. If the *AM* is operating by itself, the *AM* should be able to access the environmental readings stored in the database shown in Fig. 1.

A user, when it wants to get its position, first sends a request message to *AM*. After receiving the request, the *AM* asks the user to provide environmental readings observed at that time by the user. By running localization algorithms utilizing environmental properties, *AM* then compares the user's readings to the environmental data (provided by sensors) stored in the database and estimates the user's location.

The traditional approach for localization is to deploy enough landmarks, which measure the received signal strength, with known positions in an area of interest to assist in localization. There maybe cases, however, where there are not enough landmarks in the area of interest (e.g., due to cost limitations), to actually localize. Further, it is not always possible to deploy more landmarks due to environmental constraints. In addition, for certain applications, such as position verification in Spatial Access Control [2], [3], very high location accuracy results are not needed, so additional landmarks would be wasteful. Thus, it is advantageous to use environmental properties from sensor networks to help determine and verify positions without requiring the infrastructure of additional landmarks.

3 THEORETICAL APPROACH

In this section, we present the theoretical underpinnings behind using environmental properties for localization. We first propose a generalized measurement model, and then, provide rules to evaluate each parameter's localizing capability. Finally, we present mechanisms for parameter selection to assist in localization and position verification.

3.1 A Generalized Measurement Model

Let $E = (e_1, e_2, \dots, e_n)$ denote the vector of environmental properties that are monitored by the sensors, where e_i is the value of the i th environmental parameter. These parameters have the property that they are recorded in the spatiotemporal domain, which means that they may vary with location and time. Thus, the value of the parameter vector at position p and time t can be expressed as

$$E_{p,t} = [e_1(p, t), e_2(p, t), \dots, e_n(p, t)]. \quad (1)$$

Here, p is a spatial position, which can be one-, two-, or three-dimensional. In this study, we focus on p in a two-dimensional space. More generally speaking, p can represent a point (x, y) or a region. Let $\Omega = P \times T$ be the spatiotemporal region [2] that we are interested in, and \mathbf{E} be the domain of environmental parameter values, then there exists a mapping $f: \Omega \rightarrow \mathbf{E}$ that takes the physical position p and maps it to an environmental parameter reading $E_{p,t}$, as presented in Fig. 2. $f(p, t) = E_{p,t}$ represents the environmental readings recorded at the spatiotemporal location (p, t) . The inverse mapping from

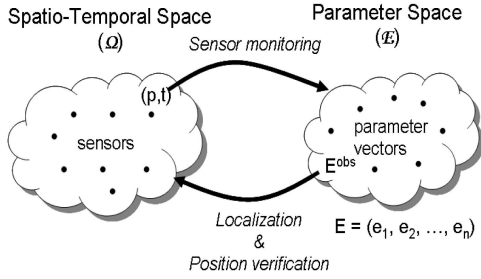


Fig. 2. Theoretical model: physical domain versus environmental properties domain.

\mathbf{E} to Ω enables localization and position verification from environmental properties.

At a fixed time t , the function $f_t(x, y) = f(x, y, t)$ induces a probability density function ρ on \mathbf{E} . We further define the function $\rho(E; \epsilon)$ to be the probability of having value $f_t(x, y) \in (E; \epsilon)$ as presented in Fig. 3, when a position (x, y) is chosen randomly from an uniform distribution from $X \times Y$. Here, $(E; \epsilon)$ denotes the ϵ -ball around E . Note that $\rho(E; \epsilon)$ is the integral of the probability density function over the region $(E; \epsilon)$. Given an environment measurement reading E , in order to find the corresponding physical position, we want to find the region $p \subset \Omega$ such that $(E; \epsilon) \subseteq \mathbf{E}$ is mapped back to p . In other words, we want to find the inverse mapping $p = f_t^{-1}(E; \epsilon) = \{(x, y) : f(x, y) \in (E; \epsilon)\}$. Usually, Ω has finite area, and we can normalize it to have $Area(\Omega) \triangleq 1$. It is clear that $Area(p) = \rho(E; \epsilon)$, as shown in [4].

We also note that in order to localize a user in a two-dimensional space, simply using a single environmental parameter is generally not sufficient. Thus, multiple environmental parameters are desirable for localization and position verification. However, using all the available environmental parameters for localization may result in high computational complexity and cost. We would like to choose subsets of parameters that consist of enough parameters to provide reasonable localization accuracy. Next, we provide an analysis to evaluate environmental parameters and derive methods for effective parameter selection.

3.2 Parameter Evaluation

Different parameters have different characteristics in terms of value changes across various physical locations and time. For certain parameters, the values may vary largely across

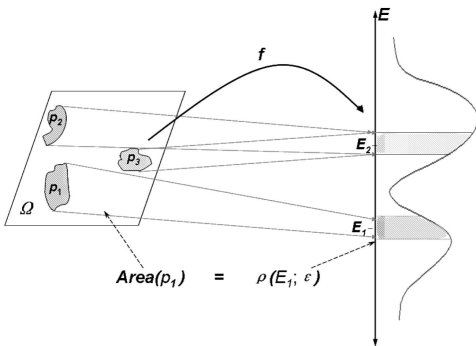


Fig. 3. Function f induces a probability density function ρ in measurement domain \mathbf{E} .

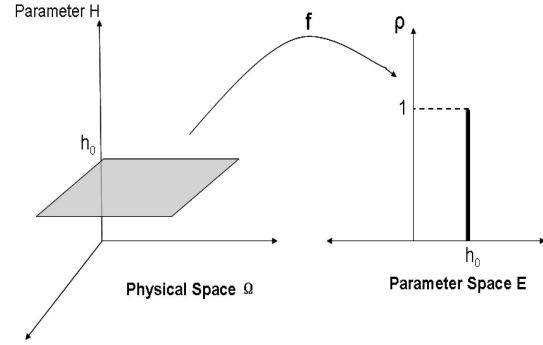


Fig. 4. An illustration of a “bad” environmental parameter that does not contribute to localization.

different locations. The physical phenomena reported by these kind of parameters can be utilized to distinguish between locations. We define such parameters as having large *discriminative power*. On the other hand, the values of some parameters may vary little within the area of interest. Fig. 4 is an illustration of a parameter H belonging to this category. It has the same value h_0 throughout the physical region, thus in the parameter space \mathbf{E} , $\rho(h_0) = 1$. Such a parameter does not have the ability to distinguish between physical locations, and thus, has poor localization accuracy.

It is important to choose a parameter subset so that the combination of the parameters in the subset has enough discriminative power to support localization. Carelessly choosing a parameter subset may even result in localization errors, as shown in Fig. 3, where two far-away regions p_2 and p_3 in the physical space have the same environmental readings $(E_2; \epsilon)$. The inverse mapping would result in $f_t^{-1}(E_2; \epsilon) = p_2 \cup p_3$. This indicates that the subset of parameters in this case is not sufficient for localization.

To summarize, in order for an environmental parameter or a subset of environmental parameters to contribute in localization and position verification, they should have the following characteristics:

1. The subset of parameters should have large discriminative power. We found that a parameter that contributes largely to location accuracy must have large variance across the environment. On the other hand, a parameter with large variance may not necessarily help to improve its localization capability. Contrary to intuition, the correlation between parameters was not an important factor in location accuracy.
2. The parameter readings with similar values should map to sensors positioned close to each other. This will eliminate large localization errors.
3. There must exist a spatial correlation for the parameter readings so that similar values for parameter readings will result in locations close in the physical domain.

3.3 Parameter Selection

Next, we develop a series of approaches to help select environmental parameters that when combined have greater capability for localization and position verification.

3.3.1 Parameter Dispersion

Conceptually, for an environmental parameter e_i , the more disperse the values are, the better the discriminative power is for this parameter. In statistics, there are several ways to measure dispersion of a parameter, such as *range*, *variance*, *standard deviation*, and *average absolute deviation*.

However, none of these measurements are complete, since they only look at the data itself and neglect the spatial relationships between the data and the physical environment. For example, if the data readings from two different environmental parameters have the same distribution, their dispersions are about the same. However, they may result in very different accuracy if used for localization. Suppose both of them have a subset of readings with the same value, but for one parameter, the same-value readings are clustered, while for the other parameter, the same-value readings are scattered among the region. The latter parameter will generate larger errors when applied in localization.

Further, we found that the cross-parameter relationship, covariance, does not heavily contribute to localizing capability. Instead, the spatial relationship dominates localization accuracy. This leads us to look for metrics that take into consideration of the spatial correlation when evaluating an environmental parameter, in addition to the parameter dispersion.

3.3.2 Data Normalization

Different from the traditional localization methods [1], [5], the data sources in our problem are from different kinds of environmental parameters, such as temperature, humidity, ambient acoustic energy, etc. Different environmental parameters have different units and different ranges of values. For example, in our experiments, the temperature readings range from 65.2F to 77.3F, whereas Received Signal Strength values range from -59.8 to -99 dBm. In order to choose a subset of environmental parameters working together for localization purpose, we need to compare and calculate the contribution of each parameter directly. We normalize the data using the classical statistical approach: $e_i^{norm} = \frac{e_i - \mu_i}{\sigma_i}$, where μ_i and σ_i are the mean and standard deviation of the parameter e_i . We then work with the normalized data e_i^{norm} for the rest of our study.

3.3.3 Spatial Correlation Weighting Mechanism

A natural approach for selecting parameters is to slice the domain of the environmental parameters \mathbf{E} into equal-sized bins. Then, for parameter readings falling within the same bin, we calculate their corresponding distances in the physical space for every pair of sensors. Based on our criteria for parameter evaluation (items 2 and 3 in Section 3.2), among all the possible combinations of fixed-size parameter subsets, a parameter subset that results in the smallest averaged physical distance per bin is the optimal parameter subset with the highest combined discriminative power for that size of parameter subsets.

The size of the bin and the number of bins are the two critical factors. We found that different parameter subsets behave differently on different bin sizes. In order to give a thorough evaluation of parameters, we need to vary the bin size and come up with a way to combine the results from different bin sizes to make a fair judgment. In addition, the

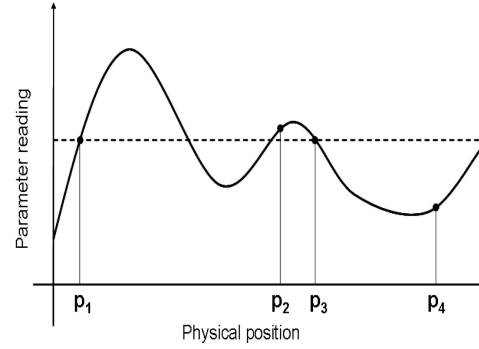


Fig. 5. Three scenarios under SCWM calculation: 1) position pair $\{p_2, p_3\}$; 2) position pair $\{p_1, p_4\}$; and 3) position pairs $\{p_1, p_3\}$ and $\{p_2, p_4\}$. The relationship is: $(w_{1,3} \cdot d_{1,3}) > (w_{1,2} \cdot d_{1,2}) \gg (w_{1,4} \cdot d_{1,4})$ and $(w_{2,3} \cdot d_{2,3})$.

disadvantage of this approach is that the number of bins (and hence, computational complexity) needed to increase exponentially with the number of environmental parameters in a parameter subset. Thus, the computational complexity increases exponentially with the size of the parameter subset. It is not desirable to use a metric that has potentially high computational complexity. To solve these problems, we have developed a method whose complexity does not change much with the size of parameter subset and inherently solves the different bin-size problem. We now present this method, the SCWM.

The important factor to take into consideration when performing parameter selection is to analyze how far away two positions can be in the physical domain \mathbf{P} , given a distance in the environmental parameter domain \mathbf{E} . The SCWM calculates a sum $W(K)$ of pairwise weighted distances in \mathbf{P} , which gives larger weight to similar parameter readings in \mathbf{E} and smaller weight to more different parameter readings. If we define K to represent a set of parameter indices chosen to form the parameter subset, $W(K)$ is defined as follows:

$$\begin{aligned} W(K) &= \sum_{p_i, p_j, i \neq j} w_{i,j} \cdot d_{i,j} \\ &= \sum_{p_i, p_j, i \neq j} w_{i,j} \cdot \|p_i - p_j\|^2, \end{aligned} \quad (2)$$

$$\text{with } w_{i,j} = \frac{1}{1 + \tau \cdot \|e_{k \in K}(p_i) - e_{k \in K}(p_j)\|^2},$$

where τ is a scaling factor. We call $w_{i,j}$ the *parameter weight*, which takes values from $(0, 1]$. The computational complexity does not dramatically increase with the number of parameters in a parameter subset when using SCWM for parameter selection.

Fig. 5 illustrates how SCWM helps to choose the parameter subset with the highest discriminative power. We describe three typical scenarios during SCWM calculation. The first scenario is shown with position pair p_2 and p_3 . The two positions are close to each other and have similar parameter readings. The contribution of the parameter weight $w_{2,3}$ is large, close to 1. But the resulting $(w_{2,3} \cdot d_{2,3})$ is small because $d_{2,3}$ is very small. Next, position pair p_1 and p_4 is farther away from each other and their parameter readings are very different. In this case, the contribution of

the $w_{1,4}$ is very small and much less than 1. The above two scenarios satisfy the theory requirements of better location accuracy described in Section 3.2. Finally, we look at a poor scenario with position pairs $\{p_1, p_3\}$ and $\{p_1, p_2\}$. Their parameter readings are the same or very similar, but they are farther away from each other. The contribution of the parameter weight is large, especially for $w_{1,3}$ which reaches its maximum, equaling to 1. Both $(w_{1,3} \cdot d_{1,3})$ and $(w_{1,2} \cdot d_{1,2})$ are also large because the distances are farther away.

For a fixed number of parameters, SCWM calculates all the pairwise weighted distances over all the possible combination of parameters. The parameter subset with most of its readings following the patterns described in the first two scenarios will result in the final value of $W(K)$ to be small. While the parameter subset having most of readings similar to the third scenario we presented, the calculated value of $W(K)$ will be large. The parameter subset that results in the minimum value of $W(K)$ is the optimal parameter combination that contains the highest discriminative power for performing localization. SCWM can sort all the possible combination of parameters under a fixed-size parameter subset in the descending order from the highest discriminative power to the lowest discriminative power for localization. In Section 5, we present experimental results utilizing SCWM.

4 ALGORITHMS

In this section, we present our algorithms for using environmental parameters, assisted by SCWM, for localization.

4.1 Overview

There are two important aspects when developing algorithms using environmental parameters for localization: one is how to choose the subset of parameters given a fixed number of parameter set, e.g., choose all at once or choose one at a time; and the other is to derive a mapping function from environmental parameters to a physical location. Taking into consideration of these two aspects, we used a two-level approach and developed an array of algorithms employing environmental properties.

First, we propose two basic algorithms, namely, *Flex-EPs* and *Prog-Flex-EPs* for selecting parameters. Given a set of parameters Γ that is chosen from all the available environmental parameters using SCWM, *Flex-EP* chooses the subset of parameters in one shot based on the results from SCWM, while *Prog-Flex-EP* progressively chooses one parameter at a time until reaching the number of parameters defined by the parameter subset. Thus, *Flex-EP* tries to find a global-optimal parameter subset, while *Prog-Flex-EP* locally customizes the best set of parameters for each testing point.

Next, we consider how position estimation is carried out. We implement the basic algorithms using two types of mapping functions, denoted by *-Dist* and *-Prob*. *-Dist* utilizes nearest neighbor matching in the parameter space, whereas *-Prob* employs a statistical maximum likelihood estimation approach. We derived four algorithms to perform location estimation and position verification: *Flex-EP-Dist*, *Flex-EP-Prob*, *Prog-Flex-EP-Dist*, and *Prog-Flex-EP-Prob*.

In addition, we developed variants of the above algorithms extending from their basic ideas. *-Avg* is a

variant for the *-Dist* function, which finds the centroid of the top k returned locations. The *-CM* variant is derived from the *-Prob* function that returns the center of mass of the top locations satisfying a threshold probability rule. The summary of the algorithms and their variants is presented in Table 1. We discuss the details of each algorithm in the following sections, and later present experimental results.

4.2 Flex-EP

Flex-EP-Dist. *Flex-EP-Dist* finds the minimum distance in the domain of environmental parameters between the observed readings E_Γ^{obs} reported by the user and the collection of measurements recorded by the sensor network, which are stored in the database, as shown in Fig. 1. *Flex-EP-Dist* reports the position of the closest sensor as the location estimate of the user.

We use *Flex-EP-Dist-Basic* to represent the basic version of *Flex-EP-Dist*. Fig. 6 presents the pseudocode that implements *Flex-EP-Dist-Basic*. Further, its variant *Flex-EP-Dist-Avg* chooses the top k sensors that are closest to the user's reading in parameter space and returns the centroid of the k locations, with $k > 1$.

Flex-EP-Prob. Instead of calculating the distance in parameter space between a user's measurement and each sensor's reading as in *Flex-EP-Dist*, *Flex-EP-Prob* takes the approach of calculating the probability that the user is at each position and choosing the one with the maximum likelihood as the estimation of the user's location. In order to support a probabilistic formulation of our localization problem, we employ several practical assumptions. First, we assume that the sensor network provides accurate sampling of the environment, and specifically, a sensor at position p_j will measure the value of the i th environmental parameter as e_i . On the other hand, in order to account for measurement errors relative to the sensor nodes, we assume that a user's measurements at a position p_j will be distributed in a Gaussian manner about the sensor's measurement at that position. That is, a user's measurement at position p_j will have mean $\mu_{I,j} = e_i(p_j, t)$, with a standard deviation σ_i . Additionally, for tractability of our formulation, we assume that the user's environmental parameters are independent of each other, and thus, at position p_j , the vector random variable $E_\Gamma = (E_1, E_2, \dots, E_k)$, where k is the cardinality of the parameter subset Γ , follows a multivariate Gaussian density:

$$\begin{aligned} f_{E_\Gamma, p_j}(\tilde{e}_1, \dots, \tilde{e}_k) &= \prod_{i=1}^k f_{E_i, p_j}(\tilde{e}_i) \\ &= \frac{1}{(2\pi)^{k/2} \prod_{i=1}^k \sigma_i} \exp\left(-\frac{1}{2} \sum_{i=1}^k \frac{(\tilde{e}_i - \mu_{i,j})^2}{\sigma_i^2}\right), \end{aligned} \quad (3)$$

with \tilde{e}_i being the user's measured values. Given a user's observed parameter vector $E_\Gamma^{obs} = (\tilde{e}_1, \tilde{e}_2, \dots, \tilde{e}_k)$, we can calculate the probability of being at position p_j by using Bayes' rule:

$$P((p_j, t) | E_\Gamma^{obs}) = \frac{P(E_\Gamma^{obs} | (p_j, t)) \times P(p_j, t)}{P(E_\Gamma^{obs})}. \quad (4)$$

Further, we may assume a uniform distribution over possible user locations, i.e., $P(p_j, t) = \frac{1}{N}$, $\forall j$, where N is the

TABLE 1
Summary of the Algorithms Employing Environmental Properties

Algorithm Abbreviation	Variant	Description
Flexibly choosing Environmental Parameters (Flex-EP)		
Flex-EP-Dist	Basic	Finds the location of the sample point that is closest to the user's reading in the parameter space.
Flex-EP-Dist	Avg	Finds the centroid of the top K locations of sample points that are closest to the user's reading in the parameter space.
Flex-EP-Prob	Basic	Finds the location of the sample point with the highest probability of correctness.
Flex-EP-Prob	CM	Finds the center of mass of the top K locations as determined by the threshold β , weighted by their probability of correctness.
Progressive Flexibly choosing Environmental Parameters (Prog-Flex-EP)		
Prog-Flex-EP-Dist	Basic	In each round, progressively chooses the most effective parameter, excludes candidate positions (that are far from the user's reading in the current subspace) based on the selection ratio γ , and returns the position that is closest to the user's reading among those candidates left in the last round.
Prog-Flex-EP-Dist	Avg	In each round, progressively chooses parameters, selects candidate positions, and returns the centroid of the top K locations of candidates remaining in the last round.
Prog-Flex-EP-Prob	Basic	In each round, progressively chooses the most effective parameter, excludes candidate positions with a low probability of correctness based on either the cumulative probability confidence α or the individual probability threshold τ , and returns the position with the highest probability among those candidates remaining in the last round.
Prog-Flex-EP-Prob	CM	In each round, progressively chooses parameters, candidate positions, and returns the center of mass of the top locations remaining in the last round as determined by the threshold β , weighted by their probability of correctness.

total number of possible locations. Also, we note that $P(E_{\Gamma}^{obs})$ is a constant; thus, we have

$$P((p_j, t)|E_{\Gamma}^{obs}) = c \cdot P(E_{\Gamma}^{obs}|(p_j, t)). \quad (5)$$

Given the fact that the user must be localized to one of the N sample locations, by using the density function from above, we can calculate the probability of being at each location:

$$P((p_j, t)|E_{\Gamma}^{obs}) = c \cdot f_{E_{\Gamma}, j}(E_{\Gamma}^{obs}), \quad (6)$$

where $c = 1/\sum_{j=1}^N f_{E_{\Gamma}, j}(E_{\Gamma}^{obs})$. Finally, the basic version of *Flex-EP-Prob*, i.e., *Flex-EP-Prob-Basic*, chooses the position with the highest probability as the location estimation.

```

input  $E_{\Gamma}^{obs}(p, t), \Gamma$ 
output
  min dist in the parameter domain
  closest sensor position in the physical domain
initialize
  minDist = maxNum
  sensorPosition = empty
loop through information reported by sensors
  for each set of information from a sensor begin
    dist =  $\|E_{\Gamma}^{obs}(p, t) - E_{\Gamma}^{sensor}(p, t)\|$ 
    if dist < minDist
      then minDist = dist and sensorPosition = sensor
  end for
end loop
return minDist, sensorPosition

```

Fig. 6. The Flex-EP-Dist-Basic algorithm.

Moreover, the variant *Flex-EP-Prob-CM* sorts the probabilities in descending order, and then, picks the top k locations $(p_{i_1}, \dots, p_{i_k})$ such that the sum of their probabilities $\sum_{m=1}^k p_{i_m} \geq \beta$, where β is an adjustable threshold. *Flex-EP-Prob-CM* returns the *center of mass* of these locations (with high probabilities) as the location estimation:

$$(\hat{x}, \hat{y}) = \left(\sum_{m=1}^k p_{i_m} x_{i_m}, \sum_{m=1}^k p_{i_m} y_{i_m} \right). \quad (7)$$

4.3 Prog-Flex-EP

Prog-Flex-EP is a successive refinement localization method. Instead of choosing all the parameters of a parameter subset at once (as described in *Flex-EP* algorithms), *Prog-Flex-EP* sequentially chooses one parameter at a time from the parameter subset. In each round, *Prog-Flex-EP* picks a parameter E_i which is most effective, i.e., with high discriminative power when combined with parameters already chosen from the previous rounds based on SCWM. Further, according to the combined parameters $E_{\Gamma} = (E_1, E_2, \dots, E_i)$, in every round, the location candidates are refined by choosing a subset from the candidate locations \tilde{P} from the previous round. This procedure repeats until it finds a solution with high confidence or reaches the maximum number of parameters given a fixed size of the subset of parameters.

Fig. 7 illustrates how the successive refining location candidates is performed. Suppose at round $i-1$, the candidate location areas are Ω_1, Ω_2 , and Ω_3 , as shown in Fig. 7, while the true location is shown as a star residing within Ω_2 . At round i , based on the newly combined parameter subset, we sort the candidate locations according

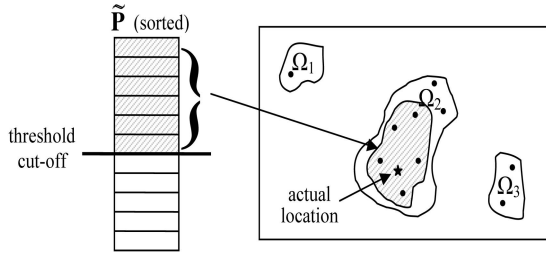


Fig. 7. The location estimation is refined by sequentially choosing new parameters, leading to successfully smaller subset of locations.

to the appropriate criteria and select the ones that are within the threshold, as the new candidate location set. We note that the threshold is a general concept: it could indicate the cumulative probability confidence α or the individual probability threshold τ in the algorithm variants of *Prog-Flex-EP*, which will be discussed shortly. As illustrated in Fig. 7, the new candidate set excludes areas Ω_1 and Ω_3 and a partial area of Ω_2 (shown as a shaded region) that contains the true location.

Therefore, in the approach of *Flex-EP*, we select the overall best K environment properties beforehand and always use that same set of properties for location estimation. Whereas in *Prog-Flex-EP*, the next selected parameter is subject to change based on the results from the former rounds. In reality, it could be that at different locations, the environmental properties that best distinguish this location from others are different. For example, in one location, the three best parameters are RSS, barometric pressure, and light, while in another location, they could be temperature, RF energy, and humidity. Thus, the approach of *Prog-Flex-EP* gives the localization process a chance to locally customize the sets of parameters used.

The key challenge for *Prog-Flex-EP* is how to evaluate the candidate locations at each round and decide on the candidates for the next round. We further developed *Prog-Flex-EP-Dist* and *Prog-Flex-EP-Prob* with regard to their different criteria used for ranking the candidate locations and setting the threshold cutoff.

Prog-Flex-EP-Dist. *Prog-Flex-EP-Dist* ranks the candidate locations at each round based on the euclidean distance between the observed readings and the readings at each location in \tilde{P} in the parameter space using the current parameter set E_Γ . Then, it selects $|P_{new}| = \lceil \gamma \cdot |\tilde{P}| \rceil$ as the number of candidates for the next round. Here, γ is an adjustable *selection ratios* with $0 < \gamma \leq 1$ (e.g., $\gamma = 0.3$), and $|\tilde{P}|$ denotes the number of locations in \tilde{P} . Fig. 8 presents the pseudocode that implements *Prog-Flex-EP-Dist-Basic*. We note that different *selection ratio* will result in different localization results. Smaller ratios allow us to refine our candidate list faster, thus permitting more effective parameter selection. However, it could also result in eliminating too many candidate locations, which increases the risk of throwing out the correct candidates prematurely. Thus, choosing an appropriate selection ratio γ will allow *Prog-Flex-EP-Dist* to be robust.

Prog-Flex-EP-Prob. Similar to *Flex-EP-Prob*, *Prog-Flex-EP-Prob* calculates the probability that the user is at each sample location, but based on the current progressively

```

initialize
 $\Gamma = \text{empty}$  //Selected environmental parameters
 $\text{sensorVectors} = \text{list of all sensor vectors}$ 
 $\gamma = \text{selection ratio}$ 

while  $|\Gamma| < \text{totalSubParams}$  and  $|\text{sensorVectors}| > 1$ 
  for each  $\text{param}_i$  not in  $\Gamma$ 
    calculate  $W_i(\{\Gamma, \text{param}_i\})$  based on SCWM
    find the  $\min W_i$ ,  $\text{newParam} = \text{param}_i$  that makes  $\min W_i$ 
     $\Gamma = \{\Gamma, \text{newParam}\}$ 
  for each sensor location  $\in \text{sensorVectors}$ 
    calculate its distance from  $E_\Gamma^{\text{obs}}$  in the parameter subspace
     $\text{dist}_\Gamma = \|E_\Gamma^{\text{obs}} - E_\Gamma^{\text{sensor}}\|$ 
  end for
  sort the remaining  $\text{sensorVectors}$  by increasing distance
  select the top  $\lceil \gamma \cdot |\text{sensorVectors}| \rceil$  entries from the sorted
   $\text{sensorVectors}$  as the new  $\text{sensorVectors}$ 
end while

return the first entry of  $\text{sensorVectors}$ 

```

Fig. 8. The Prog-Flex-EP-Dist-Basic algorithm.

obtained parameter set Γ instead of the whole parameter subset. Further, we define a *confidence* α as the threshold cutoff level. Given a confidence α , we pick the set of most likely locations such that the sum of their probabilities is larger than α .

We note that in this algorithm, α represents a cumulative threshold from each round: the larger the α , the more candidates are chosen for the next round. Ideally, a threshold would keep the sample locations that have obviously high probabilities to the observed readings, and eliminate those candidates that have much lower probabilities. In some cases, a cumulative threshold may not provide the desired effect. Fig. 9 presents three scenarios with different probability distributions. In each scenario, the shaded area shows the probabilities that accumulate to the confidence α , as an example $\alpha = 75$ percent, and the solid blue line on the bottom shows the sample locations picked by this cumulative threshold. τ is an individual probability value used as threshold. Using τ as the cutoff, every sample location with probability greater than τ is picked to enter the next round and the others are thrown away. The dotted line shows the sample locations picked by using τ as the threshold. In either Fig. 9a or Fig. 9b, there is a significant drop in each of the probability distribution. Keeping the locations before the drop and eliminating the rest would be desirable. But in Fig. 9a, we accumulated enough probability before the drop, so some “good” sample locations still with high probabilities are prematurely eliminated, whereas in Fig. 9b, many unlikely candidates are unnecessarily included since not enough probability is accumulated

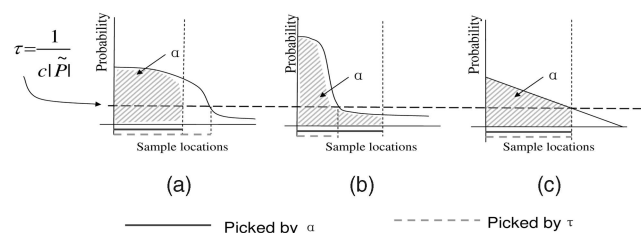


Fig. 9. Prog-Flex-EP-Prob: Comparison of candidates selection when using cumulative probability confidence α and individual probability threshold τ .

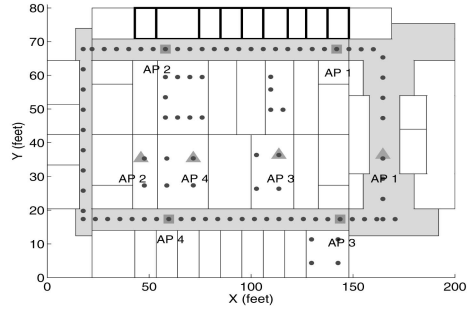


Fig. 10. Layout of the experimental floor.

before the drop. These low-probability points may become the reason to cause confusion in the next round. In these two cases, however, an individual threshold τ does a better cutoff. The individual threshold τ is related to the number of current candidates. We define the threshold τ to be

$$\frac{1}{C \cdot |\bar{P}|},$$

where C is an adjustable constant. Fig. 9c presents that under a normal situation, the selection results of using confidence α and threshold τ are the same.

Moreover, in order to compare to the *Flex-EP* approaches, two variants out of the basic algorithms: *Prog-Flex-EP-Prob-Avg* and *Prog-Flex-EP-Prob-CM*, are developed based on the current progressively obtained parameter subsets Γ .

5 EXPERIMENTAL EVALUATION

In this section, we present our experimental evaluation results.

5.1 Experimental Methodology

In order to study the effectiveness of using environmental properties for localization and position verification, we conducted experiments in a real office environment, the third floor of the Computer Science building at Rutgers University, as shown in Fig. 10. For over 100 locations on the floor, shown as small blue dots, we collected environmental readings at these locations over a one-week period of time. This simulated the setup of a sensor network consisting of 100 sensors.

The environmental parameters that we studied are temperature, humidity, acoustic noise, spectrum energy, and RSS. The RSS readings are collected from two 802.15.4 (ZigBee) networks, each with four Access Points (APs) deployed across the floor. One network has four APs deployed horizontally across the floor shown as triangles, and the other has four APs distributed in a rectangular pattern shown as squares. The access points used were Telosb motes.

To simulate a scenario where only one base station is available in the area of interest, we will choose only one RSS reading (in decibel meters) when forming the parameter subset. Further, we used a Wi-Spy spectrum analyzer [6] to record the spectral energy at each location. It records the signal amplitude (in decibel meters) versus the frequency from 2.400 to 2.485 GHz. At each testing location, we picked

TABLE 2
Summary of Environmental Parameter Measurement

Parameter	Index	Measuring Device
Temperature	1	Thermometer
Humidity	2	Digital hygrometer
Acoustic Noise	Daytime	Microphone and Dell laptop
	Night time	
Spectrum Energy	2.435GHz Max	Wi-Spy Spectrum Analyzer
	2.465GHz Max	
	2.435GHz Avg	
	2.465GHz Avg	
Received Signal Strength (RSS), Colinear	AP 1	Telosb motes and Dell laptop
	AP 2	
	AP 3	
	AP 4	
Received Signal Strength (RSS), Rectangular	AP 1	Telosb motes and Dell laptop
	AP 2	
	AP 3	
	AP 4	

two frequencies (2.435 and 2.465 GHz) and calculated their maximum and average amplitude, respectively, over the recording period. Note that the RSS is the received signal from a beacon packet, while the spectrum energy is the ambient RF energy corresponding to a specific frequency range.

For acoustic noise, our intuition is that the behavior of the parameter can vary largely during daytime and nighttime. Thus, we collected readings of ambient noise (in decibels) for both day and night. Moreover, we measured the humidity (in percentage) using a digital hygrometer and temperature (in Fahrenheit) using a thermometer, respectively, at each location. Table 2 is a summary of the parameters and the devices that we used to conduct experiments.

5.2 Evaluation of Individual Parameters

We first study the dispersion of individual environmental parameters through parameter variance. Table 3 presents the results of the variance for each individual parameter. We found that the maximum value of the spectral energy and the RSS have large variance across the area of interest, while the average value of the spectrum energy, temperature, humidity, and ambient noise do not vary much across the experimental floor. Both daytime and nighttime readings of ambient noise have smaller variance compared to other environmental parameters. For the rest of the paper, we will use the ambient noise data collected at night. The sample maps of spectral energy at 2.435 GHz, RSS from AP2, and ambient noise are shown in Fig. 11. The irregular shape of signal maps is due to the limitation of our data

TABLE 3
Results of Single-Parameter Dispersion

Parameters and Their Variance			
Temperature	Humidity	Acoustic noise daytime	Acoustic noise night time
4.15	9.3	0.01	0.0012
Spectrum energy			
2.435GHz Max	2.465GHz Max	2.435GHz Avg	2.465GHz Avg
84.36	88.21	2.09	0.08
Received Signal Strength (RSS), Colinear			
AP1	AP2	AP3	AP4
211.63	136.65	123.31	127.27
Received Signal Strength (RSS), Rectangular			
AP1	AP2	AP3	AP4
120.48	142.35	126.76	125.24

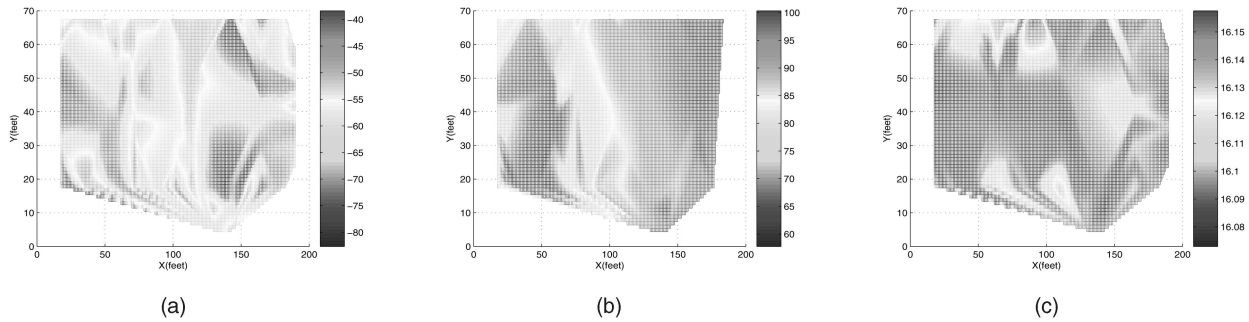


Fig. 11. Sample data maps of individual environmental parameters. (a) Max 2.435 GHz Wi-Spy (in decibel meters). (b) RSS from AP2 (in decibel meters). (c) Ambient noise (daytime) (in decibels).

collection. We can see that the sample readings of ambient noise do not change much across the whole floor, while both the maximum values of spectrum samples at 2.435 GHz and the RSS readings from AP2 present large variance indicating high discriminative power to describe the uniqueness of each location on the floor.

5.3 Effectiveness of Parameter Selection

In this section, we present the results of parameter selection using SCWM. We then evaluate the effectiveness of SCWM by comparing the cumulative distribution function (CDF) of localization errors under different sizes of parameter subsets with traditional localization methods.

Parameter selection using SCWM. Table 4 presents the results of $W(K)$ calculated from SCWM with the size of K equal to 1, 2, 3, 4, and 5, respectively. We have shown a representative subset of parameters in Table 4 with “good” and “bad” indicating that the value of $W(K)$ is smaller or larger. As we described in Section 3.3, the smaller the value of $W(K)$ for a parameter subset, the higher the discriminative power the parameter subset has. From our experimental results, we found that the parameter subset containing all the RSS parameters will result in the minimum value of $W(K)$. This is because the parameters of RSS readings have the largest variance, also have high spatial correlation, and thus, can uniquely describe the physical variability across the experimental floor. However, since we also interested in the situations where there is no localization infrastructure available, we may rely on the additional environmental properties to assist in localization and position verification. Thus, the parameter subsets displayed in Table 4 only involved at most one RSS parameter in the subset.

TABLE 4

Evaluation of SCWM with Different Size of Parameter Subsets

# of Parameters in a subset	Evaluation	Parameters: in index	SCWM calculation
1	Good:	12	20641548.6
	Bad:	8	370356046.8
2	Good:	4, 12	595033.7
	Bad:	2, 7	23284151.1
3	Good:	4, 6, 12	140758.4
	Bad:	1, 2, 8	940833.1
4	Good:	1, 4, 6, 10	72365.1
	Bad:	1, 2, 4, 7	201856.6
5	Good:	1, 4, 5, 8, 12	55198.0
	Bad:	1, 2, 4, 7, 8	112585.6

Localization using environmental parameters. Based on the parameter selection results obtained from SCWM, we further conducted localization with these parameter subsets utilizing the *Flex-EP-Dist-Basic* and *Flex-EP-Dist-Avg* algorithms. In order to compare the performance of our approach, we need to compare with a performance benchmark in the current localization research. The traditional RADAR algorithm [1] and its corresponding variants are used for our comparison, which utilize the RSS readings collected from four APs in our ZigBee network with the horizontal AP deployment.

Fig. 12 presents the CDFs of localization errors for *FLEX-EP* with the size of the parameter subset set to 2, 3, and 4, respectively. The localization results using RADAR are presented as a comparison. In each figure, the results from RADAR are presented using thinner black dash-dot lines (for regular RADAR) and black dotted lines (for RADAR-Avg algorithm), and the results from our algorithms are showed in thicker red lines. Figs. 12a and 12d are the results using two parameters in the parameter subset. The localization results when using *RSS from AP4* and *acoustic noise* are better than using *humidity* and *2.435 GHz Avg*. This is because the parameter, *RSS from AP4*, has large variance and better spatial correlation across the experimental floor. Thus, the parameter subset, $\{RSS \text{ from } AP4, \text{ acoustic noise}\}$, has smaller SCWM value than the set of $\{humidity, 2.435 \text{ GHz Avg}\}$. The performance of *Flex-EP* using two parameters is not as good as the performance of RADAR.

Next, Figs. 12b and 12e show the error CDFs when using three parameters in the parameter subset. We added one more parameter with high discriminative power, *2.465 GHz Max*, into the “good” parameter set of two parameters shown in Fig. 12a. Fig. 12b presents the results of using $\{acoustic \ noise, 2.465 \text{ GHz Max}, RSS \text{ from } AP4\}$. We found that when using two environmental parameters with high discriminative power (*2.465 GHz Max* and *RSS from AP4*) and one parameter with low discriminative power (*acoustic noise*), the performance of *Flex-EP* is qualitatively similar to traditional RADAR algorithms, which uses four RSS parameters. In Fig. 12e, each parameter in the parameter subset $\{temperature, humidity, 2.465 \text{ GHz Avg}\}$ has low discriminative power and results in a larger SCWM value, as shown in Table 4. Hence, using these properties for localization yields slightly worse performance than RADAR.

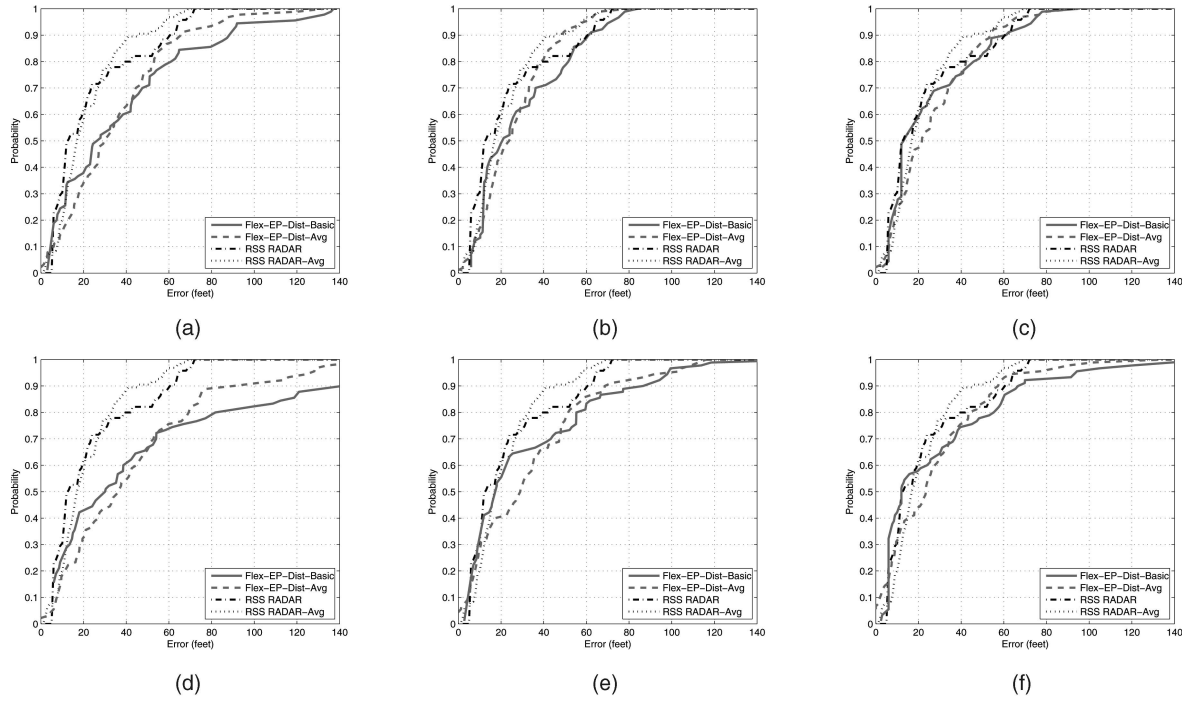


Fig. 12. Comparison of localization errors using CDF. (a) Two parameters (good): index 4, 12. (b) Three parameters (good): index 4, 6, 12. (c) Four parameters (good): index 1, 4, 6, 10. (d) Two parameters (bad): index 2, 7. (e) Three parameters (bad): index 1, 2, 8. (f) Four parameters (bad): index 1, 2, 4, 7.

Further, we examined the localization error CDFs when using four environmental parameters in Figs. 12c and 12f. In Fig. 12c, we still use two environmental parameters containing high discriminative power, *2.465 GHz Max* and *RSS from AP4*, while *acoustic noise* and *temperature* do not vary much across the experimental site. Again, we observed that the performance of *Flex-EP* is about the same as the RADAR and its variants. Moreover, under the assistance of two environmental parameters with low discriminative power, the performance is slightly improved over the three-parameter subset case, as shown in Fig. 12b.

These results indicate that choosing two environmental parameters containing high discriminative power is enough to produce comparable performance to the traditional localization approaches employing RSS with at least four access points. On the other hand, as shown in Fig. 12f, simply adding environmental parameters with low discriminative power into a parameter subset does not significantly improve the localization performance.

Fig. 13 shows the efficiency of SCWM in one picture. Given a set of parameters K , the point on the curve shows the calculated value of $W(K)$ according to SCWM and the bar below the point shows the actual average error distances using the corresponding set K . According to the theory of SCWM, we predict that the sets on the blue curve with squares, which is denoted as $W(K)$ (Bad set), will perform worse than the sets on the purple curve with triangles, which is the $W(K)$ (Good set) curve, because they have larger values of $W(K)$. The fact that the bars of Avg Err (Good set) are always lower than the bars of Ave Err (Bad set), which demonstrate that our SCWM consistently predicts the performance of parameter subsets for localization.

Since when using RSS for localization, the performance across a broad spectrum of algorithms was found to be about the same [5], we conclude that utilizing *SCWM* for parameter selection and our algorithms is effective and can achieve similar performance to a broad array of traditional localization algorithms. The similar performance is very encouraging as it indicates that utilizing environmental properties can effectively determine the position of a user.

5.4 Algorithm Performance Comparison

Flex-EP and its variants. The performance results for *Flex-EP-Dist-Basic*, *Flex-EP-Dist-Avg*, *Flex-EP-Prob-Basic*, and *Flex-EP-Prob-CM* are presented in Fig. 14a. The selection ratio γ is set to 0.5 for *Flex-EP-Prob* algorithms. In both sets of the experiments, we use four parameters in which only one is RSS. We observed that *Flex-EP* and its variants perform similarly. When using parameter set $\{1, 4, 6, 16\}$ (see

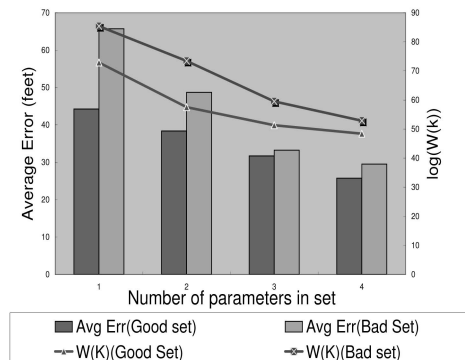


Fig. 13. Summary of the efficiency of SCWM across different parameter subsets.

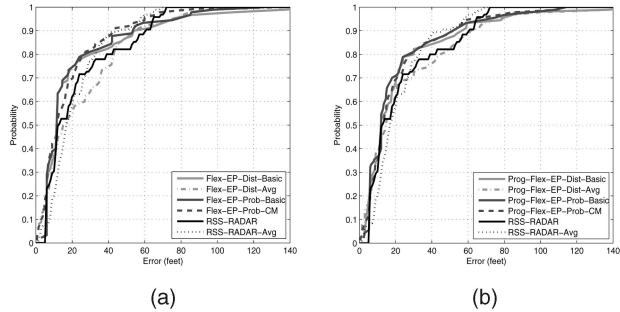


Fig. 14. Comparison of localization performance. (a) Flex-EP-Dist and Flex-EP-Prob using parameter set $\{1, 4, 6, 16\}$ with the selection ratio $\gamma = 0.5$. (b) Prog-Flex-EP-Dist and Prog-Flex-EP-Prob with the cumulative probability confidence $\alpha = 0.95$, using one RSS from Rectangular AP4.

Table 2), the results of *Flex-EP-Prob*, as shown in blue lines, are always higher than the black lines, which are the results of RADAR; thus, we observed *Flex-EP-Prob* outperforms RADAR in this setting. The result of the *Flex-EP-Prob-CM* algorithm is shown under threshold $\tau = 0.95$. We also tested on $\tau = 0.99$ or 0.75 , and the results are similar.

Prog-Flex-EP and its variants. The performance results of using *Prog-Flex-EP-Dist-Basic*, *Prog-Flex-EP-Dist-Avg*, *Prog-Flex-EP-Prob-Basic*, and *Prog-Flex-EP-Prob-CM* are presented in Fig. 14b. The results of *Prog-Flex-EP-Prob* algorithms are obtained with cumulative probability confidence α set to 0.95. In each test, we allow the use of only one RSS parameter in addition to all the other parameters. The program will determine which parameters to use as it runs. Fig. 14b shows that the progressive methods can achieve localization performance equal to or better than RADAR.

Determining threshold. Fig. 15 shows the average error distance over all the testing points when using different RSS parameters and different settings of the selection ratio γ .

The results show that usually the best performance happens when the selection ratio γ is set to 0.5 or 0.3. This trend agrees with our analysis in Section 4, which indicates that smaller ratios are more effective in eliminating the “noise” locations, and thus, improves the result; but with a too small ratio (e.g., 0.2), the error distances increase because correct locations are thrown out prematurely. We also observed that there is no obvious advantages of using the basic methods over the ones employing averaging.

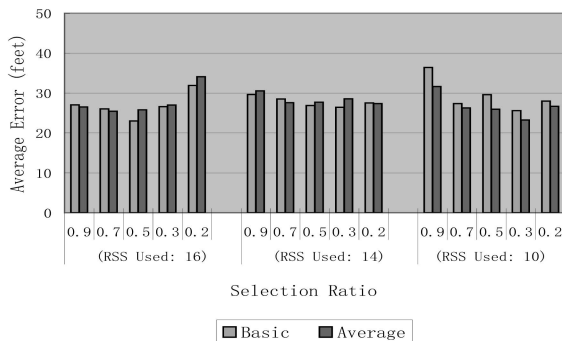


Fig. 15. Prog-Flex-EP-Dist: performance comparison when using different selection ratio γ . The size of the parameter subset is four and only one RSS is used in each experiment.

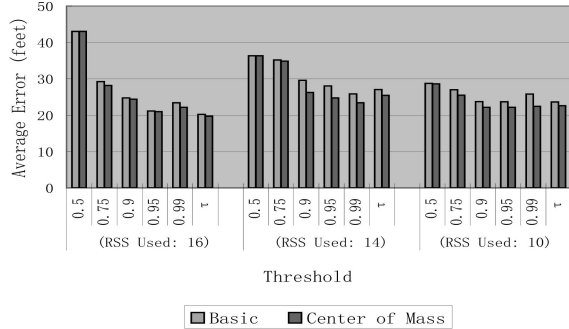


Fig. 16. Prog-Flex-EP-Prob: performance comparison of threshold values when α is set to 0.5, 0.75, 0.9, 0.95, and 0.99, respectively, and $\tau = \frac{1}{2^{|P|}}$. The size of the parameter subset is at most four and only one RSS is used in each experiment.

Fig. 16 shows the average error using *Prog-Flex-EP-Prob* with different settings of the cumulative probability confidence α and using different RSS parameters. We found the obvious trend that the larger the α , the more accurate the localization results. The average error of using the individual probability threshold method is also shown in Fig. 16 indicated by τ . The individual probability threshold means that the algorithm looks at each testing point individually. If it passes the threshold, the testing point goes to the next round. We found that, in general, using the *individual probability threshold* method achieves better performance than using the cumulative probability confidence. We also notice that *Prog-Flex-EP-Prob-CM* always performs better than *Prog-Flex-EP-Prob-Basic* under the same setting. Further, under the same input parameter sets, Table 5 shows the best average error of *Prog-Flex-EP-Dist* and *Prog-Flex-EP-Prob*, respectively, among all the threshold settings we tested on. We observed that *Prog-Flex-EP-Prob* always performs better than *Prog-Flex-EP-Dist*.

6 DISCUSSION

6.1 Refining Localization

In this section, we discuss how conventional localization results can be refined using *Flex-EP* algorithms. In a four-parameter subset, we further increased the number of parameters with high discriminative power to three by adding an additional RSS parameter into the parameter subset. Fig. 17a presents the corresponding error CDFs. We found that by utilizing three parameters with high discriminative power in a four-parameter subset, the localization performance is further refined and is almost exactly the same as RADAR.

TABLE 5
Best Average Errors for *Prog-Flex-EP-Dist* and *Prog-Flex-EP-Prob* on Different Parameter Settings

RSS used	Best Average Error	
	Prog-Flex-EP-Dist	Prog-Flex-EP-Prob
16	23.009	19.719
14	26.447	23.411
10	23.255	22.152

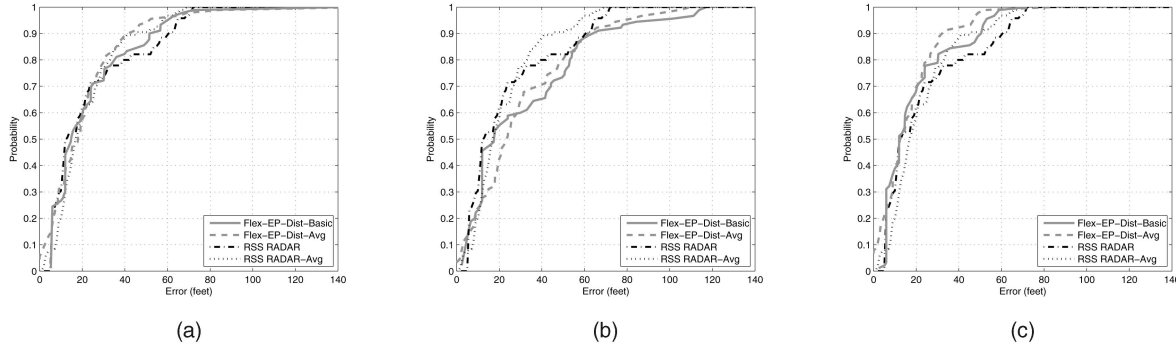


Fig. 17. Using environmental properties to refine localization results. (a) Four parameters: index 1, 5, 10, 11. (b) Five parameters: index 1, 4, 5, 8, 12. (c) Five parameters: index 1, 4, 5, 10, 11.

Further, we explored the parameter subset with five parameters. Figs. 17b and 17c show the localization error CDFs utilizing five parameters. The parameter subset in Fig. 17b still contains only two parameters with high discriminative power (*2.435 GHz Max* and *RSS from AP4*), the same as the four-parameter case in Fig. 12e, and three other parameters with low discriminative power (*temperature*, *ambient noise*, and *2.465 GHz Avg*). We observed that the localization capability is about the same as in Fig. 12e for the four-parameter case. This is in line with our previous observation, adding more environmental parameters with low discriminative power does not help much in improving the localization performance.

Turning to examine Fig. 17c, which has three parameters (*2.435 GHz Max*, *RSS from AP2*, and *RSS from AP3*) with high discriminative power in a five-parameter subset, interestingly, the localization performance has a 10 percent increase compared to the traditional RADAR algorithms, especially for *Flex-EP-Avg*, which gained over 20 percent performance improvement. In this case, only two RSS parameters are used, which means that under the assistance of other environmental parameters, only two access points are needed to achieve a better localization performance than the traditional localization algorithms employing RSS using at least four access points. This provides strong evidence that utilizing environmental properties for localization can both achieve similar performance to the traditional approaches, as well as refine conventional localization results.

6.2 Comparison of *Flex-EP* and *Prog-Flex-EP*

In this section, we provide a discussion about *Flex-EP* and *Prog-Flex-EP* algorithms in terms of computational cost, parameter selection strategy, and missing sensor readings.

Computational cost. We first look at the computational cost of the algorithms. *Flex-EP* calculates the $W(K)$ for every possible parameter combination. If we have N available parameters and want to choose a subset of K parameters to use, the computational cost is $C_N^K = \frac{N!}{K!(N-K)!}$. In our case, it is C_{16}^4 . Whereas for *Prog-Flex-EP*, it chooses one parameter at a time from the current available parameters; thus, the computation cost for parameter selection is at most $C_N^1 \cdot C_{(N-1)}^1 \cdots C_{(N-K)}^1 = \frac{N!}{(N-K)!}$. Note that we say at most is because sometimes, the algorithm stops before reaching the maximum K parameters if the desired confidence is achieved. *Prog-Flex-EP* refines the

candidate locations and calculates $W(K)$ based on new parameter sets at each round. However, the *Flex-EP* decides the parameter set offline and calculates the location estimation in one round. *Flex-EP* runs faster than *Prog-Flex-EP* in reality.

Parameter selection strategy. Examining the parameter selection, under a given parameter size, *Flex-EP* finds the overall best parameters to use for any localization tasks. On the other hand, *Prog-Flex-EP* is a locally customized method. The parameters chosen by *Prog-Flex-EP* vary based on the environment surrounding each testing point (i.e., each user).

Table 6 presents a part of our running records using the *Prog-Flex-EP-Prob* algorithm as an example. It shows for a testing point which parameter is used in each round and how many points are kept as candidates after each round. We set the maximum number of rounds to 4. We found that the parameters used for different testing points vary largely. The first round is the same for every test because at that time, we have no information about the environment around the testing point. Thus, the algorithm chooses the parameter with the strongest discriminative power. In some cases, the algorithm stopped before the total four rounds as it has already refined to one candidate location, such as the testing points 46, 49, and 50. *Prog-Flex-EP* takes the advantage of the idea that the optimal parameter set is location-dependent, thus giving it the potential to outperform *Flex-EP*, which uses a globally optimized parameter subset.

However, the “greedy” nature of *Prog-Flex-EP* drags down its performance. A “bad” parameter may be chosen at

TABLE 6
Sample Testing Output from *Prog-Flex-EP-Prob* Test, Using Parameter Set {1, 2, 3, 5, 6, 7, 8, 16}, $\tau = 0.95$

Test #		Round				Error (feet)	
		1	2	3	4	-Dist	-Avg
46	Param used	16	2	5		12.0	12.0
	# of pts left	17	6	1			
47	Param used	16	8	5	1	36.0	39.698
	# of pts left	14	13	6	2		
48	Param used	16	1	3	6	12.0	12.0
	# of pts left	19	4	2	1		
49	Param used	16	8	5		114.0	114.0
	# of pts left	16	13	1			
50	Param used	16	2	6		6.0	6.0
	# of pts left	23	7	1			

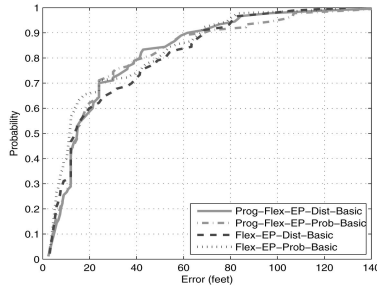


Fig. 18. Comparison of *Flex-EP-basic* using parameter set {1,3,6,14} and *Prog-Flex-EP-basic* with four parameters chosen from input parameter set {1,2,3,5,6,7,8,14} at each testing point.

an early stage, which results in excluding the “good” location candidates prematurely and leads to a larger localization error. Looking at testing point 49 in Table 6, it only uses three parameters and has already achieved the sufficient confidence; however, the error distance is as large as 114 feet. This is an example where “good” location candidates are eliminated too early. Our experimental results in Fig. 18 show that *Prog-Flex-EP* and *Flex-EP* perform quantitatively the same.

Dealing with missing environmental readings. *Prog-Flex-EP* can deal with the situation when not all environmental properties are available at all locations and times. *Prog-Flex-EP* is a sequential localization algorithm: it does not require to make the localization decision in one shot with all the environmental readings being responded by the user. Instead, it can be used in a *challenge-response*-based manner for location refinement. Such an algorithm first asks the user, “tell me the temperature at your location,” and based on the user’s answer, the algorithm localizes the user within some regions, and then, chooses the next parameter to challenge the user. Based on the subsequent answers from the user, the localization results are further refined.

By using this approach, *Prog-Flex-EP* is flexible, especially when dealing with the situation when some environmental properties are not available at specific locations and times. The unavailability can happen to both the sensor network side and the user’s side: 1) On the sensor’s side, it is possible that at a particular region, some sensor readings are unavailable or the sparse network density in that region cannot provide sufficient readings, e.g., RSS is too weak when the sensor is far away from the AP. *Prog-Flex-EP* can avoid using such insufficient parameters, and will instead choose other parameters to ask the user. 2) On the user’s side, if the user cannot provide the parameter requested, the algorithm is flexible to accept other parameters that the user can provide to continue the localization process.

Flex-EP can be modified to deal with the situation where the user cannot provide all the parameters by applying SCWM after getting the user’s vector of available parameters. However, in the case of the sensor’s side, it is possible that some parameters are not available in one region and some other parameters are not available in another region. In this case, *Flex-EP* has to totally discard the use of all these deficient parameters, which will affect the localization performance.

7 RELATED WORK

In this section, we first discuss research efforts in using spatiotemporal information in wireless sensor networks (WSNs). Then, we overview the active research in wireless localization and describe the work that is mostly related to ours.

By utilizing the radio on sensor nodes, it is possible to invert the role of sensor networks, and allow sensor nodes to actuate the environment. Chen et al. [2] utilized sensor networks in an inverted fashion to facilitate new forms of access control that are based on whether a user is located at the right place at the right time. Moreover, Vuran et al. [7] pointed out that sensor observations are highly correlated in the spatial domain. They proposed a theoretical framework to capture the spatial and temporal correlations in WSN and enable the development of efficient communication protocols in WSN utilizing these information. In this work, we explore the possibility of utilizing the physical phenomena monitored by WSN to assist in wireless localization and position verification.

Localization of nodes in WSN has become increasingly important. Localization techniques can be categorized along several dimensions. Want et al. [8] used infrared methods, and the authors in [9], [10] employed ultrasound as the basis for a localization infrastructure. On the other hand, in spite of its several meter-level accuracy, using RSS [1], [5], [11] is attractive because it can reuse the existing wireless infrastructure. Dealing with ranging methodologies, range-based algorithms involve distance estimation to landmarks using the measurement of various physical properties like RSS [1], [5], Time Of Arrival (TOA) [12], and Time Difference Of Arrival (TDOA) [9]. Range-free algorithms [13], [14] use coarser metrics to place bounds on candidate positions. Wu et al. [15] combine the range-based and range-free algorithms. Another method of classifying localization algorithms involves examining the strategy used to map a node to a location. Lateration approaches [12], [14], [16], [17] use the distances to landmarks, while angulation uses the angles from landmarks. Scene matching (or fingerprint matching) strategies [1], [5], [18], [19] use a function that maps observed radio properties to locations on a preconstructed radio map or database. Finally, another dimension of classification extends to aggregate [13], [20], [21] or singular algorithms.

The same type of physical properties is required to be used in each of the above methods to ensure the appropriate functioning of the mechanism. Our work is unique in that our localization approach is generic, i.e., we are not restricted to examining a single type of physical property. The closest works to this paper are [1], [22]. Bahl and Padmanabhan [1] developed a localization mechanism measuring the minimum euclidean distance in the signal space, and only deals with the physical property of RSS. Varshavsky et al. [22] proposed a GSM signal strength fingerprinting-based localization system to determine the current floor of a user. It addressed the problem that certain physical sources may not contribute to localization accuracy by developing a set of feature selection techniques. However, these feature selection techniques did not track the performance of each possible combination in parameter

subsets and might contain “bad” physical sources to start with. Also, Varshavsky et al. [22] only deal with one type of physical property, signal strength. By handling all kinds of physical properties, our work is broader than [1], [22], and our SCWM algorithm for parameter selection is more general than the feature selection approaches in [22]. In addition, our method is novel in that we utilize an existing sensor network to assist in localization, rather than requiring the deployment of a localization infrastructure or additional access points (or landmarks) in the area of interest.

8 CONCLUSION

In this work, we proposed to use the inherent spatial variability in physical phenomena recorded by sensor networks to support wireless localization and position verification. We formulated the problem using a theoretical measurement model to quantify the localizing capability of environmental properties. For parameter evaluation and selection, we proposed a scheme to evaluate the environmental parameters’ ability to capture the physical variability, the SCWM, which can find the optimal parameter subset with the highest discriminative power for localization under a given size of the parameter subset. Moreover, we developed a spectrum of algorithms to perform localization and position verification utilizing parameter subsets obtained from SCWM.

To evaluate the generality of our approach and the effectiveness of SCWM, we conducted experiments in a real office building by collecting various environmental parameters including temperature, humidity, ambient noise, spectrum energy, and RSS over 100 locations. We found that choosing two environmental parameters containing high discriminative power is enough to produce comparable performance to the traditional localization approaches employing RSS with at least four access points. By increasing the number of parameters with high discriminative power in a subset, we can further refine the localization accuracy and obtain better performance than conventional localization results. Thus, our experimental results provide strong evidence of the feasibility of utilizing environmental properties to assist in localization and the effectiveness of our approach by using SCWM and environmental-parameter-based algorithms. Note that there is a trade-off between the localization performance, which is related to the size of the parameter subset, and the computational cost. SCWM can help select appropriate parameter subsets which achieve the localization performance based on application requirements.

APPENDIX

Theorem. *There is no continuous injective function $f : \mathbb{R}^2 \rightarrow \mathbb{R}$.*

Proof. Consider any line l in \mathbb{R}^2 . It is trivially simply connected, and so by the Simply Connected Theorem, its mapping under f is a simply connected subset of \mathbb{R} . But since f is injective, this means that it maps to some nonzero interval on \mathbb{R} . Let z be a nonboundary point along that interval. Then, there exist points p_1 and p_2 on l such that $f(p_1) < z < f(p_2)$. Now, consider any point p_3 in \mathbb{R}^2 not on l . Assume WLOG that $f(p_3) \geq z$, and let m be the line determined by p_1 and p_3 . By the Intermediate

Value Theorem, $[f(p_1), f(p_3)]$ is contained in the image set $f(m)$. So, there exists a point q on m such that $f(q) = z$. But some points on l also map to z , and the only common point of l and m is p_1 . So, we have found two distinct points that both map to z under f . Thus, f is not injective, which is a Contradiction. \square

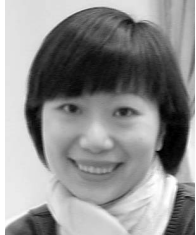
ACKNOWLEDGEMENTS

Preliminary results of this paper have been presented in part in PerCom 2008 [4].

REFERENCES

- [1] P. Bahl and V.N. Padmanabhan, “RADAR: An In-Building RF-Based User Location and Tracking System,” *Proc. IEEE INFOCOM*, pp. 775-784, Mar. 2000.
- [2] S. Chen, Y. Zhang, and W. Trappe, “Inverting Sensor Networks and Actuating the Environment for Spatio-Temporal Access Control,” *Proc. Forth ACM Workshop Security of Ad Hoc and Sensor Networks (SASN)*, pp. 1-12, 2006.
- [3] N. Michalakis, “PAC: Location Aware Access Control for Pervasive Computing Environments,” MIT Laboratory of Computer Science, <http://www.org.lcs.mit.edu/pubs/michalakis.pdf>, 2002.
- [4] S. Chen, Y. Chen, and W. Trappe, “Exploiting Environmental Properties for Wireless Localization and Location Aware Applications,” *Proc. Sixth IEEE Int’l Conf. Pervasive Computing and Comm. (PerCom)*, 2008.
- [5] E. Elnahrawy, X. Li, and R.P. Martin, “The Limits of Localization Using Signal Strength: A Comparative Study,” *Proc. First IEEE Int’l Conf. Sensor and Ad Hoc Comm. and Networks (SECON ’04)*, pp. 406-414, Oct. 2004.
- [6] “What Is Wi-Spy,” <http://www.metageek.net/Products/Wi-Spy>, 2009.
- [7] M.C. Vuran, W.B. Akan, and I.F. Akyildiz, “Spatio-Temporal Correlation: Theory and Applications for Wireless Sensor Networks,” *Computer Networks*, vol. 45, pp. 245-259, 2004.
- [8] R. Want, A. Hopper, V. Falcao, and J. Gibbons, “The Active Badge Location System,” *ACM Trans. Information Systems*, vol. 10, no. 1, pp. 91-102, Jan. 1992.
- [9] N. Priyantha, A. Chakraborty, and H. Balakrishnan, “The Cricket Location-Support System,” *Proc. ACM MobiCom*, pp. 32-43, Aug. 2000.
- [10] A. Ward, A. Jones, and A. Hopper, “A New Location Technique for the Active Office,” *IEEE Personal Comm.*, vol. 4, no. 5, pp. 42-47, Oct. 1997.
- [11] Y. Chen, J. Francisco, W. Trappe, and R.P. Martin, “A Practical Approach to Landmark Deployment for Indoor Localization,” *Proc. Third Ann. IEEE Comm. Soc. Conf. Sensor, Mesh, and Ad Hoc Comm. and Networks (SECON)*, Sept. 2006.
- [12] P. Engge and P. Misra, *Global Positioning System: Signals, Measurements and Performance*. Ganga-Jamuna Pr, 2001.
- [13] Y. Shang, W. Ruml, Y. Zhang, and M.P.J. Fromherz, “Localization from Mere Connectivity,” *Proc. Fourth ACM MobiHoc*, pp. 201-212, June 2003.
- [14] D. Niculescu and B. Nath, “Ad Hoc Positioning System (APS),” *Proc. IEEE Global Telecomm. Conf. (GLOBECOM)*, pp. 2926-2931, 2001.
- [15] H. Wu, C. Wang, and N. Tzeng, “Novel Self-Configurable Positioning Technique for Multi-Hop Wireless Networks,” *IEEE/ACM Trans. Networking*, vol. 13, no. 3, pp. 609-621, June 2005.
- [16] K. Langendoen and N. Reijers, “Distributed Localization in Wireless Sensor Networks: A Quantitative Comparison,” *Computer Networks*, vol. 43, no. 4, pp. 499-518, 2003.
- [17] Z. Li, W. Trappe, Y. Zhang, and B. Nath, “Robust Statistical Methods for Securing Wireless Localization in Sensor Networks,” *Proc. Fourth Int’l Symp. Information Processing in Sensor Networks (IPSN ’05)*, pp. 91-98, 2005.
- [18] M. Youssef, A. Agrawal, and A.U. Shankar, “WLAN Location Determination via Clustering and Probability Distributions,” *Proc. First IEEE Int’l Conf. Pervasive Computing and Comm. (PerCom)*, pp. 143-150, Mar. 2003.

- [19] T. Roos, P. Myllymaki, and H. Tirri, "A Statistical Modeling Approach to Location Estimation," *IEEE Trans. Mobile Computing*, vol. 1, no. 1, pp. 59-69, Jan.-Mar. 2002.
- [20] L. Doherty, K.S.J. Pister, and L. ElGhaoui, "Convex Position Estimation in Wireless Sensor Networks," *Proc. IEEE INFOCOM*, pp. 1655-1663, Apr. 2001.
- [21] C. Wang, H. Wu, and N. Tzeng, "RFID-Based 3D Positioning Schemes," *Proc. IEEE INFOCOM*, pp. 1235-1243, May 2007.
- [22] A. Varshavsky, A. LaMarca, J. Hightower, and E. Lara, "The Skyloc Floor Localization System," *Proc. Fifth IEEE Int'l Conf. Pervasive Computing and Comm. (PerCom)*, pp. 125-134, 2007.



Shu Chen received the BS and MS degrees in computer science and engineering from Harbin Institute of Technology, China, in 2000 and 2002, respectively. She is currently working toward the PhD degree at the Computer Science Department, Rutgers University, and is affiliated with the Wireless Information Network Laboratory (WINLAB). Her research interests include wireless networks, network and system security, and information privacy. She is a student member of the IEEE.



Yingying Chen received the PhD degree in computer science from Rutgers University. She is currently an assistant professor in the Department of Electrical and Computer Engineering at Stevens Institute of Technology. Her research interests include wireless and system security, wireless networking, and distributed systems. She is the coauthor of the book *Securing Emerging Wireless Systems* (Springer, 2009). Prior to joining Stevens Institute of Technology, she was with Bell Laboratories and the Optical Networking Group, Lucent Technologies. She received the IEEE Outstanding Contribution Award from the IEEE New Jersey Coast Section each year from 2000 to 2005. She is the recipient of the Best Technological Innovation Award from the Third International TinyOS Technology Exchange in 2006. She is member of the IEEE.



Wade Trappe received the BA degree in mathematics from The University of Texas at Austin in 1994, and the PhD degree in applied mathematics and scientific computing from the University of Maryland in 2002. He is currently an associate professor in the Electrical and Computer Engineering Department at Rutgers University, and is the associate director of the Wireless Information Network Laboratory (WINLAB). His research interests include wireless security, wireless networking, multimedia security, and network security. While at the University of Maryland, he received the George Harhalakis Outstanding Systems Engineering Graduate Student Award. He is a coauthor of the textbook *Introduction to Cryptography with Coding Theory* (Prentice Hall, 2001). He is the recipient of the 2005 Best Paper Award from the IEEE Signal Processing Society. He is a member of the IEEE Signal Processing and Communications societies, a member of the ACM, and a member of the IEEE.

▷ **For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.**