Unsupervised Learning for Crowdsourced Indoor Localization in Wireless Networks

Suk-hoon Jung, Byung-chul Moon, and Dongsoo Han, *Member, IEEE*

Abstract—Wireless Local Area Network (WLAN) location fingerprinting has become a prevalent approach to indoor localization. However, its widespread adoption has been hindered by the need for manual efforts to collect location-labeled fingerprints for the calibration of a localization model. Several semi-supervised learning methods have been applied to reduce such manual efforts by exploiting unlabeled fingerprints, but they still require some amount of labeled fingerprints for initializing the learning process. In this research, in order to obviate the need for location labels or references, we propose a novel unsupervised learning method that calibrates a localization model using unlabeled fingerprints based on a hybrid global-local optimization scheme. The method determines the optimal placement of fingerprint sequences on an indoor map, under the constraint imposed by the inner structure shown on the map such as walls and partitions. An efficient interaction between a global and a local optimization in the hybrid scheme drastically reduces the complexity of the learning task. Experiments carried out in a single- and a multi-story building revealed that the proposed method could successfully build a precise localization model without any location reference or explicit efforts to collect labeled samples.

Index Terms—Location estimation, Wi-Fi fingerprint, crowdsourcing, radio map construction, unsupervised learning

1 Introduction

The recent explosive proliferation of wireless devices and WLANs (based on the IEEE 802.11 standard) is accelerating the demand for practical location-based applications in indoor environments. In such applications, the identification of a user's location in an indoor area is critical because the Global Positioning System (GPS) is usually unavailable due to signal blocking. Instead, WLAN infrastructure allows a wireless device to be localized by referring to the Received Signal Strength Indicator (RSSI) in an indoor environment.

Among RSSI-based techniques, fingerprinting is known to be the most accurate and popular approach to indoor localization [1]. The RSSI fingerprinting-based techniques, however, require an initial training or calibration phase in which RSSI measurements are collected at known locations. Then, in the localization phase, the location of a device is estimated by matching an online RSSI measurement with the training data [2], [3].

The need for calibration is a major drawback of fingerprinting-based techniques, because this step involves laborious and time-consuming manual effort to collect location-labeled training data. The manual calibration must be conducted for every new building and should be repeated whenever the training data become outdated due to changes in the WLAN environment. The cost of manual calibration thus hinders the widespread adoption of fingerprinting-based indoor localization.

In endeavors to reduce the calibration efforts, several studies have been carried out on crowdsourcing approaches in which general users can participate in the data collection activity. Implicit crowdsourcing, besides explicit contribution approaches [4], [5], [6], has been studied to make use of RSSI measurements contributed during the normal operation of wireless devices [7], [8]. This type of data can be viewed as unlabeled samples since the true positions from which the measurements are obtained are unknown. Therefore, the issue that must be addressed is the assignment of correct location labels to the unlabeled samples for the calibration of localization models.

Additional data from inertial sensors embedded in smartphones can be used for estimating the unknown location labels [7], [8], [9], [10], [11], [12]. Although these methods can reduce the calibration efforts to some extent, the engagement of additional sensors raises new issues, such as accuracy, availability, device heterogeneity, and additional power consumption of the sensors.

Another research stream is focused on semi-supervised learning techniques that utilize both labeled and unlabeled samples [13], [14], [15], [16], [17], [18], [19]. These studies employ optimization techniques to estimate location labels of unlabeled samples based on RSSI values. However, without a good initial model, the techniques encounter difficulties in finding the global optimal solution and projecting a learned model onto an indoor map, and thus require some amount of labeled samples for constructing the initial model.

In this paper, we propose an unsupervised learning method, named Unsupervised Calibration based on a Memetic Algorithm (UCMA), to build a precise indoor localization model using only unlabeled fingerprints. UCMA

• *The authors are with the Department of Computer Science, Korea Advanced Institute of Science and Technology, N2 CS723, 291 Daehak-ro, Yuseong-gu, Daejeon 305-701, Republic of Korea. E-mail: {sh.jung, chul7672, dshan}@kaist.ac.kr.*

Manuscript received 24 Oct. 2014; revised 3 Nov. 2015; accepted 25 Nov. 2015. Date of publication 8 Dec. 2015; date of current version 28 Sept. 2016. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TMC.2015.2506585

avoids the necessity of location labels by applying a hybrid global-local optimization scheme in which the initial model is provided by a global search algorithm without reference to location information. Once the initial model is provided, location labels are estimated in a similar manner to the previous learning-based methods by a local optimization algorithm in the hybrid scheme. The global search and local optimization algorithms are integrated into a Memetic Algorithm (MA) [20], which is an evolutionary approach that provides an efficient way to address optimization problems through the interaction between global and local optimizations.

Unlike previous strategies proposed to reduce calibration efforts, UCMA does not require any prior knowledge of the WLAN environment, the involvement of additional sensors, or explicit efforts to collect labeled samples from the building of interest. As in previous studies, it is assumed that users turn on their WLAN modules to contribute their traces of RSSI measurements, i.e. *user traces*, while carrying wireless devices in a building. Another prerequisite of UCMA is an indoor map of the building for initialization. These two conditions usually are not considered to be a part of calibration efforts because online measurements and an indoor map are essential elements required in the localization phase for most location-based applications, such as navigation [8].

Unsupervised learning techniques have not been fully applied to the calibration of localization models. Two main problems should be addressed to avoid the need for location-labeled data and to perform the unsupervised learning of a localization model. The first is the mapping of a learned model onto an indoor space. In the previous learning-based methods, this has been solved by using location-labeled samples as reference points for the mapping. UCMA solves this problem by incorporating the structural information of an indoor area and human mobility constrained by the structure. Once the indoor map of a building and unlabeled user traces are given, UCMA arranges the traces to fit into the inner structure shown in the map, like fitting pieces into a puzzle. Similar approaches have been used before in a few studies [8], [10], [13]. However, the suggested approaches still depend on some amount of labeled samples [13] or data from inertial sensors [8], [10] because of the second problem.

This problem is related to the size and the complexity of a solution space to be dealt with. In general, the solution space of a location assignment problem is huge and complex since it comprises all possible assignments of location labels to given samples. Suppose that 1,000 unlabeled samples are collected from a building composed of 100 locations. There are then $100^{1,000}$ possible solutions to the assignment problem. With such a huge solution space, optimization algorithms usually fail to find the global optimal solution or do not terminate. To address this problem, an effective configuration of the global search and local optimization algorithms was deliberately devised in the proposed hybrid scheme. Under the configuration, only the solutions that do not violate the nature of signal propagation are discovered and evaluated during the interaction between the global and local optimizations. In this way, the solution space is effectively restricted to a much smaller space, and thus a localization model can be constructed via unsupervised learning.

To validate UCMA, we deployed a prototype system and conducted extensive experiments in a medium- and a large-scale building. The experimental results revealed that UCMA could build a precise localization model using unlabeled user traces. In the medium-scale building, a localization test of UCMA yielded average errors of 2.7-3.7 m under various conditions, a level comparable to the error of 1.6-2.8 m achieved with the ground-truth model. In the large-scale building where the ground-truth model yielded average errors of 2.0-3.7 m, UCMA yielded average errors of 3.1-4.6 m. These results indicate that a localization service can be provided by implicit crowdsourcing, where training data are gathered during the normal operation of wireless devices. In this way, the cost of building an indoor localization system can be dramatically reduced.

2 Related Work

2.1 RSSI-Based Indoor Localization

2.1.1 Trilateration-Based Approach

This approach estimates the location of a wireless device based on a mathematical principle called "trilateration". It assumes that APs' installed locations are known, and estimates the location of a device based on the distances between the device and the APs. Their distances can be calculated using the correlation between signal strength and distance given a propagation model [21].

Once the distance is determined, a circle (sphere) with the AP as its center, and the distance as its radius can be generated. Since the device is assumed to be located on the circumference, its location is estimated to the point where the circles are crossed one another. At least $n + 1$ APs are needed to calculate a location in n dimensions, e.g., three APs are required in 2D space, and four, in 3D.

Trilateration-based techniques are simple and require a little calibration effort. However, trilateration-based techniques are known to show comparably low accuracy [22]. Moreover, most of the APs' locations are not known in reality because many different vendors and providers are usually involved in the installation of APs for public areas such as large-scale indoor shopping malls.

2.1.2 Fingerprinting-Based Approach

RSSI fingerprint matching has been used as the basic scheme of many indoor localization systems these days. Here, indoor area of a building is usually represented as the set of discrete locations, and machine-learning techniques are often used to build a localization model from training data. In the localization phase, the methods estimate the most likely location by matching the online RSSI measurement with fingerprints in the trained model. RADAR, one of the pioneering fingerprinting-based systems, estimates the location of a device based on k-Nearest Neighbor (kNN) averaging [2]. While RADAR uses a simple, deterministic localization technique, researchers have developed more sophisticated fingerprinting-based techniques using neural networks [23], probabilistic methods [24], or fuzzy logic [25].

Although various techniques have been proposed for localization, the accuracy of localization is highly dependent on the quality and quantity of training data which comprise

RSSI measurements and their location labels. The training data set is usually collected through a labor-intensive manual calibration. Furthermore, this manual calibration must be repeated if the training data are outdated due to changes in the WLAN environment such as addition, removal, and relocation of APs.

2.2 Reducing Calibration Efforts

Researchers have started to find ways to reduce calibration efforts for RSSI fingerprinting-based localization. They have been trying to build a precise model using only a few labeled data or to collect a large amount of training data with much less cost.

Unlabeled measurements are easier to collect than labeled ones, and they also can be collected from general users while localization service is provided. When a small portion of training data are labeled with locations, semi-supervised learning techniques can propagate location labels to unlabeled measurements in the data set. Manifold learning or dimensionality reduction techniques have often been used for the semi-supervised training of a localization model [14], [15], [16], [17], [18], [19]. These techniques project high dimensional data (RSSI measurements) onto a low-dimensional space (the physical space of a building) by aligning unlabeled measurements with a small amount of labeled ones. In [13], expectation maximization algorithm was applied to exploit unlabeled measurements for the enhancement of a pre-built, incomplete localization model. The semi-supervised techniques have made a considerable progress in reducing manual labeling efforts. However, these techniques still require some amount of already-known location labels or prior knowledge of the AP deployment for a model calibration.

In addition to the semi-supervised learning techniques, researchers have developed crowdsourcing systems to collect a large amount of training data at low cost. The training data for conventional localization systems are collected by experts who can assign precise locations to collected measurements. Crowdsourcing systems usually facilitate this assignment procedure, so that even untrained users can participate in the data collection activity [4], [5]. Several systems have chosen coarse location granularity such as rooms or places for easier location assignment [6]. Some other systems encourage participants to specify prominent locations such as entrances and corners, and then infer the location labels of RSSI measurements using timestamps or inertial sensors (e.g., an accelerometer, a compass, a gyroscope) embedded in devices [26].

While inertial sensors are usually used to assist manual operations, there have been attempts to infer the location labels only from inertial sensor readings [7], [11]. Zee [8] and LiFS [9], [10] estimate the relative formation of RSSI measurements in user traces using inertial sensors and then infer their absolute positions by placing the traces on an indoor map. The constraints imposed by the indoor layout are referred to in the placement step. In addition to the sensing data from inertial sensors, occasionally captured GPS signals can be used to label RSSI measurements [12]. Although these methods can reduce the efforts of participants to some extent, they raise new issues, such as power consumption, availability, device heterogeneity, and accuracy of sensors, because of the engagement of additional sensors. Furthermore, some sensor readings should be adjusted to user-specific features (e.g., the step length). These issues limit the use of additional sensor-based systems in real fields.

UCMA does not require any knowledge of AP deployment, labels of RSSI measurements, or deliberate handling of additional sensors. Because of this advantage, UCMA can dramatically reduce the cost of training a localization model to almost zero.

3 Unsupervised Calibration Using Unlabeled User Traces

3.1 Problem Statement

Consider a localization problem in the three-dimensional indoor space of a multistory building. Formally, we model the physical area as a finite location-state space L, which is a set of physical locations with x, y, and z coordinates:

$$L = \{l_1 = (x_1, y_1, z_1), l_2 = (x_2, y_2, z_2), \ldots, l_n = (x_n, y_n, z_n)\},$$

where coordinates denote the center of a location cell.

All possible RSSI measurements are modeled as a finite observation space $O = \{o_1, o_2, \ldots, o_m\}$. Suppose k APs are installed in the building. Each observation $o$is represented as a vector of RSSI values from the APs, i.e., a k-dimensional vector,

$$o = < rssi_1, rssi_2, \ldots, rssi_k >,$$

where $rssi_i$ is the RSSI value of AP i. The RSSI value typically ranges from 0 to -100 dBm for 802.11 wireless signals.

To train a localization model, an observation o should be labeled with l, where l is the location in which o is scanned. Manual labeling is labor-intensive, whereas collecting just observations is relatively easy because they can be extracted from user traces obtained while the localization service is provided. A user trace is a sequence of unlabeled observations, and it can be represented by a vector $u = < o_1^u, o_2^u, \ldots, o_t^u >$, where t is the time index.

If the collected location information is not available, user traces cannot be directly used as calibration data for the model training. Therefore, the issue to address is to build a localization model without location labels, or to assign correct location labels to unlabeled observations.

The proposed method solves these issues by finding a hidden structure of user traces that fits into the inner structure of a building. We adopt Hidden Markov Model (HMM) to describe user traces constrained by the inner structure. The structural relation of indoor locations are modeled by a set of finite location-states with their constrained transitions in an HMM. The problem is then reduced to the HMM training problem, a problem of estimating HMM model parameters that best describe the given set of unlabeled traces.

3.2 Modeling User Traces Using HMM

HMM is a well-known technique in temporal pattern recognition, and it has been successfully applied to indoor localization and tracking [30], [31]. An HMM is a stochastic finite state machine in which the internal states are hidden, and only the outputs of the states are observable. In HMM modeling for RSSI-based indoor localization, the hidden

TABLE 1
Summary of HMM Notations

L	The location-state space for n locations in an environment, $L = \{l_1, l_2, \ldots, l_n\}$
O	The observation space, $O = \{o_1, o_2, \ldots, o_m\}$
π	An initial probability distribution on the location-states, $\pi = \{\pi_{l_1}, \pi_{l_2}, \ldots, \pi_{l_n}\}$
A	A transition probability matrix, $A = (a_{i,j})_{i,j \in L}$
B	A set of emission probabilities, $B = \{Pr(o_j \mid l_i) \mid o_j \in O, l_i \in L\}$
λ	HMM model parameter set, $\lambda = <\pi, A, B>$
u	Unlabeled user trace, $u = <o_1^u, o_2^u, \ldots, o_t^u>$, $o_i^u \in O$
U	Set of unlabeled user traces
p	Sequence of locations, simply called path
p^*	Viterbi path, $p* = <l_1^{p*}, l_2^{p*}, \ldots, l_t^{p*}>$
P^*	Set of Viterbi paths for U
$rssi_i^x$	RSSI value of AP i in an observation x
μ	Set mean observations for L, $\mu = \{\mu_{l_1}, \mu_{l_2}, \ldots, \mu_{l_n}\}$
M_l	Set of observations assigned with l

states represent the possible locations in the space of interest, and RSSI measurements of a user trace are treated as a sequence of observations emitted from the hidden states [13]. More formally, an HMM model for an indoor environment is defined by a quintuple $< L, O, \pi, A, B >$, where each element is defined in Table 1. For clarity, all the HMM notations used in this paper are summarized in Table 1.

The model is defined on location-state space L and observation space O, both of which have been discussed in Section 3.1. The initial probability distribution π provides information on the starting probability of each state; π was equally set in our model because we assumed that every location is equally likely to be a starting point.

The transition probability matrix A specifies how a user moves around in the space. In an indoor environment, user movement is subject to the constraints imposed by the indoor layout, (i.e., a user cannot walk through walls and other barriers). In addition, since users have limited mobility, they can move from a location only to its nearby locations in a time interval of two successive observations. Therefore, with the prior knowledge given by an indoor map and from human mobility settings, a transition probability matrix A can roughly be defined by assigning equal probability to the possible transitions out of a location-state. The set of emission probabilities B gives the likelihood of an observation $o \in O$ at a location $l \in L$, which will be defined in the next section. In an HMM, a triple $\lambda = <\pi, A, B>$ denotes the model parameter, and it is adjustable through training.

When a model parameter $\lambda = <\pi, A, B>$ is given, a user trace $u = <o_1^u, o_2^u, \ldots, o_t^u>$ can be tracked in the model by the well-known Viterbi algorithm. The algorithm finds the most likely state sequence $p* = <l_1^{p*}, l_2^{p*}, \ldots, l_t^{p*}>$ called a *Viterbi path*, which satisfies (1). The function (1) is called *state optimized likelihood function* [27].

$$Pr(u, p^* \mid \lambda) = \max_p Pr(u, p \mid \lambda) \tag{1}$$

4 Hybrid Global-Local Optimization Scheme for Unsupervised Calibration

In conventional HMM-based localization, HMM model parameters are calculated by supervised or semi-supervised learnings using location-labeled samples [13], [30], [31]. However, in the case of employing user traces, an HMM should be trained in an unsupervised manner because location labels are not included in the traces. When a set of unlabeled observation sequences is given for training, the HMM training has to address the problem of finding the Maximum of the Likelihood function (ML) of the observations. Although there is no analytical method to compute the global ML, it is possible to find a local ML if an initial guess of HMM model parameters $\lambda^0 = <\pi^0, A^0, B^0>$ is given [27]. Several heuristic algorithms such as the Baum-Welch [28] and the Segmental K-means (SK) [29] have been developed to train an HMM from unlabeled samples. We adopt the SK algorithm because it spontaneously solves the location-label assignment problem, which is of our main interest, during the HMM training procedure.

Typically, SK is executed several times with random initial parameters to avoid being stuck in local optima far from the global optimum [27]. However, such a scheme often fails when a complex model like localization model is involved. To overcome this local-optima problem, we developed a more advanced method based on MA, called Unsupervised Calibration using MA (UCMA). UCMA is a hybrid global-local optimization process where a global evolutionary search constructs and iteratively improves initial models, and SK performs local optimizations with the constructed initial models. More specifically, the global search is responsible to provide B^0 because the other initial parameters, π^0 and A^0, has been determined for an indoor area in Section 3.2.

4.1 Likelihood Function

As in a general HMM training, the likelihood function of an HMM is used as an evaluation function in UCMA. In a general HMM scheme, the likelihood of an observation sequence u and a state sequence p is defined as $Pr(u, p \mid \lambda)$, the joint probability with respect to a model parameter λ. It is equal to $Pr(u \mid p, \lambda) Pr(p \mid \lambda)$, and calculated using (2) and (3) [27].

$$Pr(u \mid p, \lambda) = \prod_{i=1}^{t} Pr(o_i^u \mid l_i^p). \tag{2}$$

$$Pr(p \mid \lambda) = \pi_{l_1^p} \prod_{i=2}^{t} a_{l_{i-1}^p, l_i^p}. \tag{3}$$

While (3) is computed using π and A, the calculation of (2) needs $Pr(o \mid l) \in B$ to be known. Usually, localization approaches using probabilistic matching calculate $Pr(o \mid l)$ based on RSSI distributions for each AP at location l [13], [24]. Utilizing RSSI distributions has advantages in dealing with the uncertainty in wireless signals, but it requires a large amount of training data and a long computation time, which are the main concerns in the current study. Therefore, we calculate $Pr(o \mid l)$ based on the Euclidian distance, which is one of the simplest and most widely used metrics for fingerprinting [2].

The Euclidian distance between observation o and location l can be computed in the signal space with the mean observation $\mu_l = <rssi_1^{\mu_l}, rssi_2^{\mu_l}, \ldots, rssi_k^{\mu_l}>$ assigned at location l, as follows:

$$Dist(o, l) = \sqrt{\sum_{i=1}^{k} (rssi_i^o - rssi_i^{\mu_l})^2}. \tag{4}$$

In this equation, $rssi_i^x$ refers to the RSSI value of AP i in observation x.

For indoor localization, the estimated location is the one that can yield the minimum value of (4). In our model, in contrast, the Euclidean distance is transformed further to a probability $Pr(o|l)$ for the HMM scheme. First, the Euclidian distance for a certain o is converted to a similarity measure ranging from 0 to 1 as follows:

$$Sim(o,l) = \frac{1}{1 + Dist(o,l)}. \tag{5}$$

Based on $Sim(o,l)$, a probability $Pr(l|o)$ is derived by normalization as follows,

$$Pr(l|o) = \frac{Sim(o,l)}{\sum_{i=1}^{n} Sim(o,l_i)}, \tag{6}$$

which is the probability that observation o has been obtained at location l. The probability $Pr(o|l)$ at location l is then calculated using Bayes rules, as follows:

$$Pr(o|l) = \frac{Pr(l|o)Pr(o)}{Pr(l)}. \tag{7}$$

$Pr(l)$ encodes prior knowledge about where a device may be located, and it usually is set as a uniform distribution assuming every location is equally likely to be decided. For an observation o, $Pr(o|l)$at any location l is then can be calculated using Euclidian distance (4), if the set of mean observations $\mu = \{\mu_{l_1}, \mu_{l_2}, \ldots, \mu_{l_n}\}$ is given for location-states.

Using (7), equation (2) can be rewritten as follows:

$$\begin{aligned} Pr(u|p,\lambda) &= \prod_{i=1}^{t} \frac{Pr(l_i^p|o_i^u)Pr(o_i^u)}{Pr(l_i^p)}, \\ &= c\prod_{i=1}^{t} Pr(l_i^p|o_i^u) \end{aligned} \tag{8}$$

where c is

$$c = \prod_{i=1}^{t} \frac{Pr(o_i^u)}{Pr(l_i^p)}. \tag{9}$$

For a user trace u, c is a constant for any choice of p, because $Pr(l)$ is uniform at all locations, and $Pr(o_i^u)$ is a constant related only to the given u. Since we are interested in the comparative probability of taking a location-state sequence to find the optimal state sequence of u, a constant offset is not important and can be ignored. Accordingly, for the likelihood function, the scaled probability $Pr^s(u,p|\lambda)$ defined as (10) is used instead of $Pr(u,p|\lambda)$.

$$\begin{aligned} Pr^s(u,p|\lambda) &= \frac{1}{c} Pr(u,p|\lambda) \\ &= \left(\prod_{i=1}^{t} Pr(l_i^p|o_i^u)\right) Pr(p|\lambda). \end{aligned} \tag{10}$$

4.2 Local Optimization via Segmental K-means Algorithm

When a set of unlabeled user traces U is given, the SK adjusts λ^0 to find the λ^* which maximizes $Pr^s(U,P^*|\lambda^*)$, where $p_i^* \in P^*$ is the optimal state sequence of $u_i \in U$, as given by the Viterbi algorithm on λ^*. We are interested in P^* as well as the optimized parameter λ^* itself because P^* denotes the location-label estimation of the unlabeled training data.

The SK repeatedly improves the model parameter from λ^{t-1} to λ^t such that $Pr^s(U,P^{t-1}|\lambda^{t-1}) \leq Pr^s(U,P^t|\lambda^t)$ until $P^{t-1} = P^t$. When an initial parameter $\lambda^0 = <\pi^0, A^0, B^0>$ is given, the algorithm consists of the following steps:

1. Find the Viterbi path p_i^t for each user trace $u_i \in U$ conditioning on λ^t.
2. Reassign the optimal location-state for each observation in U if it differs from that in P^{t-1}. Otherwise, if $P^{t-1} = P^t$, stop the iteration. (Henceforth, we denote with M_l^t a set of observations that are assigned with state l at time t)
3. Update the model parameter $\lambda^{t+1} = <\pi^{t+1}, A^{t+1}, B^{t+1}>$ for the next iteration.
   - 3.1 Calculate the initial probabilities and the transition probabilities:
     For each $l \in L$,
     $$\pi_l^{t+1} = \frac{|\{o_1^u \in M_l^t | u \in U\}|}{|U|}. \tag{11}$$
     For $i, j \in L$,
     $$a_{i,j}^{t+1} = \frac{|\{<o_k^u, o_{k+1}^u> \ | o_k^u \in M_i^t, o_{k+1}^u \in M_j^t, u \in U\}|}{|M_i^t|} \tag{12}$$
     .
   - 3.2 Calculate the mean observation set μ^{t+1}:
     For each $l \in L$,
     $$\mu_l^{t+1} = \frac{1}{|M_l^t|} \sum_{o \in M_l^t} o. \tag{13}$$
   - 3.3 For each location, calculate an emission probabilities of every observations in U. This is done by (4), (5), (6), and (7) defined in Section 4.1, based on the mean observations μ^{t+1} given by (13).
4. Repeat the iteration from Step 1.

Eventually, for the given λ^0, the SK produces a local optimal model λ^* describing unlabeled user traces U. In our modeling, λ^0 is specified by μ^0 with the predetermined probabilities π^0 and A^0. On an arbitrary input μ^0, we denote it with $SK(\mu^0, U)$, the SK local optimization procedure using user traces U. Along with λ^*, $SK(\mu^0, U)$ produces local optimal mean observations μ^*, local optimal location-state sequences P^*, and a scaled probability $Pr^s(U,P^*|\lambda^*)$. These outputs are referred to in the later steps.

Let n be the number of location-states and m be the number of observations in U. The time complexity of an SK is generally $O(n^2m) + T_{update}$ for each iteration, where $O(n^2m)$ is the time required for the Viterbi algorithm and T_{update} is the time required to update the model parameters. In our model, because the possible state transitions are limited to a small size by the prior knowledge on an indoor map, the time complexity of the Viterbi algorithm is $O(cnm)$, where c is the average number of possible transitions starting from a location ($c < 7$ in our experiments).

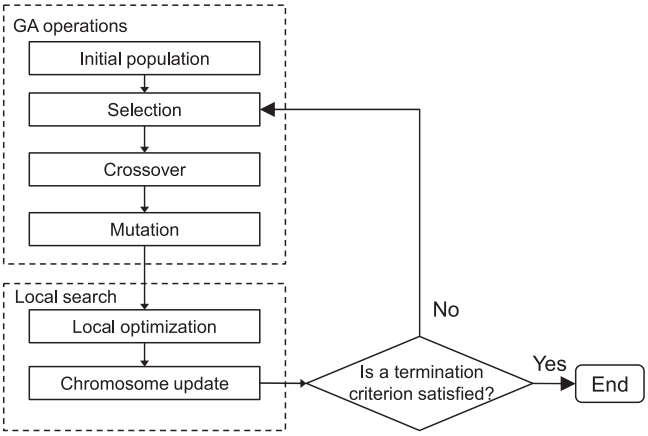


Fig. 1. Flowchart of a memetic algorithm.

The time complexity of T_{update}, especially for (4) in the emission probability calculation, is $O(knm)$, where k is the number of APs found in the environment. Hundreds of APs could be found in a large building, which increases T_{update}. In order to reduce the time complexity, we modified the algorithm to compute the emission probability based on *medoids* instead of the mean observations. The *medoid* of a location l is one of the observations in M_l^t that is closest to the mean observation μ_l. For this purpose, μ_l^{t+1} in Step 3.2 is replaced by a *medoid* after the calculation of (13) as follows:

$$\mu_l^{t+1} = \arg\min_{o \in M_l^t} Dist(o|l). \tag{14}$$

It is not necessary to recalculate distances using (4) at each iteration because a *medoid* is one of the observations in U. Instead, a pre-processed distance matrix can be reused. Consequently, the time complexity of the SK becomes $O(cnm)$ for each iteration cycle, which is linearly proportional to the number of observations and locations. It also is linearly proportional to the number of possible transitions which is the multiple of c and n.

4.3 Global Optimization via UCMA

MA is a population based evolutionary algorithm augmented with a local search [20]. Fig. 1 presents the flowchart of a general MA process. It consists of two parts: Genetic Algorithm (GA) operations and a local search. The initial population is usually generated randomly or by a specific manner. For one generation, an evolutionary process is carried out on the population by the GA operations, including selection, crossover, and mutation. After completing the GA operations, each individual in the population is further enhanced during the local search procedure. By doing so, the individuals become much closer to the global optimal solution. The enhancement of an individual is then updated to its chromosome, so that its descendants can inherit the enhanced trait, unlike usual GAs. Finally, the evolutionary process is terminated if a termination criterion is satisfied.

In the population, every individual represents a solution with a corresponding fitness value derived from a fitness function that evaluates the quality of the solution. The criterion used in the SK, $Pr^s(U, P^*|\lambda)$, is the fitness function in UCMA.

4.3.1 Genetic Representation

Designing a genetic representation adapted to a specific problem is essential for MAs and GAs. A better genetic representation can improve the performance of evolution. With a genetic representation, a solution is encoded in a *genotype* (i.e., a chromosome of an individual), and inversely the genotype also can be decoded to a solution (i.e., a *phenotype*).

To address the local-optima problem of the SK, a candidate phenotype considered in UCMA is the set of mean observations $\mu = \{\mu_{l_1}, \mu_{l_2}, \ldots, \mu_{l_n}\}$, where each element is a vector of mean RSSI values for k APs detected in a target indoor environment. A genotype may have the same format as that of the phenotype with a limited range of RSSI values. In this case, however, the genotype space is extremely large. Dealing with such a huge genotype space may not be practical, especially in the case that the environment includes a large number of locations and APs. A compact genotype representation that can enhance the evolvability and performance of UCMA is required.

UCMA referred to the physics of signal propagation to design a compact genotype representation. Wireless signals are likely to follow some propagation model in an indoor space, and therefore a signal propagation model with parameters specifying the disposition of APs can approximately describe the WLAN environment in the indoor area. Such an abstract model may not be precise enough to be used as a localization model, because signals are usually distorted from their ideal forms indoors. Fortunately, however, an initial model for the SK does not need to be precise because it will be further improved through the local optimization procedure. Based on this feature, we configure the optimization process of UCMA so that the GA explores the possible abstract models, and the SK is responsible for discovering their distortions after converting to a more sophisticated model, HMM. Under this configuration, only the solutions that do not violate the nature of radio propagation are discovered and evaluated, and therefore the complexity of the learning task is effectively reduced. Another advantage of this strategy is that the efficiency of the learning task can be further improved by incorporating an AP selection technique that reduces the number of features defining the abstract model.

To support the above process, a genotype encodes an AP deployment L^{ap} and a propagation model PM. The AP deployment L^{ap} is a location (location-state) assignment $L^{ap} = < l_1^{ap}, l_2^{ap}, \ldots, l_k^{ap} >$ for k APs found in the training data set. For the propagation model, we use a standard radio propagation model [21], defined by,

$$PW_r = PW_t + 20 \log\left(\frac{wl}{4\pi}\right) + 10e \log\left(\frac{1}{d}\right), \tag{15}$$

where PW_t is the transmit power of an AP in dBm; PW_r is its RSSI at a device; wl is the wavelength ($wl = 0.125m$ for 2.4 GHz); e is a path-loss exponent; and d is the distance between the AP and the device. Given an AP deployment L^{ap}, an attenuated RSSI, PW_r, at a location l from an AP is calculated based on the variables PW_t, wl, d, and e. The transmit power PW_t is assumed to be equal to the maximum RSSI value of the AP in the data set. The wavelength wl is known,

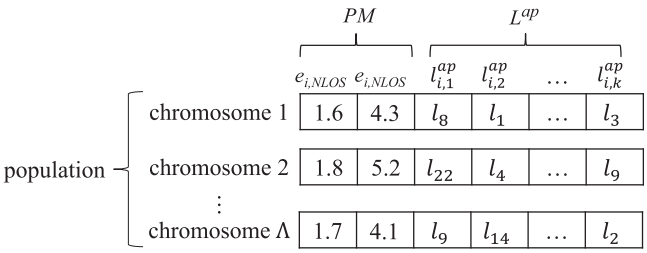


Fig. 2. Genetic representation in UCMA.

and the distance d is calculated between the location l and the AP location in L^{ap}. The only variable depending on the environment is then the path-loss exponent e. The value of e is known to vary from 1.6 to 1.8 for Line-Of-Sight (LOS) and from 4 to 6 for Non-LOS (NLOS) in indoor environments [32]. Therefore, the signal propagation model PM can be defined by the pair of path-loss exponents (e_{LOS}, e_{NLOS}) for LOS and NLOS, where $e_{LOS} \in [1.6, 1.8]$ and $e_{NLOS} \in [4, 6]$.

The genotype representation of UCMA is depicted in Fig. 2. A genotype (chromosome) consists of two parts, alleles representing a propagation model PM and alleles representing an AP deployment L^{ap}. Alleles $e_{i,LOS}$ and $e_{i,NLOS}$ denote the path-loss exponents for LOS and NLOS, respectively, in chromosome i, and the allele $l^{ap}_{i,j}$ denotes the location of AP j. This type of information encodes a hypothetical indoor WLAN environment and can be decoded to a phenotype, a set of mean observations, by computing attenuated RSSIs of each AP at each location-state using (15).

Since a genotype is a combination of L^{ap} and PM, the size of the genotype space is $d_{e_{LOS}} d_{e_{NLOS}} n^k$, where d_x is the number of possible values in the range of x. It can be much smaller than the size of the phenotype space d^{nk}_{rssi}.

We chose the simplest radio propagation model on purpose to evaluate the feasibility of UCMA under general settings. More sophisticated signal models for indoor environments, such as log-distance path loss model and ITU model [21], are other options for the genotype representation. In those cases, GA can generate initial models in a more precise form, but additional variables are required for the genotype representation.

$SK(\mu^0, U)$ enhances a phenotype μ^0 into μ^* in the local optimization procedure. The enhanced phenotype must be encoded in a genotype to be updated for the next generation. To encode a phenotype into a genotype, we approximate AP deployment L^{ap} appearing in the enhanced phenotype μ^* as follows.

$$l^{ap}_{i,j} = \arg\max_{l \in L} \left(rssi_j^{\mu^*_l} \right), \tag{16}$$

where $\mu^*_l \in \mu^*$, and μ^* is the enhanced phenotype of chromosome i.

In the prior discussion, we assumed that the transmit power of an AP is equal to its maximum RSSI value in the data set. In other words, it is regarded that the AP is located where the maximal RSSI value is observed. The equation (16) finds a location whose mean observation μ^*_l has maximum RSSI for AP j, and updates the location to chromosome i.

Path loss exponents (e_{LOS}, e_{NLOS}) appearing in μ^* are also approximated for the update. By minimizing the mean square error between the RSSI in μ^* and the estimation using (15), the path loss for AP j, e_j, for chromosome $i$can be solved by the equation in the below:

$$f(e_j) = \sum_{l \in L} \left(rssi_j^{\mu^*_l} - PW_{t,j} + 20 \log\left(\frac{wl}{4\pi}\right) + 10 e_j \log\left(\frac{1}{d_l}\right) \right)^2, \tag{17}$$

where d_l is the distance between $l$and updated AP location $l^{ap}_{i,j}$, and $PW_{t,j}$ is the transmit power of AP j (the maximum RSSI value in the data set). By equating the derivative of $f(e_j)$ to zero, an estimate of the path loss for AP j can be found [21]. In this research, (17) is calculated after dividing locations L into locations on LOS and NLOS for each AP by referring to the indoor layout. The estimated path losses for all APs are then averaged to obtain e_{LOS} and e_{NLOS}, respectively.

4.3.2 Evolutionary Process

The GA operations comprise selection, crossover, and mutation functions. UCMA adopted a general strategy for each of these three operations: roulette wheel selection, uniform crossover, and random mutation respectively.

The evolutionary process of an MA starts with an initial population of chromosomes. UCMA generates the initial population randomly and iterates as follows:

1. Compute the phenotype μ^0 for each genotype by calculating attenuated RSSIs for each AP and at each location-state using (15).
2. Enhance each phenotype μ^0 into μ^* using $SK(\mu^0, U)$.
3. Update each genotype based on the enhanced phenotype μ^* using (16) and (17).
4. Build a new population by means of the GA operations.
5. Stop if a termination criterion is satisfied, or repeat the iteration from step 1.

In the process of a local optimization, intermediate and final changes of a phenotype are evaluated, and the final evaluation score is transferred to the updated genotype. Eventually, through the evolution, UCMA finds the best fit whose phenotype μ^* is a set of estimated optimal mean observations for all location-states. At the same time, an HMM model parameter estimation λ^* and a location-label estimation P^* are produced by the local optimization procedure that has found the best fit.

4.3.3 AP Selection Automation

The efficiency of the learning task can be improved by using a small number of selected APs for the genotype representation. It is known that a small subset of APs can characterize a Wi-Fi environment [33]; therefore, indoor localization systems often adopt an AP selection technique, such as MaxMean or InfoGain, to reduce the computational cost of localization. They select the k best APs based on some criterion with a predefined parameter k, to balance the computational cost and localization accuracy. Among the techniques, InfoGain, which utilizes *Information Gain* (IG) of APs for the criterion, is recognized as one of the best methods for AP selection [33].

The fewer APs are used, the faster GA operations will be executed. Nevertheless, the selected AP set should be big enough to characterize the Wi-Fi environment of a building

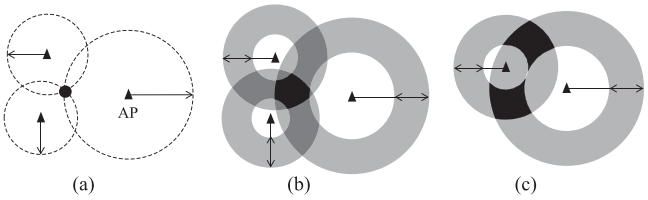


Fig. 3. Trilateration in a 2D space. The triangles indicate AP locations. The black dot or area is the intersection of the possible locations indicated by multiple APs.

so that it can support for location identification to achieve an acceptable accuracy range. Another requirement, which is most important in this research, is the automation of the selection because it is preferred for the unsupervised calibration to be performed with no human intervention. In order to satisfy these requirements, together with IG, we introduce another criterion derived from the mathematical principle of trilateration.

In the trilateration theory, a location in n dimensions can be identified by RSSIs from $n+1$ APs in an ideal environment as discussed in Section 2.1.1. Fig. 3 illustrates trilateration in a two-dimensional space. Fig. 3a shows an ideal case in which a location is identified as an intersection of the circles drawn from three APs. Signals, however, are noisy in real environments; the possible locations indicated by an RSSI thus do not form a circle but occupy the area depicted by the gray areas in the figure. In such case, the possible locations indicated by two APs would occupy two distinct areas, as shown by the black areas in Fig. 3c, while those for three APs would be clustered in the area shown in Fig. 3b. We assume that the error in the situation of Fig. 3b is acceptable, because the error seems to be recovered by local optimization. With this assumption, we acquire a set of APs such that all observations in training data have RSSI records from at least $n+1$ APs for an n-dimensional indoor space. (i.e., 3 APs in 2D areas and 4 APs in multistory buildings) As a result, the selected APs can characterize the Wi-Fi environment in that the locations of all observations are identified with acceptable errors. The AP selection algorithm is illustrated in Fig. 4.

The algorithm also uses the IG criterion but does not require a predetermined parameter k unlike with InfoGain. In this way, an AP selection is made without human intervention. Once the algorithm defines the set of APs for an indoor area, GA operates on a compact genotype representation defined with the selected APs. Meanwhile, SK local optimization does not need to use only the selected APs, because SK procedure efficiently handles a large number of APs by reusing a pre-processed distance matrix.

5 Evaluation

We evaluated the feasibility of UCMA in terms of efficiency and effectiveness in the unsupervised calibration of localization models. First, the efficiency in solving the optimization problem for unsupervised calibration was tested. The effect of the unsupervised calibration on the localization accuracy was also evaluated under various conditions. Since UCMA produces location-label estimates of unlabeled training data, any RSSI fingerprinting-based method can be used in the localization phase. A simple kNN method ($k = 3$) [3] was used for the localization test in the evaluation.

```
N = dimensionality of the indoor space of interest (2 or 3)
SELECTION = list of APs sorted by IG ascending
O = set of observations

for every AP ap in SELECTION
  O^ap = subset of O that has a RSSI record from ap
  If all o in O^ap has more than N+1 RSSI records from ap s in SELECTION
        remove ap from SELECTION
endfor

Return SELECTION
```

Fig. 4. The proposed AP selection algorithm.

When UCMA generate a calibrated model, the model can be evaluated *directly* using the result of the location-label estimation. The calibrated model can also be evaluated *indirectly* by a localization test using a test data set. The results are measured by Average Distance Error (ADE); ADE^{Loc} is used to denote the average distance between the true locations and their estimations in the localization test, and ADE^{Cal} does this for the location-label estimation of a calibrated model. Simply, ADE^{Cal} represents calibration error and ADE^{Loc} denotes localization error. The ADE^{Cal} was used to evaluate intermediate models generated during the optimization process, and the ADE^{Loc} was used to evaluate the effect of a calibrated model on localization accuracy.

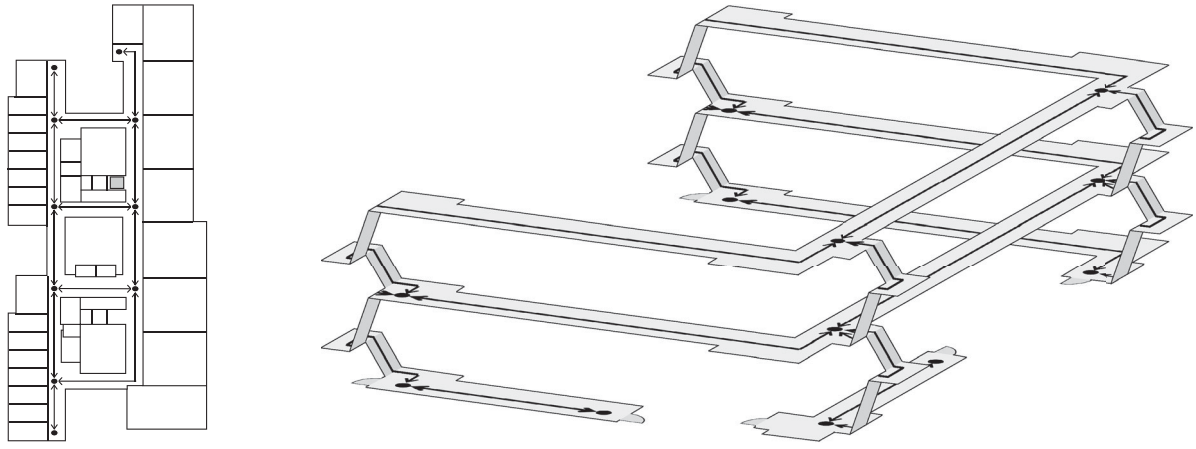
All programs for the evaluation were implemented in Java and run on a 3.40 GHz Intel Core i7 CPU with 8 GB of memory.

MAs and GAs have several parameters that influence the quality of the convergence process. The very typical approach was used for the parameter setting of UCMA. UCMA was parameterized as follows:

- The population size was fixed at 100.
- One percent of the genes of every genotype were mutated in each generation.
- Ten percent of new individuals were introduced at each generation to ensure genetic diversity.
- The number of generations was fixed at 100.

The experiments were conducted in two medium- and large-scale office buildings, N1 and N5 at Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea. Fig. 5a depicts the experimental area in N1 (testbed N1), and Fig. 5b shows the experimental area in N5 (testbed N5). Testbed N5 included three floors while testbed N1 included only one floor. Four people carrying their wireless device (Samsung Galaxy S3) walked along the hallways and collected user traces consisting of RSSI measurements with a sampling rate of 1Hz. The ground-truth coordinates of the measurements were obtained using a click-to-map-based annotation program. About four-fifths of the collection was used for the calibration and the rest were left for the localization test. The specifications of the testbeds and their data collections are listed in Table 2.

The localization area was composed of one-dimensional hallways (including stairs). The hallways were then divided by g meters, denoting location granularity. Another parameter needed for UCMA is for human mobility. We assumed that people could not move faster than 5 m/sec in the experiments.



(a) Testbed N1 (b) Testbed N5

Fig. 5. Experimental areas at KAIST, Korea: (a) testbed N1: 7th floor of N1 building; (b) testbed N5: all three floors of N5 building.

TABLE 2
Experimental Setups

Testbed	Area	Total length of corridors	# APs detected	Data collection			
				Calibration data		Test data	
				# traces	# measurements	# traces	# measurements
N1	$(80 \times 32)m^2 \times 1$ floor	196 m	196	50	1,848	14	449
N5	$(91 \times 83)m^2 \times 3$ floors	702 m	257	110	4,184	25	1,041

5.1 AP Selection Evaluation

At first, the effectiveness of the proposed AP selection technique was tested because it is the starting step of UCMA. Generally, the effectiveness of an AP selection is evaluated in terms of the localization accuracy. The ground-truth model, which was trained with the ground-truth location labels, was used for the evaluation. The results were compared with those obtained using InfoGain, the most well-known method for AP selection.

The proposed technique selected nine APs out of 196 in testbed N1, and 25 APs out of 257 in testbed N5. The localization results using the selected APs are shown in Fig. 6. As InfoGain requires a parameter for the number of APs, Fig. 6 plots ADE^{Loc} for InfoGain depending on the number of APs used. In contrast, the results of the proposed technique are shown with a point in the figure because it automatically defines the number of APs to be used. As shown in the figure, the ADE^{Loc} values of the proposed technique were lower than those of InfoGain for the same numbers of APs.

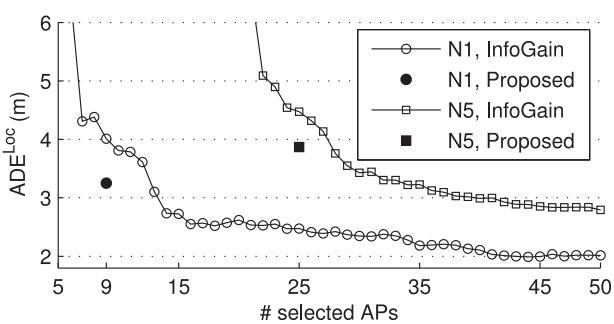


Fig. 6. Effect of AP selection techniques on localization accuracy.

We achieved 3.25 m in testbed N1 and 3.87 m in testbed N5, whereas InfoGain yielded 4.01 and 4.47 m, respectively. InfoGain has the advantage of balancing computational cost and localization accuracy with a parameter, but defining the parameter can be viewed as onerous, especially for unsupervised learning. Since the proposed AP selection technique can generate a better subset of APs in terms of localization accuracy without defining a parameter, it can be considered as an adequate method for unsupervised calibration.

5.2 Efficiency Comparison

This section shows how efficiently UCMA can solve the optimization problem for unsupervised calibration. The results of UCMA were compared with those of SK using random initialization (henceforth called RanSK for convenience), which is a general scheme for SK-based HMM training. Similar to UCMA, RanSK can incorporate the proposed AP selection technique. Therefore, four optimization schemes were compared: UCMA with(+) the AP selection, UCMA without (-), RanSK with(+), and RanSK without(-). They are denoted by UCMA+, UCMA-, RanSK+, and RanSK-, respectively. (Note that UCMA+ is the proposed unsupervised calibration scheme.) For fair comparison, RanSK+/- iteratively evaluated the same number of solutions with UCMA+/- (populations $\times$ generations).

In this experiment, the location granularity g was set to 3 m. As results, 65 locations and 398 possible transitions were generated for testbed N1, and 232 locations and 1,238 transitions for testbed N5.

Fig. 7 compares the convergence graphs of the four schemes in testbed N1. Overall, the fitness value increased with the number of iterations or generations. However, as

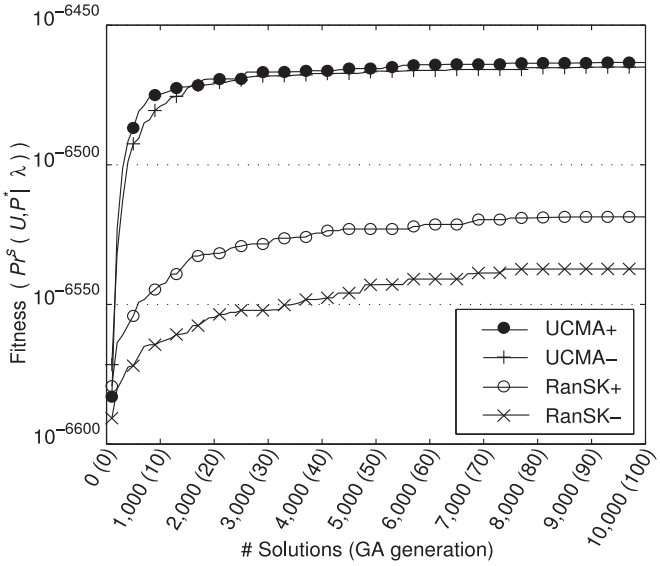


Fig. 7. Convergences of UCMAs and RanSKs in testbed N1.

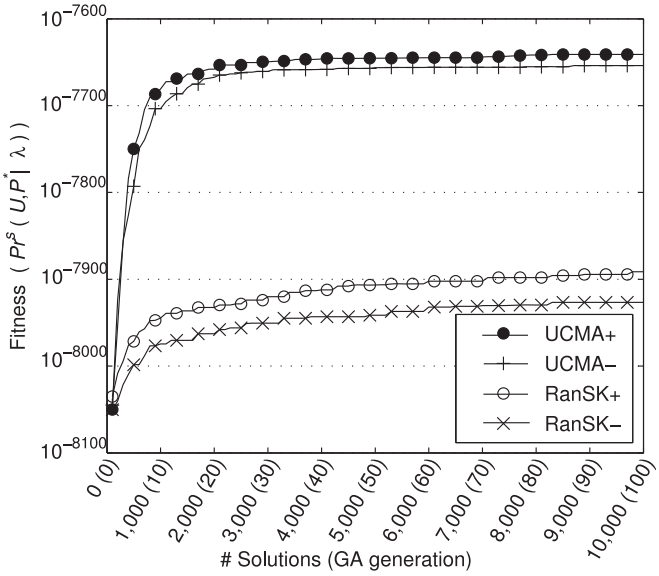


Fig. 9. Convergences of UCMAs and RanSKs in testbed N5.

can be seen in the figure, UCMA+ converged faster than the others did and found the solution with the highest fitness value. In addition, the schemes using the AP selection technique converged faster than the ones without the selection, even for the cases of RanSK schemes. The effect of the AP selection on UCMA was not significant, but the convergence became slightly faster due to the AP selection effect.

Fig. 8 plots the calibration error, ADE^{Cal}, of the models generated by UCMAs and RanSKs for the testbed N1. As can be seen in the figure, better fitness did not necessarily correspond to a better model. Nevertheless, the there was a tendency of generating a better model as fitness and the number of iterations increased. In particular, ADE^{Cal} of UCMA+ reached below 3.3 m at the 20th generation and stabilized at that level. UCMA- achieved a similar result to UCMA+, but was slightly slower. Meanwhile, RanSK+/- yielded the calibration errors of 8 m and 12.2 m, respectively.

As can be seen in Fig. 8, ADE^{Cal} of the UCMA schemes stabilized after a certain generation, whereas their fitness values slightly but continuously increased as the number of generations increased as shown in Fig. 7. This indicates that there are no significant differences in fitness among the models having ADE^{Cal} of less than 3.5 m. Noise in wireless signals is inevitable, especially in indoor environments, due to signal distortion and environmental dynamics. This may distort the correlation between the fitness and the correctness of a calibrated model when the calibrated model is close to the truth. It may be inferred from this result that there is a certain limitation in unsupervised calibration due to the noise in wireless signals.

Figs. 9 and 10 show experimental results in testbed N5. The convergence graphs in Fig. 9 show similar patterns with those for testbed N1 shown in Fig. 7. Testbed N5 is larger than N1 as it accommodates approximately 3.5 times more number of locations than N1 does (232 and 65 locations, respectively). This difference in size was reflected in the calibration errors of the RanSK schemes. While RanSK+/- yielded calibration errors of 8 m and 12.2 m in

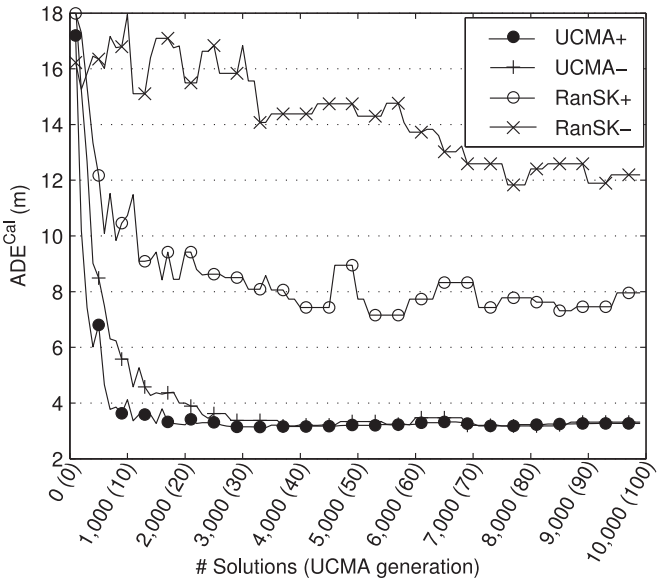


Fig. 8. Calibration errors of UCMAs and RanSKs in testbed N1.

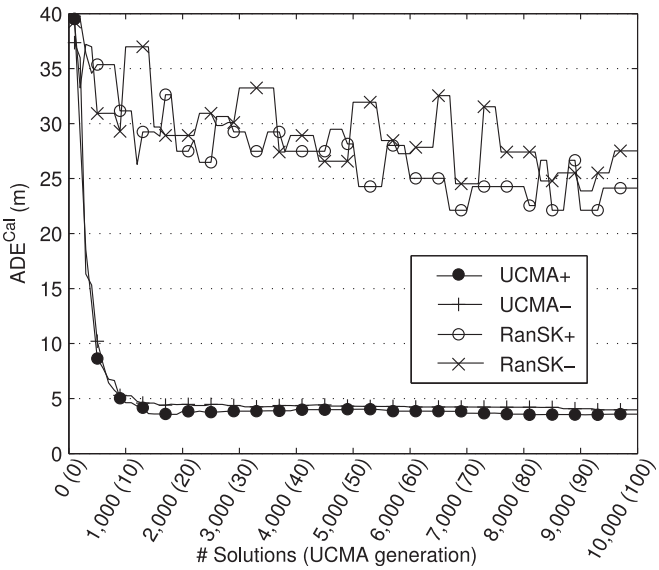


Fig. 10. Calibration errors of UCMAs and RanSKs in testbed N5.

TABLE 3
Summary of Experiment for Efficiency Comparison

Testbed	Scheme	CPU time (minute)	Avg. # iterations In SK	Best fit	
				Fitness	ADE^{Cal} (Std.)
N1	UCMA+	26	10.1	10^{-6463}	3.27 m (1.60)
	UCMA−	152	10.3	10^{-6465}	3.31 m (1.54)
	RanSK+	28	10.5	10^{-6519}	7.95 m (4.31)
	RanSK−	162	11.7	10^{-6537}	12.19 m (4.77)
N5	UCMA+	266	9.6	10^{-7641}	3.58 m (1.79)
	UCMA−	2,120	10.0	10^{-7654}	3.97 m (2.01)
	RanSK+	278	10.4	10^{-7890}	24.13 m (5.41)
	RanSK−	2,183	10.8	10^{-7927}	27.52 m (6.06)

testbed N1 (Fig. 8), they could not reduce the error below 20 m in testbed N5 (Fig. 10). Nevertheless, the effect of size difference on UCMA schemes was not significant. ADE^{Cal}s of UCMA+/- in testbed N5 were similar to those in N1, as can be seen in Fig. 10. Also, UCMA+ delivered a more accurate model faster than the others, and its error was less than 3.6 m. Detailed evaluation results on this experiment can be found in Appendix, in the form of CDF plots.

Table 3 summarizes the results of the experiments. It revealed that the UCMA+ scheme generated the best models for both the testbeds in terms of fitness value. Meanwhile, there were no significant differences in calibration errors between UCMA+ and UCMA-. As can be seen in the table, the running time of the schemes using the AP selection was much faster than that of the ones not using the AP selection. Regarding the CPU time of the UCMA schemes, UCMA+ executed approximately 5.7 times faster than UCMA- in testbed N1, and eight times faster in testbed N5. Thus the efficiency improvement by the AP selection will be a way better than that shown with respect to generations, if the CPU time were also considered.

5.3 Localization Accuracy Test

We compared the localization accuracy achieved by UCMA (unsupervised localization) with that of a model built in a supervised manner using the ground-truth location labels (supervised localization). The average error distances ADE^{Loc} were measured according to the changes of the amount of training data and location granularity.

Fig. 11 compares the errors of unsupervised and supervised localizations in testbed N1 ($g = 3$) as the increment of the training data. As shown in the figure, the supervised localization using more than 400 training samples achieved an ADE^{Loc} value of less than 2 m, and no significant improvement was observed when more than 600 samples were used. In contrast, the error of the unsupervised localization progressively decreased as more data were used for the training, and it achieved localization errors of approximately 3.2∼3.5 m with more than 1,200 samples.

The experiment in testbed N5 showed similar results to those in testbed N1, but, as the area of N5 is bigger than N1, a larger amount of training data was required to build a precise localization model. As can be seen in Fig. 12, ADE^{Loc} of the supervised localization reached less than 2.2 m with more than 1,000 data and stabilized at that level. The unsupervised localization achieved ADE^{Loc} of 3.6 m when more than 2,800 samples were used for the training. Clearly, both supervised and unsupervised localizations require some amount of training data to provide an accurate localization service, but this experiment has confirmed that an unsupervised calibration scheme requires much more data to uncover a hidden structure in unlabeled data. In the two testbeds, at least 18.5 and 12.1 samples were required in average at each location for successful unsupervised calibration. This indicates that the required amount of training data would be dependent on the environmental features of a building.

Fig. 13 plots the CPU time required for UCMA. As discussed in Section 4.2, the time complexity of an SK iteration cycle is linearly proportional to the number of observations. Other factors which affect the running time of UCMA are the numbers of generations and SK iterations. In our experiment, the number of generations was fixed to 100, and the number of SK iterations was around 10 (refer to Table 3). Accordingly, CPU time required for UCMA turned out to be linearly related to the number of observations as shown in Fig. 13.

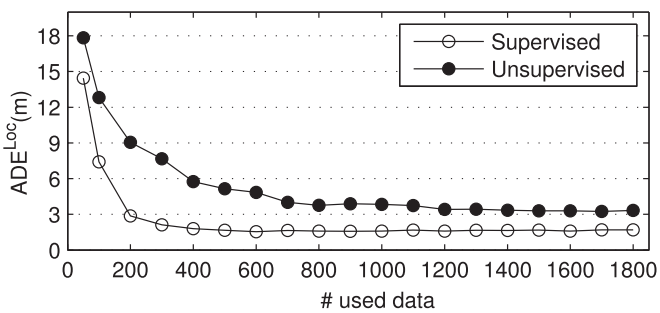


Fig. 11. Effect of the amount of training data on localization accuracy in testbed N1.

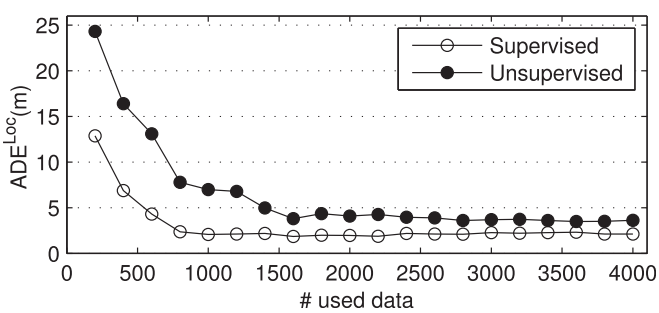


Fig. 12. Effect of the amount of training data on localization accuracy in testbed N5.

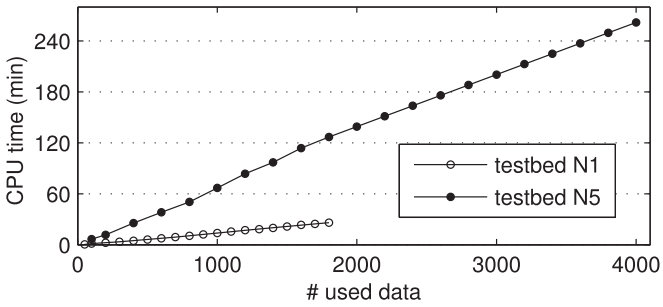


Fig. 13. Effect of the amount of training data on training time for UCMA.

Localization accuracy depends on the quality as well as the quantity of the training data. Location granularity influences the quality of the training data because it determines the resolution of the location information to be labeled on fingerprints. Fig. 14 depicts the localization errors in testbed N1 when using both supervised and unsupervised learning methods for the location granularity parameter g, and Fig. 15 shows those in testbed N5. Overall, the error decreased as the location granularity became finer. In testbed N1, the unsupervised localization achieved ADE^{Loc} values of about 2.7-3.7 m, whereas the errors of the supervised localization were observed in a range of 1.6-2.8 m. In testbed N5, the unsupervised localization achieved ADE^{Loc} values of about 3.1-4.6 m, whereas the errors of the supervised localization were observed in a range of 2.0-3.7 m. Although the unsupervised localization was not able to outperform the supervised type, they showed similar error changes depending on g. This implies that location granularity has a similar effect on both unsupervised and supervised localizations. CDF comparisons on the result of this section can be found in Appendix.

Meanwhile, location granularity affects the training time for unsupervised calibration. Finer granularity increases the number of locations. It can also increase the number of reachable locations from a location considering a given human mobility. Thus, finer granularity increases the total number of possible transitions among the locations. Fig. 16 shows the time taken in testbed N1 as the increment of the possible transitions. In the figure, each dot represents the result from an experiment for a given granularity g, and it is annotated with the number of locations, n, and the average number of transitions out of a location, c, respectively. As shown in the figure, the training time linearly increased as the number of the possible transitions increased. This result has confirmed that the training time of UCMA is linearly proportional to the number of possible transitions as discussed in Section 4.2.

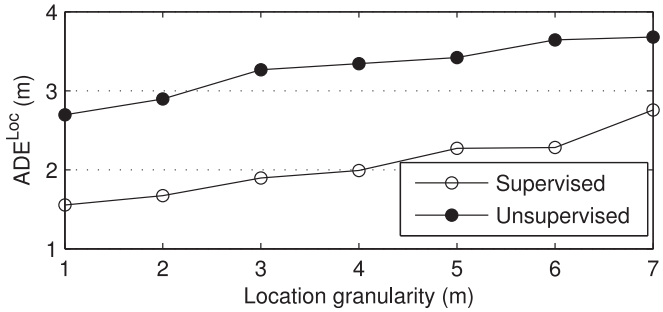


Fig. 14. Effect of location granularity on localization accuracy in testbed N1.

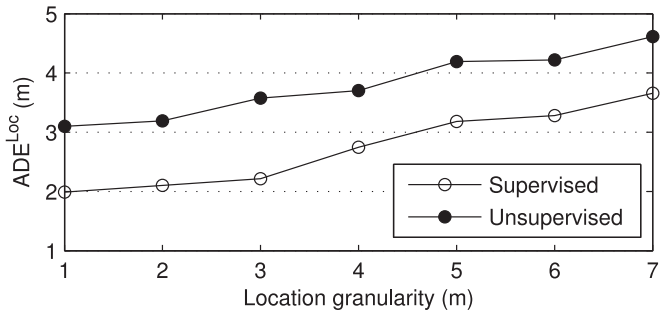


Fig. 15. Effect of location granularity on localization accuracy in testbed N5.

6 Discussion

6.1 Calibration Effort Reduction

The experiments conducted under various conditions manifested that the proposed unsupervised calibration method, UCMA, can build a precise localization model without requiring location labels of training data. The average distance error of UCMA ranged from 2.7 to 3.7 m in a medium-scale building and 3.1 to 4.6 m in a large-scale building, which were only slightly lower than those of fully supervised calibration requiring extensive labeling efforts. These results imply that an indoor localization service will be possible through implicit crowdsourcing if the data contribution were made by participants in most areas of a building.

The primary goal of this research is to minimize the cost of localization model calibration, and the cost can be estimated from the resources required in the calibration. Table 4 shows comparisons between UCMA and the previous methods for reducing calibration efforts discussed in Section 2.2, including resource requirements for each method. Note that localization accuracies listed in the table are not important for the comparisons because their accuracies are highly dependent on their own experimental settings. What we can conclude from the accuracies is that all of the methods have been successful in constructing localization models with the help of unlabeled fingerprints.

The methods for reducing calibration efforts can roughly be classified into inertial sensor-based and learning-based methods. Inertial sensor-based methods usually estimate the locations of unlabeled fingerprints included in user traces by placing the traces on a floorplan using dead reckoning techniques. Learning-based methods also estimate the placement of fingerprints, but the distribution of RSSI values is mainly considered in the estimation process.

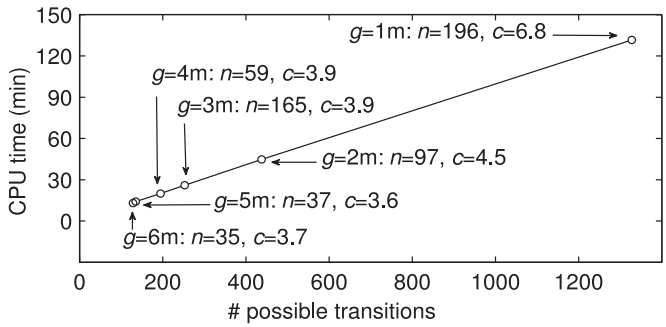


Fig. 16. Effect of the number of possible transitions on training time for UCMA in testbed N1. (g = location granularity, n = the number of locations, c = the average number of transitions out of a location).

TABLE 4
Calibration Effort Reduction Methods Using Unlabeled Fingerprints

Category		Method (author)	Calibration Algorithm	Requirements for Calibration (except unlabeled fingerprints)	Localization Accuracy (from literature)
Inertial sensor-based		ZEE [8]	Dead Reckoning	Inertial Sensors, Floorplan	3m
		LiFS [9], [10]	Dead Reckoning, Manifold Learning (Multidimensional scaling)	Inertial Sensors, Floorplan	5.9m
		WILL[11]	Dead Reckoning, Clustering algorithms (k-means, EM)	Accelerometer, Floorplan	86% (Room detection)
		Unloc[7]	Dead Reckoning, k-means clustering	Inertial Sensors, magnetometer, Floorplan (seed landmarks)	1~2m
Learning-based	Semi-supervised	Chai et al. [13]	Expectation Maximization (HMM)	Some amount of labeled fingerprints (25%~50%)	55%~90% within 3m
		EZ[12]	Genetic algorithm, Gradient Descent	Some amount of labeled fingerprints (6%, 15%)	2m, 7m
		LARM[14]	Manifold Learning (Laplacian Eigenmaps), Singular Value Decomposition, Dead Reckoning	Some amount of labeled fingerprints (5%~25%) or AP location information, Inertial sensors	20~40% (relative error distances)
		Tran et al. [15]	Manifold Learning (Dirichlet's energy minimization), Total Variation-based graph regularization	Some amount of labeled fingerprints (10%~90%)	40~45% (relative error distances)
		Sorour et al. [16]	Manifold Learning (Locally Linear Embedding)	Some amount of labeled fingerprints (10%~50%), Floorplan	2~3.7m
		Pulkkinen et al. [17]	Manifold Learning (Isomap)	Some amount of labeled fingerprints (9%)	2m
		WiFi-SLAM [18]	Manifold Learning (GP-LVM)	A good initial model (Isomap)	3.97m
	Unsupervised	**UCMA**	**Memetic Algorithm, Segmental K-means**	**Floorplan**	**2.7m, 3.1m**

The learning-based methods shown in the table can further be classified as semi-supervised or unsupervised methods depending on whether labeled fingerprints are involved in the learning process or not, respectively.

The comparison table shows that there are three types of data requirements: data from inertial sensors, floorplans, and labeled fingerprints. Among these data, the labeled fingerprints are considered as the most expensive ones because the acquisition of location labels is usually performed through laborious manual efforts. The cost of the data from inertial sensor is relatively low because the acquisition of these data usually does not require human intervention. The cost for the acquisition of floorplans is also low and sometimes assumed to be zero in this research area [8], because the floorplans, regardless of their involvement in calibrations, are prerequisites for the implementation of location-based services. In that sense, inertial sensor-based methods, which depend on inertial sensors and floorplans, are expected to be more effective in reducing calibration efforts than semi-supervised learning-based methods, which mainly depend on labeled fingerprints. The efforts can be further reduced in UCMA by using only floorplans to make use of unlabeled fingerprints in the calibration of localization model. Therefore, we can conclude that UCMA is the most effective method among the methods listed in Table 4 for the reduction of calibration efforts.

6.2 Limitations

Training time required for UCMA linearly increases depending on the number of observations and locations. When we consider that user traces are collected from numerous buildings in villages and cities all over the world, we need to develop a technique to further reduce the training time. It is better to use a small portion of the millions of crowdsourced data for unsupervised calibration and then switch to a semi-supervised learning on the rest of the data. The proposed method provides the estimated location labels, which are essential to semi-supervised learning approaches.

In our modeling, an indoor space is modeled as a set of finite location-states and constrained transitions where each state corresponds to a physical location. However, there is no guarantee of one-to-one correspondence between a model and a space if the layout of a building is symmetrical, because a symmetric layout allows multiple mappings between the model and the space. That is, the calibration may fail because of the structure of a building. In this case, a few location references such as GPS coordinates obtained near windows can play a critical role in finding a single correct mapping.

The methods that use unlabeled fingerprints for reducing calibration efforts, including UCMA, usually assume that the building in which the fingerprints have been collected is known. The building should be identified in advance if the methods are to be applied to practical crowdsourcing-based localization systems. To address this problem, a crowdsourcing-based system can use indoor/outdoor environment detection techniques [34], [35]. Typically, these techniques identify the building which a user is located in by referring to GPS data stored in devices or other ambient features.

Relatively simple model and techniques have been used for building UCMA, and it has been implemented under general assumptions. More specifically, the evolutionary

process of UCMA adopted simple strategies for GA operations, and a very common way was used for the setting of its parameters. In the HMM modeling, the simplest likelihood metric, the Euclidian distance, was used instead of using a more advanced measure. The transition probabilities of the HMM model were initialized using a static threshold on human mobility. In fact, they can be dynamically inferred using the sensing data from inertial sensors. The studies of human motion patterns can sophisticate the mobility setting as well. Furthermore, there are specialized radio propagation models for indoor environments [21], which can replace the general model (1) used in UCMA. Therefore, the proposed method can be improved further, if the more advanced techniques are used for each step.

7 Conclusion

In this paper, we presented UCMA, an unsupervised calibration method that can build an indoor localization system using unlabeled RSSI measurements. Simple modeling and optimization techniques were employed in unsupervised learning on the unlabeled measurements. The evaluation on the two office buildings confirmed that, under various conditions, the proposed method can build a precise localization model without any location reference.

An indoor map and online RSSI measurements are two essential requirements in the service phase of fingerprinting-based localization systems. UCMA uses only the two inputs for the calibration, whereas conventional approaches require extra inputs or extensive efforts. This indicates that a localization system can be implemented by UCMA without additional cost except computational cost on the server side. In that sense, the technique has the potential to make significant progress in indoor localization, especially in realizing a global indoor positioning system.

Acknowledgments

This work was supported partly by the Center for Integrated Smart Sensors funded by the Ministry of Science, ICT & Future Planning as Global Frontier Project (CISS-2012M3A6A6054195), and partly by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIP) (No. 2015R1A2A1A10052224). Dongsoo Han (dshan@kaist.ac.kr) is the corresponding author of this paper.

References

[1] J. Yim, S. Jeong, K. Gwon, and J. Joo, "Improvement of Kalman filters for WLAN based indoor tracking," *Expert Syst. Appl.*, vol. 37, no. 1, pp. 426–433, 2010.

[2] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proc. 19th Annu. Joint Conf. IEEE Comput. Commun. Soc.*, 2000, vol. 2, pp. 775–784.

[3] D. Han, S.H. Jung, M. Lee, and G. Yoon, "Building a Practical Wi-Fi-Based Indoor Navigation System," *IEEE Pervasive Comput.*, vol. 13, no. 2, pp. 72–79, Apr.–Jun. 2014.

[4] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, J. Tabert, P. Powledge, G. Borriello and B. Schilit, "Place lab: Device Positioning using Radio Beacons in the Wild," *Pervasive Computing*. Heidelberg, Germany: Springer, 2005, pp. 116–133.

[5] P. Bolliger, "Redpin-adaptive, zero-configuration indoor localization through user collaboration," in *Proc. ACM Int. Workshop Mobile Entity Localization Track. GPS-Less Environ.*, 2008, pp. 55–60.

[6] M. Lee, S. H. Jung, S. Lee and D. Han, "Elekspot: A platform for urban place recognition via crowdsourcing," in *Proc. IEEE/IPSJ Int. Symp. Appl. Internet*, Jul. 2012, pp. 190–195.

[7] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury, "No need to war-drive: Unsupervised indoor localization," in *Proc. 10th Int. Conf. Mobile Syst., Appl., Serv.*, 2012, pp. 197–210.

[8] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: Zero-effort crowdsourcing for indoor localization," in *Proc. 18th Annu. Int. Conf. Mobile Comput. Network.*, 2012, pp. 293–304.

[9] Z. Yang, C. Wu, and Y. Liu, "Locating in fingerprint space: Wireless indoor localization with little human intervention," in *Proc. 18th Annu. Int. Conf. Mobile Comput. Network.*, 2012, pp. 269–280.

[10] C. Wu, Z. Yang, and Y. Liu, "Smartphones based crowdsourcing for indoor localization," *IEEE Trans. Mobile Comput.*, vol. 14, no. 2, p. 444–457, Feb. 2015.

[11] C. Wu, Z. Yang, Y. Liu, and W. Xi, "Will: Wireless indoor localization without site survey," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 4, pp. 839–848, Apr. 2013

[12] K. Chintalapudi, A. Padmanabha, and V. N. Padmanabhan, "Indoor localization without the pain," in *Proc. Annu. Int. Conf. Mobile Comput. Network.*, 2010, pp. 173–184.

[13] X. Chai and Q. Yang, "Reducing the calibration effort for probabilistic indoor location estimation," *IEEE Trans. Mobile Comput.*, vol. 6, no. 6, pp. 649–662, Jun. 2007.

[14] J. J. Pan, S. J. Pan, J. Yin, L. M. Ni, and Q. Yang, "Tracking mobile users in wireless networks via semi-supervised colocalization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 3, pp. 587–600, Mar. 2012.

[15] D. A. Tran and P. Truong, "Total variation regularization for training of indoor location fingerprints," in *Proc. ACM Workshop Mission-Oriented Wireless Sens. Netw.*, Sep. 2013, pp. 27–32.

[16] S. Sorour, Y. Lostanlen, S. Valaee, and K. Majeed, "Joint indoor localization and radio map construction with limited deployment load," *IEEE Trans. Mobile Comput.*, vol. 14, no. 5, pp. 1031–1043, Jul. 2014.

[17] T. Pulkkinen, T. Roos, and P. Myllymäki, "Semi-supervised learning for WLAN positioning," in *Proc. 21th Int. Conf. Artif. Neural Netw.*, 2011, pp. 355–362.

[18] B. Ferris, D. Fox, and L. D. Neil, "WiFi-SLAM using Gaussian process latent variable models," in *Proc. Int. Joint Conf. Artif. Intell.*, 2007, vol. 7, pp. 2480–2485.

[19] J. Huang, D. Millman, M. Quigley, D. Stavens, S. Thrun, and A. Aggarwal. "Efficient, generalized indoor WiFi GraphSLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 1038–1043.

[20] N. J. Radcliffe and P. D. Surry, "Formal memetic algorithms," in *Evolutionary Computing*, Heidelberg, Germany: Springer, 1994, pp. 1–16.

[21] T. S. Rappaport. "Mobile radio propagation: Large-scale path loss," in *Wireless Communications: Principles and Practice*, 2nd ed, London, U.K.: Pearson, 2003.

[22] B. Cook, G. Buckberry, I. Scowcroft, J., Mitchell and T. Allen. "Indoor location using trilateration characteristics," in *Proc. London Comm. Symp.*, 2005, pp. 147–150.

[23] R. Battiti, T. L. Nhat, and A. Villani, "Location-aware computing: A neural network model for determining location in wireless LANs," Dept. Inf. Commun. Technol., Univ. Trento, Trento, Italy, Tech. Rep. DIT-02–0083, 2002.

[24] P. Kontkanen, P. Myllymaki, T. Roos, H. Tirri, K. Valtonen, and H. Wettig, " Topics in probabilistic location estimation in wireless networks," in *Proc. IEEE Symp. Pers., Indoor Mobile Radio Commun.*, Sep. 2004, vol. 2, pp. 1052–1056.

[25] J. J. Astrain, J. Villadangos, J. R. Garitagoitia, J. R. González-Mendivil, and V. Cholvim, "Fuzzy location and tracking on wireless networks," in *Proc. ACM Int. Workshop Mobility Manage. Wireless Access*, 2006, pp. 84–91.

[26] G. Shen, Z. Chen, P. Zhang, T. Moscibroda, and Y. Zhang, "Walkie-Markie: Indoor pathway mapping made easy," in *Proc. USENIX Conf. Netw. Syst. Des. Implementation*, 2013, pp. 85–98.

[27] R. Dugad and U. B. Desai. "A tutorial on hidden Markov models," Indian Inst. Technol., Tech. Rep. SPANN-96.1, 1996.

[28] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *Ann. Math. Statist.*, vol. 41, no. 1, pp. 164–171, 1970.

[29] B. Juang and L. Rabiner "The segmental K-means algorithm for estimating parameters of hidden Markov models," *IEEE Trans. Acoustics, Speech Signal Process.*, vol. 38, no. 9, pp. 1639–1641, Sep. 1990.

[30] A. M. Ladd, K. E. Bekris, A. P. Rudys, D. S. Wallach, and E. E. Kavraki, "On the feasibility of using wireless ethernet for indoor localization," *IEEE Trans. Robot. Autom.*, vol. 20, no. 3, pp. 555–559, Jun. 2004.
[31] J. Krumm, and E. Horvitz, "LOCADIO: Inferring motion and location from Wi-Fi signal strengths," in *Proc. Ann. Inter. Conf. Mobile Ubiquitous Syst.: Netw. Serv.*, Aug. 2004, pp. 4–13.
[32] J. Miranda, R. Abrishambaf, T. Gomes, P. Goncalves, J. Cabral, A. Tavares, and J. Monteiro, "Path loss exponent analysis in wireless sensor networks: Experimental evaluation," in *Proc. IEEE Conf. Int. Ind. Informat.*, Jul. 2013, pp. 54–58.
[33] Y. Chen, Q. Yang, J. Yin, and X. Chai, "Power-efficient access-point selection for indoor location estimation," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 7, pp. 877–888, Jul. 2006.
[34] P. Zhou, Y. Zheng, Z. Li, M. Li, and G. Shen. "Iodetector: A generic service for indoor outdoor detection," in *Proc. ACM Conf. Embedded Netw. Sens. Syst.*, 2012, pp. 113–126.
[35] C. Lee, G. Yoon, and D. Han. "A probabilistic place extraction algorithm based on a superstate model," *IEEE Trans. Mobile Comput.*, vol. 12, no. 5, pp. 945–956, May 2013.

Suk-hoon Jung received the PhD degree in information and communications engineering from the Korea Advanced Institute of Science and Technology (KAIST). He is a researcher at the Indoor Positioning Research Center, KAIST. His research interests include network analytics, machine learning, and their applications to pervasive computing.

Byung-chul Moon received the MS degree in information and communications engineering from the Korea Advanced Institute of Science and Technology (KAIST). He is currently working toward the PhD degree in information and communications engineering at KAIST. His research interests include indoor positioning, mobile computing, and pattern recognition.

Dongsoo Han received the PhD degree in information science from the Kyoto University. He is a professor of computer science at the Korea Advanced Institute of Science and Technology (KAIST). He is also a director at the Indoor Positioning Research Center, KAIST. His research interests include indoor positioning, pervasive computing, and location-based mobile applications. He is a member of the IEEE.