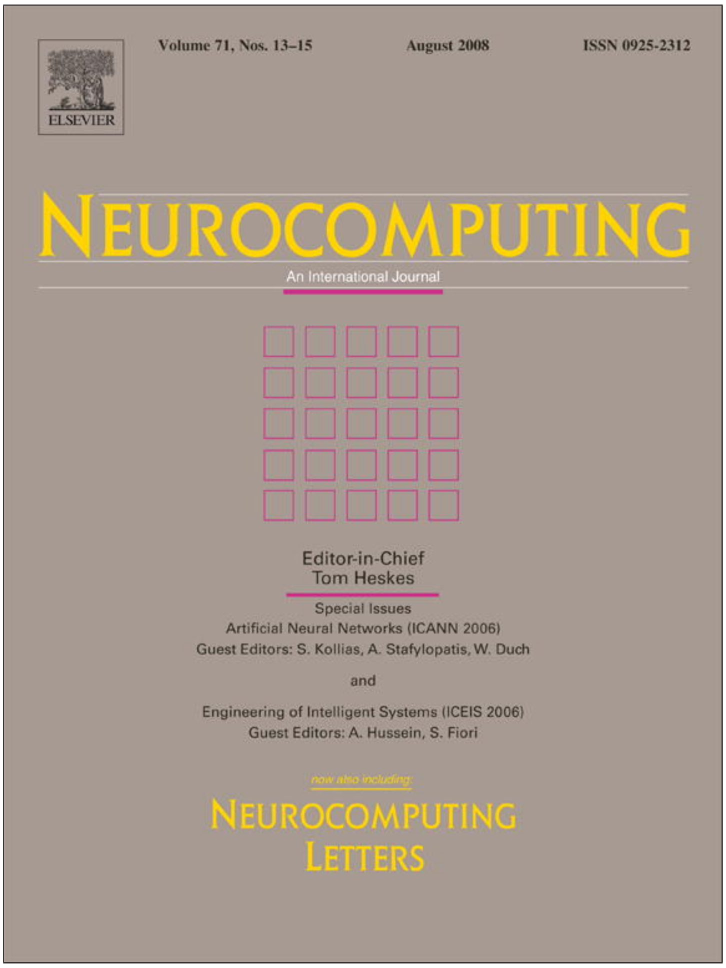
Volume 71, Nos. 13–15
August 2008
ISSN 0925-2312
ELSEVIER
NEUROCOMPUTING
An International Journal
Editor-in-Chief
Tom Heskes
Special Issues
Artificial Neural Networks (ICANN 2006)
Guest Editors: S. Kollias, A. Stafylopatis, W. Duch
and
Engineering of Intelligent Systems (ICEIS 2006)
Guest Editors: A. Hussein, S. Fiori
now also including
NEUROCOMPUTING
LETTERS

Neurocomputing 71 (2008) 2657–2669

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A modular classification model for received signal strength based location systems

Uzair Ahmad, Andrey V. Gavrilov, Sungyoung Lee, Young-Koo Lee *

Ubiquitous Computing Lab, Department of Computer Engineering, Kyung Hee University, Republic of Korea

ARTICLE INFO

Available online 16 May 2008

Keywords:
Modular classifiers
Location estimation
Received signal strength
Indoor positioning systems
Visibility matrix

ABSTRACT

Estimating location of mobile devices based on received signal strength (RSS) patterns is an attractive method to realize indoor positioning systems. Accuracy of RSS based location estimation, particularly in large target sites, is effected by several environmental factors. Especially the temporal or permanent absence of radio signals introduces *null* values rendering sparsity and redundancy in feature space. We present a *visibility matrix* based modular classification model which systematically caters for unavailable signals. This model is practically realized using two eminent classification methods: (1) multi-layer perceptron and (2) LVQ. In order to confirm robustness and applicability of this model, we developed two location systems at different sites. Experimental results in real-world environments demonstrate that modular classification model consistently achieves superior location accuracy.

1. Introduction

Location information is an integral and crucial component of ubiquitous computing applications. Indoor location estimation has been subject to costly infrastructure and special hardware devices mounted on the objects of interest [31,32,26]. Received signal strength (RSS) based indoor positioning systems can enable several location based applications in future ubiquitous computing environments. This technology offers economic viability and prospects of wide scale adoption due to pervasive availability of wireless networks and modern personal computing devices (e.g. PDA, notebooks) equipped with built in wireless network interface cards. Although RSS based location awareness particularly for indoor scenarios is desirable, accuracy of such systems has been a subject of intensive research [4,25,21,5,9,12]. These systems use standard infrastructure of IEEE 802.11, also called WiFi networks, to estimate location of any device which has a WiFi network interface card. RSS based location systems can enable several applications like automatic call forwarding to user's location, robotic localization, exploration and navigation tasks, searching, guiding and escorting systems, liaison applications, location based advertisement and positioning of entities in large warehouses.

The fundamental assumption underlying RSS based location estimation is that RSS of different access points, or signal sources, exhibit complex but distinguishable patterns at a particular location. If these patterns could be captured at each location then pattern recognition machines can be trained to learn the relationship between RSS fingerprints and each target location. From a pattern recognition standpoint, a particular set of access points constitute n-dimensional input space which is often referred to as a *radio map*. Suppose that n access points define a signal space RSS^n which covers target location space L, then this relationship can be represented as

$$F : RSS^n \rightarrow L \tag{1}$$

More specifically,

$$l_j = f(rss^n) \tag{2}$$

where $l_j \in L$ and $rss^n \in RSS^n$.

Creation of the *radio map*, also called *site calibration*, involves capturing this information and storing observed signal strengths vectors in a data store. Fig. 1 shows the methodology of the calibration process.

Once the *radio map* feature space is created, it is used to develop a mapping function, given in Eq. (2); between the jth target location l_j and respective signal strength patterns. This function is later employed to estimate location of a device with given RSS values.

WiFi radio signals follow a complex propagation model because of multi-path effects. These are caused by building structure and environmental factors such as human activity and neighboring devices. These elements as well as the distance between transmitter and receiver contribute to the loss of a signal in certain areas. In this paper, we refer to the signal availability of a particular access point at a given location as its *visibility*. Line graph of Fig. 2 shows signal strength values of four access

* Corresponding author. Tel.: +82 31 201 2514; fax: +82 31 202 2520.
E-mail address: yklee@khu.ac.kr (Y.-K. Lee).

0925-2312/$ - see front matter
doi:10.1016/j.neucom.2007.11.045

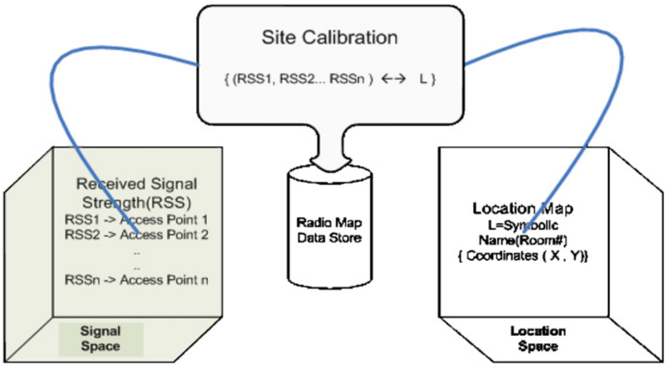

Fig. 1. Site calibration: creating the *radio map* feature space.

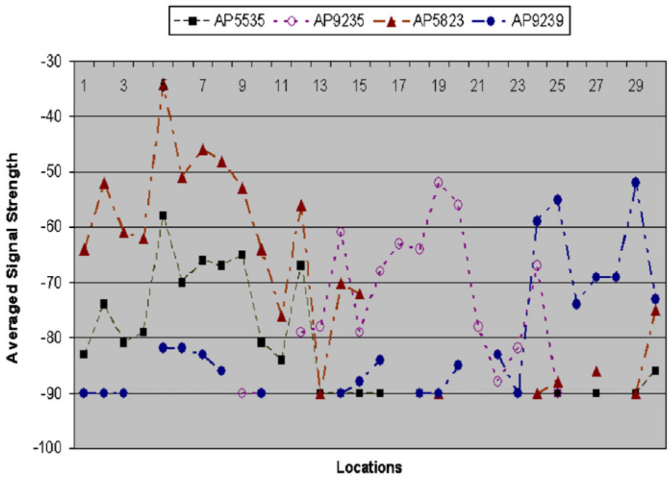

Fig. 2. Permanent loss of signals.

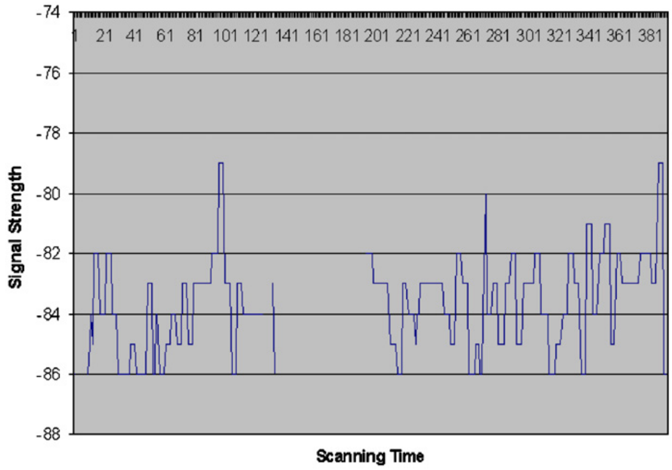

Fig. 3. Temporal loss of signals.

points,[1] listed on *y*-axis, which are received at different locations, listed on *x*-axis, by a mobile device. RSS scanning operation was performed several times for 5 days at each location and averaged signal strength are used to represent signal strength at that location. It is clearly noticeable that on many locations signal strength of each access point is too weak to be detected by mobile device. Temporal signal loss can be seen in Fig. 3 graph in which scanning time is listed along *x*-axis and signal strengths are shown along *y*-axis. It shows signal strengths of one access point received by a stationary PDA device.

Previously several pattern classification methods have been reported, as presented in Section 2, for RSS based location estimation. However, most of the previous results consider a small scale location estimation problem. Scaling up the target area introduces the visibility issue which introduces *null* values in radio map feature space. Section 2.1 addresses the limitations imposed by the absence of signals in previous approaches. We argue that the visibility issue calls for a modular approach in order to further improve the accuracy of RSS based location systems. Modularity has successfully been employed into classification systems as an effective scheme to enhance their performance as described in Section 2.2. In Section 3 a novel approach is offered to decompose the classification task into simpler, self-contained classifier modules based on the *a priori* knowledge about signal behavior. A modular classification model is presented combined with a feature selection method to develop large scale and highly accurate signal strength based location systems. Previously we reported small scale prototype location estimation system based on modular multi layer perceptron (MLP) networks [2,1]. Here we present extended experiments to validate robustness and applicability of our approach by applying it on: (a) Two classification methods (MLP and learning vector quantization (LVQ)) and (b) Two different sites. Experimental design and test set up details are given in Section 5. Results of both location systems for each site are presented in Sections 6.1 and 6.2.

2. Related work

There have been several efforts to develop RSS based location systems. Several pattern classification and machine learning methods, e.g. Bayesian classification and filtering [21], k-Nearest neighbors [4,25], GPS like triangulation [29] and Kalman filtering [12] have been employed for this purpose.

Nearest neighbors based pattern recognition technique and its derivatives have been used in pioneer works on RSS based location estimation. Microsoft's RADAR system reported 2.65 m distance error [4]. Pehlavan et al. [25] used k-nearest neighbors and achieved 2.8 m distance error. Nearest neighbor and its variants require a database of sample RSS readings at the estimation time for pattern matching. As the area and number of target locations grow, the size of the database dramatically increases prohibiting sufficient scalability.

Some research works have also employed GPS like triangulation method for location estimation. Asim et al. achieved 4.5 m location estimation error in a target area of 60 m^2 [29]. Triangulation methods work on the assumption that signal strength decays only as function of distance of receiver device from sender access points. Nevertheless, signal strength decay is a function of several factors of the indoor environments which undermine the validity of this assumption.

Probabilistic approaches like Bayesian networks have also been employed for such systems but are computationally exhaustive and difficult to scale. Ladd et al. [21] reported 1.5 m average distance error, but, only for 30 m^2 area test bed. As the area and number of target locations and wireless access points increase, the complexity of Bayesian structures grow and become computationally expensive.

[1] We use last four digits of the MAC address to identify an access point e.g., *AP*5535.

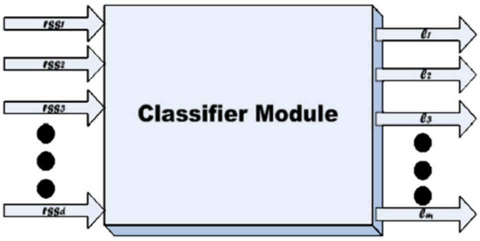

Fig. 4. *All-classes-one-classifier* approach.

Battiti et al. [5] have reported their research on using feed forward back propagation neural network on small scale (624 m^2 area using three access points) location estimation system. LVQ networks were used to develop location estimation system for 350 m^2 area using five access points [24].

2.1. Limitations of previous approaches

Previous approaches assume that all input signals are available at every location, mainly due to small problem area of location system. This assumption leads to the monolithic, large and complex classification machine for location estimation which means that classifier learns to estimate all locations, or classes, present in location space as shown in Fig. 4. This approach is referred to as *all-classes-one-classifier* in literature. It is observed that in many complex pattern classification tasks where the number of classes is large and input space is noisy, the *all-classes-one-classifier* either may not learn pattern-to-class function or may take very long to learn [28]. Due to intermittent visibility of the WiFi radio signals, all access points are not accessible at all target locations all the time. We argue that this phenomenon imposes a primary limitation on further success of *all-classes-one-classifier* approaches. Since a particular access point corresponds to a dimension of the radio map feature space, invisibility of signals introduce *null* values in RSS features. Previous approaches handle this situation by representing unknown values with very low RSS such as -100 dBm, but this practice results in redundant feature space. Redundant RSS values contain little information to influence the discrimination ability of feature space, instead they can negatively effect the performance of classifier in two ways. First, redundant features increase sample size to dimensionality ratio. It is well known that classification error is mainly determined by ratio of training sample size and feature space dimensions [17,10]. Second, unknown features do not compactly represent the target locations in terms of signal measurements. Non-representative features reduce the efficiency of classifier and contribute to increase computational complexity as well as memory requirements.

A modular classifier approach is presented in the next section which overcomes these limitations by incorporating visibility knowledge into classification system design.

2.2. Modular classification

Modularity has long been employed in computing systems. Inspired from biological evidences [28], modular classification systems attracted lot of attention from pattern recognition research community. There are numerous variants such as modular connectionless systems [14], multi-modular architectures [30], modular neural networks [6,11], modular pattern classifiers [28] and multiple classifier systems [15]. In general terms, the modular designs claim that decomposing a problem space into several subproblems enhances the learning and generalization capability of the classifier. Different methodologies for designing modular classifiers can be categorized into three schemes.

All-classes-many-classifiers refers to the methodology of training several redundant classifiers for the same problem. Later, results of all classifiers are integrated in order to obtain better performance than a single classifier can yield. Several different methods have been presented to integrate the outputs of different classifiers [33,15,7,19]. Although this scheme can produce better generalization due to the combined capabilities of multiple experts, training of such systems takes a lot longer.

One-class-one-classifier methodology is based on a class decomposition concept. It states that an N-class problem can be divided into N two-class subproblems. Later, separate modules are trained to learn each two-class subproblem. However, this approach leads to a large number of sub-classifiers which is not suitable for large scale location systems. Moreover, the discriminatory capability of the one-class-one-classifier is reported to be poor [28].

Subset of all-classes-one-classifier refers to modular design which partitions the problem space into subproblems. Each subproblem is then learned by individual classifier. Chiang [6] explained a *divide-and-conquer* learning approach based on partitioning of an input space into smaller training sets. Multiple networks learn each training set individually and network outputs are integrated to form a global output of system. Similar schemes have been reported by others [14,11].

3. Modular classification model for location estimation

Modularity is principally proven to be an effective method for improving learning and recognition performance. However, no single approach suffices for every problem. Each modular design is mainly guided by the specific nature of the problem and related goals. An intuitive approach to overcome the limitations mentioned in Section 2.1 is to decompose the location estimation task into several sub-tasks based on *a priori* knowledge about visibility of signals in the target area. This can be achieved by partitioning the input signal space into several feature spaces and training individual classification modules to learn the association between signal strengths and respective locations. Let RSS^d represent input signal space

$$RSS^d = (rss_{i1}, rss_{i2}, rss_{i3}, \ldots, rss_{id}) \tag{3}$$

where i is the ith input vector and each input vector is composed of signal strength values of d access points. This signal space provides connectivity to mobile device in a finite output space A which is divided into m disjoint locations.

$$A = \{l_1, l_2, l_3, \ldots, l_m\} \tag{4}$$

The learning procedure of RSS pattern classifier requires a training data set τ containing pairs of n input vectors and corresponding output locations.

$$\tau = (RSS_i^d, l_j)_{i=1,j=1}^{n,m}, \quad l_j \in A \tag{5}$$

In previous approaches a single large classifier is trained to learn signal-to-location associations using τ. This paper proposes to partition both input as well as output spaces based on visibility properties of radio signals in a way that the input space is divided into an arbitrary collection of nonempty subsets which constitute the actual RSS^d once combined.

$$RSS^d = (RSS_1^c \bowtie RSS_2^c \bowtie RSS_3^c \bowtie \cdots \bowtie RSS_q^c) \tag{6}$$

such that $c < d$ and q is the total the number of partitions in the input space. The output space is partitioned correspondingly and each subset RSS_i^c defines a region R which is a nonempty set of locations in the output space.

$$R_i = \{l_{i1}, l_{i2}, l_{i3} \ldots l_{il}\} \quad \forall l_{il} \in A \tag{7}$$

and

$$A = \bigcup_{i=1}^{q} R_i$$

This partitioning of the input and the output spaces is driven by visibility properties of access points in the area.

Separate classification modules can be employed, based on visibility dependency between (6) and (7), such as a subset of access points defines the input vector of each classifier and the locations in each corresponding region R_i become its outputs.

$$(R_i \longleftrightarrow RSS_i^c)_{i=1}^{q} \tag{8}$$

Fig. 5shows schematic diagram of modular classification system for RSS based location estimation. This approach leverages several desirable features of location systems which cannot be realized using previous approaches.

1. *Insulation* refers to separating the concerns by localizing the learning and recognition of *related* patterns and classes into individual classifiers. This feature ensures clarity and flexibility while developing, repairing and expanding large scale classification systems. Reparability can be more appreciated in the context of location systems because changes in the structure, such as adding or moving furniture, of a part of building can effect signal strength patterns, ultimately causing degraded location estimation performance. In case of non-modular approaches, the whole classifier is required to be retrained for small changes or additions in signal space. Whereas modular classifier model let repair the effected areas independent of other parts of the system.

Similar advantages are obtained in case of expanding a classification system. By expanding we mean to increase the area for location awareness. Even though expansibility can be achieved by *ad hoc* inclusion of new features and classes into system, unnecessary overlaps in signal space and suboptimal utilization of the discrimination ability of each access point results in increased complexity of classifier. Modular classification system can integrate new location classifiers into system without causing redesigning and retraining of previous classifier such that overlaps can be avoided and only the minimum required access points are used to train a classifier.

2. *The high dimensions* and *low sample size* contributes to learning and generalization ability of the classifier [17]. Signal strength based location estimation faces the issue of small sample size because only finite number of samples can be collected at calibration time. Collecting signal samples for longer periods of time has practical constraints especially in the case of large scale systems. Moreover, the dimensionality of input space increases as the target area expands which further aggravates the problems caused by high dimensionality and low sample size.

The modular design employs only c most influential access points from a d-dimensional input space, such as $c<d$, and sample size remains the same. This improves the dimensionality–sample size ratio and, ultimately, generalization ability resulting classifiers.

3. *Parallelism* is a common trait of modular designs which allows faster training of classification systems. Since a large and complex problem is divided into several simpler subproblems, it takes less time to learn pattern-class associations. Moreover several classifier modules can be trained in parallel, which further reduces training time.

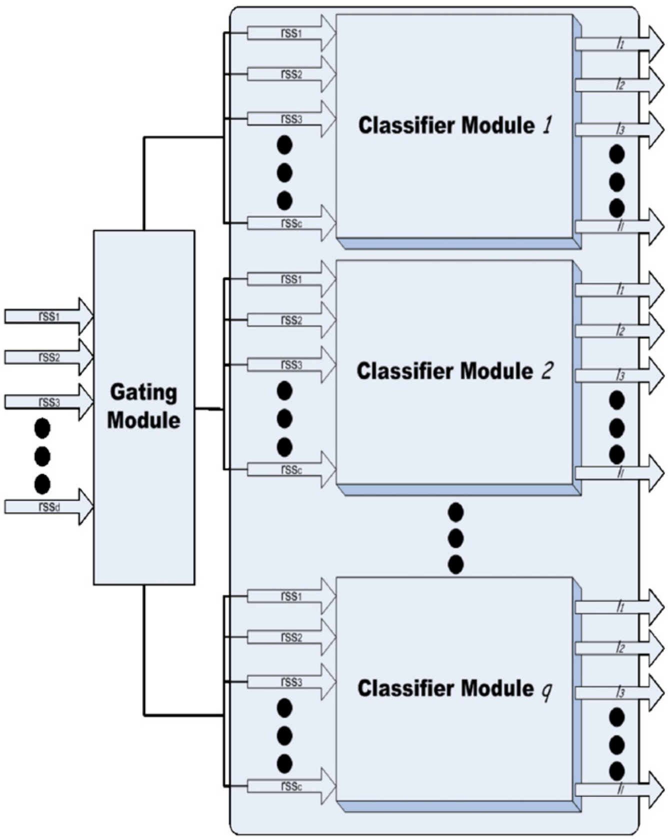

Fig. 5. Modular classification model.

4. Visibility modeling

It is needed to inspect the visibility properties of radio signals in order to support the propositions made by modular classification model in the previous section. Here we elucidate our point with the help of visibility data of 10 access points deployed in a real life environment shown in Fig. 8. The proposed visibility modeling approach captures *visibility* information about radio signals in the form of a visibility matrix. The visibility matrix provides a systematic way to select more representative features from the radio map feature space and then design separate classification modules using those features.

4.1. Visibility matrix and clusters

As mentioned in Section 2.1, in certain locations the signal strength of an access point drops to undetectable levels, e.g. less than -100 dBm. Suppose that at jth location, the signal scanning operation is performed N_j times and the signal of ith access point is detected p_i times. Then the visibility probability of a given access point i at location j is written as

$$P_{ij} = \frac{p_{ij}}{N_j} \tag{9}$$

such that $P_{ij} = 0$ implies no visibility and $P_{ij} = 1$ shows always visible signal. A visual representation of the visibility probability of 10 access points in experimental site 1 is shown using radar graphs in Fig. 6. In these graphs, the target locations are listed on the edge of each graph and P_{ij} is shown as shaded area from the center (which represents no visibility) to the edge (representing always visible) of each graph. The visibility probabilities of d

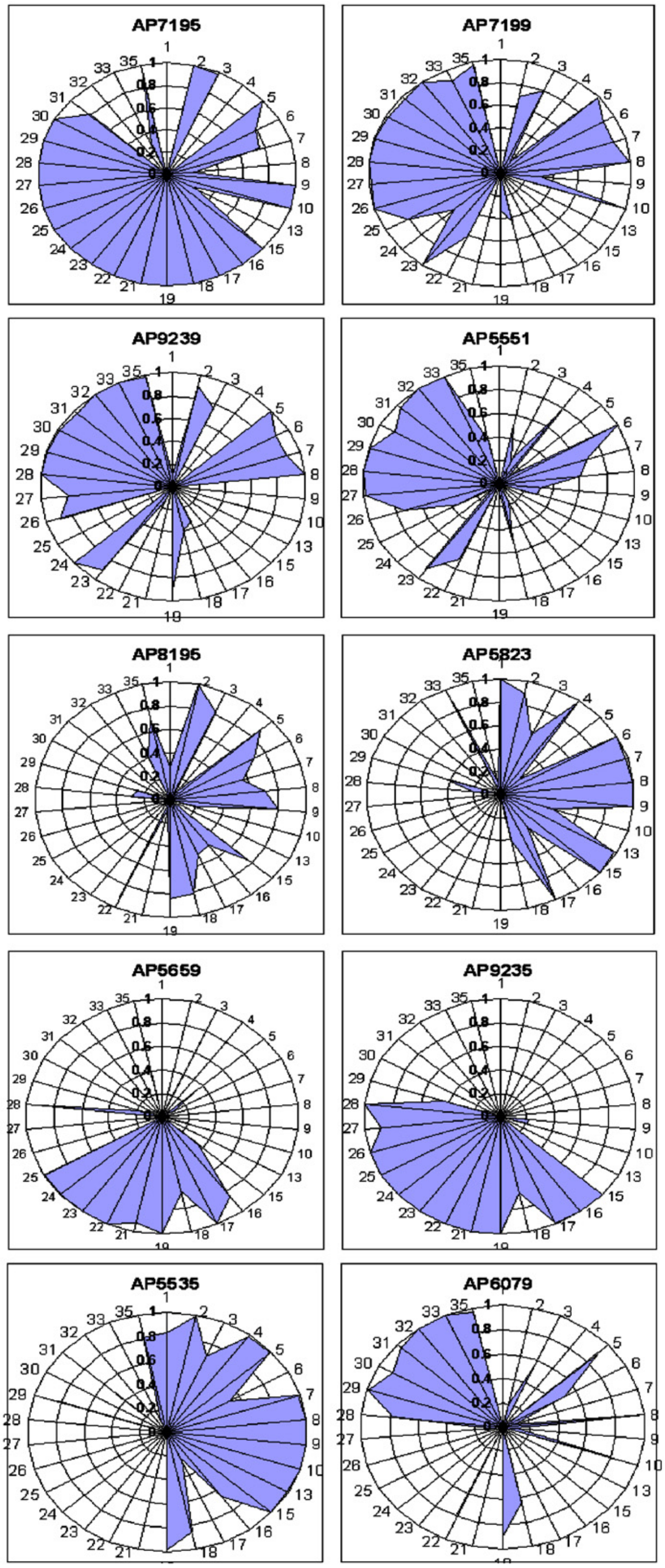

Fig. 6. *Visibility* probability graphs of 10 access points.

access points at m locations can be combined into an $m \times d$ matrix.

$$\textit{Visibility Matrix} := (P_{ij})_{m \times d}, \quad 0 \leqslant P_{ij} \leqslant 1$$

This visibility matrix resolves each location l_j into the corresponding visibility probabilities of all access points.

$$[P_{ij}]_{i=1}^{d} = \textit{Visibility Matrix } (l_j) \tag{10}$$

We calculate the visibility probability of each access point at all target locations as part of site calibration process as explained in Section 5.1. Note that locations 11, 12, 14, 20 and 34 are missing in Fig. 6 graphs because we did not have access to these locations. Careful observation of the visibility graphs, shown in Fig. 6, reveals that among total d access points there exist q subsets or clusters of c access points which are visible at different regions. Let RSS_i^c denote the ith cluster of c access points which belongs to the whole input space RSS^d. Let R_i denote the corresponding ith region where members of RSS_i^c has desirable visibility probability. Then, the *visibility clusters* define all the visibility associated pairs of *access point clusters* and their corresponding *regions* such that

$$[RSS_i^c]_{i=1}^{q} = \textit{Visibility Cluster } ([R_i])$$
$$[R_i] = \textit{Visibility Cluster } (RSS_i^c) \tag{11}$$

where $[RSS_i^c]_{i=1}^{q} \in RSS^d$ and $[R_i] \in A$ as mentioned in Eqs. (6) and (7). Notice that all visibility associations among signal space and location space, as formulated in (11), are defined over the *visibility matrix*.

4.2. *Extracting visibility clusters*

Once a visibility matrix is available, it is desirable to automatically extract: (i) those access point clusters which have more *visibility* probability than certain threshold P_τ and (ii) all locations where members of each cluster are *visible*. Let c denote the desired number of features in subspace and r denote the set of locations in corresponding region. Even though c can vary across different access point clusters, for simplicity of discussion, suppose that c remains constant for all clusters. Algorithm 1 extracts these clusters from the visibility matrix given c and P_τ.

Algorithm 1. Algorithm for extracting visibility clusters from visibility matrix.

Inputs
c: Required number of access points in a cluster
P_τ: Threshold *visibility* probability
Define Global
$L_{[j]}$: Collection of all target locations
j: Index of target location
$AP_{[k]}$: Collection of all access points
k: access point index
p_{kj}: Visibility probability of a k^{th} access point at jth location
$R_{[i]}$: Collection of regions
$RSS_{[i]}$: Collection of access points clusters
i: Index of access point clusters
newAPSet: boolean flag
APSet: temporary collection of access point
for Each Location j in $L_{[j]}$ **do**
  Initialize i, k, newAPset, and APSet
  for Each Access Point k in $AP_{[k]}$ **do**
    if p_{kj} *satisfies* P_τ **then**
      Add $AP_{[k]}$ *to APSet*
    else
      Move to next access point $AP_{[k+1]}$
    end if
  end for
  if APSet has c elements **then**
  set newAPSet flag to **true**
  for Each access point cluster i in $RSS_{[i]}$ **do**
    if *APSet is member of* $RSS_{[i]}$ **then**
      Add $L_{[j]}$ to $R_{[i]}$
      set newAPSet flag to **false**

quit for
else
Move to next access point cluster
$RSS_{[i+1]}$
end if
end for
if newAPSet is **true**
Add new APSet to $RSS_{[i]}$
Add $L_{[j]}$ to $R_{[i]}$
end if
end if
Move to Next Location $L_{[j+1]}$
end for

Even though formation of visibility clusters tends to vary in different sites due to the specific physical layout and environmental conditions at each site, we present general heuristics which largely effect this formation. Combination of three parameters influences the selection of final members of visibility clusters; (a) P_τ visibility probability threshold, (b) c features in subspace, (c) d the total number of features in the radio map, in three respects:

1. Total number of visibility clusters denoted as q. For an admissible visibility P_τ, smaller values of c and larger values of d produce several access point clusters to be visible at similar or overlapping regions. This is because if there are n access points visible in a region and $c<n$, then $n!/c!$ different combinations of access point clusters can become visible in that region. As described in Eq. (8), since each visibility cluster defines inputs and outputs of the classifier module, it is desirable to generate less number of modules in order to reduce overall complexity of the modular classification system.

2. Total number of omitted locations that cannot be covered by any of the clusters. Note that even though c can take possibly all values less than d, nevertheless, any arbitrary access point cluster does not necessarily provide enough coverage in terms of the number of locations.

3. Separability of new subspaces defined by access point clusters. It refers to those ambiguous vectors in feature space which belong to more than one location or class. Neighboring locations often receive similar signal strengths especially when situated in one room or a corridor. A straightforward solution to this issue is to incorporate more access points into feature space. Nevertheless, the exact number of access points required to achieve separability depends on a particular site. We measure separability of training samples as ratio of separable samples and total samples representing one location.

$$s^q_{i=1} = \sum_{j=1}^{m} \frac{T_j - O_j}{T_j}$$

where T_j denotes the total samples and O_j represents the number of overlapping samples in the feature space for the jth location. Summation of these ratios at every location in the ith region gives separability of the ith access point cluster. Fig. 7 shows separability of different visibility clusters.

In order to further explain above heuristics, some example visibility clusters, which exist in one of our experimental sites, are presented here. Effect of changing total number of access points used to define the radio map feature space is shown in Tables 1 and 2. It is evident from these clusters that increasing features, eight access points in Table 1 and 11 in Table 2, results in increased number of visibility clusters q. Three locations are omitted in this clustering. Nevertheless, only three features may not result in completely separable signal vectors. Table 3 shows *visibility* clustering results with different parameters. In this case q is 5 and omitted locations are 3. This clustering produces no overlap in training data of different locations as can be seen in Fig. 7. This shows that adding more access points to the feature space enriches discrimination information by making all sample training vectors separable. However, acceptable values of visibility probability P_τ restrict subspace dimensionality c to 4 and further increment of access points in c results in no visibility clusters.

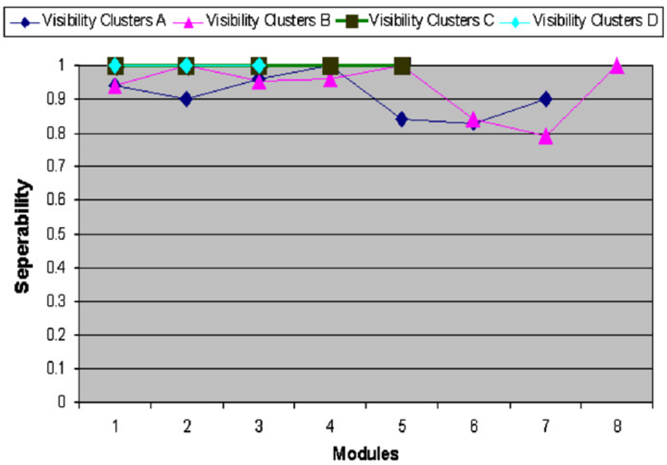

Fig. 7. Separability of visibility clusters.

Table 1
Visibility clusters A

Access point clusters	Regions
AP7195AP9239AP5659	3,19,23,24,28
AP7195AP5659AP9235	16,17,18,19,21,22,23,24,25,28
AP7195AP7199AP9239	2,3,5,6,7,10,23,26,27,28,29,30,31,35
AP5551AP7199AP9239	2,6,7,8,23,26,27,28,29,30,31,32,33
AP7195AP8195AP5823	2,3,6,7,9,15,17
AP7195AP7199AP5659	3,22,23,25,28
AP7195AP5551AP7199	2,6,7,22,23,26,27,28,29,30,31

$d = 8$, $c = 3$, $P_\tau = .55$.

Table 2
Visibility clusters B

Access point clusters	Regions
AP5551AP7199AP9239	6,7,8,23,26,27,28,29,30,31,32,33
AP7195AP5551AP7199	6,7,22,23,26,27,28,29,30,31
AP7195AP7199AP9239	2,3,5,6,7,10,23,26,27,28,29,30,31,35
AP7195AP8195AP5823	2,3,6,7,9,15
AP7195AP5659AP9235	16,17,18,19,21,22,23,24,25,28
AP7195AP7199AP5659	3,22,23,25,28
AP7195AP5659AP8135	16,17,21,22,23,24,25,28
AP7195AP9239AP5659	3,19,23,24,28

$d = 11$, $c = 3$, $P_\tau = .55$.

Table 3
Visibility clusters C

Access Point Clusters	Regions
AP7195AP5659AP9235AP8135	16,17,18,19,21,22,23,24,25,28
AP5551AP7199AP9239AP6079	6,8,28,29,30,31,32,33
AP7195AP5551AP7199AP9239	2,6,7,23,26,27,28,29,30,31
AP7195AP7199AP9239AP6079	3,5,6,10,28,29,30,31,35
AP7195AP8195AP5823AP5535	2,3,6,7,9,15

$d = 11, c = 4, P_\tau = .55$.

Table 4
Visibility clusters D

Access point clusters	Regions
AP7195AP7199AP9239AP6079AP5535	3,5,6,10,29,35
AP7195AP7199AP9239AP8195AP5535	2,3,5,6,7,35
AP5551AP7199AP9239AP6079AP9207	6,8,28,29,30,31,32,33

$d = 12, c = 5, P_\tau = .50$.

Table 5
Binary visibility decision rules

7195	7199	9239	5551	8195	5823	5659	9235	Module
1	1	1	0	0	1	0	0	m1
1	1	1	0	1	0	1	1	m2
1	0	0	0	1	1	0	0	m2
1	0	0	0	0	0	1	1	m4
1	0	0	0	1	0	1	1	m5

Consequently we increased the total number of access points d to 12. Table 4 shows visibility clusters where $c = 5$ in 12 access point radio map. Although this clustering results in only three clusters and completely separable training samples, but 16 locations are omitted.

So far it is confirmed that low values of visibility probability P_τ introduce redundant features in the radio map. This reduces the location estimation accuracy and increases the complexity of the classification system as shown in experimental results (Sections 6.1 and 6.2). On the other hand, low values of c correspond to decreased discrimination ability of access points in each cluster and higher values render either no clustering or result in a large number of omitted locations.

Since visibility clusters correspond to a set of access points which are visible at a set of locations, this information is used to develop binary decision rules, as shown in Table 5, to invoke a specific classifier module for a particular signal input. In this table the availability or absence of signal from a particular access point is represented as either 1 or 0, respectively. The gating module shown in Fig. 5 receives a d-dimensional input feature space vector rss^d, presented in Eq. (3), and converts it into appropriate subspace vectors, Eq. (6), based on visibility status of access points. This converted input vector is then routed to appropriate classifier module depending upon the decision rules in Table 5.

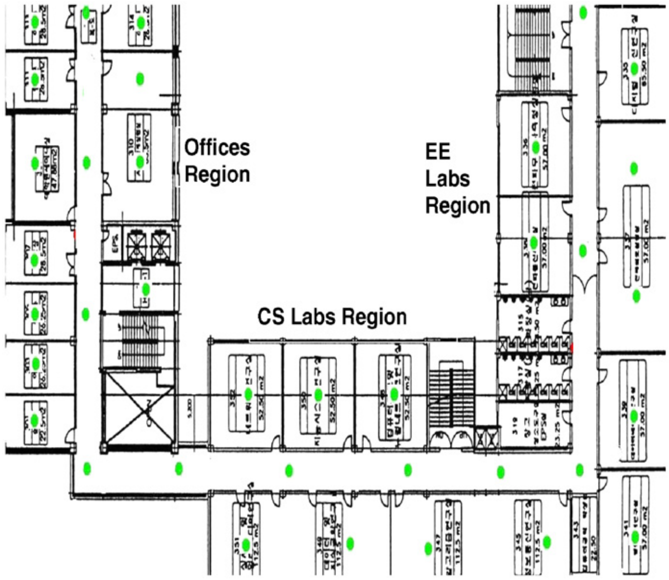

Fig. 8. Experimental site 1.

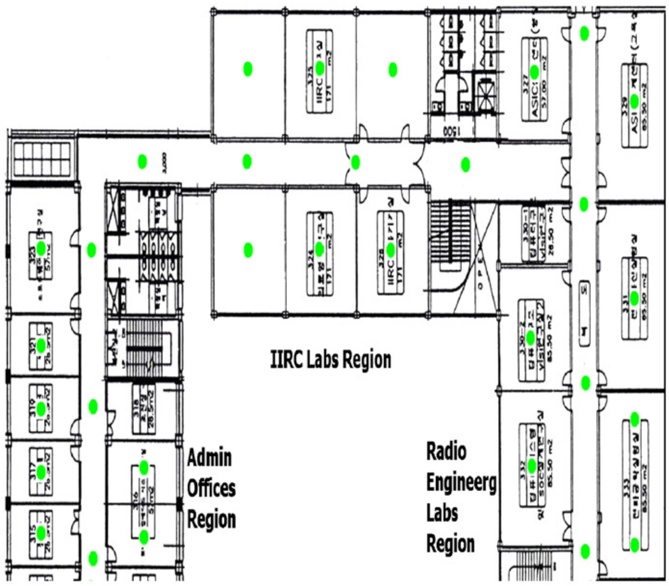

Fig. 9. Experimental site 2.

5. Experimental design

In order to evaluate modular classification model in real life environment we conducted several experiments in a campus environment. The selected sites were located on third floor of computer engineering department. This floor has a versatile environment containing class rooms, labs and offices. We used MATLAB Neural Network Tool Box [22] for training the location classifier. Nevertheless, in order to actually deploy trained classifiers onto mobile devices we developed component-based software libraries. CompoNet [8] encapsulates arbitrary size and structure of neural networks into a software object which can be invoked as a function from other programs. Experiments were carried out using a public network of 3COM IEEE 802.11 (a, b, g) WiFi access points at two different sites. In order to actually scan signal strengths, two types of consumer devices with built in WiFi Network Interface Cards (NIC) were used for RSS scanning: (1) HP iPAQ Pocket PC running Windows CE and (2) Toshiba M30 Laptops running Windows XP. Figs. 8 and 9 show map of sites 1 and 2 with target locations marked as small filled circles.

A subset of the site 2 target locations is situated in IIRC labs[2] where radio signals face high levels of electromagnetic noise generated by equipments (e.g. MRI scanner machines) causing intermittent visibility of access points. Another objective to carry out these experiments is to study the effect of low *visibility* on location estimation performance of classifiers. Visibility clusters of site 2 are presented in Table 6.

Akin to typical pattern classification tasks, RSS based location system development is divided into feature-space creation, preprocessing, classifier training and testing phases. We explain experimental design and test setup in the following sub-sections.

[2] Impedance Imaging Research Center, Kyung Hee University, http://iirc.khu.ac.kr/

Table 6
Site 2 visibility clusters

Access point clusters	Regions
AP2243AP5823AP9239	15,21,16,17,18,19,20
AP2243AP9147AP9239	5,6,7,8,9,10,11,15
AP9147AP9239AP9207	1,2,3,4
AP2243AP5823AP9147	12,13,14,15,16,17,18,28
AP2243AP5823AP5883	22,23,24,25,26,27,30
AP2243AP5823AP7195	28,29,30,31

$d = 9, c = 3, P_\tau = .75.$

5.1. Sensor data collection

Site calibration phase involves scanning RSS patterns at discrete target locations. Signal strengths of access points can be scanned passively and actively. In former case each access point periodically broadcasts an announcement packet. Mobile devices can parse that packet to know the signal source basic service set identifier (BSSID) and signal strength. The BSSID contains MAC address of access point which is used to identify different access points. In active scanning case, the mobile device broadcasts a query signal to access points. In response to the query signal each access point sends a reply signal back to querying device containing its BSSID. We used active scanning method for scanning signal strengths. For each type of device, customized software modules called Calibration Agents were developed to access NIC hardware. Calibration agent invokes the scanning process on an adjustable frequency and parses response signals of each access point. As presented in Section 4, a visibility matrix is developed during calibration phase in addition to the radio map. This is actualized through a histogram based data collection method. Apart from hardware interfacing, the calibration agent builds a histogram data structure in its memory to store scanned data. Fig. 10 shows structure of calibration data for one location and actual RSS histogram plots are shown in Fig.11. Sensor measurements contain pairs of access point BSSID and received signal strength indicator (RSSI) values.

$O^d_{i=1} = [BSSID_i, RSSI_i]$

Calibration agent parses each pair O from radio signals and establishes a separate histogram for individual access points as presented in Algorithm 2. The visibility probability of access points is computed such as for N signal scan observations at jth location, each histogram provides the total number of times the signal from ith access point is detected $p_{ji} = \sum_{i=1}^{n} x_i$. The ratio of N and p_{ji} gives visibility probability as explained in (9).

Algorithm 2. Algorithm for constructing RSS histograms.

1. **Input**
2. *rss*: Received Signal Strength of *i*th access point
3. **Define Global**
4. *new Value*: Boolean flag
5. $H[,]$: Histogram, a two dimensional array
6. S: Histogram size
7. **for** each element c in H **do**
8. **if** *rss* equals H[count,0] **then**
9. *new Value* is **false**
10. $H[c,1] \leftarrow H[\text{count},1] + 1$
11. **else**
12. *new Value* $\leftarrow$ **true**
13. **end if**
14. **end for**
15. **if** *new Value* is **true**
16. $H[S,0] \leftarrow$ rss
17. $H[S,0] \leftarrow H[S,0] + 1$
18. **end if**

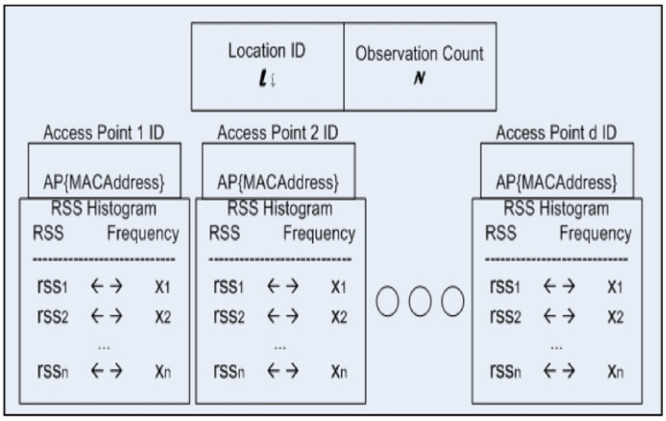

Fig. 10. Calibration data packet.

Prevalent method for site calibration uses a network card interface to extract RSS values from hardware and a graphical program with image map of site which allows developers to pinpoint their location. This practice is tiring for sufficient calibration of large sites. In order to alleviate the required labor for data collection task, we developed a distributed site calibration system which allows multiple devices to simultaneously calibrate site in short time [3]. RSS based location estimation is directly affected by the degree of how closely sample signal data represent the real life radio signals. Therefore we undertook un-customary efforts to gather sample signal strengths. Instead of creating the radio map at one time and then dividing it into training and test samples, we calibrated target sites for 7 days and at different timings of each day. Each target location was calibrated multiple times by different people and devices. Data of alternate days were used for, respectively, training and testing the classifiers. During sample data collection routine activities were taking place which are peculiar to typical indoor environments such as lecture rooms, labs and adminstration offices.

5.2. Training phase

In this phase the *radio map* feature space is used to train location estimation classifier. Two classification methods, MLP and LVQ, were employed for location estimation as suggested by Battiti et al. [5] and Ogawa et al. [24]. During training phase, preprocessing of feature space is one important step to encode inputs and outputs into a format suitable for classification method. We applied range normalization on the *radio map* feature space.

$$rss_{\text{norm}} = 2((rss - rss_{\min})/(rss_{\max} - rss_{\min})) - 1$$

where rss_{norm} is the normalized signal strength. RSS values fall in the range of -100 to -10 dBm [27], We used these values as global minima $rss_{\min}$ and maxima $rss_{\max}$ for normalizing all features.

Each location is identified in two ways: (a) unique identification number (ID) and (b) cartesian coordinates on xy-plane. The former was used for classifier training and later for error analysis. For classifier training, class labels or target locations are assigned in the range of 1 to 35 and 1 to 31, for sites 1 and 2, respectively. This ID is then encoded into sparse array which has all 0-valued elements except the one at the index of ID which contains 1. This

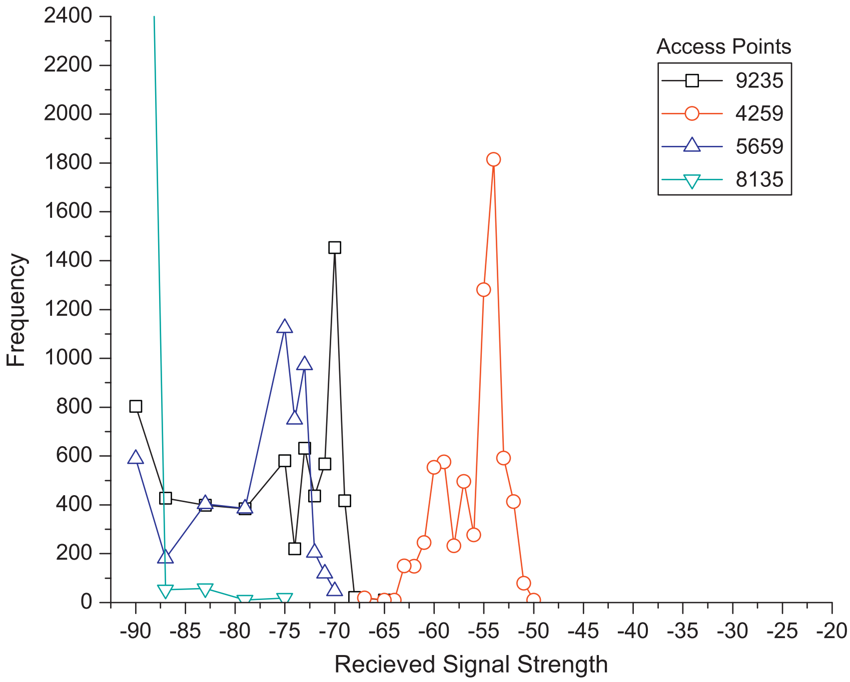

Fig. 11. RSS histogram of four access points at one location.

array has zero values in all elements for the location which was not accessible for calibration. We measure location estimation error in two aspects:

(a) Absolute deviation of location estimate from actual location is measured as mean absolute error (MAE)

$$|MAE| = \frac{1}{N}\sum_{i=1}^{N}|l_i - \hat{l}_i|$$

where N is the total number of training or test patterns, l_i is the estimated location and $\hat{l}_i$ is the true location of ith pattern. Deviation between l_i and $\hat{l}_i$ is calculated as euclidian distance

$$\Delta_i = |l_i - \hat{l}_i| = \sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2} \tag{12}$$

where (x_i, y_i) are corresponding coordinates of l_i location estimate of ith pattern and $(\hat{x}_i, \hat{y}_i)$ is the actual location.

(b) Relative deviation of location estimate, denoted as e_r, is measured relative to some tolerable error threshold denoted as Γ. It implies that relative error reflects severity of error in location estimate by allowing some deviation which is not more than Γ. An estimate is less severe if it is relatively closer to actual location than the one which is farther away. Thus E_r gives a percentage of total N estimates in which $\triangle_i$ is admissible or less than Γ.

$$E_r = \frac{1}{N}\sum_{i=1}^{N}[(\triangle_i \leqslant \Gamma) = 1]$$

where threshold $\Gamma_{i=1}^{3}$ produces three severity of error values averaged over all locations.

5.2.1. *Training multi-layer perceptron classifiers*

Separate MLP based location estimation systems were trained for both sites 1 and 2. For the sake of comparative analysis both the non-modular, represented as MLP, and modular, represented as $mMLP_i$, location estimation systems were trained. Site 1 radio map was composed of 11 access points for training MLP and for the mMLP classifiers the training data was extracted using Table 3 visibility clusters. On the other hand, nine access points constituted radio map for site 2 and the visibility clusters are shown in Table 6.

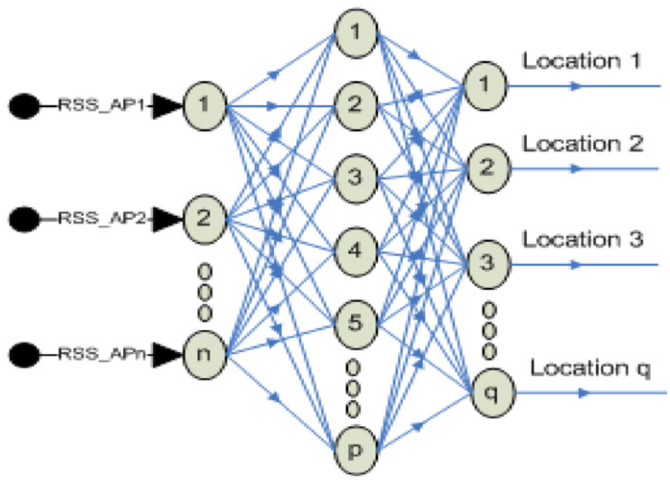

Fig. 12. Employing multi layer perceptron for RSS based location estimation.

It is established that a single hidden layer is sufficient to learn any continuous function to a desired accuracy, given that the number of hidden neurons is sufficient [16]. Fig. 12 shows an arbitrary structure of an MLP network for location estimation. Network takes individual components of the RSS vector as input and produces an estimate of most likely location of the device which is reporting these RSS values.

Several MLP networks were trained with different parameters such as hidden layer neurons, learning function, transfer functions and training epochs. Training results of only the best performing networks for site 1 location system are represented in Table 7 and Table 8 shows training results for site 2. Detailed discussion of the effects of different parameters on location estimation performance is given in [1].

Table 7
Site 1: training results of multi-layer perceptron classifiers

Module	Training function	Topology	Epochs	Training MAE
MLP	SCG	11–70–35	2000	0.021
mMLP1	LM	4–20–10	88	0.021
mMLP2	LM	4–20–8	80	0.082
mMLP3	LM	4–25–10	86	0.0184
mMLP4	LM	4–20–9	50	0.0047
mMLP5	LM	4–15–6	45	0.009

Table 8
Site 2: training results of multi-layer perceptron classifiers

Module	Training function	Topology	Epochs	Training MAE
MLP	SCG	12–200–31	1000	0.138
mMLP1	SCG	3–40–8	100	0.023
mMLP2	LM	3–40–8	64	0.001
mMLP3	LM	3–20–4	14	0.01
mMLP4	LM	3–40–8	23	0.01
mMLP5	LM	3–35–7	77	0.005
mMLP6	SCG	3–30–4	100	0.013

As results tables show in 'Training Function' column, two training algorithms were employed to learn pattern–location pairs: (i) Levenberg–Marquardt (LM) developed by Hagan and Menhaj [13] and (ii) Moller's scaled conjugate gradient (SCG) presented in [23]. In our experiments, LM algorithm achieved nearly zero training error in fewer iterations but its requirement of computational resources is prohibitive for training large networks. On the other hand SCG achieved comparable performance and did not require excessive computational as well as memory resources during training.

Topology column contains structure of respective neural network which is represented as I–H–O, where I is input neurons, H is hidden layer neurons and O is output layer neurons. Epochs column shows number of training iterations that respective networks took to achieve MAE which is listed in the last column. One common property that all networks in our experiments share is the choice of transfer functions which is *logsig*moidal function $logsig(n) = 1/(1 + e^{-n})$ at hidden layer neurons and *tansig*moidal function $tansig(n) = 2/(1 + e^{-2n}) - 1$ at the output neurons.

It is obvious from training results that modular classifier model employs networks which are simpler in structure and takes less than 100 iterations to learn pattern–location associations in comparison with non-modular networks.

5.2.2. *Training LVQ classifiers*

LVQ classifier employs non-parametric nearest neighbor pattern recognition algorithm based on Kohonen's self-organizing-maps [20]. Fig. 13 shows application of an arbitrary structure of an LVQ network for location estimation. Network takes individual components of RSS vector as input and produce an estimate of most likely location of the device which is reporting these RSS values.

Design of LVQ network specifies three layers of neurons. Input layer contains as many neurons as components of input vector I. Hidden layer contains competitive neurons to represent sub-classes and output layer consists of neurons which represent actual classes in the input space.

The hidden or competitive layer learns to classify input vectors in much the same way as the competitive layer of self-organizing-maps. Input neurons and competitive layer neurons are interconnected through IW weights. When an RSS input is applied to the network, the distance between each hidden layer neuron and input vector is computed. Then winner takes all (WTA) competition is applied to find the closest subclass, the winner, neuron IW_w. Once the input vector is classified at hidden layer, the third layer, also called linear transformation layer, transforms the output of competitive layer into target classification vectors. Learning occurs by adjusting the IW_w weights in such a way as to move it closer to input vector if classification result is correct

$$IW_w = IW_w + \alpha(I - IW_w)$$

or farther if input is incorrectly classified

$$IW_w = IW_w - \alpha(I - IW_w)$$

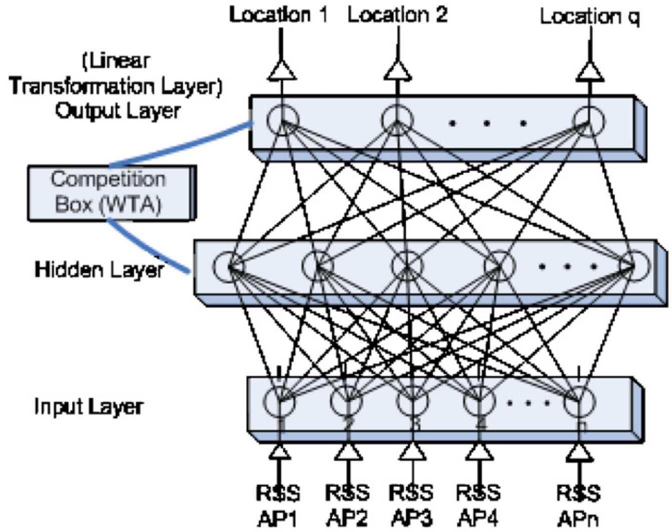

Fig. 13. Employing LVQ network for RSS based location estimation.

Table 9
Site 1: training results of LVQ classifiers

Classifier	Learning function	Topology	Epochs	Training MAE
LVQ	lvq1	8–100–35	150	0.018
mLVQ1	lvq1	5–30–10	45	0.03
mLVQ2	lvq1	4–30–8	20	0.009
mLVQ3	lvq1	4–25–10	40	0.027
mLVQ4	lvq1	4–25–9	25	0.057
mLVQ5	lvq1	4–15–6	15	0.048

Table 10
Site 2: training results of LVQ classifiers

Classifier	Learning function	Topology	Epochs	Training MAE
LVQ	lvq1	12–200–31	161	0.019
mLVQ1	lvq1	3–40–8	10	0.14
mLVQ2	lvq1	3–40–8	76	0.045
mLVQ3	lvq1	3–20–4	45	0.035
mLVQ4	lvq1	3–40–8	70	0.058
mLVQ5	lvq1	3–30–7	56	0.11
mLVQ6	lvq1	3–20–4	40	0.09

Learning rate of the network is determined by the parameter α. Number of hidden layer neurons influences the learning as well as generalization capability of LVQ networks. We developed several LVQ networks with different hidden layer neurons. Tables 9 and 10 list training results of best performing LVQ classifiers for sites 1 and 2. Analogous to the naming convention of MLP experiments, modular networks are denoted as $mLVQ_i$ and non-modular ones are denoted as LVQ in results tables.

6. Test results

As mentioned in Section 5.1 we employed RSS data of different days for testing the classifiers. The following subsections present location estimation results with test radio map for each site.

6.1. Site 1 test results

Modular classification, mMLP and mLVQ, results for site 1 are shown in Tables 11 and 12. Both classifiers exhibit similar performance, in terms of absolute and relative errors, except modules 3 and 5.

Overall performance of modular classification system is compared with non-modular classifier in Table 13. In this site MLP performed better than LVQ classifier both in terms of absolute and relative error. However, LVQ benefits from modularity significantly more than MLP with respect to absolute distance error. The mMLP and mLVQ produced relative positioning error $E_r \leqslant 1$ for 84% and 80% times, respectively. On the other hand, non-modular MLP and LVQ achieve similar performance if $E_r \leqslant 3$, which means that severity of error of modular classification system is three times better in this site than monolithic counterparts. As we increased the tolerable error threshold up to three positions, modular classification system consistently provided superior results in comparison with non-modular approach.

Table 11
Site 1: modular approach results of modular MLP

Classifier	Test patterns	Test MAE	$E_r \leqslant 1$ (%)	$E_r \leqslant 2$ (%)	$E_r \leqslant 3$ (%)
mMLP1	140	0.4140	85	90	91
mMLP2	140	0.7286	91	92	94
mMLP3	165	0.4957	81	95	96
mMLP4	115	0.556	91	92	99
mMLP5	103	0.9806	68	88	88

Table 12
Site 1: modular approach results of modular LVQ

Classifier	Test patterns	Test MAE	$E_r \leqslant 1$ (%)	$E_r \leqslant 2$ (%)	$E_r \leqslant 3$ (%)
mLVQ1	140	0.5571	87	91	91
mLVQ2	140	0.7214	91	91	92
mLVQ3	165	0.9760	66	92	93
mLVQ4	115	0.4087	93	95	95
mLVQ5	103	1.25	61	78	94

Table 13
Site 1: summarized results

Classifier	Test MAE	$E_r \leqslant 1$ (%)	$E_r \leqslant 2$ (%)	$E_r \leqslant 3$ (%)
MLP	1.0	70	78	85
mMLP	0.6349	84	92	95
LVQ	2.4	54	68	78
mLVQ	0.7831	80	90	93

Table 14
Site 2: modular approach results of modular MLP

Classifier	Test patterns	Test MAE	$E_r \leqslant 1$ (%)	$E_r \leqslant 2$ (%)	$E_r \leqslant 3$ (%)
mMLP1	109	1.13	68	85	95
mMLP2	105	0.33	93	99	100
mMLP3	65	0.06	100	100	100
mMLP4	194	1.17	60	82	91
mMLP5	194	0.77	72	88	99
mMLP6	87	0.44	100	100	100

Table 15
Site 2: modular approach results of modular LVQ

Classifier	Test patterns	Test MAE	$E_r \leqslant 1$ (%)	$E_r \leqslant 2$ (%)	$E_r \leqslant 3$ (%)
mLVQ1	109	1.43	54	56	83
mLVQ2	105	0.15	93	99	99
mLVQ3	65	0.04	100	100	100
mLVQ4	194	0.92	65	90	100
mLVQ5	194	1.08	61	83	92
mLVQ6	87	0.32	100	100	100

Table 16
Site 2: summarized results

Classifier	Test MAE	$E_r \leqslant 1$ (%)	$E_r \leqslant 2$ (%)	$E_r \leqslant 3$ (%)
MLP	1.56	54	69	91
mMLP	0.63	82	88	98
LVQ	1.49	72	75	85
mLVQ	0.66	79	88	96

6.2. Site 2 test results

The generalization performance of mMLP and mLVQ on site 2 radio map is presented in Tables 14 and 15, respectively. Note that, for both mMLP and mLVQ, performance of 1,4 and 5 modules is aggravated in comparison with other modules. This is mainly because the target locations of these modules are situated in an area which is permeated with high levels of radio noise as mentioned in Section 5. Due to the specific environmental conditions, mobile devices in this area face frequent visibility problems.

Summarized test results for site 2 are shown in Table 16 with respect to all classification methods. Note that even though modules 1, 4 and 5 produce relatively higher distance error than other modules, these mMLP produce lower error in comparison with MLP and LVQ classifiers. It concurs the tendency of modular classifier to show more resilience to noise and render better location estimates than non-modular MLP and LVQ. It is interesting to note that the mMLP based location systems continued to perform better than mLVQ as in site 1, which implies that MLP demonstrates better competence than LVQ. However, LVQ based location system surpassed MLP in terms of absolute error and produced similar relative error.

Based on these results, it is clear that modular classification model improves location estimation accuracy. This performance holds across different sites both in terms of absolute location error and relative error. Moreover, different classification methods show similar trend to gain benefit from modularity.

7. Conclusions

RSS based location systems are poised to enable indoor positioning systems due to their economic viability. Intrinsically, signal strength based location estimation is a pattern classification problem. Large scale realization of these systems face with the visibility issue. This introduces null values in the radio map feature space resulting in sparsity and redundancy. A modular classification model is presented to overcome the visibility problem by incorporating the prior knowledge about signal visibility into design of the classification system. This is achieved by partitioning high-dimensional and sparse radio map feature space into low-dimensional but compact subspaces. Two location systems were developed for different sites in order to confirm the applicability and robustness of modular approach in real life environments. It is demonstrated through development of location systems in two real life environments that signal visibility based decomposition of radio map enables development of location systems in arbitrarily large target sites.

The location estimation performance is compared with monolithic, non-modular classification approach using two famous neural networks: (1) multi layer perceptron and (2) LVQ. Same neural networks were employed to realize the modular classification model. Comparative results show superiority of modular approach in terms of both absolute error and relative error across different sites. The noteworthy improvement in location estimation accuracy is observed to be consistent across different sites. With respect to absolute error measure, in case of site 1, compared to non-modular approach the accuracy of modular approach is improved 1.58 and 3 times, respectively, for MLP and LVQ classifiers. Similarly, in case of site 2, the accuracy is 2.47 and 2.55 times better than monolithic classifiers. Both classifiers benefit from the modular approach in terms of relative error in the same way. On the basis of extensive experimental results we conclude that modular classification model achieves significant improvement in location estimation accuracy as well as enables systematic expansion in coverage area of location systems.

Acknowledgements

This research was supported by the MIC (Ministry of Information and Communication), Korea, Under the ITFSIP (IT Foreign Specialist Inviting Program) supervised by the IITA (Institute of Information Technology Advancement).

References

[1] U. Ahmad, et al., In-building localization using neural networks, in: IEEE International Conference on Engineering of Intelligent Systems, Islamabad, Pakistan, March 2006.
[2] U. Ahmad, et al., Modular multilayer perceptron for WLAN based localization, in: IEEE International Joint Conference on Neural Networks, Vancouver, Canada, July 2006.
[3] U. Ahmad, et al., Location-ware: a distributed wireless site calibration system for location awareness, in: The 2006 IFIP International Conference on Embedded and Ubiquitous Computing, Seuol, Korea, August 2006.
[4] P. Bahl, et al. RADAR: an in-building RF-based user location and tracking system, in: IEEE INFOCOM 2000, March 2000, pp. 775–784.
[5] R. Battiti, A. Villani, T. Le Nhat, Neural network model for intelligent networks: deriving the location from signal patterns, in: The First Annual Symposium on Autonomous Intelligent Networks and Systems 2002.
[6] C.C. Chiang, A divide-and-conquer methodology for modular supervised neural network desien, in: Proceedings of 1994 International Joint Conference on Neural Networks, pp. 119–124.
[7] S.-B. Cho, Combining multiple neural networks by fuzzy integral for robust classification, IEEE Trans. Syst. Man Cybern. 25 (2) (1995).
[8] CompoNet: Componentization of Neural Networks ⟨http://www.http://sourceforge.net/projects/componet/⟩.
[9] F. Dieter, et al., Bayesian filters for location estimation, in: IEEE CS and IEEE ComSoc 1536-1268/03.
[10] C. Ding, H. Peng, Minimum redundancy feature selection for gene expression data, in: Proceedings of the IEEE Computer Society Bioinformatics Conference (CSB 2003), Stanford, CA, August 2003, pp. 523–529.
[11] S.-U. Guan, et al., Neural network task decomposition based on output partitioning, Journal of the Institution of Engineers, Singapore 44 (3) (2004).
[12] I. Guvenc, et al., Enhancements to RSS based indoor tracking systems using kalman filters, in: GSPx International Signal Processing Conference, Dallas, TX, March 31–April 3, 2003.
[13] M.T. Hagan, M. Menhaj, Training feedforward networks with the Marquardt algorithm, IEEE Trans. Neural Networks 5 (6) (1994) 989–993.
[14] J.B. Hampshire, et al., The Meta-Pi network: Building distributed knowledge representations for robust multisource pattern recognition, IEEE Trans. Pattern Anal. Mach. Intell. 14 (7) (1992).
[15] T.K. Ho, Decision combination in multiple classifier systems, IEEE Trans. Pattern Anal. Mach. Intell. 16 (I) (1994) 66–75.
[16] K. Hornik, Approximation capabilities of multilayer feedforward networks, Neural Networks 4 (1991) 251–257.
[17] A.K. Jain, B. Chandrasekaran, Dimensionality and sample size considerations in pattern recognition practice, in: Handbook of Statistics, vol. 2, North-Holland, Amsterdam, 1987, pp. 835–855.
[19] J. Kittler, et al., On combining classifiers, IEEE Trans. Pattern Anal. Mach. Intell. 20 (3) (1998).
[20] T. Kohonen, Learning vector quantization, in: The Handbook of Brain Theory and Neural Networks, MIT Press, Cambridge, MA, 1995, pp. 537–540.
[21] A.M. Ladd, et al., Using wireless ethernet for localization, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, Lausanne, Switzerland, September 2002.
[22] MATLAB Neural Network Toolbox ⟨http://www.mathworks.com/products/neuralnet/⟩.
[23] M.F. Moller, A scaled conjugate gradient algorithm for fast supervised learning, Neural Networks 6 (1993) 525–533.
[24] T. Ogawa, S. Yoshino, M. Shimizu, Location determination method for wireless systems based on LVQ, NTT Network Innovation Laboratories, vol. 1, no. 9, Japan.
[25] K. Pehlavan, et al., Indoor geolocation science and technology, IEEE Communications Magazine 40 (2) (2002) 112–118.
[26] N.B. Priyantha, A. Chakraborty, H. Balakrishnan, The cricket location-support system, in: Proceedings of the Sixth ACM MOBICOM, Boston, MA, August 2000.
[27] Steve Pope, ''Issues related to RSSI measurement, IEEE 802.11-02/520r0 ⟨www.grouper.ieee.org/groups/802/11/Minutes/Cons_Minutes_Nov-2003.pdf⟩.
[28] M. Sarkar, Modular pattern classifiers: a brief survey, in: Proceedings of IEEE International Conference on Systems, Man, and Cybernetics, 2000, pp. 2878–2883.
[29] Smailagic, et al., Location sensing and privacy in a context aware computing environment, Pervasive Computing, 2001.
[30] F. Soulie, Multi-modular neural network-hybrid architectures: a review, in: Proceedings of 1993 International Joint Conference on Neural Networks, pp: 2231–2236.
[31] R. Want, et al., The active badge location system, ACM Trans. Inf. Syst. 10 (1) (1992) 91–102.
[32] A. Ward, et al., A new location technique for the active office, IEEE Pers. Commun. 4 (5) (1997) 42–47.
[33] L. Xu, et al., Methods of combining multiple classifiers and their applications to handwriting recognition, IEEE Trans. Syst. Man Cybern. 22 (3) (1992).

Uzair Ahmad recieved Post Graduate Diploma (PGD-IT) and masters degree (MIT) in Information Technology in 2000 and 2002 from NUST Institute of Information Technology (NIIT), Pakistan. Currently he is a Ph.D. student in Department of Computer Engineering, Kyung Hee University, Republic of Korea. His research interests are autonomous systems, machine learning, web technologies and bio-inspired intelligence. He is student member of IEEE and International Neural Network Society.

Andrey V. Gavrilov received diploma of system engineer in automation control systems from Novosibirsk State Technical University in 1974 and the Ph.D. degree in engineering (in computer systems and networks) from Institute of Mathematics of Siberian Branch of AS of USSR in 1979. His current research interests are in neural networks and its application in ubiquitous computing systems and robotics, hybrid intelligent systems and tools and natural language processing.

Sungyoung Lee received his B.S. from Korea University, Seoul, Korea. He got his M.S. and Ph.D. degrees in Computer Science from Illinois Institute of Technology (IIT), Chicago, Illinois, USA, in 1987 and 1991, respectively.

He has been a professor in the Department of Computer Engineering, Kyung Hee University, Korea, since 1993. His current research focuses on ubiquitous computing and applications, context-aware middleware, sensor operating systems, real-time systems and embedded systems. He is a member of the ACM and IEEE.

Young-Koo Lee got his B.S., M.S. and Ph.D. in Computer Science from Korea Advanced Institute of Science and Technology, Korea. He is a professor in the Department of Computer Engineering at Kyung Hee University, Korea. His research interests include ubiquitous data management, data mining and databases. He is a member of the IEEE, the IEEE Computer Society and the ACM.