Power-Efficient Access-Point Selection for Indoor Location Estimation

Yiqiang Chen, Qiang Yang, *Senior Member*, *IEEE*, Jie Yin, and Xiaoyong Chai

Abstract—An important goal of indoor location estimation systems is to increase the estimation accuracy while reducing the power consumption. In this paper, we present a novel algorithm known as `CaDet` for power-efficient location estimation by intelligently selecting the number of Access Points (APs) used for location estimation. We show that by employing machine learning techniques, `CaDet` is able to use a small subset of the APs in the environment to detect a client's location with high accuracy. `CaDet` uses a combination of information theory, clustering analysis, and a decision tree algorithm. By collecting data and testing our algorithms in a realistic WLAN environment in the computer science department area of the Hong Kong University of Science and Technology, we show that `CaDet` (*C*lustering *a*nd *De*cision *T*ree-based method) can be much higher in accuracy as compared to other methods. We also show through experiments that, by intelligently selecting APs, we are able to save the power on the client device while achieving the same level of accuracy.

Index Terms—Data mining in mobile wireless networks, power efficient computation.

1 Introduction

In today's pervasive computing applications, location-estimation systems are becoming increasingly important as well as practical. Such systems can be used to support many location-based services, such as content delivery and object and people tracking. For indoor location estimation, a challenging problem is how to estimate a client's locations from the signals received using a wireless device with limited computational and power resources. In this paper, we show how to use data mining techniques to perform indoor location estimation with a focus on saving the power usage on a client device. The question we ask is: Is it possible to apply data mining techniques to wireless data to detect the locations of a client while using a minimal amount of battery power?

Our answer is positive. In this paper, we conduct a comprehensive study on realistic wireless data to compare different methods in probabilistic location estimation. Our analysis shows that for wireless data sets collected in indoor wireless environments, in order to apply probabilistic location estimation, information theory provides the best feature selection methods for identifying the most important access points and for minimizing the online samples needed for decision making. We also show that to ensure energy-efficient computation, which is a new constraint for real-time data mining systems, a multiple-decision-tree-based approach can be used. We develop an algorithm based on this approach which we call `CaDet`, which stands for (Clustering and Decision-tree-based method). We relate the accuracy of location estimation with sampling time and energy consumption.

Our work builds on the previous work on location estimation, based on the use of inexpensive wireless local area network (LAN) as the fundamental infrastructure. To detect user locations, the signals from different access points (AP) are collected and used as a basis for location estimation [1], [2], [3], [4], [5]. Systems that utilize the estimated location for further analysis of user goals and objectives are also emerging [6], [7]. In the indoor environment where clients only sense the signal-strength values from different APs, the location estimation problem is full of uncertainty. It is therefore not surprising that a major accepted practice is to apply probabilistic techniques. For example, in the system developed by Ladd et al. [1], it was reported that the location of a client can be estimated to be within 1.5 meters with 83 percent of confidence. However, while most previous probabilistic location estimation works are based on data mining methods, such as clustering and regression, one important question remains: how to ensure the consumption of energy on a client device while achieving a high-level of accuracy?

To the best of our knowledge, our work on `CaDet` is the first that links energy consumption with the data mining methods used in the wireless domain. Our major contribution is to introduce energy consumption as an objective in the design of data mining algorithms for building prediction models. In this area, we propose a client-based architecture on which to build `CaDet` where the client processes the signals sent by various APs in location estimation. An energy efficient prediction model is installed on the client. We show the advantage of this architecture in Section 3.3. In addition, we present a multiple-decision-tree-based approach, where a collection of decisions are built, one for each cluster, in an offline phase. The decision trees

- *Y. Chen is with the Institute of Computer Technology, Chinese Academy of Sciences, Beijing, China. E-mail: yqchen@ict.ac.cn.*
- *Q. Yang and J. Yin are with the Department of Computer Science, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong. E-mail: {qyang, yinjie}@cs.ust.hk.*
- *X. Chai is with the Department of Computer Science, Purdue University, 250 N. University Street, West Lafayette, IN 47907. E-mail: chai@cs.purdue.edu.*

Manuscript received 8 Aug. 2005; revised 5 Dec. 2005; accepted 23 Jan. 2006; published online 18 May 2006.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-0301-0805.

1041-4347/06/$20.00 © 2006 IEEE Published by the IEEE Computer Society

allow a minimum of access points to be used, thus reducing the computation and wake-up time on the client. We demonstrate the impact on energy savings on the client device. Our design makes a contribution to pervasive computing as well since client-based location estimation is shown to be effective for preserving the privacy of clients, but takes more power in computation in general. Thus, an important issue in pervasive computing is how to save energy while producing effective predictions.

In the data mining area, `CaDet` is one of the first works to apply feature selection to access point selection, and to apply a multidecision-tree-based model for wireless data. A special character of wireless data is the high level of uncertainty associated with the data, such that even at the same location, the signals change a lot. We show that with this data, although the accuracy of the model is correlated to the sampling time, an important issue that is how to achieve the best balance between sampling time and accuracy. Saving in sampling time can save energy. We show that using information theory-based feature selection methods in `CaDet`, it is possible to optimize the location-estimation accuracy and reduce the number of samples needed for high accuracy computation. We carry out comparisons with a variety of other methods, including maximum likelihood and clustering methods, on realistic wireless data that we have collected.

The paper is organized as follows: Section 2 discusses related work. Section 3 introduces the problem domain. Section 4 presents the algorithms used in our analysis. Section 5 presents the experiments. The paper concludes in Section 6 with a discussion of future work.

2 Related Work

Various techniques have been proposed in recent years to determine a user's location using radio frequency (RF) signals. In surveying the related work, we consider two different aspects, namely, the location estimation work and the power-efficiency work.

2.1 Previous Work on Location Estimation

In general, the location estimation research can be classified into two categories: deterministic techniques and probabilistic techniques. Deterministic techniques [8], [4], [5], [9] use deterministic inference methods to estimate a user's location. The RADAR system developed by Microsoft Research [8], [4] proposes nearest-neighbor heuristics and triangulation methods to infer a user's location. It maintains a radio map which tabulates the signal strength received from different access points at selected locations. Each signal-strength measurement is then compared against the radio map and the coordinates of the best matches are averaged to give the location estimation. The accuracy of RADAR is about three meters with 50 percent probability. The LANDMARC system [5] exploits the idea of reference points to alleviate the effects caused by the fluctuation of RFID signal strength. The accuracy is roughly one to three meters. However, the placement of reference tags should be carefully designed since it has a significant effect on the performance of the system. Moveover, the RFID readers are so expensive that it is infeasible for localization in a large area. In [9], an online procedure based on feedback from users was employed to correct the location estimation of the system.

Another branch of research is the probabilistic techniques [10], [3], [11], [1], [6] which construct a conditional probability distribution over locations in the environment of interest. In [1], Ladd et al. use probabilistic inference methods for localization. They first use Bayesian inference to compute the conditional probability over locations, based on received signal-strength measurements from nine access points in the environment. Then, a postprocessing step, which utilizes the spatial constraints of a user's movement trajectories, is used to refine the location estimation and reject the results with significant change in the location space. Depending on whether the postprocessing step is used or not, the accuracy of this method is 83 or 77 percent within 1.5 meters. In addition, Roos et al. [11] compare the performance of the nonprobabilistic nearest-neighbor method with that of two probabilistic approaches. The results show that the two probabilistic approaches produce better results than the nearest-neighbor method and the average location estimation error is below two meters. Furthermore, the time-series analysis technique [3] was introduced to study the correlation among consecutive samples received from the same access point over time. The authors reported that better accuracy can be achieved by taking such correlation into account.

While probabilistic techniques provide more accurate results than deterministic techniques, which has been proven formally in [12], a trade-off between computational overhead and accuracy has been introduced.

2.2 Previous Work on Ensuring Energy Efficiency

Because the client devices are usually small, self-maintained devices that depend on battery power, the question of how to save energy has attracted much attention from various research teams. In the pervasive computing area, there are two major research problems regarding energy consumption: One concerns hardware and the other software. In hardware design, a major problem is how to make mobile devices lighter and more compact without adding more power consumption. There has been much work on hardware power management which focuses on different components such as the network [13], [14], disk [15], [16], and CPU [17], [18].

For the software side of the issue, the mobile software continues to grow in complexity, hence increasing the energy demand. There is a lot of work that addresses energy savings from two different aspects: communication components and computation components. In order to reduce power consumption, researches are focused on optimizing the communication cost by deactivating radio as much as possible or by trading off computation for communication. For example, in the wireless data broadcast protocol, mobile devices turn on the radio only during the arrival time of the requested data frames [19], [20]. Similarly, in sensor networks, a localized network architecture [21] is proposed to achieve power savings by allowing most of the sensor nodes to stay in the sleep mode and by reducing the amount of long-range transmissions. In addition, a low-energy adaptive clustering hierarchy [22] is

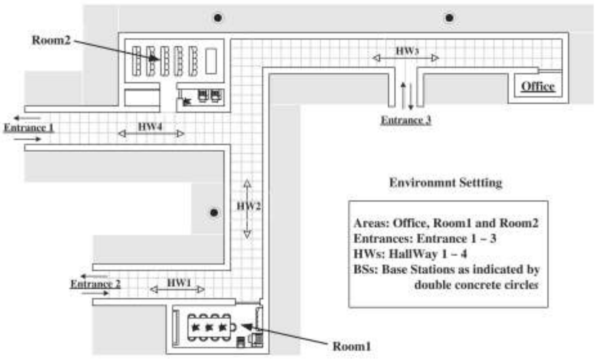

Room2
HW3
Office
Entrance 1
HW4
Entrance 3
Environmnt Settting
HW2
Areas: Office, Room1 and Room2
Entrances: Entrance 1 – 3
HWs: HallWay 1 – 4
BSs: Base Stations as indicated by double concrete circles
Entrance 2
HW1
Room1

Fig. 1. The layout of the office area of the Computer Science Department of Hong Kong University of Science and Technology.

presented to reduce the amount of information that must be transmitted. However, while computing components consume less power than the communication components, they are still important sources of energy dissipation, especially after the communication cost is optimized. Thus, a prediction-based energy-saving scheme [23] is proposed to reduce the energy consumption in the computational components of sensor networks for object tracking.

In the location estimation area, little work has tackled the issue of how to reduce the computational overhead during online location estimation. Youssef et al. uses a joint clustering technique to group locations in order to reduce the computational cost of the system [10]. The method defines a cluster as a set of locations sharing the same set of access points. The location determination process is as follows: After a signal-strength measurement is made, the strongest access points are used to determine one cluster to search within for the most probable location, and then the maximum likelihood (ML) method is used to estimate the most probable location within the cluster. However, their method suffers from two disadvantages: First, the clustering step only selects the same set of access points with the strongest signal strength to represent a cluster; however, the discriminative ability of different access points toward locations and different signal values from the same set of access points have not been considered. Second, the ML method requires the multiplications of a few conditional probability distributions. This is still demanding in power-constrained client devices. Our work, in the energy savings respect, contributes to intelligently selecting access points for the purpose of clustering and then applying an efficient estimation method to reduce the computational cost.

3 Wireless Environment

In this section, we begin with a description of out experimental setup. We then discuss the RF signal propagation and the noisy wireless channel characteristics, which make location estimation a challenging task.

3.1 Overview of the Environment

Our experimental test-bed is set up in the faculty office area of the Computer Science Department in the Academic Building of the Hong Kong University of Science and Technology. The building is equipped with an IEEE 802.11b wireless Ethernet network in the 2.4 GHz frequency bandwidth. The layout of the floor is shown in Fig. 1. This area has a dimension of 64 meters by 50 meters. Experiments were carried out in the four hallways (HW1 ~ HW4) and two rooms as labeled in the figure. The four hallways measure 19.5 for HW1, 37.5 for HW2, 46 for HW3, and 21 for HW4 in meters.

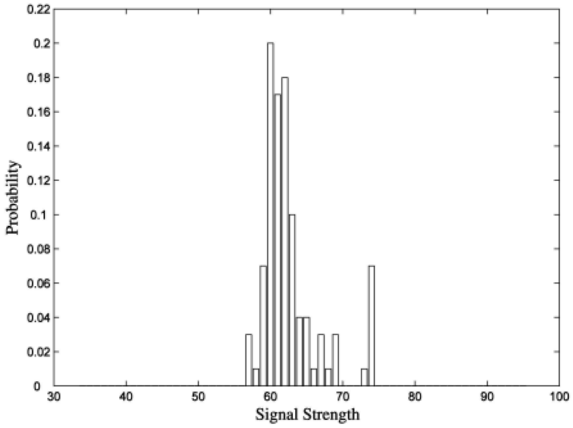


Fig. 2. An example of signal strength distribution.

There are a total of 25 access points that are detectable in the environment, of which three APs distributed within the area are marked with concrete circles in the figure. Among the other 22 APs, some are located on the same floor outside of the area while the others are located on different floors. Using the device driver and API we developed, the signals from these APs were recorded by an IBM laptop with a standard wireless Ethernet card. The laptop was carried by the user for data collection and online location estimation when operating within the office area.

3.2 Characteristics of Signal Propagation

The IEEE 802.11b standard works over the radio frequencies in the 2.4 GHz band. It is widespread since the band is license-free in most places around the world. It is attractive because the RF-based techniques are popular and inexpensive, providing much ubiquitous coverage and requiring little overhead.

However, accurate location estimation using measurements of signal strength is a longstanding difficult task due to the *noisy characteristics* of signal propagation. Subject to reflection, refraction, diffraction, and absorption by structures and even human bodies, signal propagation suffers from severe multipath fading effects in an indoor environment [24]. As a result, a transmitted signal can reach the receiver through different paths, each having its own amplitude and phase. These different components combine and reproduce a distorted version of the original signal. Moreover, even changes in the environmental conditions, such as temperature or humidity, also affect the signals to a large extent. As a consequence, the signal strength received from an access point at a fixed location varies with time and its physical surroundings.

Fig. 2 gives a typical example of the normalized histogram of the signal strength received from an access point at a fixed location. Several hundred measurements were sampled to construct the histogram. It is obvious that the signal strength received from the same AP varies with time, even at a fixed location. Furthermore, the number of APs covering a location also varies with time.

3.3 Rationale of Our Approach

A key novelty of our work is the ability to save energy for location-estimation systems through the application of data mining algorithms. We adopted three approaches. First, we adopted an architecture in which we only receive signals from various APs, rather than sending them. As we will see later, this mode of sensing saves much battery power. Second, we optimize the location estimation algorithm by both reducing the number of APs that must be sensed, thus reducing the amount of data we handle and saving the computational time. The third and the most effective method is to ensure the wake-up time of the client device is minimal.

Our architecture is a client-server-based system. In this system, the client device is held in the hand of the user, which is battery powered. The hand-held client system usually consumes much energy, which cuts down the lifetime of the location-estimation system, especially when we wish to make the whole operation as pervasive as possible. Papers [25] and Ebert et al. [26] examined the power consumption of various wireless local-area network (LAN) cards in the 802.11 range. They, in particular, measured the power consumption of the system in different modes, including *Sleep* Mode, in which the system hibernates, the *Idle Mode*, in which the system does nothing active in transmission, and the two transmission modes. In the transmission modes, the system can transmit packages (TX) or receive packages (RX). In the Sleep mode, the average power consumption is around 20MW. In the Idle mode, the power is 110MW; in the RX mode, the power is 900MW while in the TX mode, the power consumption is 2,500 MW. Other factors affecting the power consumption include the size of packets and the speed of transmission. Generally, when the packets are large and the transmission speed is fast, the system uses less energy; however, large packets also increase the chance of errors, which in turn increases the energy consumption.

Therefore, in a client-based system, it is possible to save the transmission energy consumption by adopting a packet-receiving mode. However, in a server-based system, packets must be sent to the servers in order to locate the client. Thus, in order for the client-based system to save energy, one way is to reduce the amount of signals transmitted between a client system and a server system. On one hand, in the location estimation systems that we surveyed, the RADAR system [8] and the commercially available Ekahau system [27] need to transmit signals to a server, which then makes the prediction on the client's location. These systems have the advantage that they require less offline training, but they require a large amount of battery power on the client devices. On the other hand, the client-based system that we propose in this paper is one that only receives packages sent by the APs (i.e., use the RX mode), and uses the signal strengths and an offline-obtained radio map to decide where it is currently located. Such a client-based architecture requires less energy. An additional advantage is that it is easier to protect the identity of the client bearer since the main computation is done on-site.

TABLE 1
Power Dissipation on HP Smartbadge IV [28]

Subsystem	Power (mW)	Percentage
CPU	694	21%
Memory	1115	34%
802.11b	1500	45%
Total	3309	100%

A second method to reduce the consumption of energy is to reduce the amount of computation that is done on the client system. In our approach, we first reduce the number of APs that are required to obtain signals from, because we use a decision-tree-based model after applying clustering. As we will later show, reducing the number of APs corresponds to dimensionality reduction in data mining, which reduces the number of multiplications that must be done on the system. In our experimental system, we will demonstrate the computational effort reduced by this method.

A third method is to ensure that the amount of time the system is in idle or sleep mode is maximized. Our approach ensures that this is the case by minimizing the number of samples that must be received in real time before the system can make a credible decision. The 802.11b interface operates at a maximum bit rate of 11 Mbps with a maximum range of 100 meters. Delaney [28] used a PCMCIA 802.11b interface card and measured the average current going into the interface to get the power dissipation. He used an on/off-scheduling algorithm to reduce the total energy consumption of the 802.11b device. While operating in the 802.11b power management mode, a WLAN card goes into an *idle* state. For every 100 ms, it *wakes up* and receives a traffic indication map, which is used to indicate when the base station will be transmitting data to this particular mobile host. When there is heavy broadcast traffic, which happens when the client device is conducting signal transmission, the WLAN interface will rarely be in the idle state and it will consume as much power as if it were in the always-on mode. This is because the time required to analyze the broadcast packets is larger than the sleep mode. This increase in power consumption will happen even if there are no applications running on the mobile host.

In the experimental section, we will show that our system outperforms others in terms of accuracy for a fraction of the samples that they use for location estimation; this shows that our system can have longer sleep time during the operation.

Table 1 (data quoted from [28]) shows the power measurements of an HP Smartbadge IV embedded system using 802.11b to transmit signals From the table, we can see that the wireless communication takes up almost half the energy, and the CPU activities take about 20 percent. This means that if we can apply a more intelligent algorithm that increases the sleep and idle time of the device and reduces the amount of computation, we can achieve our goal of saving energy.

4 CaDet Algorithm Description

Our CaDet algorithm for location estimation is divided into two phases:

- The first phase is done offline, where the main purpose is to perform intelligent AP selection. We divide this phase into the following steps:
  1. First, a feature selection algorithm is applied to find a subset S of APs that can give the best performance. This subset will then be used as the basis for subsequent computation.
  2. A subsequent clustering analysis is then applied to the set S and data collected in the offline phase, in order to partition the grid space into clusters. Each cluster will then provide a subsequent location model.
  3. Finally, a decision tree model is constructed for each cluster, based on the APs given in S. For each cluster only a subset of APs from S is selected, which further reduces the number of APs needed for location estimation within each cluster.
- The second phase is done online, in which a new trace of signal-strength values is taken as input and the current location is estimated. This phase is done in two steps:
  1. First, the signal strength values from the selected APs from the set S is used to determine that the cluster of the current client is most likely located within.
  2. Then, the decision tree from the identified cluster is used to determine, at a finer level, which grid the client belongs to. This step will use a subset of the APs given in S, which further reduces the number of APs used in a computation. In addition, the APs that are used only involve arithmetic comparison, which is one of the cheapest computations as computational energy is concerned.

4.1 Offline AP Selection in CaDet

4.1.1 AP Selection Using Information Theory

Normally, in an environment, signals from many APs are detectable here or there within the area of concern. For example, nine APs were detectable in the region of Duncan Hall at Rice University which was used as the test-bed for experiments [1]. Among them, five were located within the region while the others were located outside, including those on other floors. In many other environments, such as ours, many more APs can be detected. As shown in Section 3.1, there are a total of 25 APs detectable. Signals from each AP provide some information for location estimation, and it is a natural way to use as many APs as possible to improve the accuracy in a location estimation system. However, the increase of accuracy is at the cost of adding more computational burden to the system. As a consequence, such a location system not only has poor scalability but also is power-insufficient when energy is constrained on the computational unit. Therefore, it is important to only use the number of APs that a target system can afford while maintaining as high a level of accuracy as possible.

To find a trade-off point between the number of APs used and the accuracy they can achieve, we propose an Information Gain-based AP selection method (InfoGain for short) [29]. The idea of AP selection using InfoGain is as follows: Suppose in a grid-based location system, n is the number of grids and m is the total number of APs detectable. Each AP ($AP_i, 1 \leq i \leq m$) is viewed as a feature and each grid ($G_j, 1 \leq j \leq n$) is described by these m features. To a particular grid G^*, signal samples from the APs are collected offline and the average signal strength from AP_i is taken as the value of the ith feature of G^*. It is also possible that some APs may not be detected in G^* because of their physical locations and also the characteristics of signal propagation. In this case, the features of the corresponding missing APs take a default value, which is set to -95, the minimum strength of the signal received in the environment. The InfoGain criterion for AP selection is to evaluate the worth of each feature (i.e., AP) in terms of its discriminative power and select the highest ones. The discriminative power of feature AP_i is measured by the information gain when its value is known. Specifically, it is calculated as the reduction in entropy as follows:

$$InfoGain(AP_i) = H(G) - H(G|AP_i), \tag{1}$$

where $H(G) = -\sum_{j=1}^{n} Pr(G_j) \log Pr(G_j)$ is the entropy of the grids when AP_i's value is not known. Here, $Pr(G_j)$ is the prior probability of grid G_j, which can be uniformly distributed if a user can be equally likely in any grid. $H(G|AP_i) = -\sum_v \sum_{j=1}^{n} Pr(G_j, AP_i = v) \log Pr(G_j|AP_i = v)$ computes the conditional entropy of grids given AP_i's value. v is one possible value of signal strength from AP_i and the summation is taken over all possible values of AP_i.

For each AP_i, we compute the information gain using (1). The top k APs with the highest value are selected. Compared with the traditional selection method which selects the APs having the most strongest signals in the environment, our InfoGain method has the following advantages. InfoGain bases the selection of APs directly on their abilities to discriminate the grids by their signal values. As a consequence, the top k APs are the best at distinguishing one grid from another. On the other hand, the traditional method only considers the strength of the signals from APs and selects the strongest APs. Although in general, the APs having strong signals covering the region are preferable, they may not be the best to be selected, as we will see in Section 5.2.

4.1.2 Offline Location Clustering in CaDet

After k most discriminative APs are selected, the next step is to cluster the locations which are modeled as grids in the environment. Clustering is the unsupervised classification of patterns into groups [30]. The idea of location clustering is that locations where the received signals have similar characteristics form a cluster. Location clustering is important because the complexity of the location estimation algorithms can be greatly reduced by first identifying a cluster to which an unknown sample belongs and then

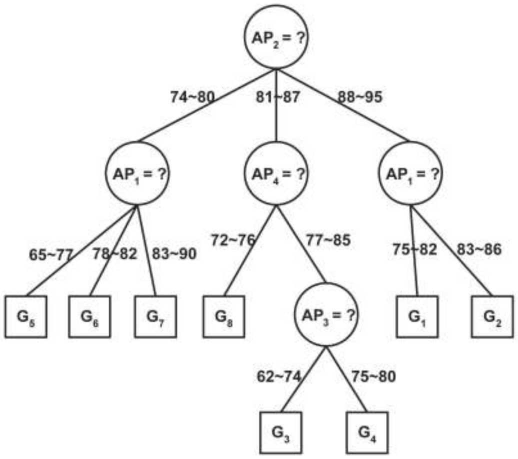


Fig. 3. A decision tree for location determination in CaDet.

determining the exact location of a user within the cluster. A similar claim was made in [10] where a joint clustering technique was proposed.

We define a location cluster as a set of grids that receive similar signals from k selected APs. Each grid can be represented by a vector of k signal strength, where the $i\text{th}$ element is the average signal strength received at this grid. Intuitively, the signals received in grids within a cluster are more similar to each other than they are to the signals of a grid belonging to a different cluster. The similarity of two grids can be measured in terms of the Euclidean distance between their signal strength vectors. Compared with the notion of joint cluster in [10], where a cluster is defined as a set of locations sharing a common set of APs, our definition not only considers the aspects of different coverage of APs over the grids as in [10], but also reflects the difference in the values of signal strength.

In this paper, we adopt the *K-means* clustering algorithm [30]. K-means is a widely used clustering algorithm that iteratively forms clusters. By specifying k, the number of clusters desired, the algorithm begins by arbitrarily selecting k grids as k cluster centroids. In each iteration, each grid is assigned to the nearest cluster centroid by measuring the distance between its signal strength and that of the centroids. After all the grids are assigned, the centroid of each cluster is recalculated by taking the average of the signal strength of the grids belonging to it. This iteration process continues until convergence is achieved, where the k centroids no longer shift. A grid is finally associated with one of the k cluster whose centroid is the closest to that grid. Therefore, after all grids are assigned, they are grouped into clusters whose number is significantly less than that of the grids. As we will see in the experiment (Section 5.3), the use of K-means algorithm is justified by the clustering results that grids geometrically close to each other are clustered together. However, the number of clusters k remains an open issue. If we specify k to be too large, there are too many clusters. As a result, there is much redundant computation and the number of APs is not reduced. However, if we specify too few APs, then we cannot take advantage of localized patterns from locations. In the following, we will empirically vary the number clusters k in order to find the best balance.

TABLE 2
A Test Sample T

Access Point	AP_1	AP_2	AP_3	AP_4
Signal Strength	91	84	65	78

4.1.3 Intracluster Estimation with Decision Trees

After a cluster is identified, CaDet determines a user's location at a coarse level (low resolution). In the next stage, grids in the same cluster need to be distinguished so as to make location estimation at a finer level, leading to a high resolution while at the same time, reducing the number of APs used.

CaDet uses a simple but effective approach based on decision trees [31]. Decision trees have been used in a wide range of pattern classification applications. The idea behind a decision tree is natural and intuitive: classify a test sample through a sequence of questions. As an example, a decision tree built over grids ($G_1 \sim G_8$) of a location cluster is shown in Fig. 3. In the figure, each internal node corresponds to a test question on the value of signal strength of a particular AP. Several subtrees are branched out from a internal node, each corresponding to a different range of values. Starting from the root node, the test sample will be asked a sequence of questions until it reaches a leaf node. A leaf node at the lowest level represents the decision on which grid the sample belongs to. More detail will be given through an example in Section 4.2.

The advantages of using decision trees to determine the location within a cluster are as follows: First, decision trees are efficient and the decisions made at each step when walking through them are intuitive and easily understood. The significance of decision trees also lies in that they are of low computational complexity and therefore power-efficient since only comparison of signal strength is needed.

To build a decision tree over the grids in each cluster, we again use the AP selection method introduced in Section 4.1.1. Since different APs have different coverage and also have different discrimination power in each cluster, the selection process will further reduce the computational burden. After a set of APs are selected for each cluster, decision trees can be built using C4.5 [31]. Decision trees similar to the one shown in Fig. 3 are obtained.

4.2 Online Application of the Models in CaDet

Once the models are built, we can then apply them to online location estimation. For a given received signal sample, the signal-strength values from the selected APs are used to determine the cluster that the current client is most likely located in. Subsequently, the decision tree associated with the identified cluster is used to determine, at a finer level, which grid the client belongs to. Ties are broken arbitrarily. We now illustrate this process using an example.

As an example, suppose that a vector of four APs ($AP_1 \sim AP_4$) is selected using the InfoGain algorithm. The signal strength of a test sample T is listed in Table 2. Let C_j denote the centroid of the $j\text{th}$ clusters among the total M clusters given by the offline-clustering algorithm. To determine which cluster the test sample belongs to, the distance from the test sample T to each cluster centroid is calculated. The distance $D(T, C_j)$ from T to C_j is given by:

$$D(T, C_j) = \sum_{i=1}^{4} (SS_i(T) - SS_i(C_j))^2,$$

where $SS_i(\cdot)$ is the value of signal strength from the AP_i. The cluster C^* whose centroid is the closest to the test sample ($C^* = \arg\max_j D(T, C_j)$) is identified and associated to the test sample. Suppose the cluster centroid C_2 is the nearest and the decision tree for cluster 2 is as shown in Fig. 3. Applying this decision tree, the classification of the sample begins at the root node, which asks for the value of signal strength from AP_2. The different branches from the root node correspond to a different range of values. Since AP_2's value of the sample is 84, classification follows the middle branch. The next step is to check the value for AP_4 at the subsequent node, which turns out to be 78. The same process continues until a leaf node is reached. The leaf nodes bear labels of each grid to which samples are assigned grids to. In the example, the leaf node G_3 is reached and the sample T is determined to be from grid G_3.

4.3 Analysis on Energy Savings in CaDet

As stated in the related work section, one of the few location estimation works that considered the power saving issue is that of [10], which adopted a probabilistic framework of maximal likelihood. In this approach, a set of probabilistic distributions are modeled in every grid, with one distribution corresponding to one AP. In all, $|G| \times |AP|$ distributions need to be built, where $|G|$ is the number of grids and $|AP|$ is the number of APs. The grid with the maximum likelihood of observing the signal strength of the test sample is determined as where a user is. Although high accuracy can be achieved (83 percent within 1.5 meters [1] and over 90 percent within 2 meters [10]), the probabilistic approaches suffer from high computational cost, which leads to high power consumption. To calculate the likelihood of the user in one grid, $O(|AP|)$ times float-point multiplications are taken and in sum $O(|G| \times |AP|)$, multiplications are needed. It is computationally costly.

In contrast, in CaDet, we only require a small fraction of all the available APs in location estimation computation. Furthermore, the decision trees within each cluster require an even smaller number of APs as compared to the whole set. Thus, by reducing the number of APs that are involved in the computation, we reduce the power consumption as well. In our case, the number of APs used will never be higher than that of the Joint Clustering approach, and is often much lower. The exact number of APs that are selected and used through decision tree construction is an empirical question, which we will answer in the next section.

5 Experimental Results

In this section, we discuss the experimental test-bed and evaluate the performance of CaDet, and compare it with others. First, the effectiveness of offline AP selection and clustering is shown (Sections 5.2 and 5.3). Then, in the online phase of location estimation, a comparison is made with previous work in terms of accuracy and computational cost (Sections 5.4 and 5.5).

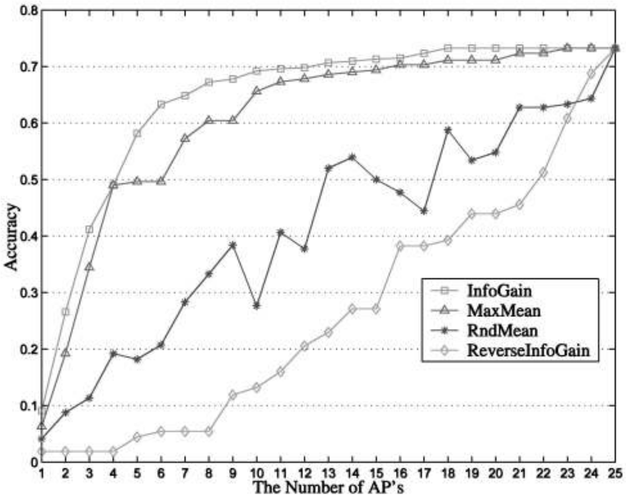


Fig. 4. Testing the effect of AP selection criteria on accuracy in CaDet.

5.1 Experimental Test-Bed

We performed our experiment in the office area of the Computer Science Department in the Academic Building of the Hong Kong University of Science and Technology (HKUST) as shown in Fig. 1. The environment is modeled as a space of 99 locations, each representing a 1.5-meter grid cell. There are a totsl of 25 access points that can be detected, three of which are distributed within this area and the others are from adjacent areas on the same floor or different floors. We label these access points from 1 to 25 in our experiment. On average, each location is covered by five access points.

Using the device driver and the API we developed, we collected 100 samples at each location, one per second, and we used them to test the performance of our proposed method. For the validity of experimental results, we ran the experiments based on 10-fold cross validation. It partitions the whole data set into 10 independent folds, each time it uses nine folds for training and the other one fold for testing, and finally reports the average result.

5.2 Offline AP Selection

In this section, we study the effect of different AP selection criteria on the accuracy of location estimation. Fig. 4 compares the accuracy within 1.5 meters using the ML method over all the locations with respect to four AP selection criteria: *InfoGain*, *MaxMean*, *RndMean*, and *ReverseInfoGain*. The *InfoGain* criterion ranks APs in descending order of their InfoGain values using our algorithm described in Section 4.1.1. The *MaxMean* criterion ranks APs in descending order of their average signal-strength values, which has been used to select APs in [10]. For the purpose of analysis, we also apply the other two criteria. The *RndMean* criterion randomly selects a few APs regardless of their signal-strength values. The *ReverseInfoGain* criterion selects APs in reverse order of the way the *InfoGain* criterion does.

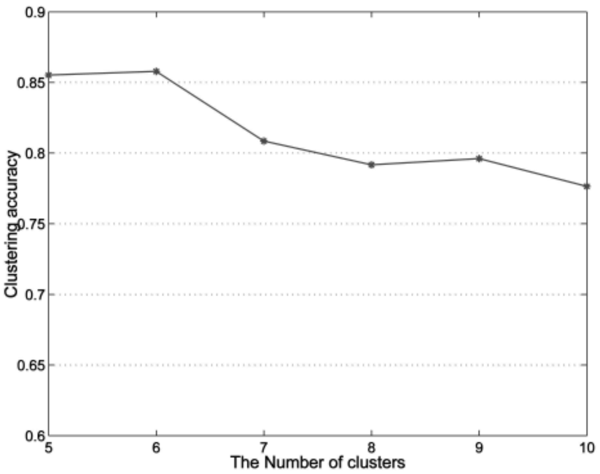


Fig. 5. Clustering accuracy versus the number of clusters in `CaDet`.

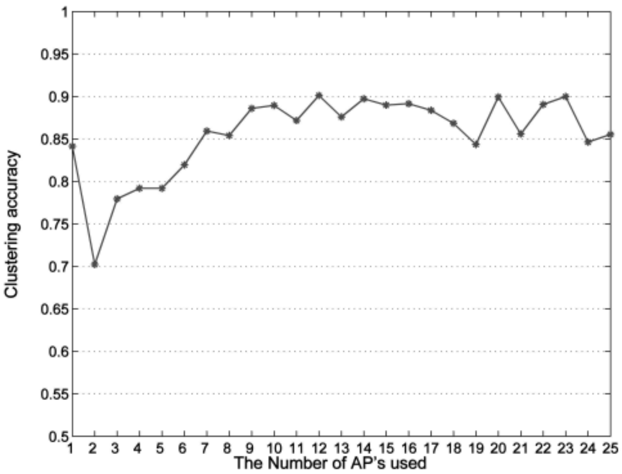


Fig. 6. Clustering accuracy versus the number of APs in `CaDet`.

We can see from Fig. 4 that, as the number of APs increases, the accuracy of using the *InfoGain* criterion increases faster than the other three criteria. In other words, in order to achieve the best accuracy using the same ML-based method, the *InfoGain* criterion uses the smallest subset of APs while the *RndMean* and *ReverseInfoGain* criteria need to use all 25 APs. This is because the subset of APs with the same size selected by the *InfoGain* criterion does have the best discriminative abilities towards different locations. On the contrary, the *ReverseInfoGain* criterion performs worst since it reverses the discriminative abilities of APs. This shows that the *InfoGain* criterion has the advantage of using the fewest APs to achieve the same accuracy, which in turn reduces the computational cost required by each location estimation.

Let us further compare the performance of the *InfoGain* criterion with that of the *MaxMean* criterion. The first four APs selected by both of the criteria are consistent except for their relative order, so as the number of APs increases from one to four, the accuracy of the two criteria remains approximately the same. The main difference of the two criteria lies in the selection of the other 21 APs. The *InfoGain* criterion selects the APs in descending order of their discriminative abilities toward different locations. However, the *MaxMean* criterion just considers the average signal-strength values received in different locations, instead of their inherent discriminative abilities. Therefore, if an access point can distinguish some parts of locations accurately, the *InfoGain* criterion will rank it near the front of all the APs while the *MaxMean* criterion may possibly rank it near the end. This is because if the signal-strength values of this AP are small in the other locations, its average value may be smaller than another AP which has large signal-strength values over all the locations. Moreover, in the extreme case, if the signal-strength values of an AP are uniformly large but differ a little over all the locations, the *MaxMean* will rank this AP near the front by priority although it does not contribute to distinguishing the locations. On the contrary, the *InfoGain* criterion does rank this AP near the end since its discriminative ability toward different location is inherently low. We can see from the figure that the accuracy of the ML method using *InfoGain* increases faster than that of using *MaxMean* especially when the number of APs ranges from 4 to 12. This is because the *InfoGain* criterion can select a subset of APs with higher discriminative abilities than the *MaxMean* criterion. As a consequence, this nice property guarantees that the *InfoGain* criterion can achieve the same accuracy as the *MaxMean* but use fewer APs.

5.3 Offline Clustering Result

In this section, we study the performance of our location clustering method in `CaDet` as discussed in Section 4.1.2. In particular, we discuss the effect of the number of clusters and the number of APs used in clustering on its performance.

Fig. 5 shows the effect of the number of clusters on performance. This experiment is taken over all the 25 APs to see the change of clustering accuracy with respect to the number of clusters. In order to reduce the computational cost of online location estimation, we expect to obtain several clusters while ensuring the high accuracy of locating samples to clusters. For this purpose, we define the clustering accuracy as the number of signal-strength samples which are located to the correct cluster divided by the total number of samples. It can be seen from the figure that the best accuracy can be achieved when the number of clusters is equal to six.

Fig. 6 shows the effect of the number of APs used in clustering on performance. In order to reduce the computational cost used for clustering, we expect to choose as few APs as possible to achieve the best accuracy. It can be seen from the figure that when the number of APs is equal to 12, the best accuracy of 90 percent can be achieved.

For the rest of the paper, we choose the number of clusters to be 6 and the number of APs used in clustering to be 12 since this setting leads to the best performance of our location clustering method in our experiment. As shown in Fig. 7, six clusters are labeled with different colors, respectively. Moreover, the center of each cluster is represented as a black square. It can be seen from the figure that the locations contained within each cluster is physically adjacent. This is because physically adjacent locations may receive more similar signal values from the

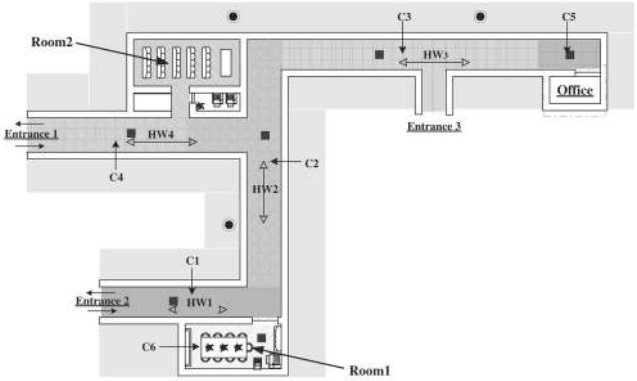


Fig. 7. The clustering result in CaDet.

same set of APs. In our experiment, when a signal-strength measurement is made, the clustering method can move it to the correct cluster with the accuracy of 90 percent. This shows that the most difficulty lies in distinguishing the locations within the same cluster.

5.4 Online Comparison of Accuracy

In this section, we compare the performance of the decision tree method in CaDet with that of the ML method [10] implemented in the same test-bed.

Fig. 8 compares the accuracy within 1.5 meters using three approaches, the decision tree method and two ML methods, in six clusters, respectively. Here, ML1 is the ML method using a different number of APs while ML2 is the ML method using all of the APs in each cluster. In this experiment, we rank APs in descending order of their InfoGain values. Note that we only consider the relationship between accuracy and different number of APs with nonzero InfoGain values because any access point with zero InfoGain values does not contribute to the increase of accuracy. For example, in Fig. 8b, we only consider the first nine APs in hallway2. In each cluster, we consider a different number of APs depending on their computed InfoGain values.

We can see from the figure that the decision tree method outperforms the two ML methods in every cluster. On the average, the decision tree method can achieve the best accuracy of 83.4 percent within 1.5 meters, while the best accuracy of ML is 74.9 percent. For a location estimation, as more APs are used, the accuracy of ML1 becomes closer to that of ML2 that uses all the APs. Compared with ML2, the decision tree method can achieve the same accuracy but use fewer APs. As more APs are used later, decision tree outperforms ML2. This shows that by selecting APs based on the *InfoGain* criterion, the decision tree can maintain the same accuracy as the ML method without access-points selection while reducing the number of APs. In addition, the accuracy of decision tree is even higher than that of ML1 using the same number of APs.

Let us study the performance of decision tree in detail. It can be seen from the figure that as the number of APs increases in each cluster, the accuracy of decision tree increases approximately monotonically because we have more information due to the addition of APs. For the same reason, the computational cost required by each location estimation increases. However, as more APs are involved, the accuracy of decision tree increases more slowly and converges to a certain value. This is because the decision tree algorithm inherently exploits the *InfoGain* criterion to select the APs in descending order of their discriminative abilities toward locations. Therefore, if an access point can distinguish different locations within a cluster more accurately, it will be chosen first by a decision tree to reduce the search space of locations. On the contrary, if the discriminative ability of an AP is weak, it will be chosen later, or never be chosen in the case that it does not provide any information to distinguish locations. Now, we select Fig. 8a for illustration. As the first five APs are involved one by one, the accuracy of decision tree increases. However, the accuracy remains almost the same even if the other seven APs are added later. It shows that, after the clustering step, only a subset of APs further contribute to distinguishing different locations within each cluster. Therefore, by further reducing the number of APs, the computational cost can be reduced while the accuracy is maintained at the same level.

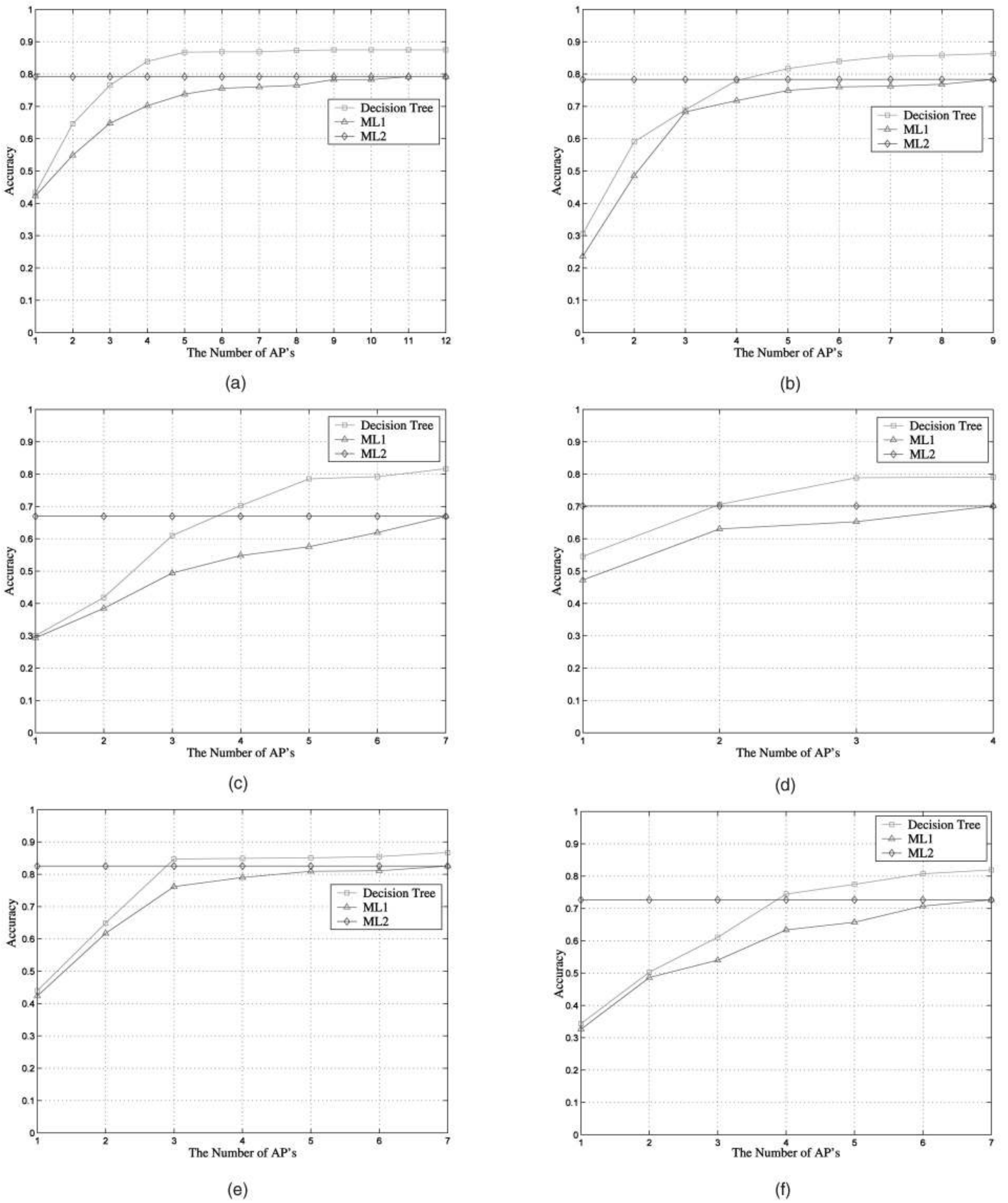
Table 3 shows the selected APs within each cluster. It can be seen from the table that the optimal number of APs in each cluster is four or five on average. Based on these selected APs, the decision tree can achieve the accuracy of about 83 percent within 1.5 meters.

5.5 Comparison of Online Computational Cost

In this section, we compare the computational cost required by three techniques: our proposed approach, the Joint Clustering (JC) technique, and RADAR. The computational cost is measured by the average number of operations (multiplications) performed for a single location estimation.

Fig. 9 shows the expected computational cost over all of the locations using three techniques with respect to different numbers of APs. After clustering and AP selection, the resulting clusters are of different sizes (consisting of a different number of grids) and each cluster has its own set of APs selected. As a result, the number of operations for estimating a sample is not the same, depending on which cluster is associated in the first step. Therefore, the computational cost is calculated by taking the expectation over all possible grids in the environment. That is, the number of operations for locating each grid is first accumulated and then divided by the total number of grids, which gives the expected computational cost.

We can see from the figure that the computational cost required by three techniques increases as the number of APs increases. The figures also show that, compared with RADAR, our approach and the Joint Clustering method reduce the computational cost by using clustering. However, for each location estimation, our approach requires a lower computational cost than Joint Clustering. This is because after locating a signal-strength measurement to a location cluster, our approach uses the decision tree algorithm to determine the most likely location within the cluster, which just requires the comparison operations when walking along the tree. However, Joint Clustering uses the maximum likelihood method to determine the location within a cluster, which requires the multiplication of conditional probabilities in proportion to the number of APs and the locations within each cluster.



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 8. The Online comparison on the accuracy of the decision tree method and two ML methods in six clusters in `CaDet`. (a) Hallway1. (b) Hallway2. (c) Hallway3. (d) Hallway4. (e) Hallway5. (f) Room1.

5.6 Reducing the Online Sampling Time in `CaDet`

In Section 3.3, we considered a third method to ensure that energy is saved by increasing the Idle or sleep time of the system. If we can obtain the same accuracy in a short amount of time, then we can allow the system to wake up once every several seconds for location estimation, thus saving a great amount of energy. In this section, we demonstrate that our system can indeed give the same amount of accuracy while using a smaller amount of wake-up time.

The experimental comparisons are shown in Fig. 10 and Fig. 11. In this experiment, we use only three APs in each hallway to do location estimation, which is more realistic, because in a typical environment, we can only expect a small number of APs to be available to cover each area. We

TABLE 3
The AP Selection within Each Cluster in CaDet

Cluster	AP #	Selected APs	Accuracy
C1	5	5,10,17,4,19	86.4%
C2	6	21,7,6,10,2,3	84.6%
C3	5	1,7,3,2,6	78.5%
C4	3	1,2,3	78.5%
C5	3	14,12,8	81.0%
C6	5	10,16,11,5,3	84.6%

compare three methods, joint cluster (JC), RADAR, and our method. The first phase of CaDet is cluster selection, which is the same as joint clustering, we only show the comparison result of the second phase in these figures. As we can see from the figures, using our system, we can indeed save from 3 to 4 seconds for each estimated location while maintaining the same level of location-estimation accuracy. That means that we can leave the client system off for 3 to 4 seconds for each location, thus saving battery power when operating online.

6 Conclusions and Future Work

In this paper, we have presented a new approach to increase the location-estimation accuracy in an indoor wireless environment while reducing the power consumption. Our method intelligently selects the number of APs used for location estimation by employing machine learning techniques. Through information theory, clustering, and decision tree algorithms, we are able to use a small subset of the APs in the environment to detect a client's location with high accuracy. An important consequence is the ability to use only a small fraction of the computational power as compared with previous techniques. In the future, we plan to experiment with different clustering and feature selection techniques for access point selection. In addition, we wish to use similar techniques to schedule optimal layout maps for AP distribution in an indoor environment.

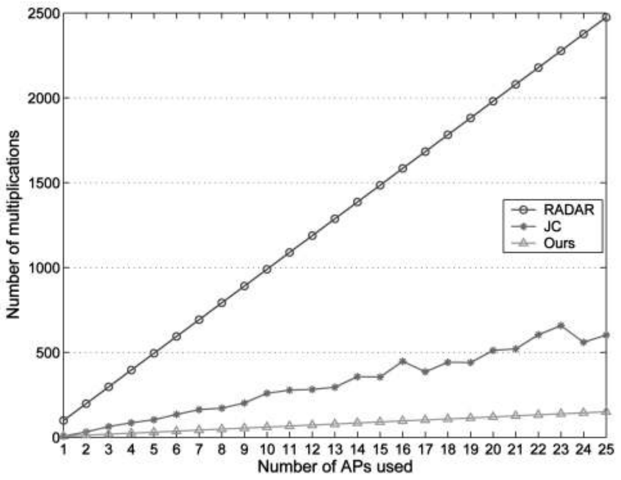


Fig. 9. The comparison of online computational cost using three techniques.

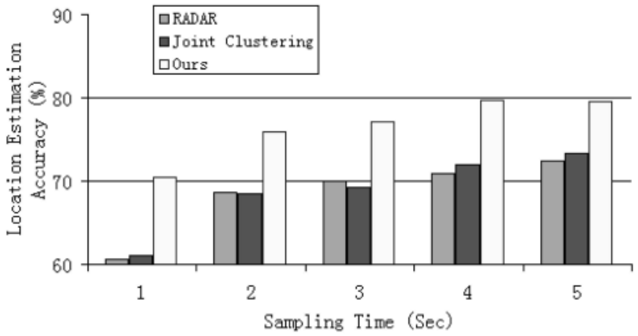


Fig. 10. Location-estimation accuracy versus sampling time in hallway 4 by different systems, using 3 Aps.

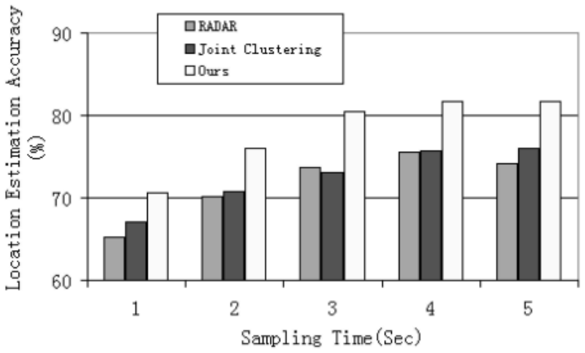


Fig. 11. Location estimation accuracy versus sampling time in hallway 5 by different systems, using 3 Aps.

Acknowledgments

Yiqiang Chen is supported by the Natural Science Foundation of China (No. 60303018) and Beijing New Science and Technology Star Plan (2005B54). Qiang Yang, Jie Yin, and Xiaoyong Chai are supported by a grant from Hong Kong RGC (RGC Project HKUST 6187/04E). The authors thank Jeff Junfeng Pan, Dr. Lionel Ni, and Mr. Min Gao for their valuable comments.

References

[1] A. Ladd, K. Bekris, G. Marceau, A. Rudys, L. Kavraki, and D. Wallach, "Robotics-Based Location Sensing Using Wireless Ethernet," *Proc. MOBICOM2002 Conf.*, pp. 227-238, Sept. 2002.

[2] C. Gentile and L.K. Berndt, "Robust Location Using System Dynamics and Motion Constraints," *Proc. IEEE Conf. Comm.*, vol. 3, pp. 1360-1364, June 2004.

[3] M. Youssef and A. Agrawala, "Handling Samples Correlation in the Horus System," *Proc. IEEE InfoCom 2003 Conf.*, vol. 2, pp. 1023-1031, Mar. 2004.

[4] P. Bahl, A. Balachandran, and V. Padmanabhan, "Enhancements to the RADAR User Location and Tracking System," technical report, Microsoft Research, Feb. 2000.

[5] L.M. Ni, Y. Liu, Y.C. Lau, and A.P. Patil, "Landmarc: Indoor Location Sensing Using Active RFID," *Proc. IEEE Int'l Conf. Pervasive Computing and Comm. 2003*, pp. 407-415, Mar. 2003.

[6] D. Fox, J. Hightower, L. Liao, and D. Schulz, "Bayesian Filtering for Location Estimation," *IEEE Pervasive Computing*, vol. 2, no. 3, pp. 24-33, 2002.

[7] J. Yin, X.Y. Chai, and Q. Yang, "High-Level Goal Recognition in a Wireless Lan," *Proc. 19th Nat'l Conf. Artificial Intelligence (AAAI '04)*, pp. 578-584, July 2004.

[8] P. Bahl and V.N. Padmanabhan, "RADAR: An In-Building RF-Based User Location and Tracking System," *Proc. IEEE INFOCOM2000 Conf.*, pp. 775-784, 2000.

[9] E.S. Bhasker, S.W. Brown, and W.G. Griswold, "Employing User Feedback for Fast, Accurate, Low-Maintenance Geolocationing," *Proc. IEEE Int'l Conf. Pervasive Computing and Comm. 2004 (PerCom '04),* pp. 111-120, Mar. 2004.

[10] M. Youssef, A. Agrawala, and U. Shankar, "WLAN Location Determination via Clustering and Probability Distributions," *Proc. IEEE Pervasive Computing,* pp. 143-152, Mar. 2003.

[11] T. Roos, P. Myllymaki, H. Tirri, P. Misikangas, and J. Sievanen, "A Probabilistic Approach to WLAN User Location Estimation," *Int'l J. Wireless Information Networks,* vol. 9, no. 3, pp. 155-164, July 2002.

[12] M. Youssef and A. Agrawala, "On the Optimality of WLAN Location Determination Systems," *Proc. Comm. Networks and Distributed Systems Modeling and Simulation Conf.,* Jan. 2004.

[13] R. Kravets and R. Krishnan, "Power Management Techniques for Mobile Communication," *Proc. Fourth Ann. ACM/IEEE Int'l Conf. Mobile Computing and Networking (MOBICOM'98),* pp. 157-168, Oct. 1998.

[14] M. Stemm and R.H. Katz, "Measuring and Reducing Energy Consumption of Network Interfaces in Handheld Devices," *IEICE Trans. Fundamentals of Electronics, Comm., and Computer Science,* vol. 80, no. 8, pp. 1125-1131, Aug. 1997.

[15] I. Hong and M. Potkonjak, "Power Optimization in Disk-Based Real-Time Application Specific Systems," *Proc. 1996 IEEE/ACM Int'l Conf. Computer-Aided Design,* pp. 10-14, Nov. 1996.

[16] S. Gurumurthi, A. Sivasubramaniam, M. Kandemir, and H. Franke, "DRPM: Dynamic Speed Control for Power Management in Server Class Disks," *ACM SIGARCH Computer Architecture News,* vol. 31, no. 2, May 2003.

[17] M. Weiser, B. Welch, A. Demers, and S. Shenker, "Scheduling for Reduced CPU Energy," *Proc. First USENIX Symp. Operating System Design and Implementation (OSDI),* pp. 13-23, Nov. 1994.

[18] J.R. Lorch and A.J. Smith, "Scheduling Techniques for Reducing Processor Energy Use in Macos," *Wireless Networks,* vol. 3, no. 5, pp. 311-324, 1997.

[19] W.-C. Lee and D.L. Lee, "Using Signature Techniques for Information Filtering in Wireless and Mobile Environments," *J. Distributed and Parallel Databases,* vol. 4, no. 3, pp. 205-227, July 1996.

[20] N. Shivakumar and S. Venkatasubramanian, "Efficient Indexing for Broadcast Based Wireless Systems," *ACM/Baltzer Mobile Networks and Applications (MONET),* vol. 1, no. 4, pp. 433-446, Dec. 1996.

[21] Y. Xu and W.-C. Lee, "On Localized Prediction for Power Efficient Object Tracking in Sensor Networks," *Proc. First Int'l Workshop Mobile Distributed Computing (MDC),* pp. 434-439, May 2003.

[22] W.R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," *Proc. Hawaii Int'l Conf. System Sciences (HICSS),* pp. 1-10, Jan. 2000.

[23] Y.Q. Xu, J. Winter, and W.-C. Lee, "Prediction-Based Strategies for Energy Saving in Object Tracking Sensor Networks," *Proc. IEEE Int'l Conf. Mobile Data Management (MDM '04),* pp. 346-357, Jan. 2004.

[24] H. Hashemi, "The Indoor Radio Propagation Channel," vol. 81, no. 7, pp. 943-968, 1993.

[25] A.W. Papers, "Power Consumption and Energy Efficiency Comparisons of WLAN Products," technical report, Atheros Inc., 2003.

[26] J.-P. Ebert, B. Burns, and A. Wolisz, "A Trace-Based Approach for Determining the Energy Consumption of a WLAN Network Interface," *Proc. European Wireless 2002 Conf.,* pp. 230-236, Feb. 2002.

[27] T.E. system, http://www.ekahau.com, 2006.

[28] B. Delaney, "Reduced Energy Consumption and Improved Accuracy for Distributed Speech Recognition in Wireless Environments," PhD dissertation, Georgia Inst. of Technology, 2004.

[29] T. Mitchell, *Machine Learning.* McGraw-Hill, 1997.

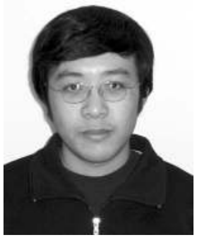
[30] R. Duda, P. Hart, and D. Stork, *Pattern Classification.* Wiley, 2001.

[31] J.R. Quinlan, *C4.5: Programs for Machine Learning.* Morgan Kaufmann Publishers, 1993.

Yiqiang Chen received the BASc and MA degrees from the University of Xiangtan in 1996 and 1999, respectively, and the PhD degree from the Institute of Computing Technology, Chinese Academy of Sciences in 2002. In 2004, he was a visiting scholar researcher in the Department of Computer Science at the Hong Kong University of Science and Technology (HKUST). He is now an associate professor of ICT and Vice Director-General of the Shanghai Branch of the Institute of Computing (ICT), The Chinese Academy of Sciences (CAS). His research interests include data mining and mobile multimedia.

Qiang Yang received the PhD degree from the University of Maryland, College Park. He is a faculty member in the Hong Kong University of Science and Technology's Department of Computer Science. His research interests are AI planning, machine learning, case-based reasoning, and data mining. He is a senior member of the IEEE and an associate editor of the *IEEE Transactions on Knowledge and Data Engineering* and *IEEE Intelligent Systems*.

Jie Yin received the BE degree in computer science from the Xi'an Jiaotong University in 2001. Since the the fall of 2001, she has been a PhD student in the Department of Computer Science at the Hong Kong of Science and Technology. Her research interests include artificial intelligence, data mining, and pervasive computing.



Xiaoyong Chai received the MPhil degree in computer science from the Hong Kong University of Science and Technology in 2005. Currently, he is a PhD student in the Department of Computer Sciences at Purdue University. His research interests include artificial intelligence and data mining.