

On Clustering RSS *Fingerprints* for Improving Scalability of Performance Prediction of Indoor Positioning Systems



Nattapong Swangmuang
Graduate Program in Telecom/Networking
University of Pittsburgh
nswang@mail.sis.pitt.edu

Prashant Krishnamurthy
Graduate Program in Telecom/Networking
University of Pittsburgh
prashant@mail.sis.pitt.edu

ABSTRACT

We previously developed an analytical model in [8] to predict the precision and accuracy performance of indoor positioning systems using location fingerprints. A by-product of the model is the ability to **eliminate unnecessary fingerprints to reduce the number of fingerprints in a radio map for comparison without loss in performance.** This model enables computation of an **approximate probability distribution of location selection,** by employing a **proximity graph to extract neighbor and non-neighbor sets** of a given fingerprint. However, employing the model ‘as is’ in a system with many location fingerprints requires determining a single large proximity graph derived from all location fingerprints which may involve significant computational effort. In this paper, we consider two techniques to **divide location fingerprints into smaller clusters.** **Separate proximity graphs for each cluster are now used to predict performance and eliminate unnecessary location fingerprints.** Results show that the computational effort can be reduced, creating a more scalable analytical model, while still predicting and enabling good precision performance.

Categories and Subject Descriptors

C.2.1 [Computer System Organization]: Network Architecture and Design—*Wireless Communication*; G.3 [Mathematics of Computing]: Probability and Statistics—*Statistical Computing*; I.5.3 [Pattern Recognition]: Clustering—*Algorithms*

General Terms

Algorithms, Performance

Keywords

Indoor Positioning, Location Fingerprinting, WLANs, Clustering Methods

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MELT’08, September 19, 2008, San Francisco, California, USA.
Copyright 2008 ACM 978-1-60558-189-7/08/09 ...\$5.00.

1. INTRODUCTION

The popularity of wireless local area networks (WLANs) inside and around buildings makes positioning systems based on readily available received signal strength (RSS) from access points (APs) desirable. *Location Fingerprinting* is a pattern matching technique that exploits the unique characteristics of the RSS from different APs measured at physical locations for location estimation. This technique can be simply deployed based only on an existing WLAN infrastructure [1], as it requires neither extra spectrum for positioning nor a line-of-sight (LOS) to an AP. Obstacles inside buildings lead to multipath with time and angular dispersion that significantly bias time or angular measurements. These biases directly degrade performance of other positioning techniques (i.e., time-of-arrival or angle-of-arrival), but has little impact and sometimes even aids location fingerprinting.

Fingerprinting based positioning can be divided into the *offline* and *online* phases. In the offline phase, *location fingerprints* (or *fingerprints*) are collected by site-surveys of RSS from multiple APs at different locations. The database of RSS data is often called a *radio map*. Each location fingerprint is typically represented as a **vector consisting of the means of RSS values from N visible APs at a particular location,** denoted by $\vec{R} = [\rho_1, \rho_2, \rho_3, \dots, \rho_N]$. Here ρ_i (in dBm) represents the mean of RSS samples from the i^{th} AP at the specific location. The *grid spacing* (in m or ft) is the distance between two closest physical locations during the site-survey. Note that, **without a good survey plan and a clear performance objective, the offline phase can be very laborious and time-consuming, especially for large indoor areas with many locations.**

A mobile station’s (MS) location is determined during the online phase. First, a MS will measure a *sample fingerprint*, as denoted by $R = [r_1, r_2, r_3, \dots, r_N]$. Here, r_i is a sample RSS value from the i^{th} AP at the MS’s location. This sample fingerprint is compared to location fingerprints stored in the radio map for estimating the location. Fingerprint comparisons can be performed at a MS or a location server in the WLAN infrastructure. While many methods can be used [3, 4], the most common method to estimate the location is to compute the Euclidean distance between the sample fingerprint vector and each location fingerprint in the radio map. The co-ordinates of a location (i.e., a grid point) associated with the fingerprint having the smallest Euclidean distance to R is reported as the estimate of the location.

It is important to have an analytical model for predicting *accuracy* and *precision* of a system before actual deployment. Here, ‘accuracy’ reports the average *error distance*

between the estimated location and the actual MS location, whereas ‘precision’ reports the probability of successful (or unsuccessful) location estimates with a given accuracy¹. We developed the first analytical model for this purpose in [8]. The model employs proximity structure between fingerprints and proximity graphs (Gabriel graph) to extract information about neighbor and non-neighbor sets (details are presented later). The model allows computation of an approximate probability distribution of error distance given a RSS location fingerprint database and its associated statistics (results were validated in [8]). It is also desirable to build a radio map such that it contains only necessary location fingerprints without sacrificing performance. Our model enables identifying and eliminating unnecessary location fingerprints stored in the database, thereby saving on computation while performing location estimation in the online phase.

The computational requirement for constructing the proximity graph can be high when location fingerprints from a large number of grid points are collected in an actual deployment. From our previous study, we see that fingerprints in RSS space are typically non-uniformly distributed even though they are collected on a uniform grid of physical locations. Visually, fingerprints may be even clustered together [8]. Given a sample RSS vector, the probability of selecting a fingerprint not within the same cluster is quite low. From this observation, one plausible way to scale the proximity graph construction is to apply it only to clusters of fingerprints. Fingerprint clustering reduces the effort for computation by substituting multiple smaller proximity graphs, one per cluster, in place of a single large proximity graph (the computational complexity is discussed later in Table 1). The analytical model for determining approximate probabilities of selecting a location given an measured RSS vector can then be applied using proximity graphs of smaller sizes. Dividing fingerprints into small clusters can also reduce number of comparison pairs (between a sample fingerprint and location fingerprints in the radio map) performed during the online phase by considering only those fingerprints in a particular cluster.

However, questions that arise are (a) whether using proximity graphs based on smaller clusters impacts the prediction of precision performance or the actual performance in the online phase and (b) whether there are tradeoffs between precision and computational complexity. Ideally, we want fingerprint clusters to be evenly distributed (i.e., the number of fingerprints or the online comparison pairs are the same for each cluster), yet they should reflect actual signal distance relationships (do they actually form a cluster?). Also, the clustering method should be simple and not create additional computational burden. While we only consider two clusters in this paper, this approach can be extended to multiple clusters. At the two extremes, we have a single cluster or each fingerprint belongs to its own cluster. In this work, we have not evaluated when an increasing number of clusters begins to impact the precision performance or computational complexity negatively.

The remainder of this paper is organized as follows. An overview of the analytical model is presented in Section 2. Section 3 describes two clustering methods used to separate fingerprints into clusters. Performance evaluation of the analytical model with clustering and computational compar-

isons for fingerprint clustering are presented in Section 4. Finally, discussion and conclusion of the paper are presented in Section 5.

2. OVERVIEW OF ANALYTICAL MODEL

Consider an indoor positioning system overlaid on a WLAN. There are N “perceptible” APs in the area under consideration. Assume a total of M predetermined grid points. Any estimate of a MS’s location is limited to one of these grid points. During the offline phase, the mean RSS vector for each of the M grid points is determined by measurement and forms the corresponding location fingerprint. During the online phase, a sample vector RSS values measured at the MS’s current location is collected. Each of RSS elements (i.e., $r_i, i = 1, 2, \dots, N$) of a sample vector is assumed to be a Gaussian random variable and they are all mutually independent. For mathematical tractability as well as based on measurements, this assumption has been shown to be acceptable [7].

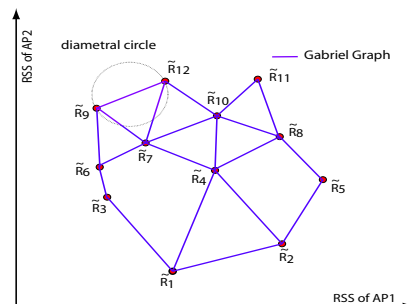


Figure 1: The Gabriel Proximity Graph

Before describing the analytical model, first we describe the concept of a Gabriel graph (GG) which is later used to determine neighbors of a given fingerprint. A GG is a graph that connects location fingerprints in Euclidean (RSS) space. Two fingerprints \tilde{R}_i, \tilde{R}_j are connected by the GG edge if a diametral circle (one with a line segment $\tilde{R}_i\tilde{R}_j$ as its diameter) contains no other fingerprints \tilde{R}_k . An example of a GG is shown in Figure 1. By brute force search for possible pairs of \tilde{R}_i, \tilde{R}_j and testing with all \tilde{R}_k , construction of a GG needs $O(NM^3)$ operations [2]. We define a *neighbor fingerprint* as a fingerprint that has a GG edge to a given fingerprint (e.g., in Figure 1, the GG neighbor set of fingerprint \tilde{R}_{12} is $GG_{nb}^{12} = \{\tilde{R}_7, \tilde{R}_9, \tilde{R}_{10}\}$).

The goal of our analytical model is to find the probability of selecting one fingerprint against others to evaluate the “influence level” of a fingerprint on the probability of correctly selecting a fingerprint. In other words, given that the MS is at a particular location, the probability distribution of fingerprint selection is desired. Computing the exact probabilities of selecting fingerprint is not simple. Given a MS at the i^{th} grid point, our model finds the approximate probability of selecting both the correct fingerprint \tilde{R}_i and other fingerprints \tilde{R}_k as follows [8]:

$$Prob\{\text{Select } \tilde{R}_i\} = \prod_{j \in GG_{nb}^i} Pr\{C_j \leq 0\} \quad (1)$$

¹With 100% accuracy (error distance = 0m), the location precision is equal to the probability of returning the correct location.

$$\text{Prob}\{\text{Select } \tilde{R}_k\} = \text{PEP}(\tilde{R}_i, \tilde{R}_k) \times \prod_{j \in GG_{nb}^i} \text{PCP}(\tilde{R}_k, \tilde{R}_j). \quad (2)$$

$C_j = \|\tilde{R}_i - R_i\| - \|\tilde{R}_j - R_i\|$. $\text{PEP}(\tilde{R}_i, \tilde{R}_k)$ is the *pairwise error probability* (PEP) that gives the probability that a sample vector R_i is closer to \tilde{R}_k than \tilde{R}_i . $\text{PCP}(\tilde{R}_k, \tilde{R}_j)$ is the *pairwise correct probability* (PCP) that gives the probability that \tilde{R}_k is selected over \tilde{R}_j . More details of this model and its validation can be found in [8]. Note that in the model, only a single GG is constructed and used regardless of the number of fingerprints in the system.

Once the approximate probability distribution of picking a fingerprint for each MS's location $i \in \{1, 2, \dots, M\}$ is calculated, it can be used to find fingerprints or locations that, if retained in the radio map, can actually degrade the overall performance of location estimation. The idea is as follows: for each of the probability distributions for a given MS location i , if there exists another location, such that its fingerprint has a higher or equivalent probability of being selected compared to the fingerprint of location i , then the fingerprint of location i is identified as an unnecessary and must be eliminated from the radio map. So pairwise comparisons are performed for finding all unnecessary fingerprints. Due to space constraints, we refer to [8] for details of the fingerprint elimination procedure.

In the next section we will describe two fingerprint clustering methods – Median clustering and K-Mean clustering to reduce the complexity of GG construction.

3. FINGERPRINT CLUSTERING METHODS

Median Clustering Method: The Median clustering method, as the name implies, divides all fingerprints in a radio map into two clusters using the median of the fingerprint's RSS values. The method produces one cluster with a half of all fingerprints having its RSS values below the median and another cluster with the remaining half. The median RSS is derived from one dimension (i.e., one AP) of location fingerprint vectors. Such a dimension (or an AP) is selected to have the largest RSS range (the difference between the maximum and a minimum of the RSS in the radio map from that AP). Intuitively, the “largest range” AP provides a large *signal distance* on average. Thus it provides a better ability to distinguish among fingerprints. If needed, the median clustering can be re-applied within clusters to further divide fingerprints (i.e., using the AP with the second largest RSS range within the former two clusters to produce four clusters). This method is relatively simple.

K-Mean Clustering Method: The K-Mean clustering method is a well-known pattern recognition clustering method [5]. Here, natural centers (known as *centroids*) of fingerprint clusters are determined. Each fingerprint is assigned to the cluster having the nearest centroid. K indicates the total number of clusters. The objective is to minimize total intra-cluster variance, which is the sum of squares of distances between fingerprints and the corresponding cluster centroids. The Euclidean distance is commonly used to measure proximity.

A simple algorithm for K-Mean clustering can be described as follows: (we use $K = 2$)

- 1) Initialize K cluster centers (μ_i for $i = 1, \dots, K$) by randomly picking K fingerprints
- 2) (2a) For all fingerprints, assign each fingerprint \tilde{R} to the

nearest centers and corresponding clusters

(2b) Recompute new mean centers;

$$\mu_i = (1/K_c) \times \sum_{\tilde{R}_c \in i^{th} \text{ cluster}} \tilde{R}_c,$$

for all $i = 1, \dots, K$ and $K_c = \text{number of fingerprints in a cluster}$
 3) Repeat step 2 until clustering converges.

Here \tilde{R}_c represents a current fingerprint member of a cluster. Note that a derived center is a mean vector of fingerprints within the same cluster. Also, centers from all clusters are needed to determine cluster membership for a sample RSS vector during the online phase. The K-Mean clustering method is iterative and it can be used even if many clusters (for $K > 2$) are desired. However, this is a tradeoff between additional clustering computation and improvement in predicting precision performance. We evaluate results from clustering methods in the following section.

4. EVALUATING CLUSTERING METHODS

4.1 Clustering Experiment Setup

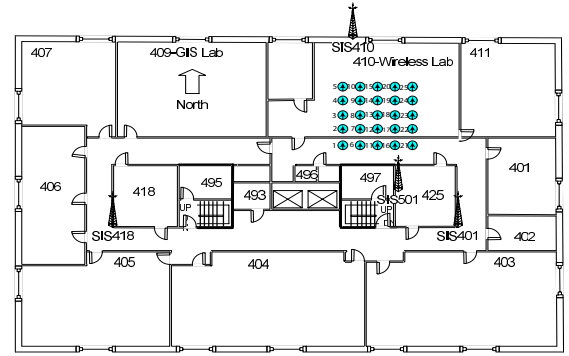


Figure 2: Measurement Setup: Scenario 1

To evaluate clustering methods, we consider two grid systems from previous measurements of location fingerprints [6]. Scenario 1 corresponds to a square grid of 25 locations as shown in Figure 2. 25 RSS fingerprints were collected at grid points in the 4th floor office environment of the Information Sciences (IS) building at Pitt. The grid spacing is about 1 meter. Numbers from 1 to 25 indicate the location labels of each grid point. Visible APs at these grid points are also shown in the figure. Scenario 2 consists of a larger grid system where fingerprints were collected from 71 grid points at the Hillman Library building as shown in Figure 3. Here, all RSS measurements were done inside the area on the 1st floor, where there is a large open space that shares the ceiling with the 2nd floor. Measurement grid points, as labeled by numbers 1 to 71, are shown by small arrows in Figure 2. 6 APs, placed on different floors, are shown and overlaid in the figure. 2 APs are placed each on the ground floor, the 2nd floor, and the 4th floor. The number (or alphabet) after “hl” indicates the floor in the building where an AP is located (i.e. 2 = 2nd floor, g = ground floor). Although signals from all 6 APs can be detected on the 1st floor, their coverage is not complete throughout the floor. The grid spacing between locations is non-uniform (> 2 m). This is because grid points were picked according to loca-

tions of reading tables inside the library. More details about this measurement can be found in [6].

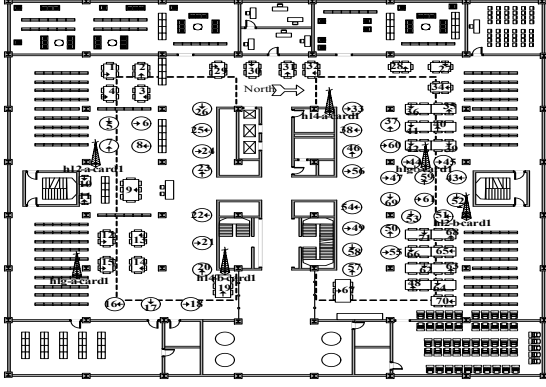


Figure 3: Measurement Setup: Scenario 2

Next we apply the Median clustering and the K-Mean clustering methods to both scenarios. Figure 4 shows Median clustering and the associated GGs (one graph for each cluster) in Scenario 1. There is a cluster with 12 fingerprints and another with 13 fingerprints. Markers with dots (or squares) indicate fingerprints within the same cluster. Each fingerprint is based on 2 APs (SIS410 and SIS501) shown in Figure 2. A fingerprint is represented as the RSS vector $[SIS410, SIS501]$. SIS410 is selected for the median for clustering. RSS values from SIS410 in the fingerprints provide the highest range. The RSS median value is -47.99 dBm. Note that each GG will be used separately in computing the probability distributions. Later the fingerprints needed to be eliminated within each cluster are determined. The K-Mean clustering and the GGs for the Scenario 1 are shown in Figure 5. The K-Mean clustering method results in 9 fingerprints in one cluster and 16 fingerprints in the other cluster. The centers for the two clusters are $[-51.86, -76.12]$ and $[-46.14, -80.31]$ (dBm) respectively.

In Scenario 2, fingerprints are based on 2 APs (hl2_b_card1 and hl4_b_card1) shown in Figure 3. A fingerprint is represented by the RSS vector $[hl2_b_card1, hl4_b_card1]$. We use 42 out of 71 locations that can see both APs. With median clustering in Scenario 2, fingerprints are divided into clusters

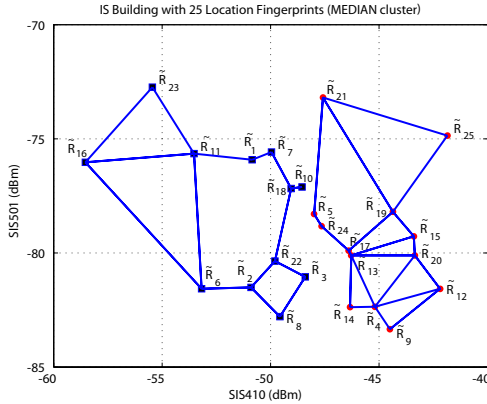


Figure 4: Median Clustering and GGs: Scenario 1

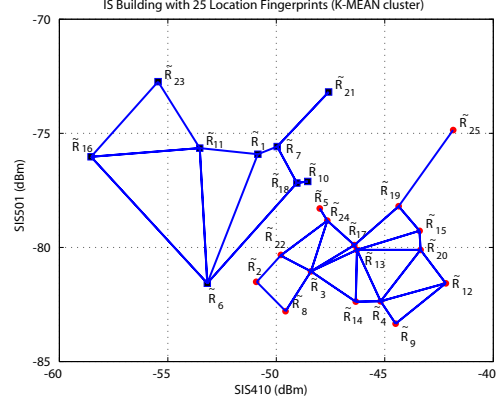


Figure 5: K-Mean Clustering and GGs: Scenario 1

of 21 fingerprints each indicated by dots and squares in Figure 6. The AP hl2_b_card1 was used to derive the median (-67.31 dBm) for clustering. Two GGs are constructed but not shown in the figure for a better view of the clustering. The K-Mean clustering method for Scenario 2 is shown in Figure 7. The clusters have 18 and 24 fingerprints. Centers for the two clusters in this scenario are $[-85.69, -89.42]$ and $[-60.74, -81.46]$ (dBm) respectively.

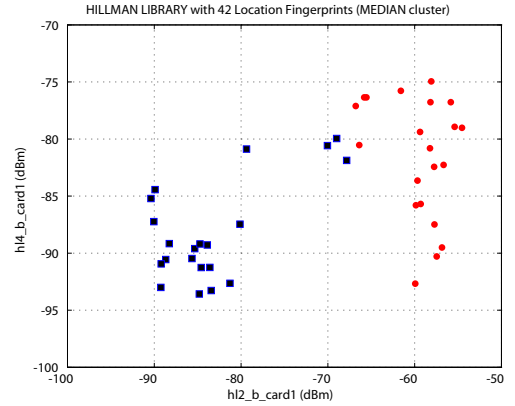


Figure 6: Median Clustering: Scenario 2

4.2 Results with Fingerprint Clustering

In each scenario, for each cluster, we use the separate GGs to find the approximate probability distributions of picking a location and eliminate unnecessary fingerprints as discussed in Section 2. We also simulate the system and obtain results for the average CDFs of error distance after fingerprint elimination. We compare the two results when clustering is used and without clustering where a single GG with all fingerprints is constructed. The average CDFs of the error distance in Scenarios 1 and 2 are shown in Figure 8 (a) and (b) respectively.

From Figure 8 (a), we can see that simulation results of the CDF of error distance with the Median and K-Mean clustering methods (indicated by the “elim-median” and “elim-kmean” lines respectively) show only a small difference (less than 0.05) from the results without clustering (indicated by

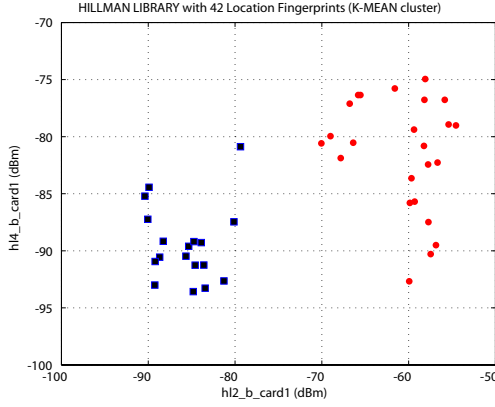


Figure 7: K-Mean Clustering: Scenario 2

the “elim-nocluster” line). In both cases, unnecessary fingerprints were eliminated. The CDF when all 25 fingerprints are used for positioning (indicated by the “nonelim” line) is shown as a baseline performance. The number of eliminated fingerprints are lower when the clustering methods are used; 9 from no clustering, 6 from the Median clustering, and 8 from the K-Mean clustering. This can be explained as follows. Because of the separate GGs used for determining the approximate probability distributions, pairwise comparison for eliminating fingerprints is only applied to fingerprints within the same cluster. As such, a fingerprint at the border of one cluster completely ignores the chance of picking a border fingerprint from the other cluster, although they may have a small Euclidean distance between them. Therefore, the elimination procedure results in fewer numbers of eliminated fingerprints as compared to the case where a single GG is used without clustering. In some ways what we would accomplish by eliminating fingerprints is accomplished with minimal penalty by clustering, since fingerprints from other clusters are eliminated automatically from comparison. Such elimination however is performed without considering pairwise error probabilities. However, it is quite possible that a measured RSS vector was associated with a wrong cluster during the online phase. The minimum error distance in such a case is between a pair of fingerprints, one from each cluster, with the smallest Euclidean distance. The maximum error could be larger, but is highly unlikely to occur because the probability that a measured RSS vector is associated with a fingerprint in the wrong cluster that has a large Euclidean distance to the correct fingerprint in the correct cluster is negligible. This minimum physical error is smaller with median clustering than K-Mean clustering (1m in both cases in Scenario 1; 8 and 12m respectively in Scenario 2). Further, simulations show that average CDFs of error distance for MSs located at different points cannot be distinguished from one another whether or not clustering was used, or based on the clustering method, at error distances greater than 2m. So we can conclude that, by applying clustering methods, we can still maintain almost the same precision performance as without clustering.

This is also true for Scenario 2 as indicated by the results in Figure 8 (b). In Scenario 2, the number of eliminated fingerprints are 12, 10, and 11 with no clustering, Median clustering, and K-Mean clustering respectively. The differ-

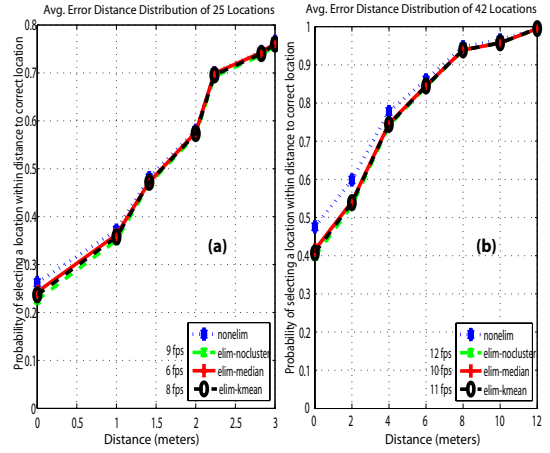


Figure 8: The Average CDF of Error Distance with Clustering in (a) Scenario 1 and (b) Scenario 2

ences in the average CDFs of the error distance between the two clustering methods and without clustering is minimal (less than 0.05, for error distances less than 4m and negligible at larger distances). In other words, no significant difference in the probability of selecting a location is observed if clustering is used.

4.3 Comparison of Computational Effort With and Without Clustering

We also analyze the effort for the total computation with fingerprint clustering and without clustering in both the offline and online phases. This will provide us a basis for concluding whether or not clustering helps in saving on computation (we already know that there is minimal impact on the performance based on the error CDF results shown earlier). With K clusters, we assume M_{ci} as the number of fingerprints in cluster C_i . Ideally, $M_{ci} = M/K$ and we use this below although the following can be done with knowledge of individual M_{ci} .

During the offline phase, there are 3 sequential tasks: 1) clustering 2) GG construction and 3) pairwise comparison for fingerprint elimination:

Clustering Setup: Here, either the Median or the K-Mean clustering is performed. With Median Clustering, for each of N APs, we first sort M fingerprints using a sorting algorithm, find the RSS range (max - min), and find the median. Sorting takes $O(M \log M)$ while finding the range and median ($O(1)$) can be ignored. Totally, this step needs $N \times O(M \log M) = O(NM \log M)$ operations. Next, we sort the computed ranges from N APs and find the largest which takes $O(N \log N)$ operations. Finally, we assign a cluster to each of M fingerprints by comparing each to the median. For K clusters (or K medians), this needs $O(KM)$ operations. Hence, the total number of operations with Median clustering is $O(NM \log M + N \log N + KM)$. If $N, K \ll M$ as is usually the case, the above result reduces to $O(NM \log M + KM)$. If K-Mean clustering is used, we first randomly pick K cluster centers from all fingerprints, which takes $O(K)$ operations. Next we assign a cluster to each of M fingerprints by computing Euclidean distances to K centers and pick the smallest. Computing K Euclidean distances to a fingerprint takes $K \times O(N) = O(KN)$ (because

Table 1: Computation of Fingerprint Clustering

Task	Clustering		No Clustering
	Median	K-Mean	
1)	$O(NM \log M + KM)$	$O((KN + K + N)Mt)$	-
2)	$O(NM^3/K^2)$	$O(NM^3/K^2)$	$O(NM^3)$
3)	$O(M^2/K)$	$O(M^2/K)$	$O(M^2)$
4)	$O(K)$	$O(KN)$	-
5)	$O(NM/K)$	$O(NM/K)$	$O(NM)$

of the N dimensions in the Euclidean distance) and picking the smallest distance takes $O(K)^2$. To compute K new centers, we need to consider M fingerprints each with N dimensions for a total of $O(NM)$ steps. Then each fingerprint has to be assigned to the clusters again. Given t iterations before convergence, totally we need $O(KM(N+1)t + NMt)$ steps. Hence, the total number of operations with K-Mean clustering is $O(K + KM(N+1)t + NMt)$. If $N, K \ll M$, the above result reduces to $O((KN + K + N)Mt)$.

GG Construction: Without clustering, the computation of a single GG takes $O(NM^3)$. With clustering and K GGs, we need $O(NM_{c1}^3 + \dots + NM_{cK}^3)$ operations. Ideally, with M/K fingerprints/cluster, we need $O(NM^3/K^2)$ steps.

Pairwise Fingerprint Elimination: With no clustering, we need $O(M^2)$ steps for the probability distribution comparisons. With clustering, we need $O(M^2/K)$ operations.

During the online phase, to determine a MS's location, a system with fingerprint clustering performs two sequential tasks: 4) determine cluster membership and 5) find nearest neighbor in signal space (NNSS)[8].

Determining Cluster Membership: To fix cluster membership, with Median clustering, the measured RSS sample in the dimension of the pre-selected "highest RSS range" AP, is compared with the pre-computed median (i.e., -47.99 dBm in Scenario 1 or -67.31 dBm in Scenario 2). With K clusters, we need $O(K)$ operations for this task. If K-Mean clustering is used, the Euclidean distance (N dimensions) between the measured RSS vector and the K cluster centers are computed. The cluster whose center has the smallest Euclidean distance to the sample vector is picked. Hence, determining the membership takes $O(KN)$ operations.

NNSS: Once the cluster membership is determined, the Euclidean distance from a sample vector to all fingerprints in this cluster is computed. The nearest fingerprint points to the estimate of the MS's location. Hence, on average, we need $O(NM/K)$ for the NNSS operation. Without clustering, we need $O(NM)$ operations. In fact, since some fingerprints were already eliminated from the offline phase, the number of comparisons needed is actually lower than that shown above. Table 1 summarizes the computational complexity of the five tasks.

To illustrate this numerically, we compare the number of operations that a system needs to complete both the online and offline tasks in Scenarios 1 and 2. The number of operations are based on the "Big O" notation described above. Based on Figure 4 and 5, the following parameters are used to compute the number of operations needed in Scenario 1: $M = 25$, $K = 2$, $N = 2$, $M_{c1} = 12$ (Median) or 9(K-Mean), $M_{c2} = 13$ (Median) or 16(K-Mean), and $t = 4$ (from

²We assume distance computation in one dimension and comparisons are equivalent in complexity.

simulations). Likewise, the following parameters are used to compute the number of operations needed in Scenario 2: $M = 42$, $K = 2$, $N = 2$, $M_{c1} = 21$ (Median) or 18(K-Mean), $M_{c2} = 21$ (Median) or 24(K-Mean), and $t = 6$.

Numerical results shown in Table 2 indicate that we save about one-third of the total operations with fingerprint clustering during the offline phase (8,284 or 10,789 operations compared to 31,875 operations in Scenario 1). Also, we could save about half the number of operations with fingerprint clustering in the online phase (27 or 29 operations as compared to 50 operations in Scenario 1).

Table 2: Operations Comparison

Method	Scenario 1		Scenario 2	
	Offline	Online	Offline	Online
Median	8,284	27	38,147	44
K-Mean	10,789	29	42,230	46
No Clustering	31,875	50	149,940	84

5. CONCLUSIONS

In this paper, fingerprint clustering is shown to help reduce computational effort for a location fingerprint system making the performance prediction and actual position determination scalable. Ongoing work is focussed on generalizing this approach further and quantifying its impact.

6. ACKNOWLEDGMENTS

Authors want to thank Dr. Kamol Kaemarungsi for providing measurement data and reviewers for their comments that improved the paper.

7. REFERENCES

- [1] P. Bahl and V. N. Padmanabhan. RADAR: An In-Building RF-based User Location and Tracking System. In *INFOCOM'00*, IEEE, March 2000.
- [2] B. Bhattacharya, R. Poulsen, and G. Toussaint. Application of Proximity Graphs to Editing Nearest Neighbor Decision Rules. In *The Int'l Sym. on Information Theory*, Santa Monica, 1981.
- [3] M. Brunato and R. Battiti. Statistical Learning Theory for Location Fingerprinting in Wireless LANs. *Computer Networks*, 47(6), 2005.
- [4] P. Castro, P. Chiu, T. Kremenek, and R. R. Muntz. A Probabilistic Room Location Service for Wireless Networked Environments. In *The Int'l Conference on Ubiquitous Computing Proceedings*, 2001.
- [5] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience Publication, November 2000.
- [6] K. Kaemarungsi. *Design of Indoor Positioning Systems based on Location Fingerprinting Technique*. PhD thesis, University of Pittsburgh, February 2005.
- [7] K. Kaemarungsi and P. Krishnamurthy. Properties of Indoor Received Signal Strength for WLAN Location Fingerprinting. In *MOBIQUITOUS'04*, IEEE, August 2004.
- [8] N. Swangmuang and P. Krishnamurthy. Location Fingerprint Analyses Toward Efficient Indoor Positioning. In *PerCom'08*, IEEE, March 2008.