The Horus WLAN Location Determination System

Moustafa Youssef and Ashok Agrawala
Department of Computer Science
University of Maryland
College Park, Maryland 20742
{*moustafa, agrawala*}*@cs.umd.edu*

Abstract

We present the design and implementation of the *Horus* WLAN location determination system. The design of the *Horus* system aims at satisfying two goals: high accuracy and low computational requirements. The *Horus* system identifies different causes for the wireless channel variations and addresses them to achieve its high accuracy. It uses location-clustering techniques to reduce the computational requirements of the algorithm. The lightweight *Horus* algorithm helps in supporting a larger number of users by running the algorithm at the clients.

We discuss the different components of the *Horus* system and its implementation under two different operating systems and evaluate the performance of the *Horus* system on two testbeds. Our results show that the *Horus* system achieves its goal. It has an error of less than 0.6 meter on the average and its computational requirements are more than an order of magnitude better than other WLAN location determination systems. Moreover, the techniques developed in the context of the *Horus* system are general and can be applied to other WLAN location determination systems to enhance their accuracy. We also report lessons learned from experimenting with the *Horus* system and provide directions for future work.

1 Introduction

Horus is an RF-based location determination system. It is currently implemented in the context of 802.11 wireless LANs [25]. The system uses the signal strength observed for frames transmitted by the access points to infer the user location. Since the wireless cards measure the signal strength information of the received frames as part of their normal operation, this makes the *Horus* system a software solution on top of the wireless network infrastructure. There are two classes of WLAN location determination systems: client-based and infrastructure-based. Both have their own set of applications. *Horus* is currently implemented as a client-based system. A large class of applications [10], including location-sensitive content delivery, direction finding, asset tracking, and emergency notification, can be built on top of the *Horus* system.

WLAN location determination is an active research area [5, 6, 8, 9, 12, 13, 15, 17, 20–22, 29–31, 33, 34]. WLAN location determination systems usually work in two phases: an *offline* training phase and an *online* location determination phase. During the offline phase, the system tabulates the signal strength received from the access points at selected locations in the area of interest, resulting in a so-called *radio map*. During the location determination phase, the system use the signal strength samples received from the access points to "search" the radio map to estimate the user location.

Radio-map based techniques can be categorized into two broad categories: deterministic techniques and probabilistic techniques. *Deterministic techniques* [5, 6, 22] represent the signal strength of an access point at a location by a scalar value, for example, the mean value, and use non-probabilistic approaches to estimate the user location. For example, in the *Radar* system [5, 6] the authors use nearest neighborhood techniques to infer the user location. On the other hand, *probabilistic* techniques [8, 9, 13, 17, 20, 21, 29–31, 33, 34] store information about the signal strength distributions from the access points in the radio map and use probabilistic techniques to estimate the user location. For example, the Nibble system [8, 9] uses a Bayesian Network approach to estimate the user location.

The *Horus* system lies in the probabilistic techniques category. The design of the *Horus* system aims at satisfying two goals: high accuracy and low computational requirements. The *Horus* system identifies different causes for the wireless channel variations and addresses them to achieve its high accuracy. It uses location-clustering techniques to reduce the computational requirements of the algorithm. The lightweight *Horus* algorithm allows it to be implemented in energy-constrained devices. This

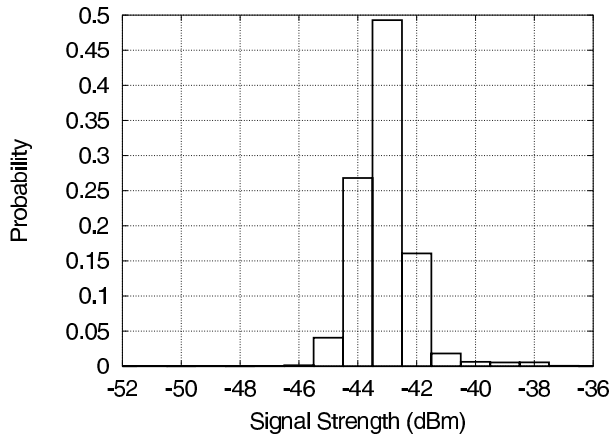


Figure 1: An example of the normalized signal strength histogram from an access point.

non-centralized implementation helps in supporting a larger number of users. In this paper, we present the different components of the *Horus* system and show how they work together to achieve its goals. We discuss our *Horus* implementation under two different operating systems and evaluate its performance on two different indoor testbeds.

The rest of the paper is structured as follows: in the next section, we describe the different causes of variations in the wireless channel. In Section 3 we present the different components of the *Horus* system that deal with the noisy characteristics of the wireless channel. We present the results of testing the *Horus* system on two different testbeds in Section 4. Section 5 presents our experience while building the *Horus* system. In Section 6 we discuss related work. Finally, Section 7 concludes the paper and provides directions for future work.

2 Wireless Channel Characteristics

In this section, we identify the different causes of variations in the wireless channel quality and how they affect the WLAN location determination systems. We are mainly concerned with the variations that affect the received signal strength. We start by describing our sampling process. Then, we categorize the variations in the wireless channel as temporal variations and spatial variations. We performed all the experiments in this section in a typical office building, measured during the day when people are around.

2.1 Sampling Process

A key function required by all WLAN location determination systems is signal-strength sampling. We used a Lucent Orinoco silver network interface card (NIC) supporting up to 11 Mbit/s data rate [3]. The *Horus* system is implemented under both the Linux and Windows operating systems.

For the Linux OS, we modified [1] the *Lucent Wavelan* driver so that it returns the signal strength of probe response frames received from all access points in the NIC range using active scanning [25]; our driver was the first to support this feature.

The scanning process output is a list of the *MAC* addresses of the access points associated with the signal strength observed in this scan (through probe response frames). Each scan's result set represents a sample.

We also developed a wireless API [1] that interfaces with any device driver that supports the wireless extensions [2]. The device driver and the wireless API have been available for public download and have been used by others in wireless research.

For the Microsoft Windows operating system, we used a custom-built NDIS driver to obtain the signal strength from the wireless card (using active scanning). This gives us more control over the scanning process as described in Section 5.

We now describe the different causes of variations in a wireless channel. We divide these causes into two categories: temporal variations and spatial variations.

2.2 Temporal variations

This section describes how the wireless channel changes over time when the user is standing at a fixed position.

2.2.1 Samples from one access point

We measured the signal strength from a single access point over a five minute period. We took the samples one second apart for a total of 300 samples. Figure 1 shows the normalized histogram of the received signal strength. In our experience, the histogram range can be as large as 10 dBm or more. This time variation of the channel can be due to changes in the physical environment such as people moving about [23].

These variations suggest that the radio map should reflect this range of values to increase the accuracy. Moreover, during the online phase, the system should use more than one sample in the estimation process to have a better estimate of the signal strength at a location.

2.2.2 Samples Correlation

Figure 2 shows the autocorrelation function of the samples collected from **one access point** (one sample per second) at a fixed position. The figure shows that the autocorrelation of consecutive samples ($lag = 1$) is as high as 0.9. This high autocorrelation is expected as over a short period of time the signal strength received from an access point at a particular location is relatively stable (modulo the changes in the environment discussed in the previous section).

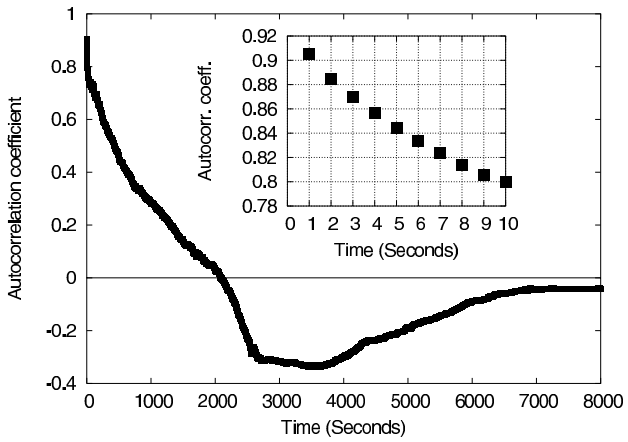


Figure 2: An example of the autocorrelation between samples from an access point (one sample per second). The sub-figure shows the autocorrelation for the first 10 seconds.

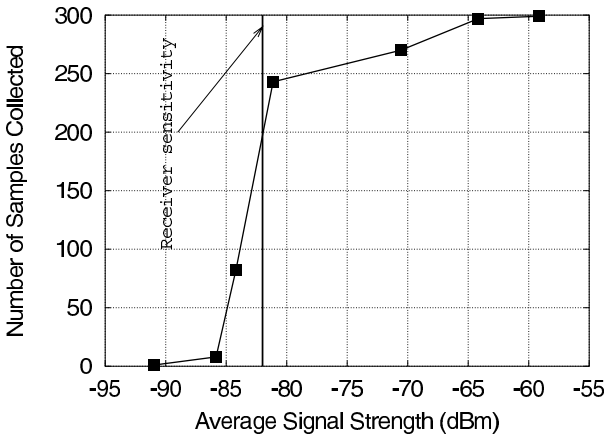


Figure 3: Relation between the average signal strength of an access point and the percentage of samples received from it during a 5-minute interval.

This high autocorrelation value has to be considered when using multiple samples from one access point to enhance accuracy. Assuming independence of samples from the same access point leads to the undesirable result of degraded system performance as the number of samples is increased (as explained in Section 4) as in a typical WLAN environment samples from the same AP are highly correlated.

2.2.3 Samples from different access points

We performed an experiment to test the behavior of access points with different average signal strength at the same location. During this experiment, we sampled the signal strength from each access point at the rate of one sample per second. Figure 3 shows the relation between the average signal strength received from an access point and the percentage of samples we receive from it during a period of 5 minutes. The figure shows that the number of samples collected from an access point is a monotonically increasing function of the average signal strength of this access point. Assuming a constant noise level, the higher the signal strength, the higher the signal to noise ratio and the more probable it becomes that the 802.11b card will identify the existence of a frame. The sharp drop at about -81 dBm can be explained by noting that the receiver sensitivity (minimum signal power required to detect a frame) for the card we used was -82 dBm.

2.3 Spatial characteristics

These variations occur when the receiver position is changed. We further divide these variations into large-scale variations and small-scale variations.

2.3.1 Large-Scale Variations

Figure 4 shows the average signal strength received from an access point as the distance from it increases. The signal strength varies over a long distance due to attenuation of the RF signal.

Large-scale variations are desirable in RF-based systems as they lead to changing the signature stored in the radio map for different locations and, hence, better differentiation between these locations.

2.3.2 Small-Scale Variations

These variations happen when the user moves over a small distance (order of wavelength). This leads to changes in the average received signal strength. For the 802.11b networks working at the 2.4 GHz range, the wavelength is 12.5 cm and we measure a variation in the average signal strength up to 10 dBm in a distance as small as 7.6 cm (3 inches) (Figure 5).

Dealing with small-scale variations is challenging. To limit the radio map size and the time required to build the radio map, selected radio map locations are typically placed more than a meter apart. This means that the radio map does not capture small-scale variations leading to decreased accuracy in the current WLAN location systems.

In the next section, we indicate how the *Horus* system handles these temporal and spacial variations.

3 The *Horus* System

In this section, we present the different components of the *Horus* system.

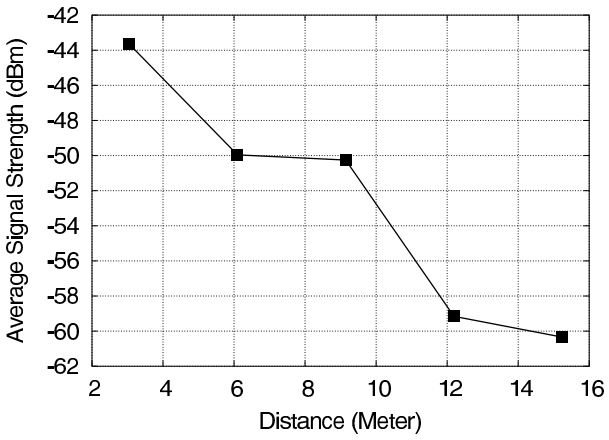


Figure 4: Large-scale variations: Average signal strength over distance.

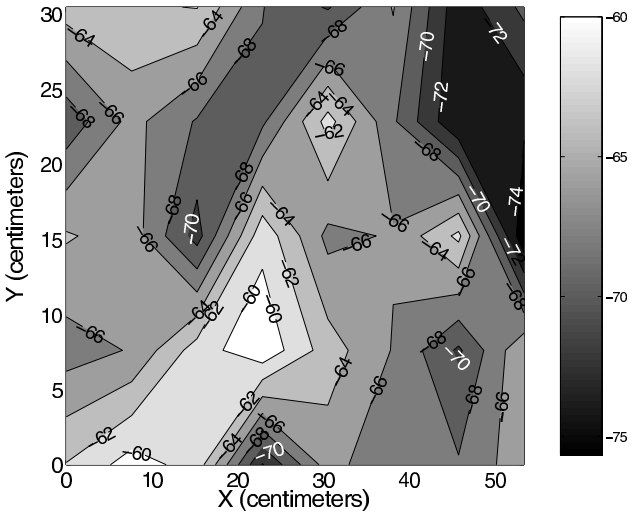


Figure 5: Small-scale variations: Signal strength contours from an AP in a 30.4 cm (12 inches) by 53.3 cm (21 inches) area.

3.1 Overview

Figure 6 shows the overall system. The *Horus* system works in two phases:

1. *Offline phase*: to build the radio map, cluster radio map locations, and do other preprocessing of the signal strength models.

2. *Online Phase*: to estimate the user location based on the received signal strength from each access point and the radio map prepared in the offline phase.

The radio map stores the distribution of signal strength received from each access point at each sampled location. There are two modes for operation of the *Horus* system: one uses non-parametric distributions and the other uses parametric distributions.

The *Clustering* module is used to group radio map locations based on the access points covering them. Clustering is used to reduce the computational requirements of the system and, hence, conserve power (Section 3.7).

The *Discrete Space Estimator* module returns the radio map location that has the maximum probability given the received signal strength vector from different access points (Section 3.3).

The *Correlation Modelling and Handling* modules use an autoregressive model to capture the correlation between consecutive samples from the same access point. This model is used to obtain a better discrete location estimate using the average of n correlated samples (Section 3.4).

The *Continuous Space Estimator* takes as an input the discrete estimated user location, one of the radio map locations, and returns a more accurate estimate of the user location in the continuous space (Section 3.5).

The *Small-Scale Compensator* module handles the small-scale variation characteristics of the wireless channel (Section 3.6).

We start by laying out the mathematical framework for the approach then give details about different components of the system.

3.2 Mathematical Model

Without loss of generality, let $\mathbb{X}$ be a 2 dimensional physical space. At each location $x \in \mathbb{X}$, we can get the signal strength from k access points. We denote the k-dimensional signal strength space as $\mathbb{S}$. Each element in this space is a k-dimensional vector whose entries represent the signal strength readings from different access points. We denote samples from the signal strength space $\mathbb{S}$ as s. We also assume that the samples from *different* access points are independent.

The problem becomes, given a signal strength vector $s = (s_1, ..., s_k)$, we want to find the location $x \in \mathbb{X}$ that maximizes the probability $P(x/s)$.

In the next section, we assume a discrete $\mathbb{X}$ space. We discuss the continuous space case in Section 3.5.

3.3 Discrete Space Estimator

During the offline phase, the *Horus* system estimates the signal strength histogram for each access point at each location. These histograms represent the *Horus* system's radio map. Now consider the online phase. Given a signal strength vector $s = (s_1, ..., s_k)$, we want to find the location $x \in \mathbb{X}$ that maximizes the probability $P(x/s)$,

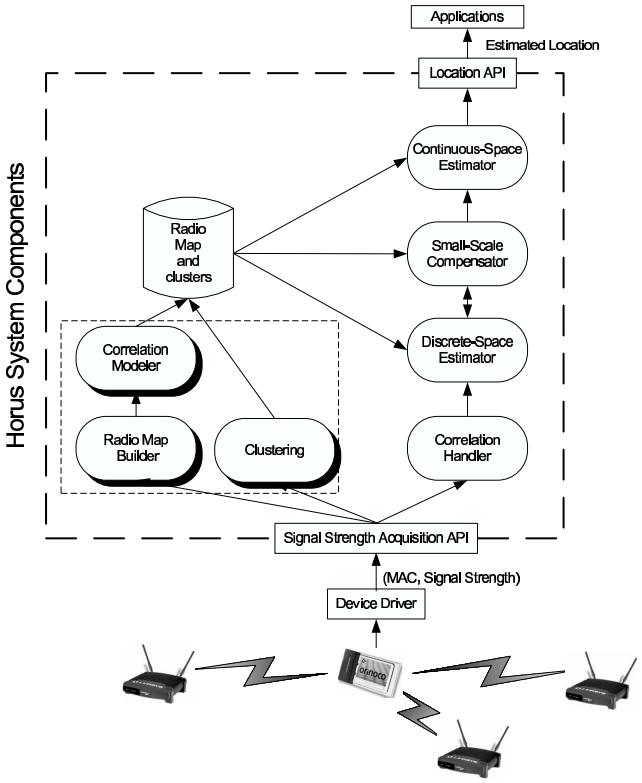


Figure 6: *Horus* Components: the arrows show information flow in the system. Shadowed blocks represent modules used during the offline phase.

i.e., we want

$$\text{argmax}_x[P(x/s)] \tag{1}$$

Using Bayes' theorem, this can be shown to be equivalent to [33]:

$$\text{argmax}_x[P(x/s)] = \text{argmax}_x[P(s/x)] \tag{2}$$

$P(s/x)$ can be calculated using the radio map as:

$$P(s/x) = \prod_{i=1}^{k} P(s_i/x) \tag{3}$$

The signal-strength histogram can be approximated by a parametric distribution such as the Gaussian distribution. We compare the performance of the discrete-space estimator based on the parametric and non-parametric distributions in the Section 4.

3.4 Correlation Handling

To account for the temporal signal-strength variations, it is important to average multiple signal strength samples from the same access point. As we showed in Figure 2, the autocorrelation of successive samples collected from one access point is as high as 0.9. Assuming independence of samples from the same access point leads to the undesirable result of degraded system performance as the number of averaged samples is increased (as we demonstrate below, in Section 4).

3.4.1 Mathematical model

We use an autoregressive model to capture the correlation between different samples from the same AP.

Let s_t be the *stationary* time series representing the samples from an access point, where t is the discrete time index. s_t can be represented as a *first order* autoregressive model [7] as:

$$s_t = \alpha s_{t-1} + (1-\alpha)v_t \quad ; 0 \leq \alpha \leq 1 \tag{4}$$

where v_t is a noise process, independent of s_t, and α is a parameter that determines the degree of autocorrelation of the original samples. Moreover, different samples from v_t are independent and identically distributed (i.i.d.).

The model in Equation 4 states that the current signal strength value (s_t) is a linear aggregate of the previous signal strength value (s_{t-1}) and an independent noise value (v_t). The parameter α gives flexibility to the model as it can be used to determine the degree of autocorrelation of the original process. For example, if α is zero, the samples of the process s_t are i.i.d.'s, whereas if α is 1 the original samples are identical (autocorrelation=1).

Assuming that the signal strength distribution of samples from an access point is Gaussian with mean μ and variance σ^2, we have shown in [29, 31] that the distribution of the average of n correlated samples is a Gaussian distribution with mean μ and variance given by:

$$\frac{1+\alpha}{1-\alpha}\sigma^2 \tag{5}$$

3.4.2 Correlation modeler

The purpose of the correlation modeler component is to estimate the value of α in the autoregressive model and to pre-calculate the parameters of the distribution of the average of n correlated samples during the offline phase. In a previous work [29,31], we have shown that α can be approximated using the autocorrelation coefficient with lag 1. The variance of the distribution can be calculated using Equation 5. These distribution parameters (μ, σ, and α) are then stored in the radio map.

3.4.3 Correlation handler

During the online phase, the correlation handler module averages the value of n consecutive samples from an access point and passes this information to the discrete-space estimator, which uses the distributions stored in the radio map (taking correlation into account using the information in Section 3.4.1) to estimate the user location.

3.5 Continuous Space Estimator

The discrete-space estimator returns a single location from the set of locations in the radio map. To increase the system accuracy, the *Horus* system uses two techniques to obtain a location estimate in the continuous space.

3.5.1 Technique 1: Center of Mass of the Top Candidate Locations

This technique is based on treating each location in the radio map as an object in the physical space whose weight is equal to the normalized probability[1] assigned by the discrete-space estimator. We then obtain the center of mass of the N objects with the largest mass, where N is a parameter to the system, $1 \leq N \leq ||\mathbf{X}||$.

More formally, let $p(x)$ be the probability of a location $x \in \mathbb{X}$, i.e., the radio map, and let $\bar{\mathbb{X}}$ be the list of locations in the radio map *ordered* in a descending order according to the normalized probability (the location with lower ID comes first for locations with equal probability). The center of mass technique estimates the current location x as:

$$x = \frac{\sum_{i=1}^{N} p(i)\bar{\mathbb{X}}(i)}{\sum_{i=1}^{N} p(i)} \quad (6)$$

where $\bar{\mathbb{X}}(i)$ is the i^{th} element of $\bar{\mathbb{X}}$

Note that the estimated location x need not be one of the radio map locations.

3.5.2 Technique 2: Time-Averaging in the Physical Space

The second technique uses a time-average window to smooth the resulting location estimate. The technique obtains the location estimate by averaging the last W locations estimates obtained by either the discrete-space estimator or the continuous-space estimator discussed in the previous section.

More formally, given a stream of location estimates $x_1, x_2, ..., x_t$, the technique estimates the current location $\bar{x_t}$ at time t as:

$$\bar{x_t} = \frac{1}{\min(W,t)} \sum_{t-\min(W,t)+1}^{t} x_i \quad (7)$$

We compare the two techniques in Section 4.

3.6 Small-Scale Compensator

Dealing with small-scale variations (Figure 5) is challenging. Since the selected radio map locations are typically placed more than a meter apart, to limit the radio map size, the radio map does not capture small-scale variations. This contributes significantly to the estimation errors in the current systems. The *Horus* system uses the *Perturbation* technique to handle the small-scale variations. The technique is based on two sub-functions: detecting small-scale variations and compensating for small-scale variations.

3.6.1 Detecting small-scale variations

In order to detect small-scale variations, the *Horus* system uses the heuristic that users' location cannot change faster than their movement rate. The system calculates the estimated location using the standard radio map and the inference algorithm, then calculates the distance between the estimated location and the previous user location. If this distance is above a threshold, based on the user movement rate and estimation frequency, the system decides that there are small-scale variations affecting the signal strength.

3.6.2 Compensating for small-scale variations

To compensate for these small-scale variations, the system perturbs the received vector entries, re-estimates the location, and chooses the nearest location to the previous user location as the final location estimate. For example, if one sample includes a signal-strength observation from each of k access points $(s_1, s_2, \ldots, s_k)$, the system tries all 3^k combinations in which each of the k observations i is replaced by one of three values, s_i, $s_i(1+d)$, or $s_i(1-d)$; we explore the parameter d in Section 4.2.4. An enhancement of this approach is to perturb a subset of the access points. The effect of the number of access points to perturb and the value of d on accuracy is described in Section 4.

3.7 Clustering Module

This section describes the *Incremental Triangulation* (IT) clustering technique used by the *Horus* system to reduce the computational requirements of the location determination algorithm. We define a *cluster* as a set of locations sharing a common set of access points. We call this common set of access points the *cluster key*. The problem can be stated as: Given a location x, we want to determine the cluster to which x belongs. The noisy characteristics of the wireless channel described in Section 2 make clustering a challenging problem because the number of access points covering a location varies with time.

The *IT* approach is based on the idea that each access point defines a subset of the radio map locations that are covered by this access point. These locations can be viewed as a cluster of locations whose key is the access point covering the locations in this cluster. If during the location determination phase we use the access points incrementally, one after the other, then starting with the first access point, we restrict our search space to the locations covered by this access point. The second access point chooses only the locations in the range of the first

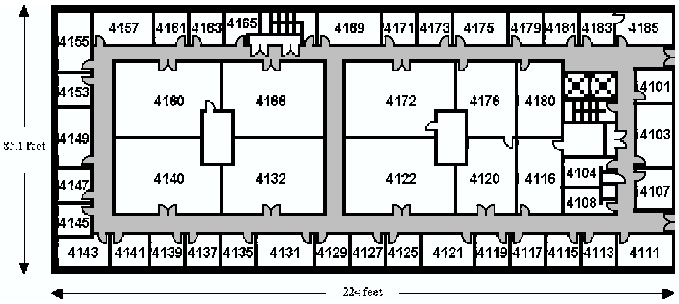


Figure 7: Floor plan for the first testbed. Readings were collected in the corridors and inside the rooms.

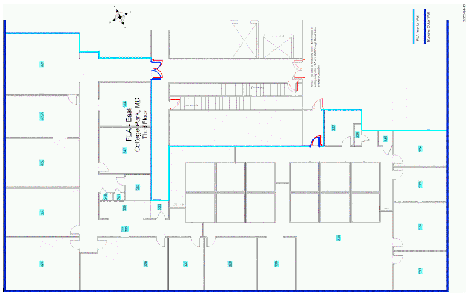


Figure 8: Floor plan of the office space where the second experiment was conducted. Readings were collected in the corridors and inside the rooms.

access point and covered by the second access point and so on, leading to a multi-level clustering process.

Notice that no preprocessing is required in the offline training phase. During the online phase, a location x belongs to a cluster whose key is access point a if there is information about access point a at location x in the radio map.

The algorithm works as follows. Given a sequence of observations from each access point, we start by sorting the access points in descending order according to the average received signal strength. For the first access point, the one with the strongest average signal strength, we calculate the probability of each location in the radio map set given the observation sequence from this access point alone. This gives us a set of candidate locations (locations that have non-zero probability). If the probability of the most probable location is "significantly" higher (according to a threshold) than the probability of the second most probable location, we return that most probable location as our location estimate, after consulting only one access point. If this is not the case, we go to the next access point in the sorted access point list. For this access point, we repeat the same process again, but only for the set of candidate locations obtained from the first access point. We study the performance of the *IT* approach in Section 4.

4 Experimental Evaluation

In this section we start by showing the effect of each module independently on the the accuracy of the basic algorithm. We then show the effect of using all the components together on the performance of the *Horus* system.

4.1 Experimental Testbed

We performed our experiment in two different testbeds.

4.1.1 Testbed 1

We performed our first experiment in the south wing of the fourth floor of the A. V. Williams building in the University of Maryland at College Park. The layout of the floor is shown in Figure 7. The wing has a dimension of 68.2 meters by 25.9 meters. The technique was tested in the University of Maryland wireless network using Cisco access points. 21 access points cover the multi-story wing and were involved in testing.

The radio map has 110 locations along the corridors and 62 locations inside the rooms. On the average, each location is covered by 6 access points. The *Horus* system was running under the Windows XP professional operating system.

4.1.2 Testbed 2

We performed the second experiment in another office space (Figure 8). The area of the experiment site is approximately 11.8 meters by 35.9 meters covering corridors, cubicles, and rooms. Five LinkSys access points and one Cisco access point cover the test area.

We have a total of 110 locations in the radio map. On the average, each location is covered by 4 access points. The *Horus* system was running under the *Linux* (kernel 2.5.7) operating system.

4.1.3 Data collection

The radio map locations were marked on the floor before the experiment and the user clicked on the map to point the location of the radio map points. We collected 100 samples, spaced 300 ms apart, at each radio map location. We expect an error of about 15-20 cm due to the inaccuracies in clicking the map.

The training data was placed 1.52 meters (5 feet) apart for the first testbed and 2.13 meters apart for the second testbed (7 feet).

For each testbed, we selected 100 test locations to random cover the entire test area (none of them coincide with a training point). For both testbeds, the test set was

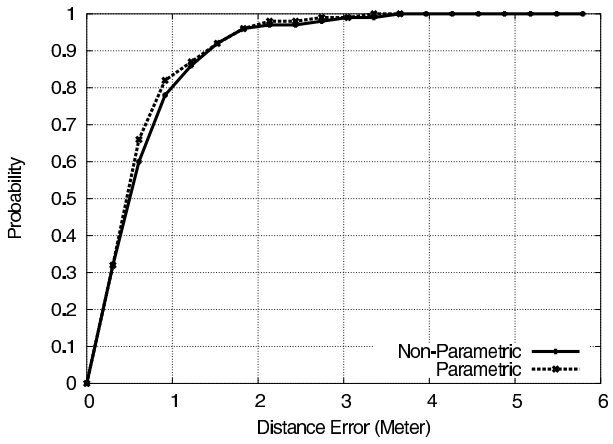


Figure 9: Performance of the basic algorithm of the *Horus* system for the first testbed.

collected by different persons on different days and times of day than the training set. This difference presents a realistic testbed and should, if anything, decrease the measured accuracy of our approach because it lessens the likelihood that the test data is a close match to the training data.

4.2 Results

We show the effect of each module independently on the performance of the discrete-space estimator and present the overall system performance in Section 4.3.

4.2.1 Discrete-space estimator (Basic algorithm)

Figure 9 shows the performance of the basic algorithms of the *Horus* system for the first testbed. The system can achieve an accuracy of 1.4 meters 90% of the time. The performance of the parametric and non-parametric methods is comparable with a slight advantage for the parametric method. Using a parametric distribution to estimate the signal-strength distribution smooths the distribution shape to account for missing signal strength values in the training phase (due to the finite training time). This smoothing avoids obtaining a zero probability for any signal strength value that was not obtained in the training phase and hence enhances the accuracy.

Table 1 shows the summary of the results for the two testbeds. Details for the second testbed can be found in [28].

4.2.2 Correlation handler

Figure 10 shows the performance of the *Horus* system when taking the correlation into account and without taking the correlation into account for the first testbed. We estimated the value of α to be 0.9. The figures show that under the independence assumption, as the number of averaged samples increases, the performance degrades. The minimum value at $n = 2$ can be explained by noting that there are two opposing factors affecting the system accuracy:

1. as the number of averaged samples n increases, the accuracy of the system should increase.

2. as n increases, the estimation of the distribution of the average of the n samples becomes worse due to the wrong independence assumption.

At low values of n ($n = 1, 2$) the first factor is the dominating factor and hence the accuracy increases. Starting from $n = 3$, the effect of the bad estimation of the distribution becomes the dominating factor and accuracy degrades.

Using the modified technique, the system average accuracy is enhanced by more than 19% using five signal-strength samples.

Table 1: Summary of the percentage enhancement of different components on the basic algorithm

Technique	Testbed 1	Testbed 2
Correlation Handling	19%	11%
Center of Mass	13%	6%
Time Averaging	24%	15%
Small-Scale Compensator	25%	21%

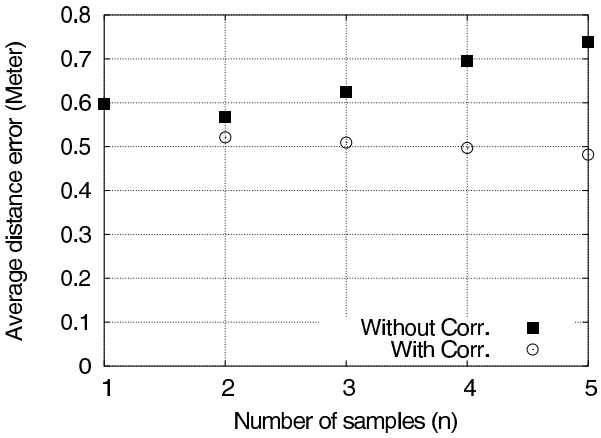


Figure 10: Average distance error with and without taking correlation into account for the first testbed.

4.2.3 Continuous space estimator

Center of Mass Technique: Figure 11 shows the effect of increasing the parameter N (number of locations to interpolate between) on the performance of the center of mass technique for the first testbed. Note that the special

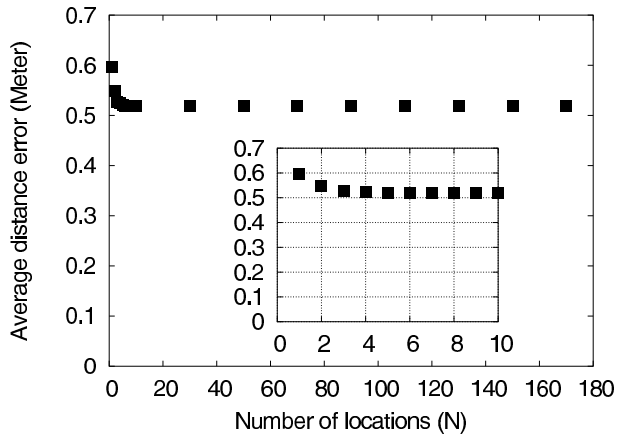


Figure 11: Average distance error using the center of mass technique for the first testbed.

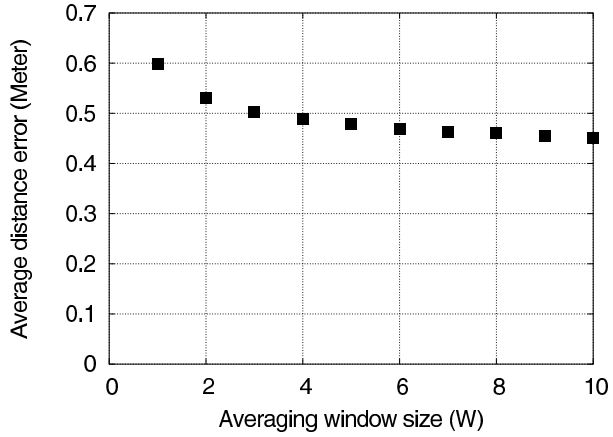


Figure 12: Average distance error using the time-averaging technique for the first testbed.

case of $N = 1$ is equivalent to the discrete-space estimator output. The figures show that the performance of the *Horus* system is enhanced by more than 13% for $N = 6$.

Time-averaging Technique: Figure 12 shows the effect of increasing the parameter W (size of the averaging window) on the performance of the time-averaging technique. The figures show that the larger the averaging window, the better the accuracy. The performance of the *Horus* system is enhanced by more than 24% for $W = 10$.

4.2.4 Small-scale compensator

For the purpose of detecting small-scale variations, we assume a maximum user speed of two meters per second.

Figure 13 shows the effect of changing the perturbation fraction (d, which is the amount by which to perturb each access point) on average error. We can see from this figure that the best value for the perturbation fraction is 0.05 for the first testbed. We use these values for the rest of this section.

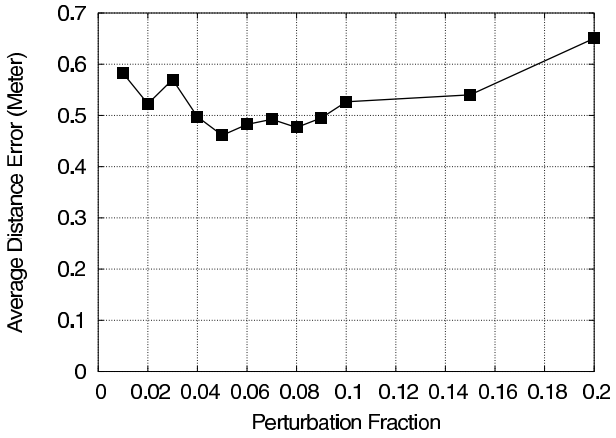


Figure 13: Effect of changing the perturbation fraction on average distance error.

Figure 14 shows the effect of increasing the number of perturbed access points on the average distance error. The access points chosen at a location are the strongest access points in the set of access points that cover that location. The figure shows that perturbation technique is not sensitive to the number of access points. This means that perturbing one access point only is sufficient to enhance the performance.

Figures 15 shows the effect of using the perturbation technique on the basic *Horus* system. The perturbation technique reduces the average distance error by more than 25% and the worst-case error is reduced by more than 30%.

4.2.5 Clustering module

Figures 16 and 17 shows the effect of the parameter $Threshold$ on the performance. For small values of the $Threshold$ parameter, the decision is taken quickly after examining a small number of access points. As the threshold value increases, more access points are consulted to reach a decision. As the number of access points consulted increases, the number of operations (multiplications) per location estimate increases and so does the accuracy.

4.3 Overall System Performance

In the previous sections, we studied the effect of each component of the *Horus* system separately on the performance. In this section, we compare the performance of the full *Horus* system, to the performance of a deterministic technique (the *Radar* system [5]) and a probabilistic

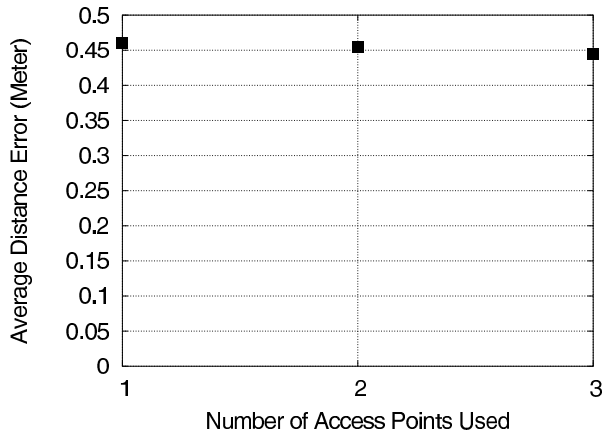


Figure 14: Effect of changing the number of perturbed access points on average distance error.

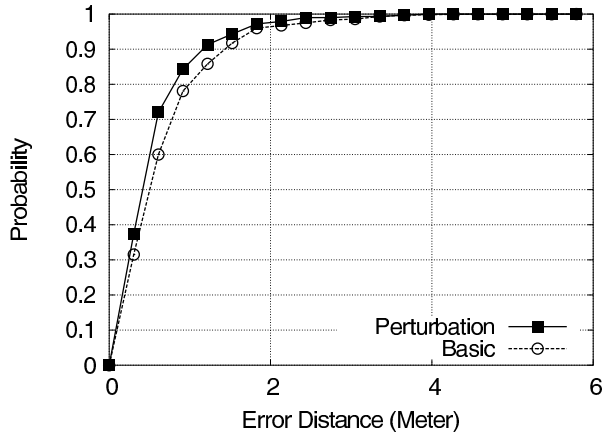


Figure 15: CDF for the distance error for the first testbed.

technique [21]. We use the parametric distribution technique. Table 2 shows the values of different parameters.

Figures 18 shows the comparison for the two testbeds (the curve for the *Radar* system is truncated). Tables 3 summarizes the results. The table shows that the average accuracy of the *Horus* system is better than the Radar system by more than 89% for the first testbed and 82% for the second testbed. The worst case error is decreased by more than 93% for the first testbed and 70% for the second testbed.

Comparing the probabilistic system in [21] to the *Horus* system shows that the average error is decreased by more than 35% for the first testbed and 27% for the second testbed. The worst case error is decreased by more than 78% for the first testbed and 70% for the second testbed. These results show the effectiveness of the proposed techniques.

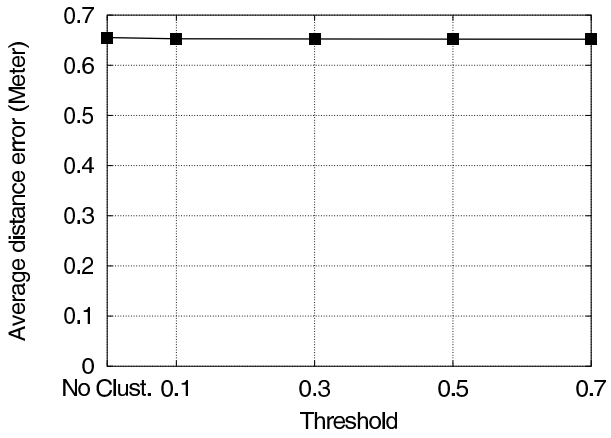


Figure 16: Effect of the parameter $Threshold$ on the average distance error for the first testbed.

Table 2: Estimation parameters for the two testbeds

Parameter	**Test. 1**	**Test. 2**
Correlation Degree (α)	0.9	0.7
Num. of avg. samples (n)	10	10
Num. of loc. used in interp. (N)	6	6
Averaging window (W)	10	10
Threshold	0.1	0.1

The performance of the three systems is better in the first testbed than the second testbed as the first testbed has a higher density of APs per location and the calibration points were closer for the first testbed.

Moreover, the *Horus* system leads to more than an order of magnitude savings in the number of multiplications required per location estimate compared to the other systems (250 multiplications for Horus compared to 2708 for the other two systems).

We also applied the enhancement discussed in this paper (without correlation handling) to the Radar system. We summarize the results is Table 3. These results show the effectiveness of the techniques proposed in the paper and that these techniques are general and can be applied to other WLAN location determination systems to enhance their accuracy.

5 Discussion

In this section, we highlight some of our experience with the *Horus* system.

Table 3: Comparison of the *Horus* system and other systems (error in centimeters)

	Testbed 1					Testbed 2				
	Median	**Avg**	**Stdev**	**90%**	**Max**	**Median**	**Avg**	**Stdev**	**90%**	**Max**
Horus	39	42	28	86	121	51	64	53	132	289
System [21]	48	65	63	143	578	72	86	77	181	991
Radar	296	400	326	853	1757	341	361	184	611	967
Radar with Horus tech.	161	193	107	302	423	142	195	106	332	483

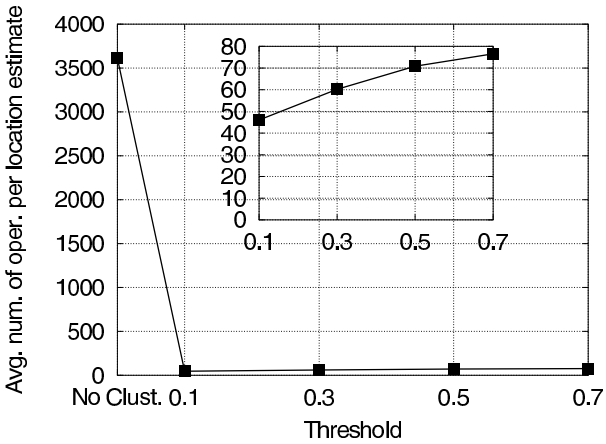


Figure 17: Effect of the parameter $Threshold$ on the average number of operations per location estimate for the first testbed. The sub-figure shows the same curve for $Threshold = [0.1, 0.7]$.

5.1 Parametric vs Non-Parametric Distributions

The *Horus* system can model the signal strength distributions received from the access points using parametric or non-parametric distributions. The main advantage of the non-parametric technique is the efficiency of calculating the location estimate, while the parametric technique reduces the radio map size and smooths the distribution shape which leads to a slight computational advantage of the parametric technique over the non-parametric technique.

5.2 Location Estimation Latency

The correlation handling and the continuous space estimator modules each use more than one sample to increase the accuracy of the system. However, a side effect of this increased accuracy is that the latency of calculating the location estimate increases. In general, we have a tradeoff between the accuracy required and the latency of the location estimate. The higher the required accuracy, the higher the number of samples required and the higher the latency to obtain the location estimate. This decision is dependent on the application in use.

Latency can be reduced by presenting the location estimate incrementally using one sample at a time. The system need not to wait until it acquires the n samples all at once. Instead, it can give a more accurate estimate of the location as more samples become available by reporting the estimated location given the partial samples it has. Other factors that affect the choice of the value of n are the user mobility rate and the sampling rate. The higher the user mobility rate or the sampling rate, the lower the value of n.

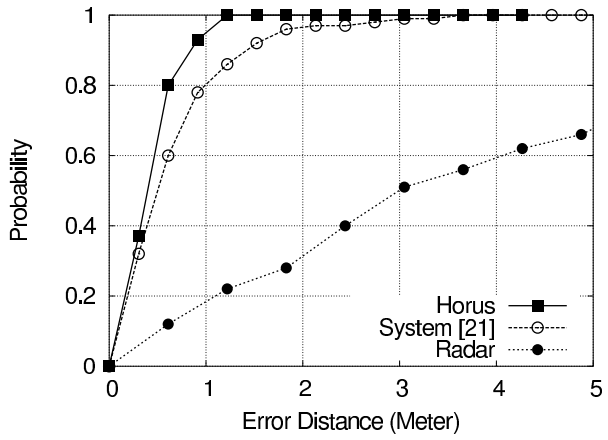
5.3 User Profile

A common assumption of WLAN location determination systems is that the user position follows a uniform distribution over the set of possible locations. Our analysis and experimentation [32] show that knowing the probability distribution of the user position can reduce the number of access points required to obtain a given accuracy. However, with a high density of access points, the performance of the *Horus* system is consistent under different probability distributions for the user position, i.e., the effect of the user profile is not significant with a high density of access points. Systems can use this fact to reduce the energy consumed in the location determination algorithm by not using the user profile in the estimation process.

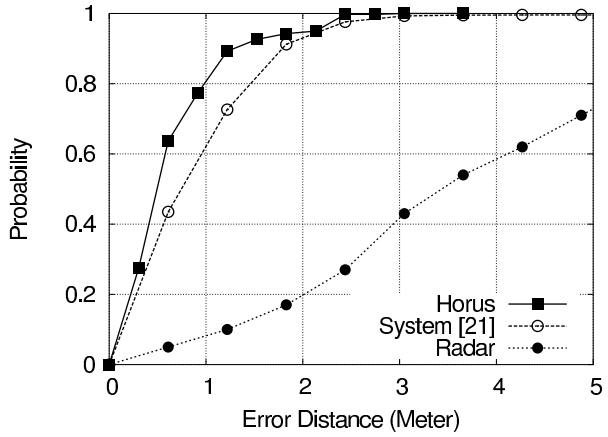
5.4 Effect of Different Hardware

One of the hardware related questions is whether different hardware from different manufacturers are compatible. That is, how does using different APs, mobile devices, or wireless cards affect the accuracy?

Our experience with the *Horus* system shows that the Laptop or PDA used for the calibration has no effect on the accuracy if a different device is used in tracking. APs from different manufacturers can be used without affecting the accuracy since the radio map captures the signature of the AP at each location (note that the second testbed uses mixed types of APs). The 802.11h specifications, however, require APs to have transmission power control (TPC) and dynamic frequency selection (DFS). This presents an open research direction for the current WLAN location determination systems as they assume that the AP transmission power does not change over



(a) First testbed



(b) Second testbed

Figure 18: CDF of the performance of the *Horus* system and the *Radar* and the probabilistic system.

time.

The main factor that may affect the accuracy when changing hardware is the wireless card. Our experience shows that cards from the same manufacture are interchangeable. The good news is that a linear mapping exists between different NICs [13]. Unfortunately, some of the cards in the market are so noisy [27] that with this linear mapping the obtained radio map is not representative of the environment. We found that Orinoco cards and Cisco cards are stable, in terms of signal-strength measurements.

5.5 Operating System Interface

We implemented the *Horus* system under both Linux and Windows. The main functionality we require from the OS is support for issuing scan requests and returning the results. Under the Linux OS, the wireless extensions [1,2] give us a common interface to query different drivers that support that interface. Under the Windows OS, NDIS allowed us to perform the same functions.

Our experience with both systems shows that drivers under Linux conform to the Wireless Extensions APIs better than Windows Drivers do with the NDIS. For example, under the Windows, some cards, like the Cisco card, respond to scans with low frequency (every 2-3 seconds) and return only one AP. We hope that future versions of the driver will have better support for the NDIS interface. Moreover for both systems, better active scanning techniques needs to be developed to reduce the scanning overhead.

6 Related Work

Many systems over the years have tackled the problem of determining and tracking the user position. Examples include GPS [11], wide-area cellular-based systems [24], infrared-based systems [4, 26], ultrasonic-based systems [19], various computer vision systems [16], and physical contact systems [18].

Compared with these systems, WLAN location determination systems are software based (do not require specialized hardware) and may provide more ubiquitous coverage. This feature adds to the value of the wireless data network.

The Daedalus project [14] developed a system for coarse-grained user location. A mobile host estimates its location to be the same as the base station to which it is attached. Therefore, the accuracy of the system is limited by the access point density.

The RADAR system [5] uses the RF signal strength as an indication of the distance between the transmitter and receiver. During an offline phase, the system builds a radio map for the RF signal strength from a fixed number of receivers. During normal operation, the RF signal strength of the mobile client is measured by a set of fixed receivers and is sent to a central controller. The central controller uses a K-nearest approach to determine the location from the radio map that best fits the collected signal strength information.

The Aura system proposed in [22] uses two techniques: pattern matching (PM) and triangulation, mapping and interpolation (TMI). The PM approach is very similar to the RADAR approach. In the TMI technique, the physical position of all the access points in the area needs to be known and a function is required to map signal strength onto distances. They generate a set of training points at each trained position. The interpolation of the training data allows the algorithm to use less training data than the PM approach. During the online phase, they use the approximate function they got from the training data to generate contours and they calculate the intersection between different contours yielding the signal space position of the user. The nearest set of mappings from

the signal-space to the physical space is found by applying a weighted average, based on proximity, to the signal space position.

The Nibble location system, from UCLA, uses a Bayesian network to infer a user location [8]. Their Bayesian network model include nodes for location, noise, and access points (sensors). The signal to noise ratio observed from an access point at a given location is taken as an indication of that location. The system also quantizes the SNR into four levels: high, medium, low, and none. The system stores the joint distribution between all the random variables of the system.

Another system, [21], uses Bayesian inversion to return the location that maximizes the probability of the received signal strength vector. The system stores the signal strength histograms in the radio map and uses them in the online phase to estimate the user location. Yet, another system, [17], applies the same technique to the robotics domain.

The *Horus* system is unique in defining the possible causes of variations in the received signal strength vector and devising techniques to overcome them, namely providing the correlation modeler, correlation handler, continuous space estimator, and small-space compensator modules. Moreover, it reduces the computational requirements of the location determination algorithm by applying location-clustering techniques. This allows the *Horus system* to achieve its goals of high accuracy and low energy consumption.

7 Conclusions

In this paper, we presented the design of the *Horus* system: a WLAN-based location determination system. We approached the problem by identifying the various causes of variations in a wireless channel and developed techniques to overcome them. We also showed the various components of the system and how they interact.

The *Horus* system models the signal strength distributions received from access points using parametric and non-parametric distributions. By exploiting the distributions, the *Horus* system reduces the effect of temporal variations.

We showed that the correlation of the samples from the same access point can be as high as 0.9. Experiments showed that under the independence assumption, as the number of averaged samples increases, the performance degrades. Therefore, we introduced the correlation modeler and handling modules that use an autoregressive model for handling the correlation between samples from the same access point. Using the modified technique, the system average accuracy is enhanced by more than 19% for the first testbed and 11% for the second testbed.

The *Horus* system uses the *Perturbation* technique for handling small-scale variations. The perturbation technique enhances the average distance error by more than 25% for the first testbed and more than 21% for the second testbed. Moreover, the worst-case error is reduced by more than 30% for the two testbeds.

The basic *Horus* technique chooses the estimated location from the discrete set of radio map locations. We described two techniques for allowing continuous-space estimation: the *Center of Mass* technique and the *Time-Averaging* technique. Using the *Center of Mass* technique, the accuracy of the *Horus* system was increased by more than 13% for the first testbed and by more than 6% for the second testbed compared to the basic technique. The *Time-Averaging* technique enhances the performance of the *Horus* system by more than 24% for the first testbed and more than 15% for the second testbed. The two techniques are independent and can be applied together.

We also compared the performance of the *Horus* system to the performance of the *Radar* system. We showed that the average accuracy of the *Horus* system is better than the Radar system by more than 89% for the first testbed and 82% for the second testbed. The worst case error is decreased by more than 93% for the first testbed and 70% for the second testbed. Comparing the probabilistic system in [21] to the *Horus* system shows that the average error is decreased by more than 35% for the first testbed and 27% for the second testbed. The worst case error is decreased by more than 78% for the first testbed and 70% for the second testbed. These results show the effectiveness of the proposed techniques. In terms of computational requirements, the *Horus* system is more efficient by more than an order of magnitude.

The proposed modules are all applicable to any of the current WLAN location determination systems. We show the result of applying the techniques of the *Horus* system to the *Radar* system. The results show that the average distance error is reduced by more than 58% for the first testbed and by more than 54% for the second testbed. The worst case error is decreased by more than 76% for the first testbed and by more than 48% for the second testbed.

As part of our ongoing work we are experimenting with different clustering techniques. Automating the radio-map generation process is a possible research area. The *Horus* system provides an API for location-aware applications and services. We are looking at designing and developing applications and services over the *Horus* system. A possible future extension is to dynamically change the system parameters based on the environment, such as changing the averaging window size as the user speed changes or using a time-dependent radio map. We are also working on the theoretical analysis of different components of the system.

Our experience with the *Horus* system showed that it has achieved its goals of:

- High accuracy: through a probabilistic location determination technique, using a continuous-space estimator, handling the high correlation between samples from the same access point, and the perturbation technique to handle small-scale variations.

- Low computational requirements: through the use of clustering techniques.

The design of Horus also allows it to achieve scalability to large coverage areas, through the use of clustering techniques, and to large number of users, through the distributed implementation on the mobile devices and due to the low energy requirements of the algorithms.

Moreover, the techniques presented in this paper may be applicable to other RF-technologies such as 802.11a, 802.11g, HiperLAN, and BlueTooth.

Acknowledgments

This work was supported in part by the Maryland Information and Network Dynamics (MIND) Laboratory, its founding partner Fujitsu Laboratories of America, and by the Department of Defense through a University of Maryland Institute for Advanced Computer Studies (UMIACS) contract.

Availability

The MAPI API and the Linux device drivers are available for download at [1].

References

[1] http://www.cs.umd.edu/users/moustafa/Downloads.htm.
[2] http://www.hpl.hp.com/personal/Jean_Tourrilhes/.
[3] http://www.orinocowireless.com.
[4] AZUMA, R. Tracking requirements for augmented reality. *Communications of the ACM 36*, 7 (July 1997).
[5] BAHL, P., AND PADMANABHAN, V. N. RADAR: An In-Building RF-based User Location and Tracking System. In *IEEE Infocom 2000* (March 2000), vol. 2, pp. 775–784.
[6] BAHL, P., PADMANABHAN, V. N., AND BALACHANDRAN, A. Enhancements to the RADAR User Location and Tracking System. Tech. Rep. MSR-TR-00-12, Microsoft Research, February 2000.
[7] BOX, G. E. P., JENKINS, G. M., AND REINSEL, G. C. *Time Series Analysis: Forcasting and Control*, third ed. Prentice Hall, 1994.
[8] CASTRO, P., CHIU, P., KREMENEK, T., AND MUNTZ, R. A Probabilistic Location Service for Wireless Network Environments. *Ubiquitous Computing 2001* (September 2001).
[9] CASTRO, P., AND MUNTZ, R. Managing Context for Smart Spaces. *IEEE Personal Communications* (OCTOBER 2000).
[10] CHEN, G., AND KOTZ, D. A Survey of Context-Aware Mobile Computing Research. Tech. Rep. Dartmouth Computer Science Technical Report TR2000-381, 2000.
[11] ENGE, P., AND MISRA, P. Special issue on GPS: The Global Positioning System. *Proceedings of the IEEE* (January 1999), 3–172.
[12] GWON, Y., JAIN, R., AND KAWAHARA, T. Robust Indoor Location Estimation of Stationary and Mobile Users. In *IEEE Infocom* (March 2004).
[13] HAEBERLEN, A., FLANNERY, E., LADD, A., RUDYS, A., WALLACH, D., AND KAVRAKI, L. Practical Robust Localization over Large-Scale 802.11 Wireless Networks. In *10th ACM MOBICOM* (Philadelphia, PA, September 2004).
[14] HODES, T. D., KATZ, R. H., SCHREIBER, E. S., AND ROWE, L. Composable ad hoc mobile services for universal interaction. In *3rd ACM MOBICOM* (September 1997), pp. 1–12.
[15] KRISHNAN, P., KRISHNAKUMAR, A., JU, W. H., MALLOWS, C., AND GANU, S. A System for LEASE: Location Estimation Assisted by Stationary Emitters for Indoor RF Wireless Networks. In *IEEE Infocom* (March 2004).
[16] KRUMM, J., ET AL. Multi-camera multi-person tracking for Easy Living. In *3rd IEEE Int'l Workshop on Visual Surveillance* (Piscataway, NJ, 2000), pp. 3–10.
[17] LADD, A. M., BEKRIS, K., RUDYS, A., MARCEAU, G., KAVRAKI, L. E., AND WALLACH, D. S. Robotics-Based Location Sensing using Wireless Ethernet. In *8th ACM MOBICOM* (Atlanta, GA, September 2002).
[18] ORR, R. J., AND ABOWD, G. D. The Smart Floor: A Mechanism for Natural User Identification and Tracking. In *Conference on Human Factors in Computing Systems (CHI 2000)* (The Hague, Netherlands, April 2000), pp. 1–6.
[19] PRIYANTHA, N. B., CHAKRABORTY, A., AND BALAKRISHNAN, H. The Cricket Location-Support system. In *6th ACM MOBICOM* (Boston, MA, August 2000).
[20] ROOS, T., MYLLYMAKI, P., AND TIRRI, H. A Statistical Modeling Approach to Location Estimation. *IEEE Transactions on Mobile Computing 1*, 1 (January-March 2002), 59–69.
[21] ROOS, T., MYLLYMAKI, P., TIRRI, H., MISIKANGAS, P., AND SIEVANEN, J. A Probabilistic Approach to WLAN User Location Estimation. *International Journal of Wireless Information Networks 9*, 3 (July 2002).
[22] SMAILAGIC, A., SIEWIOREK, D. P., ANHALT, J., KOGAN, D., AND WANG, Y. Location Sensing and Privacy in a Context Aware Computing Environment. *Pervasive Computing* (2001).
[23] STALLINGS, W. *Wireless Communications and Networks*, first ed. Prentice Hall, 2002.
[24] TEKINAY, S. Special issue on Wireless Geolocation Systems and Services. *IEEE Communications Magazine* (April 1998).
[25] THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC. IEEE Standard 802.11 - Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications.
[26] WANT, R., HOPPER, A., FALCO, V., AND GIBBONS, J. The Active Badge Location System. *ACM Transactions on Information Systems 10*, 1 (January 1992), 91–102.
[27] YEO, J., BANERJEE, S., AND AGRAWALA, A. Measuring traffic on the wireless medium: experience and pitfalls. In *Technical Report, CS-TR 4421, Department of Computer Science, University of Maryland, College Park* (Dec. 2002).
[28] YOUSSEF, M. *Horus: A WLAN-Based Indoor Location Determination System*. PhD thesis, University of Maryland at College Park, May 2004. Submitted for SigMobile Dissertation Page.
[29] YOUSSEF, M., ABDALLAH, M., AND AGRAWALA, A. Multivariate Analysis for Probabilistic WLAN Location Determination Systems. In *The Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services* (July 2005).
[30] YOUSSEF, M., AND AGRAWALA, A. Small-Scale Compensation for WLAN Location Determination Systems. In *IEEE WCNC 2003* (March 2003).
[31] YOUSSEF, M., AND AGRAWALA, A. Handling Samples Correlation in the Horus System. In *IEEE Infocom* (March 2004).
[32] YOUSSEF, M., AND AGRAWALA, A. On the Optimality of WLAN Location Determination Systems. In *Communication Networks and Distributed Systems Modeling and Simulation Conference* (January 2004).
[33] YOUSSEF, M., AGRAWALA, A., AND SHANKAR, A. U. WLAN Location Determination via Clustering and Probability Distributions. In *IEEE PerCom 2003* (March 2003).
[34] YOUSSEF, M., AGRAWALA, A., SHANKAR, A. U., AND NOH, S. H. A Probabilistic Clustering-Based Indoor Location Determination System. Tech. Rep. UMIACS-TR 2002-30 and CS-TR 4350, University of Maryland, College Park, March 2002. http://www.cs.umd.edu/Library/TRs/.

Notes

[1]The normalization is used to ensure that the sum of the probabilities of all locations equals one.