# Integrating probabilistic techniques for indoor localization of heterogeneous clients

Antonio J. Ruiz-Ruiz, Oscar Canovas
Department of Computer Engineering
University of Murcia
Murcia, Spain
antonioruiz@um.es, ocanovas@um.es

***Abstract*—This work integrates well-known proposals for indoor location of wireless devices using signal strength on commodity hardware. During the last years, remarkable contributions have been made by the research community to enable location-aware services for indoor scenarios. Location fingerprinting has been proved to be a promising technique of exploiting already existing infrastructures based on IEEE 802.11. In this paper, we combine several approaches in order to design a location estimator which is able to provide good accuracy and performance for different hardware devices, such as laptops, smart phones and wireless tags. Some of the techniques that we have implemented are: error estimation, clustering, probabilistic inference to estimate the location of a device, hidden Markov model, handling of heterogeneous hardware through the least-squares method, and path-restricted location. Our selection has been made after an exhaustive analysis of the existing proposals, pursuing a good balance between accuracy and performance. The experimental testbed has an area of 1050 squared meters, with several corridors, offices and labs. Our main intention is to determine whether this set of techniques can be used to build a ready-to-use location service and to investigate the need for integrating other sensors that would enhance the results. Signal strength will be used to determine a cluster of physical points, or zone, where the device seems to be. Taking into account that we are also working with smart phones, this work has to be considered as a starting point for a multi-sensor architecture able to incorporate accelerometers and cameras for better estimation.**

***Keywords*—Wireless networks, 802.11, probabilistic techniques**

## I. Introduction

The widespread adoption of devices like smart phones is confirming the essential role of location-based applications. For a diverse set of areas including tracking, geographic routing or entertainment, location-sensing systems have been an active research field. Though the Global Positioning System (GPS) is the predominant outdoor positioning system, it suffers from several obstacles blocking the radio signals indoor.

However, wireless devices, like those based on IEEE 802.11, include the hardware necessary to measure the received signal strength intensity (RSSI) of transmitted packets. Using this widely-deployed off-the-shelf hardware, several previous works have demonstrated that a significant accuracy can be obtained by means of location fingerprinting techniques [6], each associated with distinct tradeoffs between accuracy and scalability.

Nowadays, the increasing number of sensors on mobile devices presents new opportunities for localization [1][8][25]. In-built accelerometers or cameras may be useful in inferring coarse-grained user motion and the nature of particular places, respectively. Our final goal is to design an architecture able to fusion data from different sensors in order to provide several levels of accuracy, depending on the application. RSSI plays a major role in our proposal, since it constitutes the primary data to limit the amount of information to be examined. Using fingerprinting methods, we obtain a cluster of physical points where there is a high probability of finding the device. Further refinement, for example by means of images, will be constrained to the data related to that particular cluster.

Therefore, in order to accomplish our work, we have analyzed and implemented several well-known proposals for indoor location based on RSSI fingerprinting. The primary contribution of this paper is the analysis and the integration of several existing techniques for location estimation. For our particular scenario, we wanted to know which technique provides a higher accuracy, how to improve the performance, how to support different devices, and how the scenario may be optimized when path restrictions apply. As we show, we have mainly focused on location techniques based on Bayesian inference. We find especially interesting the obtained balance between accuracy and performance, which constitutes a solid basis to integrate other sensors.

The rest of this paper is structured as follows. Section II gives an overview of the techniques that inspired our work. Section III describes our experimental setup. Section IV presents that way we have managed different devices. Section V presents the results we obtained with different estimation techniques. Section VI introduces the system model based on Markov. Section VII analyzes how we can improve performance in terms of locations per second. Section VIII describes a method for obtaining better accuracy when path is restricted. Section IX depicts that clustering favors integration of multiple sensors. Section X provides information about the accuracy provided by our system when using several devices in real time estimations. Finally, Section XI presents our main remarks and future directions.

## II. Related Work

Indoor positioning is a research field that has been addressed by many different authors and disciplines. Several types of signals (radio, light, sound) and methods have been used to infer location. Each method has specific requirements as to what types of measurements are needed. Different methods make use of the propagation speed of signals in order to collect distance-related measurements. Lateration methods, such as Time-Of-Flight (TOF) [33] and Time-Difference-Of-Arrival (TDOA) [28], estimate positions from distance-related measurements to fixed sensors with known positions. Angle-Of-Arrival (AOA) methods [27] work by observing what angle a signal from a sensor arrives in. Both lateration and angulation require special sensors or hardware to be installed in the covered area. However, most of the pattern recognition methods, like fingerprinting, estimate locations by recognizing position-related patterns in measurements using commodity hardware. Fingerprinting is based on radio maps containing patterns of RSSIs, which are obtained using 802.11, Zigbee, Bluetooth or any other widespread wireless technology. Maps can be manually obtained by collecting signal samples or can be derived from radio propagation models [11][30]. Compared to other types of positioning methods, fingerprinting is not able to provide the centimetre accuracy realized with other proposals, which is not necessary for most location-based applications. As we will see in this paper, we can obtain an accuracy ranging from 0.5 to 3 meters using fingerprinting.

Fingerprinting can be classified into two main categories: deterministic techniques and probabilistic techniques. Deterministic techniques [2][32] represent the signal by a scalar value and use some pattern-matching method to estimate the user location, for example by means of nearest neighbor. However, probabilistic techniques [4][10][35] store information about the signal strength distributions from the access points and represent user positions as probability vectors. For example, one of the main methods to infer location is the Bayesian inference. In this paper we are going to analyze the results obtained using both set of techniques.

On the other hand, there are several options to implement location systems using 802.11, depending on the division of responsibilities between wireless clients, access points, and servers. The three main categories are network-based, client-assisted and client-based, and they differ in who sends out beacons, who makes measurements and who stores the radio map and estimates locations. Network-based systems [2][5][17] offer better support for limited wireless clients, since measurements are collected by access points and forwarded to location servers. Most fingerprinting systems were built client-assisted or client-based [3][29][34], which are more suitable tu support privacy since clients measure RSSI and might estimate locations using the radio map. As we show in this work, our system is both network-based and client-assisted, depending on the type of client we are using (tags, smart phones or laptops).

## III. Experimental environment

### A. Physical environment

The testbed where our experiments were conducted is located on the third floor of our Faculty. The dimension of the testbed is 35 meters by 30 meters, and includes 26 rooms. We have selected 94 cells where the users could be located, spaced out 1.5 meters, according to recommendations made by King et al. in [12].

Our location system works in two phases. First, an off-line or training phase is performed to build the radio signal map and to obtain the signal distribution models. Then, it is during the on-line phase when we are able to estimate the user location.

As we will show, in order to compare the accuracy of our system depending on the number of access points and the number of samples used to build the radio map, we carried out several tests using two different testbed configurations. Initially, we distributed four access points along our dependencies (in Figure 1 they are indicated as red dots). During the corresponding training phase we collected 60 observations[1] at each cell. Later, we carried out a second set of tests, by adding two more access points (blue dots in Figure 1) and collecting 250 observations at each cell. There are several

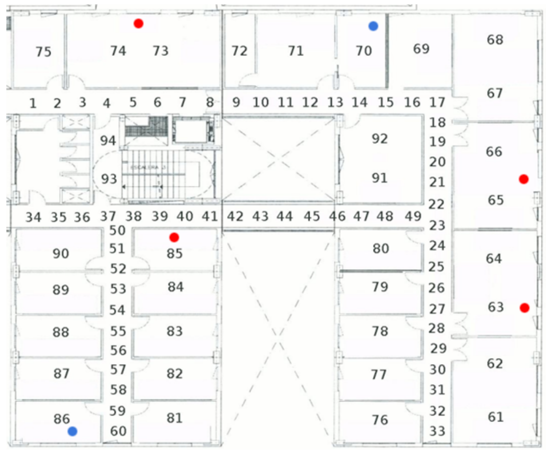


Fig. 1. Experimental environment map

works proposing how to automate the training phase. Chen et al. [7] or LaMarca et al. [21] provide techniques for the automatic generation of fingerprinting maps. The former approach was developed using RFID sensors, while the latter studies the pattern of WiFi signals. Though we have not integrated these proposals within our testbed, as it is relatively small, they should be considered in order to improve the scalability for bigger scenarios.

### B. Hardware and software

Our experiments were carried out using several hardware devices. The training observations were captured with an Asus Eee 1201 laptop with a Realtek TRL8191SE Wireless LAN 802.11n card. In addition, during the online phase we have also used a HP iPAD hx2400 series using Windows Mobile 2005, a HTC Desire smart phone with Android and Aeroscout T2 wireless tags. With the exception of wireless tags, we developed the appropriate software client for each device in order to collect RSSIs and to send them to a repository. Applications were programmed in C++ and Java, depending on the requirements imposed by each device. Furthermore, we implemented several estimation techniques in Java. According to the different nature of our devices, the system was designed to support both a client-assisted and a network-based infrastructure, that is, RSSI can be collected by the end-user devices or by the 802.11 access points.

We have used Linksys WRT54G access points with 802.11abg support. Their locations were chosen so as to provide consistent coverage throughout the entire scenario. In addition, the firmware was modified to work in monitor mode, thus providing support for special devices with limited computing resources, like the already mentioned wireless tags.

[1]An observation is a set of RSSIs collected from all the reachable access points at the same cell and during a particular scan.

## IV. CALIBRATION

Besides accuracy or performance, one of the imposed requirements of our proposal is the support for heterogeneous hardware clients. Due to the wide range of devices on the market, we do not want to restrict the performance of our location system to specific hardware. However, different devices provide different intensity readings, depending on antennas, transmission power and many other factors. Gwon and Jain proposed in [9] a calibration-free location algorithm that eliminates offline RSSI measurements. However, mean error distance is about 5.4 meters. Several proposal such as [10][14][15] provide calibration mechanisms improving this distance error.

On the one hand, Haeberlen et al. [10] propose a calibration function based on the following linear relationship:

$$c(i) = c_1 \cdot i + c_2 \tag{1}$$

where $i$ is the observed signal intensity value by the new device and c($i$) is the value that would have been observed by the training device. Computing the least-squares fit between the observations obtained by the new device on the calibration cells[2] and the corresponding values from the sensor map, we can obtain the parameters $c_1$ and $c_2$. The authors proposed several methods for manual, quasi-automatic and automatic calibration. On the other had, Kjaergaard [15] proposes a Hyperbolic location fingerprinting to solve the signal-strength difference problem and an automatic technique [14] for adapting an indoor localization system based on signal strength to the specific hardware and software of a wireless network client. In relation to our scenario, the best calibration parameters were obtained with the proposal from Haeberlen et al. As Figure 2 shows, unadjusted RSSIs do not fit to the training laptop signal. Nevertheless, once we have calibrated all the devices, Figure 3, signals are quite similar.

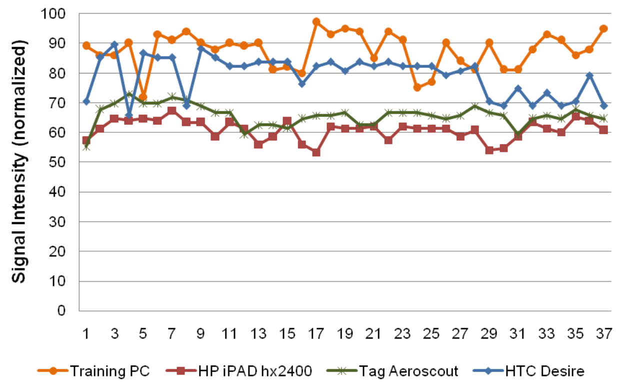


Fig. 2. Signal intensity before calibration

## V. ANALYSIS OF ESTIMATION METHODS

As we mentioned before, our first intention was to explore the accuracy of the system as we varied the amount of access points and the number of observations used to build the radio maps. We have accomplished several tests using two different techniques in order to compare their results. On the one

[2]A set of cells previously established to get a heterogeneous set of observations.

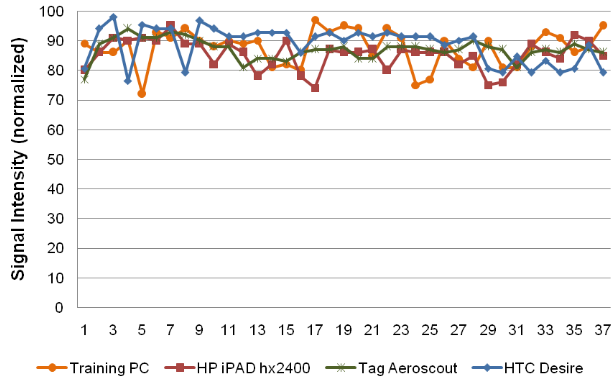


Fig. 3. Signal intensity after calibration

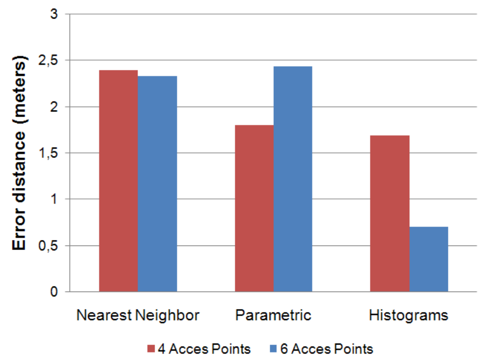


Fig. 4. Mean error for different configurations

hand we have used a deterministic technique based on nearest neighbor and Euclidean distance of RSSIs. We implemented the proposal from [2] and it is able to estimate location with a mean error distance between 2 and 3 meters, about the size of a typical office room. On the other hand, we have also represented the position as a probability distribution using a Bayesian inference technique discussed in [10][13][20]. This algorithm estimates posterior distributions and can be applied in the case of sensors that have non-Gaussian noise distributions, such as our signal strength sensor.

We have to take into account that signal propagation in an indoor environment is noisy since it is affected by reflection, diffraction, and scattering of radio waves caused by structures within the building. These dynamic environmental influences can cause the observed signal strength to vary considerably and this makes very difficult to estimate the location using a single signal observation. So, using historical information about the previous locations of the user, we may get better results by means of probabilistic methods, as you can see in Figure 4. Hereinafter, we will focus our tests on probabilistic methods since they offer several possibilities to improve the performance and accuracy of our system.

Being $C = \{c_1, .., c_m\}$ the set of cells that make up the finite space state and $\pi$ a probability distribution vector over each cell, for each observation $O_j$, the probability to take a measurement from the access point $a_\beta$ at reference cell $c_i$ with a signal strength $\lambda_\beta$ can be expressed by the conditional probability:

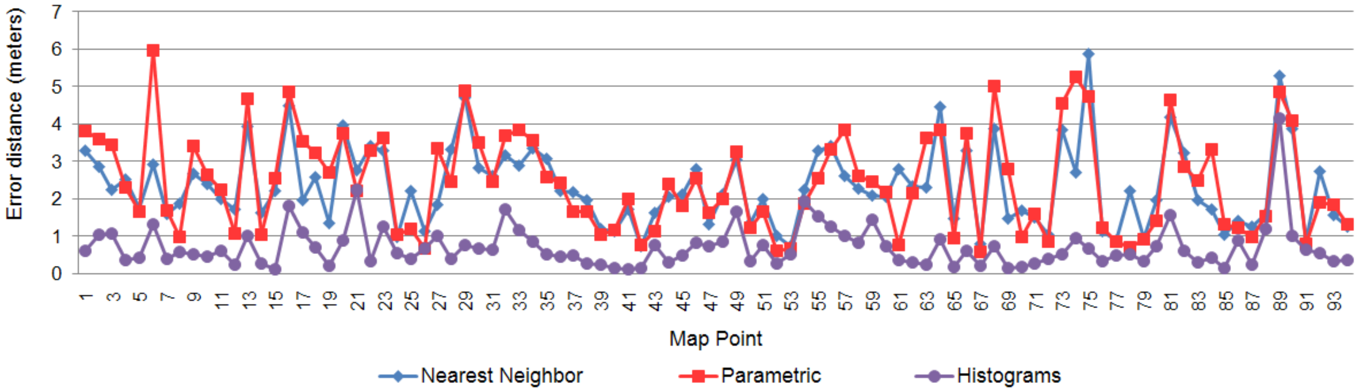


Fig. 5. Mean error at each cell

$$Pr(O_j|c_i) = \prod_{\beta=1}^{n} Pr(\lambda_\beta|a_\beta, c_i) \tag{2}$$

These conditional probabilities are used to update the probability vector $\pi$ by applying Bayes' Rule:

$$\pi_i' = \frac{\pi_i Pr(O_j|c_i)}{\sum_{\alpha=1}^{m}(\pi_\alpha Pr(O_j|c_\alpha))} \tag{3}$$

We also compare taking a Gaussian fit of signal strength to using the full histogram of signal strength. Parametric-based distribution is built by modeling the signal intensity as a normal distribution defined at each cell and for every base station by its mean and standard deviation. Histograms represent the sensor model explicitly.

As you can see in Figure 4, there are some techniques that perform better using four access points but it is clear that the result obtained using the histogram-based probabilistic method and six access points provides the higher accuracy. WiFi signals have a very unpredictable behavior so the main cause of histograms to perform better than parametric is because signals are not fit to a parametric based probability distribution, therefore using a histogram based probability distribution it is easier to obtain a correct probability estimation. Therefore, we are going to analyze the results obtained from this configuration of 6 base stations and 250 training observations. Additionally, in order to provide more detailed information, Figure 5 shows the mean error for each cell after estimating the user position. The shape of the histogram sometimes is particularly sensitive to the number of bins. In order to find the right number of bins there are several aspects that we have to take into account. If the bins are too wide, important information might get omitted. However, if the bins are too narrow, what seems to be meaningful information may be due to random variations that show up because of the small range in a bin. In conclusion, there is no best number of bins since different bin sizes can reveal different features of the data. So, to determine whether the bin width is set to an appropriate size, different bin widths should be tested to determine the sensitivity of the histogram shape with respect to it size. It is worth mentioning that every bin is considered to contain at least one sample, in order to discard zero probability.

Figure 6 shows the error distance obtained using a distribution model based on histograms varying the number of bins. When a greater number of bins is used, accuracy improves since each bin is formed by a lower range of samples, giving more importance to those that are more representative, and lowering the error down to 0.7 meters.

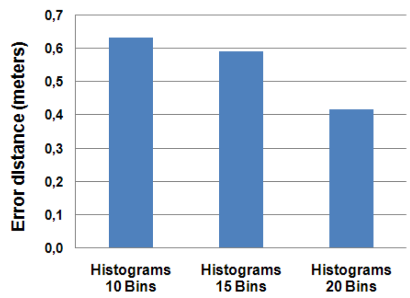


Fig. 6. Accuracy depending on bins

## VI. System Model

Until now, the way we have used RSSIs is not rich enough to track the location of a mobile device since we should include additional information to infer motion. Considering that, at this stage, we have not integrated inertial sensors (like an accelerometer) into our system, we might take into account several proposals integrating sensor readings over time to track mobile users. Krumm and Horvitz [19] measure the variance of the signal strength of the strongest access point to infer whether the user is still or moving. Muthukrishnan et al. [26] presents an inference system based on euclidean distances between signals. Despite both proposals are good motion estimators, we implemented an algorithm which takes the output of the estimation method as a stream of observations and stabilizes the distribution by modeling the usual behavior of users within our scenario.

This algorithm is based on a Hidden Markov Model (HMM) [16][31] and it has been used in several proposal such as [10][20], where it has been proved as a good system model. Given a user position, this method spreads probability over those points that are reachable during the next interval of time. The performance we can obtain from HMM depends on the design of the Markov chain, which encodes assumptions about how the user can move from state to state, referring to a state

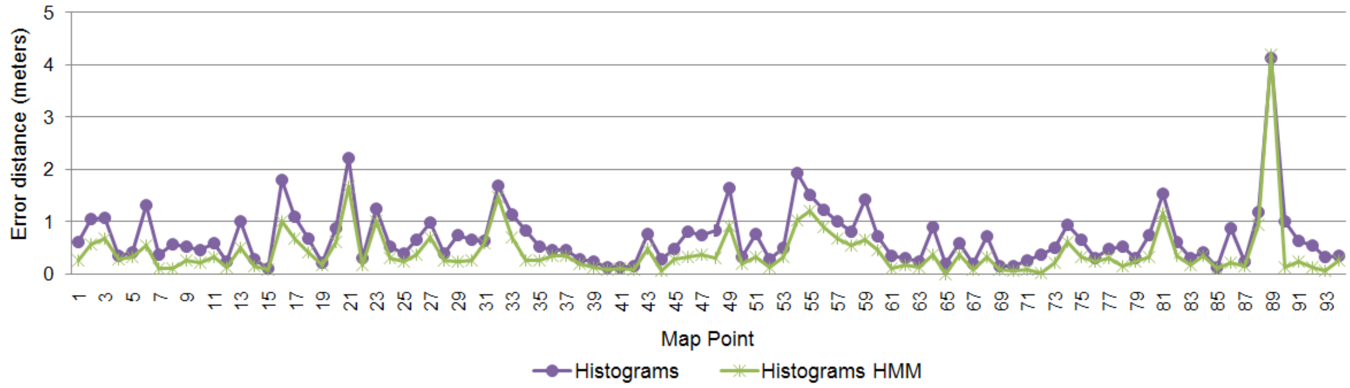


Fig. 7. Histogram-based position estimation error with HMM at each cell

as a cell in our scenario. This chain specifies the probability of remaining still at a cell or moving to a nearby one. One of the more critical points of using HMM is to define the matrix describing how the system being modeled evolves with time. In order to create the chain that best fits to our environment, we took into account several considerations. We have designed a matrix $A$ that encodes the HMM chain considering the normal behavior of users around our scenario. As we saw in equation 3, if $\pi$ is a probability distribution vector over $S$, then $\pi' = A\pi$ will be the probability distribution vector at the following instant time.

Our scenario is mainly static, since it will not suffer relevant changes over time. Thus, definition of $A$ will be carried out only once. Additionally, taking into account that our scenario is mainly formed by offices and laboratories where people usually stays static, probability of moving should be lower than remaining at the same point. Also, we assume that people do not exceed a speed of 2 meters per second. Once the HMM matrix is designed, we carried out some tests using histograms and 20 bins in order to check whether better results are obtained. Figure 7 confirms that the best location estimation technique is based on histograms and including HMM, since it usually reduces the error down to 0.41 meters on average.

## VII. Performance Analysis

In order to reduce the computational cost of our location estimation system, to minimize the number of operations per location estimation, and thus to get a greater number of locations per second, we studied the contributions made by Youssef et al. in [34][35], paying more attention to the *Incremental Triangulation (IT)* clustering technique. This technique is based on the idea that the strongest signals come from the nearest access points. Therefore we can assume that those signals are more stable and more reliable. So, when we estimate the location of a user using the received signals ordered by their intensity, it means that we evaluate the signals ordered by their usefulness. During the location estimation process we use the access points iteratively, one after the other, then starting with the first access point. Therefore, we restrict our search space to the cells covered by this access point. In reduced scenarios, like ours, this might not suppose any important improvement since we do not discard so many cells. Nevertheless, in bigger spaces with a higher number of access points, like an airport or a hospital, this can suppose a huge time reduction. As it is presented in [34], given a sequence of observations from each access point, we start by sorting the access points in descending order according to received intensity. If the probability of the most probable location is meaningfully higher (threshold) than the probability of the second most probable location, we return that most probable location as our location estimation, and we do not take into consideration the next access points. If we come back to equation 2, when we calculate the probability of being at each cell we reduce the number $n$ of access points.

Before analyzing the *IT* results, we would like to note that our main intention is to compare performance in relative terms. However, for the sake of completeness we provide the details of the used computing platform: CPU Pentium(R) Dual-Core E5300(2M Cache, 2.60 GHz, 800 MHz FSB), 2GB RAM memory and Windows XP Professional. Table I summarizes the obtained results applying this algorithm. For lower threshold values (1$^{st}$ column), the decision is taken quickly after examining a small number of access points, no more than 3 access points on average (3$^{rd}$ column). As the threshold value increases, a higher number of access points has to be evaluated. Consequently, as the number of considered access points increases, the number of operations increases, which reduces the number of location estimations per second (4$^{th}$ column), but the average accuracy increases (2$^{nd}$ column). We have carried out this test using the histogram-based probability distribution technique with HMM. As we can see, using *IT* we can obtain similar results in terms of accuracy to those obtained previously. Using a threshold of 0.4 we are able to reduce the number of analyzed access points, from 6 to 3.5 on average. This involves a speed-up of 15.73% on average. System accuracy is adversely affected by a few centimeters, from an error distance of 0.41 m. to 0.57 m., what is acceptable to estimate the location of a user into our

| IT Threshold | Error distance | Access Points | Locations/sec | Optimization Locations/sec | Improvement |
|---|---|---|---|---|---|
| No IT | 0,419 | 6,0 | 1772 | - | - |
| 0.1 | 1,224 | 2,3 | 2645 | 2670 | 33,64% |
| 0.2 | 0,854 | 2,8 | 2219 | 2437 | 27,31% |
| 0.3 | 0,668 | 3,2 | 2240 | 2337 | 24,19% |
| 0.4 | 0,571 | 3,5 | 2102 | 2180 | 18,72% |

Tabla I
PERFORMANCE ANALYSIS

scenario.

Despite we obtained a good improvement with *IT*, we designed a further optimization. This optimization tries to improve system performance without compromising accuracy. We avoid to evaluate unnecessary cells (at each iteration of the *IT* algorithm) where probability is meaningfully low. For example, if using the signal received by one of the strongest access points the cell probability is under a threshold, we will not evaluate this cell again using the next access points. This threshold is determined by the minimum density of histogram distributions.

This optimization reduces the required cells in vector $\pi$ (equation 3), and therefore the number of locations per second increases. The 5$^{th}$ column in table I shows the results of applying this optimization to the *IT* technique, always offering better results. The 6$^{th}$ column shows the speed-up of using *IT* in relation to the absence of any improvement (1$^{st}$ row). As we can see, we can improve our performance up to 18.72% without having an adverse effect on accuracy.

## VIII. Path-Restricted Location

There are scenarios where users have restricted access to some dependencies[3]. To reflect these restrictions, we have to discard those points where the user cannot be located. One approach is to label each cell indicating its access level. Since our scenario is within a Faculty, we have conducted some tests assuming two different types of users: professors and students. Usually students will move primarily along the corridors, so the cells belonging to those dependencies are labeled as public. The rest of dependencies are labeled as private, and only professors can gain access to them.

Consequently, we propose another optimization with the aim of minimizing the number of cells where a user could be located. We called it Path-Restricted Location (PRL). IT-based and PRL optimizations may be complementaries, but we prefer to show them in an independent way. We carried out some tests assuming that the user was a student and therefore he had no access to private rooms. To carry out these tests we have used the histogram-based probability distribution with 20 bins and HMM. The path we have covered during the test goes from cell 33, through 49 and 50, to cell 60 (it can be see in Figure 1. As you can see in Figure 8, PRL still improves accuracy due to average error is reduced to 0.31 meters. This makes sense if we think that the number of cells analyzed is lower than in the previous tests, around 33%, discarding the possibility of being in private dependencies.

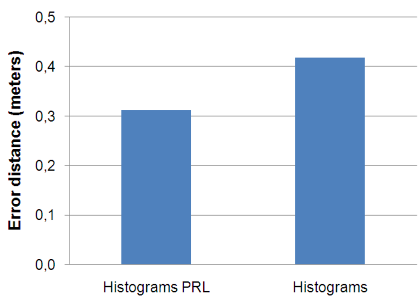


Fig. 8. Accuracy using PRL

[3]Referred to a set of cells that form a corridor, a laboratory, an office, etc.

## IX. Clustering

Clustering techniques have been applied in several ways. On the one hand, Youssef et al. [35] propose the Joint Clustering algorithm that uses joint probability distributions of the RSSI of different access points to find the most probable user location. They try to reduce the computational cost by grouping the cells into clusters according to the access points providing coverage, at the expense of loosing accuracy. Each cell belonging to a cluster has in common the order in which signals are received, according to their intensity, from those of the strongest visible access points $q$ choose for clustering. This technique is further applied during the online phase, using the $q$ strongest access points to select the cluster of cells that will be analyzed to determine the most probable location. A similar proposal was made by Krumm and Hinckley [18] to obtain a coarse-grained proximity between users.

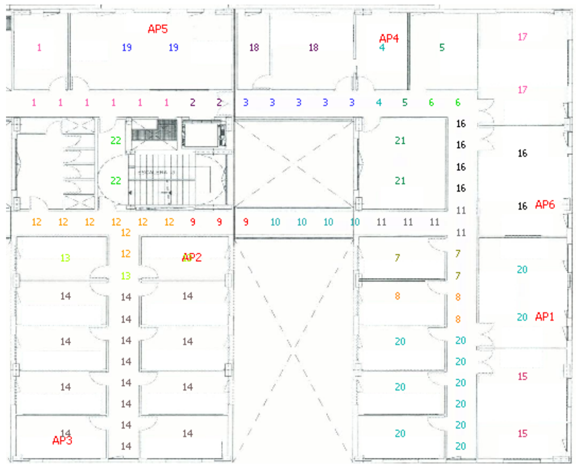


Fig. 9. Clusters

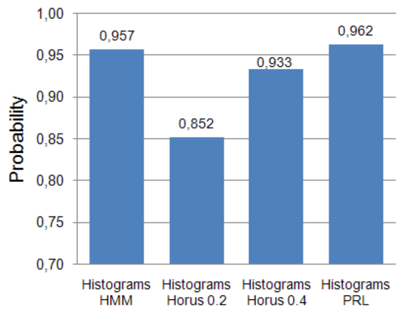


Fig. 10. Cluster hit probability

On the other hand, Lemelson et al. propose in [22] four algorithms to estimate the position error that is inherent to 802.11-based positioning systems. One of them, Fingerprint Clustering algorithm, makes use of the training RSSIs to find clusters. It is based on the idea that the signal collected in nearby cells tend to cover only a limited range of the possible values. So, if we find an area with similar signal properties, the position estimation error will be higher because of the number of similar fingerprints is high. However, we have a high probability to estimate the location of a user within those areas, with a maximum error distance less than or equal to distance between the two furthest points from the cluster in which user has been located. We are especially interested in this second proposal, since it will help us to define medium-sized zones, joining adjacent clusters, where WiFi-based location might be further refined by means of other sensors. We can define the clusters once the training phase have been finished, and it does not require a high computational effort.

In order to show how this proposal can be applied to our interest, we have calculated the clusters, shown in Figure 9, and then we have carried out several tests. The clusters hit probabilities obtained from those tests are shown in Figure 10. As you can see, most of the already-analyzed techniques obtain a high cluster hit percentage, up to 93%. These results will have important implications for our future work.

## X. Tracking tests

Previous sections have presented analytic results that were calculated using the observations obtained during the training phase as inputs to our location system. Nevertheless, in order to validate our location system, we realized that we have to demonstrate its accuracy carrying out real time test, i.e. trying to estimate a user location using the RSSI captured while the user is walking around the scenario. The estimation method we used is also histogram-based with HMM.

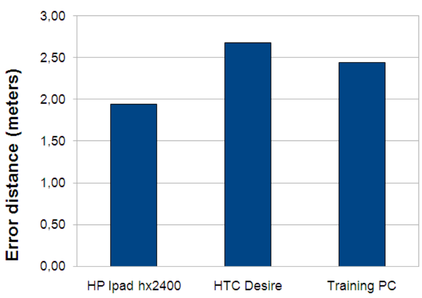


Fig. 11. Mean error while tracking

Therefore, we walked around our scenario carrying three different devices: PDA, smart phone and laptop (the one used during the training phase). We took one observation at each cell along the path. Figure 11 presents the results on average. There are several conclusions we can derive from those tests. Firstly, we show that the error distance is lower than 2.5 meters on average. This accuracy is quite similar to that obtained in referenced proposals. Also we have to take into account that we move through a complex environment made of several materials and with moving persons. Secondly, both Figures highlight the importance of the calibration process, since the results obtained with both the PDA and the smart phone are very similar to those obtained with the training laptop.

## XI. Summary and Future work

In this paper, we have analyzed widely known research works for indoor location in order to evaluate them and to design a system for heterogeneous clients. This heterogeneity is given by the possibility of using a wide range of devices. The results shown in the calibration section allow us to be optimistic about it, as we have been able to adapt the signals collected by different devices to those of the training laptop.

We modeled the signal strength distributions received from access points using deterministic and probabilistic techniques (by means of parametric and non-parametric based probability distributions). This allowed us to demonstrate that probabilistic methods fit better to signal behavior since they reduces the effect of temporal variations. Therefore, we decided to use the Bayesian inference technique using a 20 bins histogram-based probability distribution as default algorithm for our next experimental tests, because it reduces the error down to 0.7 meters on average. Thereinafter, we have added several optimizations to our location estimation system that offer better accuracy and performance results.

The integration of HMM, discussed in section VI, improves the accuracy of our system. In the absence of inertial sensors, the HMM allows us to estimate user movements. Using this widely-deployed technique we have improved the accuracy of the system up to 40% on average, reducing the error to 0.41 meters.

In addition, we have analyzed several proposals in order to improve the system performance, and we have carried out our own tests to validate their benefits within our scenario. On the one hand, in section VII we have analyzed the *Incremental Triangulation (IT)* clustering technique, that allows us to reduce the number of required operations to infer the user location. Furthermore, we proposed an optimization to *IT* that improves the system performance up to 18.72% without having an adverse effect on accuracy. On the other hand, test results from section VIII, where we discussed the Path-Restricted Location optimization, show that using environment information we avoid the evaluation of unnecessary cells, and we are able to improve the average accuracy error. This results demonstrate the need for an appropriate context model, whose design we are already defining.

From previous sections, we are able to state the degree of accuracy a WiFi sensor can offer. Indeed, considering a cluster level accuracy around 93% on average, we will concentrate our efforts on integrating several sensors within our location system. Some proposals, like Azizyan et al. [1], introduced several methods to join data from different sensors of existing smart phones. Our future direction to exploit the sensor fusion goes in a different way.

Using the camera of the smart phone and the Scale-Invariant Feature Transform (or SIFT) algorithm proposed by Lowe [23][24] we are able to detect and describe local features in images. Some initial tests show that combining the information from both sensors, WiFi and camera, we are able get better accuracy. However, the main drawback of using images is the elevated computational cost. This gets more importance when we need to locate a user in large scenarios, since the number of images to analyze is excessive, involving serious scalability problems. As we have previously mentioned, the use of clustering algorithms, such as Fingerprint Clustering, reduces the number of cells to be analyzed to those contained in the cluster. Once we have used RSSIs to determine the cluster, we can process a reduced set of images to perform a fine-grained localization, improving scalability.

Finally, in order to check if our system works properly in real time conditions we carried out some tracking test, discussed in section X. After analysing this tests results we can get some conclusions. We confirm that selected position estimation technique gets good results to locate a user in motion. Moreover, we demonstrate that it has been able to adapt the signals collected by different devices since all of them have similar behaviour, so it means that calibration works properly.

Summarizing, once we are able to decide which RSSI based technique obtains the best results, we are experimenting in order to design a multi sensor location estimation using as much information from sensor as we can as well as making use of available context information.

## REFERENCES


[1] M. Azizyan, I. Constandache, and R. Roy Choudhury. SurroundSense: Mobile Phone Localization via Ambience Fingerprinting. In Proceedings of the 15th Annual International Conference on Mobile Computing and Networking, MOBICOM, pages 261-272, Beijing, China, September 2009.

[2] P. Bahl and V. N. Padmanabhan. RADAR: An In-Building RF-based User Location and Tracking System. In Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies, volume 2, pages 775-784, Tel Aviv, Israel, March 2000. IEEE INFOCOM.

[3] G. Borriello, M. Chalmers, A. LaMarca, and P. Nixon. Delivering Real-World Ubiquitous Location Systems. Communications of the ACM, 48:36-41, March 2005.

[4] P. Castro, P. Chiu, T. Kremenek, and R. R. Muntz. A Probabilistic Room Location Service for Wireless Networked Environments. In Proceedings of the 3rd International Conference on Ubiquitous Computing, UbiComp, pages 18-34, Atlanta, Georgia, USA, 2001.

[5] R. Chandra, J. Padhye, A. Wolman, and B. Zill. A Location-Based Management System for Enterprise Wireless LANs. Technical report, Microsoft Research, 2007.

[6] G. Chandrasekaran, M. A. Ergin, J. Yang, S. Liu, Y. Chen, M. Gruteser, and R. P. Martin. Empirical Evaluation of the Limits on Localization Using Signal Strength. In Proceedings of the 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, SECON, pages 333-341, Rome, Italy, June 2009.

[7] Y.-C. Chen, J.-R. Chiang, H.-h. Chu, P. Huang, and A. W. Tsui. Sensor-Assisted Wi-Fi Indoor Location System for Adapting to Environmental Dynamics. In Proceedings of the 8th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems, MSWiM, pages 118-125, Montreal, Quebec, Canada, 2005.

[8] S. Diverdi and T. HÃ¶llerer. Groundcam: A Tracking Modality for Mobile Mixed Reality. In Proceedings of IEEE Virtual Reality Conference, VR, pages 75-82, 2007.

[9] Y. Gwon and R. Jain. Error Characteristics and Calibration-free Techniques for Wireless LAN-based Location Estimation. In Proceedings of the 2nd International Workshop on Mobility Management and Wireless Access Protocols, MobiWac, pages 2-9, Philadelphia, PA, USA, 2004.

[10] A. Haeberlen, E. Flannery, A. M. Ladd, A. Rudys, D. S. Wallach, and L. E. Kavraki. Practical Robust Localization over Large-Scale 802.11 Wireless Networks. In Proceedings of the 10th Annual International Conference on Mobile Computing and Networking, MOBICOM, pages 70-84, Philadelphia, PA, USA, 2004.

[11] H. Hashemi. Indoor Radio Propagation Channel. In Proceedings of the IEEE, volume 81,7, pages 943-968, Washington, DC, USA, August 1993. IEEE Computer Society.

[12] T. King, T. Haenselmann, and W. Effelsberg. Deployment, calibration, and Measurement Factors for Position Errors in 802.11-based Indoor Positioning Systems. In Proceedings of the 3rd International Conference on Location-and Context-Awareness, LoCA, pages 17-34, Oberpfaoenhofen, Germany, 2007.

[13] T. King, S. Kopf, T. Haenselmann, C. Lubberger, and W. Effelsberg. COMPASS: A Probabilistic Indoor Positioning System Based on 802.11 and Digital Compasses. In Proceedings of the 1st International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization, WiNTECH, pages 34-40, Los Angeles, CA, USA, 2006.

[14] M. Kjaergaard. Automatic Mitigation of Sensor Variations for Signal Strength Based Location Systems. In M. Hazas, J. Krumm, and T. Strang, editors, Location and Context Awareness, volume 3987 of Lecture Notes in Computer Science, pages 30-47. Springer Berlin / Heidelberg, 2006.

[15] M. B. Kjaergaard and C. V. Munk. Hyperbolic Location Fingerprinting: A Calibration-Free Solution for Handling Diferences in Signal Strength. In Proceedings of the 2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications, PerCom, pages 110-116, Hong Kong, China, 2008. IEEE Computer Society.

[16] K. Konolige and K. Chou. Markov Localization using Correlation. In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, IJCAI, pages 1154-1159, Stockholm, Sweden, 1999. Morgan Kaufmann Publishers Inc.

[17] P. Krishnan, A. Krishnakumar, W.-H. Ju, C. Mallows, and S. Ganu. A System for LEASE: Location Estimation Assisted by Stationary Emitters for Indoor RF Wireless Networks. In Proceedings of the Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, IEEE INFOCOM, Hong Kong, China, 2004.

[18] J. Krumm and K. Hinckley. The NearMe Wireless Proximity Server. In Proceedings of the Sixth International Conference on Ubiquitous Computing, UbiComp, pages 283-300, Nottingham, England, September 2004. Springer.

[19] J. Krumm and E. Horvitz. LOCADIO: Inferring Motion and Location from Wi-Fi Signal Strengths. In First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, Mobiquitous, pages 4-13, Boston, MA, USA, August 2004.

[20] A. M. Ladd, K. E. Bekris, A. Rudys, G. Marceau, L. E. Kavraki, and D. S. Wallach. Robotics-Based Location Sensing Using Wireless Ethernet. In Proceedings of the 8th Annual International Conference on Mobile Computing and Networking, MOBICOM, pages 227-238, Atlanta, Georgia, USA, 2002.

[21] A. Lamarca, J. Hightower, I. Smith, and S. Consolvo. Self-Mapping in 802.11 Location Systems. In Proceedings of the Seventh International Conference on Ubiquitous Computing (Ubicomp 2005), Lecture Notes in Computer Science, pages 87-104, Tokyo, Japan, 2005.

[22] H. Lemelson, M. B. Kjaergaard, R. Hansen, and T. King. Error Estimation for Indoor 802.11 Location Fingerprinting. In Proceedings of the 4th International Symposium on Location and Context Awareness, LoCA '09, pages 138-155, Tokyo, Japan, 2009.

[23] D. G. Lowe. Object Recognition from Local Scale-Invariant Features. In Proceedings of the International Conference on Computer Vision, Volume 2, ICCV '99, pages 1150-1157, Kerkyra, Greece, September 1999. IEEE Computer Society.

[24] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision, Volume 60, pages 91-110, November 2004.

[25] A. Mulloni, D. Wagner, I. Barakonyi, and D. Schmalstieg. Indoor Positioning and Navigation with Camera Phones. IEEE Pervasive Computing, 8:22-31, 2009.

[26] K. Muthukrishnan, M. Lijding, N. Meratnia, and P. Havinga. Sensing motion using spectral and spatial analysis of WLAN RSSI. In Proceedings of the 2nd European Conference on Smart Sensing and Context, EuroSSC, pages 62-76, Kendal, England, 2007.

[27] D. Nath and B. Niculescu. Vor base stations for indoor 802.11 positioning. In Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM, pages 58-69, 2004.

[28] R. Ogino, A. Tamaki, T. Uta, T. Matsuzawa, N. Kalo, and T. Yamasaki. TDOA location system for IEEE 802.11b WLAN. In IEEE Wireless Communications and Networking Conference, WCNC, pages 2338-2343, 2005.

[29] S. Pandey, F. Anjum, B. Kim, and P. Agrawal. A Low-cost Robust Localization Scheme for WLAN. In Proceedings of the 2nd Annual International Workshop on Wireless Internet, WICON '06, Boston, Massachusetts, 2006.

[30] T. S. Rappaport. Wireless Communications-Principles and Practice. Prentice Hall Communications Engineering and Emerging Technologies Series, 2 edition, 2005.

[31] W. B. S. Thrun and D. Fox. Probabilistic Robotics. The MIT Press, 2005.

[32] A. Smailagic, D. P. Siewiorek, J. Anhalt, D. Kogan, and Y. Wang. Location Sensing and Privacy in a Context Aware Computing Environment. IEEE Wireless Communications, 9:10-17, 2001.

[33] A. Ward, A. Jones, and A. Hopper. A New Location Technique for the Active Office. IEEE Personal Communications, 4:42-47, 1997.

[34] M. Youssef and A. Agrawala. The Horus WLAN Location Determination System. In Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services, MobiSys, pages 205-218, Seattle, Washington, 2005.

[35] M. A. Youssef, A. Agrawala, and A. U. Shankar. WLAN Location Determination via Clustering and Probability Distributions. In Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, PERCOM, pages 143-150, Dallas-Fort Worth, Texas, USA, March 2003.