Automatic Identification of Fingerprint Regions for Quick and Reliable Location Estimation

Hendrik Lemelson, Sascha Schnaufer, and Wolfgang Effelsberg
Department of Computer Science IV
University of Mannheim, Germany
{lemelson,schnaufer,effelsberg}@informatik.uni-mannheim.de

***Abstract*—One of the drawbacks of location fingerprinting systems is the effort that is necessary to set up and update the fingerprint database. In this paper, we propose a novel approach to significantly reduce this effort. We split the area of operation into a grid of quadratic cells and then combine these cells into larger regions of similar signal properties using a clustering algorithm and a novel similarity measure. Thus, less training data is required, and it can be collected in a more efficient way: We move through the area of operation on predefined trajectories and interpolate the approximate position for each measurement. In addition, by storing only one fingerprint for each region, we reduce the computational requirements of the location fingerprinting algorithm considerably. Since the radio measurements are quite similar in such a region, it is hard to estimate the exact location within the region; thus we do not lose much accuracy by clustering. An evaluation of our approach shows that it achieves an accuracy that is sufficient for most location-based services and at the same time reduces the effort for the collection of the training data to a mere walk of the area of operation.**

I. Introduction

During the last few years, mobile devices have become more powerful in terms of processing power, available memory and battery runtime, and also the price for high bandwidth mobile data communication has dropped to reasonable values. These developments have fostered the rise of a novel class of applications called *Location-based Services* (LBSs). They take information about the user's current whereabouts into account to improve their delivered services.

The major source for location information used by LBSs nowadays is the satellite-based *Global Positioning System* (GPS) [1]. But whereas GPS is easy to use and offers a good positioning accuracy and reliability in outdoor open-sky environments, it performs poorly in situations where the mobile device has no line of sight to at least three GPS satellites. In these situations, like for example inside of buildings or in the street canyons of large metropolitan cities, GPS either delivers imprecise location estimates or even no estimates at all.

To be able to estimate a user's position under such adverse conditions, several alternative techniques have been proposed during the last few years (e.g., RightSPOT [2], Cricket [3], PlaceLab [4], Active Badge [5]). But most of these systems either require specialized hardware or dedicated infrastructure to estimate a user's position or only offer coarse positioning accuracy. Position estimation with *802.11* [6] and *location fingerprinting* (LF) offers a promising alternative for estimating a user's position in cases where GPS is not available:

- With 802.11 and LF, an average positioning error of less than two meters can be achieved. This makes the technique suitable for most LBSs.
- Almost all modern mobile devices offer an 802.11 interface.
- It is possible to receive the signals of at least a few 802.11 access points in most places where people live or work in the developed world [7].
- LF can be used in situations where other systems – especially GPS – fail.

While positioning systems based on location fingerprinting offer a viable alternative to other positioning techniques, they still have one major drawback: To achieve an accuracy beyond a coarse proximity estimation, a dense grid of reference spots is needed, and a high number of measurements have to be collected at each reference spot [8]. As a result, a high effort is necessary to set up such a positioning system as well as to keep it up-to-date.

Our contributions in this paper are twofold: First, we introduce a novel location fingerprinting algorithm that aims at significantly reducing the effort needed for setup and operation while retaining a good positioning accuracy and reliability. It identifies regions of similar signal properties and uses these regions for location estimation. With our approach, the training data no longer has to be collected at pre-defined reference spots. Instead, it can be collected while moving through the area of operation at walking speed. This is a great improvement over grid-based systems. In addition, with our approach, only one fingerprint is created per region. This reduces the computational load when computing a position estimate as the search database is much smaller compared to a grid-based approach. By identifying regions of similar signal properties, our algorithm achieves a high reliability of the location estimates.

Second, we provide an extensive evaluation of our novel LF algorithm by means of emulation with real-world data collected in our testbed and show that with our algorithm, we can achieve an accuracy sufficient for most LBSs while reducing the effort to set up the system to a mere walk of the covered area.

The remainder of this paper is structured as follows: In the next section we introduce our novel LF algorithm, the metric it uses during position estimation, our measurement clustering approach, and the special similarity measure used for clustering. Subsequently, we give an overview of our experimental setup and methodology in Section III. This is followed by experimental results in Section IV and a discussion of the results in Section V. Section VI then gives a brief overview of related work in the area of location fingerprinting. Finally, Section VII concludes the paper and gives an outlook on further research.

II. Positioning Algorithm

This section gives an overview of the probabilistic metric our positioning algorithm uses. It further introduces our clustering technique and explains our special similarity measure used during the clustering process.

A. Probabilistic Metric

During the position determination phase, our algorithm uses a probabilistic metric introduced by Haeberlen et al. [9]. The approach is split into several steps and based on a Hidden Markov Model:

- In the beginning, each fingerprint in the database has an equal probability of the user being at the position of the fingerprint.

- Then, for each access point from which signals are received at the user's current position, a probability value is computed for each fingerprint in the database by matching the signal strength that has been measured at the user's current position for the specific access point against the signal strength distribution stored for that access point in the currently considered fingerprint.
- Finally, for each fingerprint the single probabilities are combined and the algorithm selects the fingerprint with the highest overall probability. The position of that fingerprint is returned as the position estimate.

B. Measurement Clustering

We have made the observation that in structurally limited areas the signals received from 802.11 access points tend to stay within a limited range of all possible values.

Our algorithm exploits this effect by automatically identifying these regions of similar signal properties and by using them for a more reliable position estimation. As inside one region a further distinction of cells is hard to achieve because of the similar signal properties [10], the whole region is returned as a location estimate. After the collection of the training data, our algorithm performs the following steps:

- First, the area covered by the positioning system is divided into a grid of quadratic cells. In the beginning, each cell is a single cluster comprising the measurements that were made in the cell.
- Next, for each cluster the similarity to all its neighbors is computed. This is done using our novel similarity measure (refer to Section II-C). We then select the pair of neighboring clusters with the highest similarity value, and – if that value lies above the given similarity threshold – merge the two clusters.
- The second step is repeated with the remaining set of clusters until, for all clusters, no more merges take place.
- Finally, each remaining cluster represents a physically connected region of similar signal properties. The size and shape of the region is defined by the grid cells the cluster is comprised of.

In the position determination phase, the user's mobile device measures the signal properties at its current position and matches them against the fingerprints in the database. As we have only one fingerprint per region, our algorithm does not return a position estimate but rather a region as a location estimate to the user.

C. Similarity Measure

To estimate the similarity of two clusters A and B, we use a newly developed similarity measure. The computation of the measure consists of several steps:

- For each access point i of which signal strength measurements are contained in both currently considered clusters, we independently compute the average signal strength and standard deviation of the signal strength measurements on a per-cluster basis.
- We then use these values to create two normal distributions (ρ_{A_i}, ρ_{B_i}). Each distribution represents the approximate signal strength density for the access point i and one of the two clusters A or B.
- We then compute the intersection area of the two density distributions (see Equation 1).

$$area(A_i, B_i) = \int_{-\infty}^{+\infty} \left(min \left\{ \begin{array}{c} \rho_{A_i}(x) \\ \rho_{B_i}(x) \end{array} \right. \right) \qquad (1)$$

  Here, $\rho_{A_i}(x)$ is the density distribution for access point i and cluster A and $\rho_{B_i}(x)$ is the density distribution for access point i and cluster B.
- The sum of the intersection areas for all common access points divided by the number of all access points contained in either of the clusters is finally taken as the similarity measure for the two clusters (see Equation 2).

$$similarity(A, B) = \frac{\sum_n \; area(A_n, B_n)}{m} \qquad (2)$$

  where n is the set of common access points and m is the number of all access points contained in either of the fingerprints for the clusters A and B.
  Considering not only the common but also the other access points ensures that clusters having only a few access points in common are not falsely rated similar.

Instead of using common statistical approaches we decided to use this way of comparing the distributions as it very well reflects the way the positioning algorithm itself – during the position determination phase – performs the matching of the fingerprints.

III. Experimental Setup

This section describes the environment in which we performed the evaluation of our algorithm. Also, the metric applied and our experimental methodology are introduced, and we give an overview of the way we collected the data used for the evaluation.

A. Local Test Environment

We deployed our 802.11-based positioning system on the second floor of the office building A5-B on the campus of the University of Mannheim. The deployment area consists of many offices and three long hallways (see Figure 1). It is nearly 57 meters long and about 32 meters wide.

Within the test environment 34 access points are installed in total. Of these, twelve are administered by the computer center of the university, eleven were installed by us, and the remaining eleven are located in nearby offices and buildings and are beyond our control. Our data shows that most of the access points cover only parts of the operation area. In fact, some access points where received only at very few positions, and only two access points cover the operation area completely.

B. Data Collection

To evaluate our algorithm, we collected different sets of training data. For the first set, we applied a grid of reference spots with a grid spacing of 1.5 meters to the operation area. These spots are marked by the black dots in Figure 1a. At each spot we collected 110 signal strength measurements while standing still to which we will refer to as *static training measurements* for the remainder of this paper.

The second and third set contain signal strength measurements that were collected while in motion. We initially defined paths within our area of operation. For each path, we continuously collected signal strength measurements at pedestrian speed. Based on the time passed while moving from the beginning of the path to its end and the measurements' timestamps we interpolated the physical coordinates where each measurement was collected. An example for the result of this procedure is given in Figure 1b.

We repeated the collection ten times per path moving at normal walking speed ($\approx 1.0\frac{m}{s}$) and ten times moving quite slowly ($\approx 0.5\frac{m}{s}$). We call these kinds of measurements *in-motion training measurements*.

The normal walking speed was selected as we consider it to be the most intuitive speed for in-motion data collection. After we realized the low density of measurements at this speed, a second set of in-motion training data was collected at a lower speed.

We further randomly selected 46 more spots in the area of operation (marked as gray dots in Figure 1a) and collected 110 signal strength measurements – or *position determination measurements* – at each of these spots. This data is used to emulate a user requesting a position estimate during the position determination phase.

C. Metric

Using a region as a location estimate for a user is settled between position estimation and location estimation. The difference between these two is that for position estimation a coordinate in a known coordinate system is returned as the result of the estimation process. For location estimation, on the other hand, the result is a logical description like *"cafeteria"* or *"office 221"* (also refer to Kjaergaard [11]).

As our algorithm is a combination of both positioning and locating, neither using the *accuracy* (the difference between real and estimated position) – as is usually done for positioning systems – nor the *reliability* (the rate of correct location estimations) – as often done for location estimation systems – alone would serve as a suitable performance metric.

So besides looking at the reliability, we decided to also combine these two metrics and form a new one: Whenever our algorithm returns an estimated region, we check whether the real position lies within that region. If this is the case, the error is counted as 0. If the real position is outside, we use the distance between the real position and the border of the estimated region as the error value. Thus we can get an impression of how far away our estimated regions are from the user's real position.

D. Methodology

To analyze the properties of our algorithm, we used our suite of positioning-related tools [12] and modified it to suit our needs.

Our basic experiment consists of the following steps:

- At first, the positioning algorithm is initialized with either 20 or four randomly selected static training measurements per reference spot or one set of in-motion training measurements per path to create the regions and build up the fingerprint database. We will refer to the number of measurements, respectively sets, as *training set size* (TSS) from now on.
- This is followed by providing the algorithm – for each test position – with three randomly selected position determination measurements. For the subsequently estimated region, the error is computed according to our metric and stored for later reference.

For the evaluation of the performance of our algorithm we used our three different sets of training data. Doing so, we could compare the performance of the algorithm when using training data that was collected while moving, to that when using statically collected training data. The number of training and position determination measurements were – as far as applicable – chosen according to King et al. [8]. We also varied the similarity threshold in 0.01 steps from 0.01 to 0.99. To achieve statistically stable results, we repeated our basic experiment 250 times for each setup.

IV. Experimental Results

In this section, we describe our experimental results. We explain the influence the similarity threshold has on the performance of our algorithm and show the effects of smaller or larger regions on the position determination. Afterwards, we look at the consequences of using training data that was collected when moving and the clustering process.

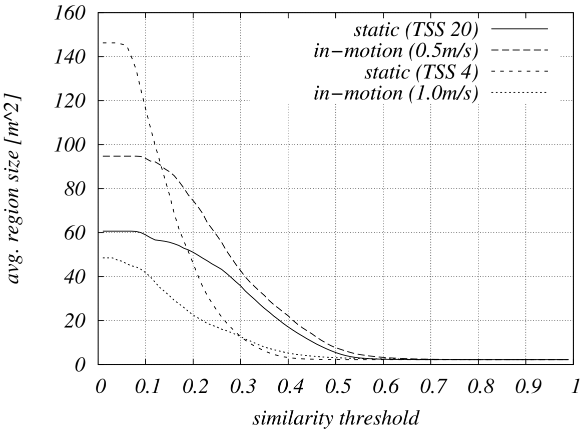


Fig. 2. Influence of the similarity threshold on the region size.

A. Similarity Threshold and Region Size

The factor that has the largest influence on our algorithm is the similarity threshold that is used to determine whether two adjacent clusters should be merged. By varying this factor, we can adjust the overall rate of correct region estimates and the average error.

As shown in Figure 2, a very low threshold results in few but large regions. But this also increases the rate of correct region estimations (see Figure 3). If we, for instance, set the similarity threshold to a value of 0.40, our algorithm – being given the training data that was collected at slow walking speed – estimates the correct region in about 82% of all cases. This comes at the cost of a decreasing *absolute* accuracy, though. Even if the system estimates the correct region it has no further information where inside the region – the average region size is roughly $22m^2$ in our example – the user is located.

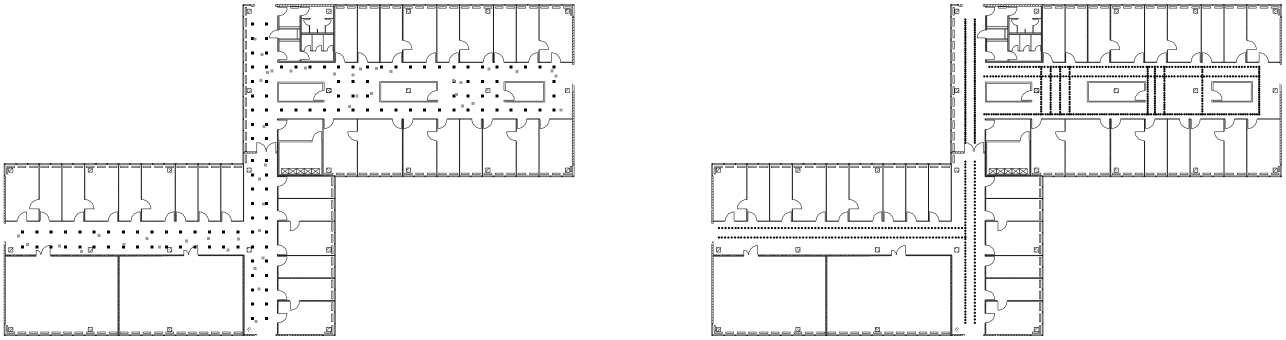
In contrast, if a high value is selected, the regions stay small and ultimately only comprise one single cell each. In such a case of many very small regions, the positioning results of our region-based algorithm are similar to those of a fingerprinting-based positioning algorithm that uses a grid of reference spots. Each region or cell then corresponds to one of the reference spots inside the grid.

B. Motion and Motion Speed

Another important factor for the performance of our algorithm is the way the training measurements are collected. It influences both the density of and also the signal variation between consecutively collected samples.

When using the set of static training measurements and a training set size of 20 or four we have a density of 20 respectively four measurements per grid cell as each grid cell contains exactly one reference spot. For the training data collected while moving, the density depends on the time the collecting device spends within the area of each cell. In our case with a cell size of $1.5\ m \times 1.5\ m$, the density of the measurements collected while moving at a speed of about $0.5\frac{m}{s}$ is seven measurements per grid cell. When moving faster at $1.0\frac{m}{s}$, this value drops to roughly three measurements per grid cell.

Compared to the static training data, the in-motion training data also has an inherently higher signal variance. This in combination



(a) Layout of the reference and position determination spots.

(b) Layout of the in-motion measurements.

Fig. 1. Reference and position determination spots (Figure 1a) and moving paths (Figure 1b) within the area of operation.

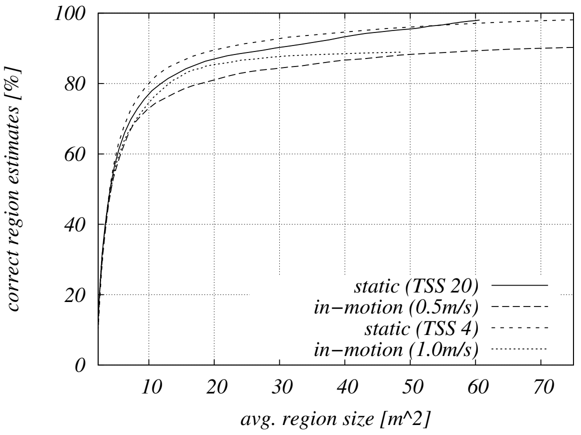


Fig. 3. Influence of the region size on the rate of correct region estimates.

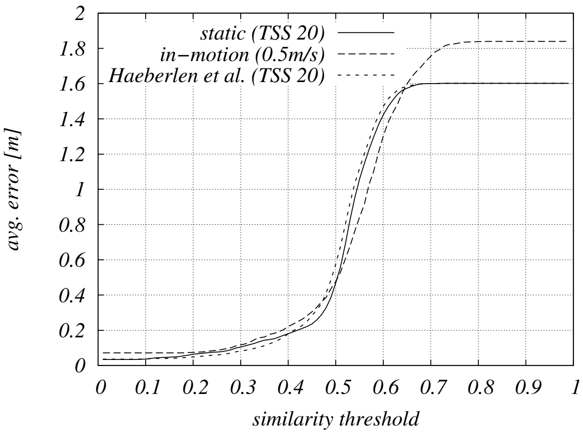


Fig. 4. Comparison of the performance of our algorithm being given training measurements that were collected when standing still or when moving.

with the sample density results in a varying number of access points that were received in the different grid cells. On the average ten access points are contained in each fingerprint for the $1.0\frac{m}{s}$ in-motion training measurements, about twelve access points for the $0.5\frac{m}{s}$ in-motion training measurements, 14 for the static training measurements when using a TSS of four, and about 16 access points in each fingerprint when using a TSS of 20.

For the sets of measurements used for positioning during the position determination phase, the average number of contained access points is ten for the measurements from the static dataset and eight for the measurements from the in-motion dataset.

Looking at the actual positioning performance of our algorithm when being given training measurements that were collected while standing still and those that were collected while in motion, we can see that our algorithm handles both almost equally well. Only for higher values of the similarity threshold – and therefore when the regions become small – we achieve better results when using statically collected training data (see Figure 4). The reason for this behavior is the low number of measurements per grid cell in combination with the higher signal variation. Especially for small regions, in such a case, the overall number of measurements per region is not sufficient to create stable distributions for the fingerprints (also refer to [8]). If the data has additionally been collected while moving, this further decreases the distribution quality.

This higher variance in the signals is also the reason why the maximum average region size (see Figure 3) is lower for the two cases where we used the training data with a TSS of four respectively the one that was collected at normal walking speed. For the latter case, for instance, the maximum average size of the regions was only about $50\ m^2$.

C. Clustering

As a further benchmark for our own algorithm, we also implemented the original LF algorithm from Haeberlen et al. [9]. During each run of the emulation, we fed the same training data to our own clustering algorithm to let it create the set of regions and to the algorithm from Haeberlen et al. to create its fingerprint database. During the position determination phase, we then checked in which region the position estimated by the algorithm from Haeberlen et al. lay and used our metric to compute the error. As the algorithm from Haeberlen et al. cannot handle training data that was collected when moving, we only used the statically collected data in this case. The results of this comparison can be seen in Figure 4. As shown there, using one fingerprint per region – as our algorithm does – results in a slightly higher amount of correct region estimates most of the time, whereas the approach of using the algorithm from Haeberlen et al. to estimate the position and matching it afterwards to the regions is slightly inferior in our case.

V. Discussion

This section briefly discusses the implications of our findings for LF systems and LBSs. We consider both the standpoint of the service provider and the standpoint of the service user and also some other aspects of our system.

TABLE I
MEASURING TIMES FOR THE AREA OF OPERATION

training data	static (TSS 20)	static (TSS 4)	$0.5 \frac{m}{s}$	$1.0 \frac{m}{s}$
time [sec]	2150	430	465	280

TABLE II
COMPUTATION TIMES

	offline [ms]	online [ms]
Haeberlen et al.	22	2655
Fingerprint clustering	270	114

A. Accuracy and Reliability

Different applications have different requirements regarding the accuracy of the delivered position estimates and their reliability. Our algorithm accounts for these differences by offering the similarity threshold as a parameter to adapt the algorithm's performance to the application's requirements (see Figure 3).

By reducing the similarity that is required for two clusters to be merged during the training phase, we can directly influence the average size of the clusters and thus adapt the coverage area of the single fingerprints to the required accuracy of the application.

Increasing the size of the clusters also increases the reliability of our algorithm. This comes at the cost of a decreased absolute accuracy. When, for instance, the similarity threshold is set to a very low value, the regions finally cover full floors. The system, in this case, estimates the correct region with a reliability of almost 100%, but we have no further information about where the user is exactly located in the estimated region. Therefore, for applications that require a high reliability, a lower similarity threshold would be the appropriate setting.

B. Position Refinement

To overcome the loss of accuracy for larger regions, we tried to further refine the position estimation by introducing a second position determination step: After the estimation of the region the user is located in, we created a second fingerprint database with one fingerprint for each cell in the estimated region. We then matched the collected live data against the fingerprints in that second, limited database and selected the cell with the best matching fingerprint as an estimate for the position inside the cluster.

Especially for higher values used for the similarity threshold, this did not improve the positioning accuracy, though. The reason for this result is that when the signal strength properties from within the region are similar anyway, such a refinement step hardly performs better than a random selection of a cell from within the region.

C. Usability

In terms of usability, our algorithm is a considerable improvement. It delivers an accuracy and reliability that is sufficient for most LBSs. At the same time, it offers the possibility to reduce the amount of time that needs to be spent for setup and maintenance of the fingerprint database to a fraction of what other systems need. The main reason for this is that the training data no longer needs to be collected while standing still but can be collected by simply walking through the area of operation the positioning system shall cover.

Table I gives an overview of the measurement collection times for our different datasets. It should be noticed that for the in-motion approach these numbers do reflect the real effort quite well. For the grid-based approach, though, the time needed to exactly position the measuring device on the reference spots is not included in these numbers. If we take an average time of 10 seconds to reposition the collecting device on each reference spot an amount of 1300 seconds would have to be added to reflect the real effort.

Furthermore, compared to systems where fingerprints are created only for special locations, our approach offers the possibility to cover the complete area of operation like it is generally done when using a grid-based approach. By automatically detecting regions of similar signal properties, we avoid the error-prone process of letting the operator manually define regions or locations himself. This prevents the definition of regions that are connected in physical space but might not be consistent in signal space.

Considering the often mentioned problem of keeping the fingerprint database up to date once the positioning system is operational, our algorithm has several advantages, too. First, the operator can – in the case of environmental changes like added or removed access points – identify the affected regions and recreate the fingerprint database only for these regions in the same effort-saving manner as when initially setting up the system. Second, our system, as it can handle in-motion training data, could also be extended to take advantage of user supplied training data as proposed by e.g., Chai et al. [13].

Finally, depending on the requirements of the application, the performance of the our algorithm can be adjusted for accuracy or reliability by simply varying the similarity threshold.

D. Computational Requirements

Taking a look at the computational requirements, our approach is advantageous as well. It combines similar cells into regions and creates only one fingerprint per region. Therefore the overall number of fingerprints for an area to cover decreases compared to a classical grid-based approach where each cell or reference spot would have its own fingerprint. Whereas the creation of the region map is an additional effort that has to be performed once during the training phase, the system can take advantage of the lower number of fingerprints and the therefore decreased computational effort for every following position estimation during the position determination phase. For instance, using a similarity threshold of 0.53 results in a set of 19 regions and thus 19 fingerprints. Covering the same area with a grid of reference spots using a gridsize of $1.5\ m$ results in a total of 130 fingerprints. As, for each position estimation during the position determination phase, the collected data has to be matched against all fingerprints, this means a reduction of the necessary computational effort by more than 80% when using our algorithm.

Table II exemplarily shows the measured average computation times for the above described setup on one of our emulation systems. The offline time is the average time the algorithm needed to compute its fingerprint database, and in the case of our algorithm to create the region map in advance. The online time is the average time that was needed to compute one round of position estimations for our set of position determination spots (46 spots).

VI. RELATED WORK

The position estimation with 802.11 and location fingerprinting has been under research for quite some time. Early systems like RADAR [14] use a *deterministic* approach – namely the Euclidean distance in signal space – to compare the measurements and to find the fingerprint that offers the closest match to the data collected in the position determination phase. Novel systems, for instance the system introduced by Haeberlen et al. [9], use a *probabilistic* metric based on signal strength distributions to compare the measurements. This approach has several advantages; for example, it offers a better robustness with respect to noisy signals. Even generally, probabilistic systems can deliver better results according to Youssef et al. [15]. For this reason we used the latter type of system as a starting point for our own developments.

Considering the reduction of the effort needed for setup and maintenance of 802.11-based LF systems, Chai et al. [13] examine the effects of reducing either the density of the reference spots or the number of training measurements collected at each spot. Their result is that both have a negative influence on the accuracy of the positioning system. To compensate for this, Chai et al. propose the use of user-collected traces that contain consecutive measurements that the devices of the users collect while the positioning system is already operational. Even though there is no verified position information available for these measurements, Chai et al. suggest a method to use them to enhance the training data and by such the positioning accuracy of the system.

Our own previous work [16] handles a possible reduction of the amount of training measurements in another way. We have shown that the overall amount of training measurements that has to be collected at each reference spot can be reduced without losing accuracy by using an approach we call *quick fingerprinting*. This is achieved by not only considering measurements that have been collected at the reference spot itself but also at adjacent spots in the grid.

The idea of using regions in the context of positioning systems has been accounted for in the past as well. In [17], Youssef et al. use a clustering technique based on the set of access points from which signals are received in the position determination phase to group training measurements and by such reference spots into clusters. These clusters then are used to reduce the overall computational complexity of the position estimation process by only considering fingerprints that match the current set of received APs. Such a reduction of the computational complexity is especially desirable for devices such as PDAs or cellular phones.

A further use of regions has been pointed out by Kjaergaard et al. in [18]. There, zones are defined for the coverage areas of LBSs. Information about the signal properties within these zones can then be used by mobile devices to determine when they enter or leave the sphere of influence of an LBS. This helps to avoid the regular beaconing of position information to the services which would be otherwise needed to determine whether the mobile device is still within the service area. Such an approach helps to reduce the overall amount of data that needs to be sent, saves precious energy and is also desirable from a privacy point of view.

VII. Conclusion

In this paper, we have introduced a novel algorithm to estimate the location of a user with 802.11 and fingerprinting. Compared to other approaches, our approach has the major advantage that it clusters the collected training measurements into regions of similar signal properties by using a novel similarity metric and thus needs significantly less training data to achieve satisfying results. The clustering also allows us to collect the training data by simply walking through the area of operation on defined trajectories. This is a major improvement over other systems that need measurements at fixed reference spots.

As our algorithm uses one fingerprint for each region instead of creating one fingerprint for each reference spot the overall computational load – one of the limiting factors especially on mobile devices – can be significantly reduced.

Our algorithm delivers an accuracy that is sufficient for most types of LBSs and at the same time maintains a high reliability. It offers the possibility to adjust its output for either higher reliability or higher accuracy by varying the similarity threshold.

In the future, we intend to further ease the effort of training data collection by using sensor fusion. By this approach we hope to no longer be bound to given trajectories when collecting the training data and instead to be able to move completely free through the area of operation during the training phase.

References

[1] E. Kaplan and C. Hegarty, *Understanding GPS: Principles and Applications*, 2nd ed. Artech House Incorporated, December 2005.

[2] J. Krumm, G. Cermak, and E. Horvitz, “Rightspot: A novel sense of location for a smart personal object,” in *Proc. of the 5th Int. Conf. on Ubiquitous Computing (Ubicomp)*, October 2003.

[3] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, “The cricket location-support system,” in *Proc. of the 6th Int. Conf. on Mobile Computing and Networking (MobiCom)*, August 2000.

[4] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, J. Tabert, P. Powledge, G. Borriello, and B. Schilit, “Place lab: Device positioning using radio beacons in the wild,” in *Proc. of the 3rd Int. Conf. on Pervasive Computing (Percom)*, March 2005.

[5] R. Want, A. Hopper, V. Falcão, and J. Gibbons, “The active badge location system,” *ACM Transactions on Information Systems*, January 1992.

[6] I. for Electrical and I. Electronics Engineers, “Ansi/ieee standard 802.11: Wireless lan medium access control (mac) and physical layer (phy) specifications,” Website: http://standards.ieee.org/getieee802/, 1999.

[7] V. Bychkovsky, B. Hull, A. Miu, H. Balakrishnan, and S. Madden, “A measurement study of vehicular internet access using in situ wi-fi networks,” in *Proc. of the 12th Int. Conf. on Mobile Computing and Networking (MobiCom)*, September 2006.

[8] T. King, T. Haenselmann, and W. Effelsberg, “Deployment, calibration, and measurement factors for position errors in 802.11-based indoor positioning systems,” in *Proc. of the 3rd Int. Symposium on Location- and Context-Awareness (LoCA)*, September 2007.

[9] 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 *Proc. of the 10th Int. Conf. on Mobile Computing and Networking (MobiCom)*, September 2004.

[10] H. Lemelson, M. B. Kjaergaard, R. Hansen, and T. King, “Error estimation for indoor 802.11 location fingerprinting,” in *Proceedings of the Fourth International Symposium on Location and Context Awareness (LoCA 2009)*, Tokyo, Japan, Mai 2009.

[11] M. B. Kjaergaard, “A taxonomy for radio location fingerprinting,” in *Proc. of the 3rd Int. Symposium on Location- andContext-Awareness (LoCA)*, September 2007.

[12] T. King, T. Butter, H. Lemelson, T. Haenselmann, and W. Effelsberg, “Loc{lib,trace,eva,ana}: Research tools for 802.11-based positioning systems,” in *Proc. of the 2nd Int. Workshop on Wireless Network Testbeds, Experimental evaluation and CHaracterization (WiNTECH)*, September 2007.

[13] X. Chai and Q. Yang, “Reducing the calibration effort for location estimation using unlabeled samples,” in *Proc. of the 3rd Int. Conf. on Pervasive Computing and Communications (PerCom)*, March 2005.

[14] P. Bahl and V. N. Padmanabhan, “Radar: An in-building rf-based user location and tracking system,” in *Proc. of the 19th Int. Conf. on Computer Communications (InfoCom)*, March 2000.

[15] M. Youssef and A. Agrawala, “On the optimality of wlan location determination systems,” in *Proc. of the Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS)*, January 2004.

[16] H. Lemelson, T. King, and W. Effelsberg, “Pre-processing of fingerprints to improve the positioning accuracy of 802.11-based positioning systems,” in *Proc. of the 1st Int. Workshop on Mobile Entity Localization and Tracking in GPS-less Environments (MELT)*, September 2008.

[17] M. A. Youssef, A. Agrawala, and A. U. Shankar, “Wlan location determination via clustering and probability distributions,” in *Proc. of the Int. Conf. on Pervasive Computing and Communications (Percom)*, March 2003.

[18] M. B. Kjaergaard, G. Treu, and C. LinnhoffPopien, “Zone-based rss reporting for location fingerprinting,” in *Proc. of the 5th Int. Conf. on Pervasive Computing (Pervasive)*, May 2007.