REIHE INFORMATIK
TR-2007-008

On-Demand Fingerprint Selection for 802.11-based Positioning Systems

Thomas King, Thomas Haenselmann, Wolfgang Effelsberg
University of Mannheim
– Fakultät für Mathematik und Informatik –
Praktische Informatik IV
A5, 6
D-68159 Mannheim, Germany

On-Demand Fingerprint Selection for 802.11-based Positioning Systems

Thomas King, Thomas Haenselmann, Wolfgang Effelsberg
{king,haenselmann,effelsberg}@informatik.uni-mannheim.de
Department of Computer Science
University of Mannheim, Germany

Abstract

Fingerprinting is a popular technology for 802.11-based positioning systems: Radio characteristics from different access points are measured at various positions and stored in a database. The database is copied to all mobile devices, and when a position is needed, the devices compares its currently measured radio characteristics with the database entries. In this paper, we present two on-demand fingerprint selection algorithms to avoid the cumbersome and time-consuming approach of manually copying all fingerprints. Our algorithms only request those fingerprints from the database that are currently required to compute a position. The two algorithms differ in the way they shape the region for which fingerprints are requested. On-demand selection also allows storage-restricted mobile devices to utilize the positioning system. We carefully evaluate our algorithms in a real-world experiment. The results show that our algorithms do not harm the position accuracy of the positioning system. In addition, we analyze the space requirements of our algorithms and show that the typical constraints of mobile devices are met.

1 Introduction

In recent years, there have been considerable improvements in down-sizing computer hardware and in increasing the capacity of rechargeable batteries, as well as the advent of wireless networks for the mass markets. These technologies allow manufacturers to build mobile devices that have a similar performance as desktop computers had several years ago. The benefit of these mobile devices can be leveraged by so-called *location-based services*: Applications that act differently depending on the location of the device. Location-based services are currently a hot topic in research, and are considered to be a promising market.

Nowadays, the *Global Positioning System* (GPS) [9] is the predominant positioning system. Whereas GPS works well in many scenarios, it suffers from blocked radio signals caused by walls, ceilings, or skyscrapers creating shielded street canyons. To provide users with positioning information even in indoor environments, various research groups proposed different positioning technologies (e.g., [7]).

One of the most promising technologies that could be an equivalent to GPS for indoor applications are *802.11-based positioning systems* (e.g., [1], [4]). Nowadays, 802.11 hardware is readily available and installed nearly everywhere where people live and work. 802.11 is a wireless local area network technology that is used to provide Internet access to mobile users; however, it can be used for positioning purposes at the same time [13]. Furthermore, more and more modern PDAs, cellphones and laptops are capable to communicate with 802.11 infrastructure because they are shipped with built-in 802.11 network technology.

The best positioning results can be achieved with 802.11-based positioning systems that utilize the so-called *fingerprinting* approach. This approach consists of a two stage mechanism: A *training phase* and a *position determination phase*. During the training phase, specific radio characteristics from nearby access points are gathered at pre-defined reference spots in the operation area and stored together with their physical coordinates in a database: They are called *fingerprints*. In the position determination phase, the user's mobile device samples specific radio characteristics at its (unknown) position and searches for similar patterns in the database. The closest match is selected and its coordinate returned as a position estimate. A wide range of radio characteristics can be used for fingerprinting: For instance, nearly all positioning systems work with signal strength (e.g., [1]), others additionally utilize signal to noise ratio (e.g., [3]), or response rate (e.g., [15]).

Following Kjærgaard [14], most fingerprinting systems can be categorized as so-called *infrastructure-based* and *terminal-based* systems. Infrastructure-based means that fixed infrastructure such as 802.11 access points are utilized to generate the radio characteristic in question. Further, in terminal-based positioning systems, the user's mobile device is used to sample characteristic data and to compute a position estimate. There are two good reasons why terminal-based systems are prevailing: First, terminal-based systems are easy to set up. In most cases, a sensor to sample radio characteristics such as a 802.11 network card is already part of the user's mobile device. To make use of this sensor, only additional software needs to be installed. Furthermore, the infrastructure can be used without any changes and without even knowing that it is used in a completely new way. This makes prototype installations as well as system-wide roll-outs quickly realizable and cheap. Second, privacy concerns remain a major barrier to adoption of location-based services. If the data required to calculate a position estimate is sampled and processed on the user's device, the user is in control of disclosing her position to whom, when and with which level of granularity she wants.

One major drawback of most fingerprinting systems, especially the terminal-based ones, is that the fingerprint database has to be copied on each mobile device that is supposed to use the positioning system. So far, an administrator manually copies the fingerprint database from a central repository such as PlaceLab [17] onto the mobile devices. We call this way of handling the fingerprint data the *administrator's occupational therapy*. While this cumbersome and time-consuming approach is applicable in lab settings, it leads to a lot of problems in large-scale roll-outs and daily-use:

- *Update of fingerprints:* It might happen once in a while that fingerprints taken from a building have to be updated because the building has been reconstructed or access points inside the building have been relocated. Furthermore, fingerprints from newly constructed buildings have to be added and fingerprints from buildings that have been demolished should be removed. If fingerprint data on mobile devices are not updated, the position accuracy at least decreases and in the worst case, no positioning is feasible for areas that are covered by stale or missing fingerprints.
  So far, no way exists to automatically update all mobile devices that carry a fingerprint database.
- *Limited storage capacities of mobile devices:* Hightower et al. [8] roughly estimate that a fingerprint database containing all 802.11 access points in the world would at least require a few dozen gigabytes of memory. From a positioning system point of view, a major restriction of mobile devices is their storage capacities. If we compare storage capacities of mobile devices, we see two major device classes: Mobile devices in the *smart-phone* class, such as PDAs, smartphones, and laptops usually provide at least few hundred megabytes of fixed-disk storage and about a few dozens megabytes of main memory. Tiny devices of the *sensor node* class typically contain no fixed-disk storage and only a few hundred kilobytes of main memory.
  While it might be possible to store a world-wide fingerprint database on devices of the smart-phone class, nobody wants to allocate a large portion of the fixed-disk storage for data that is never used. Even worse, it is impossible to store a world-wide fingerprint database on a sensor node.

In this paper, we provide solutions for the problems listed in the previous itemization. We present two novel algorithms to select only fingerprints from a central repository that are required to compute a mobile device's position. The fingerprints are dynamically selected based on the access points within communication range of the device. This keeps the fingerprint data on mobile devices fresh and makes sure that always the latest available fingerprints are used. Our novel on-demand fingerprint selection algorithms are of common use which means every positioning algorithm that relies on fingerprints can be combined with our algorithms. Further, a major design goal of these algorithms is to reflect the storage capabilities of the aforementioned classes of mobile devices. The performance of our algorithms in terms of their impact on the positioning system, frequency of fingerprint data queries, size of regions and space requirements are evaluated in a real-world testbed.

Looking at research work that has been done in the area of 802.11-based positioning systems it can be seen that besides work on the theoretical basis of positioning algorithms (e.g., [22]), a lot of experimental work has been carried out (e.g., [1], [6], [17]). Additionally, large testbeds up to a scale of cities have been set up to research how 802.11-based positioning systems perform in these environments (e.g., [16], [17], [18]). However, best to our knowledge, we are the first who present different algorithms to select only the fingerprints that are required to compute a position estimate while respecting the storage capabilities of mobile devices. Although we have already presented preliminary work as a demo [10], here we present the complete algorithms together with enhancements and also provide an in-depth experimental evaluation. This work is another building block in making 802.11-based positioning systems easily usable and ready for productive usage.

The remainder of this paper is structured as follows: We present our two novel on-demand fingerprint selection algorithms in the following section. In Section 3, the experimental setup and research methodology are described that are used to evaluate the on-demand fingerprint selection algorithms. Subsequently, the experimental results are presented in Section 4. Section 5 presents the relevant related work. Finally, we conclude the paper and discuss future work in Section 6.

2 On-Demand Fingerprint Selection Algorithms

As 802.11-based positioning systems rely on 802.11, it can be generally assumed that a broadband Internet connection is offered by these networks [2]. Most location-based services such as a friend-finder application or an asset tracking system make only sense if data can be exchanged between these kinds of applications. We utilize this fact and connect mobile devices to the fingerprint repository by means of 802.11 networks. This allows a mobile device to query the repository for fingerprints covering its current region. We define the *region* of a mobile device as an area wherein it is located and that is encircled by the access points within its communication range.

Our two novel on-demand fingerprint selection algorithms differently shape the size of a mobile device's region. Generally speaking, the footprint of fingerprints covering a region depends on its size. So, by differently shaping the size of a device's region, the storage capabilities of the aforementioned device classes can be met. Furthermore, the size of the region in general defines the area wherein a mobile device can move around without triggering a fingerprint data query. The following sections introduce the novel on-demand selection algorithms and discuss how they work.

2.1 Union of Access Points

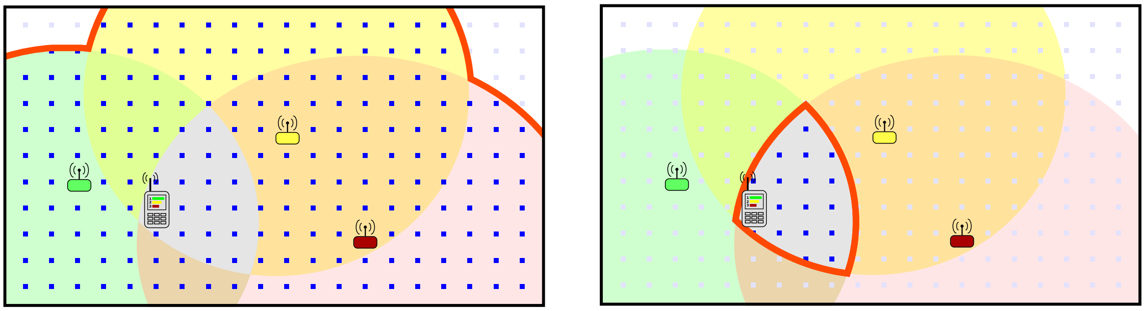
The advances in miniaturization of memory technology allow to build mobile devices categorized as laptop class devices that offer plenty of storage. For these devices, a on-demand fingerprint selection approach should not try to minimize the footprint of stored fingerprints, because these devices can easily handle larger sets of data. Instead, in such a scenario, the prime reason for a fingerprint selection algorithm is to keep the data on mobile devices up-to-date. To achieve this goal, all fingerprints that contain any of the access points within communication range of the user's device are selected. The region of a mobile device is therefore shaped by the union of the coverage areas of all access points within its communication range (see Figure 1(a)). Hence, we call this algorithm the *Union of Access Points (UAP)* approach.

We describe the algorithm in more detail: After a mobile device sampled its proximity for access points, it queries the repository by providing the sample results. The repository replies with the fingerprints of each reference spot that contains at least one fingerprint corresponding to any of the access points listed by the mobile device. If the mobile device moves out of the communication range of an access point, the fingerprints of this particular access point are removed. In case unrecognized access points come into communication range, the mobile device requests fingerprints for all access points it is aware of. If one of the access points is unknown by the repository (e.g., it has lately been deployed), it is just omitted while constructing the union.

2.2 Intersection of Access Points

Considering sensor nodes leads to the *Intersection of Access Points (IAP)* algorithm. Mobile devices of the class of sensor nodes are in general extremely limited in terms of storage. In this case, the footprint of fingerprints stored on a mobile device should be as small as possible. Given only the access points in communication range of a mobile device and the access points' coverage areas, the intersection of these areas defines the smallest area wherein the mobile device can be located. So, we shape the region of a mobile device by intersecting the coverage areas of the access points within its communication range (see Figure 1(b)).

The IAP algorithm utilizes this fact: A mobile device scans for access points in communication range and reports the access points to the repository. The repository



(a) The mobile device's region according to the Union of Access Points approach.

(b) The Intersection of Access Points algorithm is used to calculate the region of the mobile device.

Figure 1. These figures exemplify how a mobile device's region is shaped by the two fingerprint section algorithms. The figures show the same scenario: Three access points and their coverage areas are plotted in red, green, and yellow, respectively. The gray and blue squares depict reference spots where fingerprints are sampled. The blue squares are framed by an orange line and represent the reference spots used to select fingerprints according to the on-demand selection algorithm in question.

computes the intersection of the access points' coverage areas. Only for the reference spots inside this intersection, the fingerprints of the reported access points are transferred to the mobile device. Each time a mobile device moves out of the coverage area of a known access point or into the coverage area of an unrecognized access point, the procedure is repeated. In case an access point is unknown by the repository, its presence is ignored.

3 Experimental Setup and Measurement Methodology

In this section, we describe the experimental setup and the measurement methodology.

3.1 Local Test Environment

We deployed our 802.11-based positioning system on the second floor of our office building on the campus of the University of Mannheim. The operation area is nearly 57 meters in width and 32 meters in length; approximately 221 square meters are covered. The floor plan of the operation area is shown in Figure 2.

3.2 Hardware and Software Setup

Initially, the test environment was covered by twelve access points. Seven of them are administered by the computer center of our university. The other five are installed in nearby buildings and offices. We additionally installed thirteen access points. Our data shows that most of the access points cover only parts of the operation area. In fact, only two access point cover the operation area completely. One of these access points is the one marked in the middle of the storage room in the horizontal hallway in the right part of the building. This access point is located in a suspended ceiling on top of this room. The position of the second access point is in an office one floor below of our operation area. The positions of the access points that are located on the same floor and inside the same building parts as our operation area are marked by orange circles in Figure 2.

As a client, we used a Lucent Orinoco Silver PCMCIA network card supporting 802.11b. This card was plugged into an IBM Thinkpad R51 running Linux kernel 2.6.13 and Wireless Tools 28pre. To collect signal strength samples, we implemented our own set of tools [11].

3.3 Data Collection

The grid of reference spots in the operation area includes 130 spots with a spacing of 1.5 meters (see the blue marks in Figure 2). During the training phase, we collected 110 signal strength samples at each reference spot. For the position determination phase, we randomly selected 46 spots. Again, we collected 110 signal strength samples for each positioning spot. In Figure 2, the positioning spots are marked by purple dots. We spent over ten hours to collect all the data.

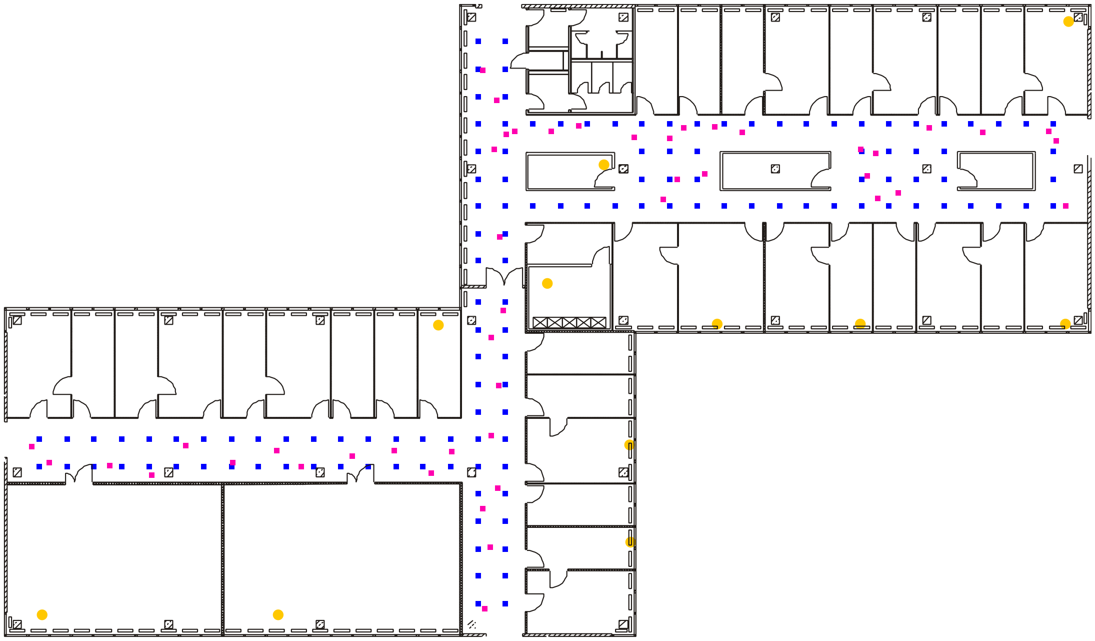


Figure 2. Floor plan of the local test environment.

3.4 Metrics

The metric that is of most interest for all positioning systems is the *position error*. As we consider a two-dimensional operation area, the position error is defined as the Euclidean distance between the real physical position and the estimated position. In most cases, our on-demand fingerprint selection algorithms select only a subset of the available fingerprints and hence reduce the data on which positioning algorithms operate to compute a position estimate. So, we are interested in the impact of the fingerprint selection algorithm on the position accuracy.

Metrics of relevance for fingerprint selection algorithms are the *frequency of fingerprint data queries*, the *size of regions* and the *space requirements*. We define the former metric as the number of fingerprint data queries per radio characteristic measurement. The frequency should be small because each query means that the 802.11 network is utilized to transfer the query from the mobile device to the repository and fingerprints vice versa.

The size of a region defines the area in which a mobile device can move around while positioning itself utilizing the fingerprints is stores. The size of a region is defined by the number of reference spots it contains.

The space requirements are measured in terms of fingerprints. We keep this metric abstract because the exact size of a fingerprint depends on the selected positioning algorithm. Different algorithms aggregate raw measurement data differently to a fingerprint [14].

3.5 Experiments

To analyze our on-demand fingerprint selection algorithms we utilize the data we have collected as described in Section 3.3. We define a *basic experiment* that is used as a basis for the subsequent studies. If a study of a certain on-demand fingerprint selection algorithm requires an extension of the basic experiment, the changes are described in the corresponding section.

The basic experiment is defined as follows: As our positioning algorithm we utilize the algorithm proposed by [16]. This algorithm is probabilistic, and Youssef et al. proved in [21] that this class of algorithms is superior to other classes in terms of position accuracy. For each reference spot, 20 samples are randomly selected out of the 110 samples. In the position determination phase, only one signal strength measurement is selected out of the 110 samples taken at each positioning spot. These numbers are derived from the recommendations stated in [12]. This experiment is repeated 1000 times to achieve statistically stable results.

4 Experimental Results

In this section, we present our evaluation results.

4.1 Position Accuracy

We combine the basic experiment with our on-demand fingerprint selection algorithms to see if they interfere with the positioning algorithm, and what the consequences are. Figure 3 depicts the cumulative distribution function (cdf) of the position error for the plain basic experiment as well as the combinations with our selection algorithms.

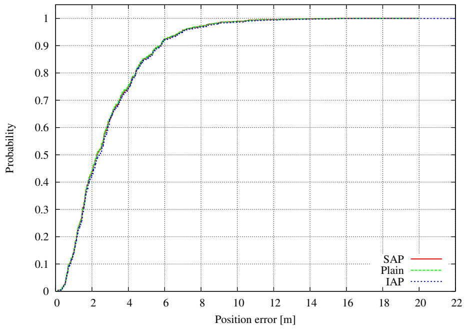


Figure 3. Cumulative distribution function of the position error.

The figure shows the the cdfs of the position error for all three experiments are practically equal. The average position error for the plain basic experiment is 2.86 meters, UAP shows the same value. The reason for this is that UAP selects all fingerprints from the reference spots covered by any of the access points visible to the mobile device. This is what basically happens in the basic experiment. Although the complete fingerprint database is available on the mobile device, only the fingerprints that contain any of the access points the mobile device is aware of can be used by the positioning algorithm for further processing. So, the fingerprints used by the positioning algorithm to calculate a position estimate are identical and hence the cdfs are equal. The very small differences come from statistical variations.

The IAP approach shows a slightly worse average position error of 2.97 meters. The reason for this is that sometimes the real position of the user is outside the intersection, and in fact closer to a reference spot that is also outside the intersection than to any reference spot of the intersection. The positioning algorithm applied to IAP has to choose one the reference spots of the intersection which means the position error is a little bit larger than what can be achieved in the other experiments. A similar fact causes the large long tail of IAP compared to the other cdfs. In very rare occasions, the radio characteristics sampled outside the intersection do not match a nearby reference spot and hence the algorithm gets confused and wrongly selects a reference spot far away. However, the difference in the average position errors is only 0.09 meters which corresponds to about three percent.

Even although we see minor differences in the cdfs of the experiments, we conclude that the impact of the fingerprint selection algorithm on the position error is negligibly small.

4.2 Frequency of Fingerprint Data Queries

During our experiments we realized that sometimes the set of access points a mobile device samples changes over time, though the device remains at a single spot. The reasons for this are manifold: A measurement packet gets lost due to packet collisions or a noise source such as a microwave increasing the noise level so that a receiver cannot clearly hear the packet. Another reason is that structural changes in the environment (e.g., a door is closed) worsen radio propagation.

For both algorithms a fingerprint data query is executed if the set of access points changes in two consecutive measurements. We counted how often the same set of access points can be found in two consecutive measurements. We found that on average only 2.28 consecutive measurements contain an equal set of access points. In other words, on the average, 106.72 consecutive measurements show different sets of access points. This corresponds to a frequency of 0.98 fingerprint data queries per radio characteristic measurement. To reduce the high frequency of queries we came up with the following approach: The access points from the n-latest measurements are grouped together and compared with the set of access points obtained from the n measurements taken before. Only if the two sets of access points differ, a fingerprint data query is performed. This enhancement is dubbed n-*Group* (n-*G*).

Table 1 lists the number of inequalities that occur if each set of access points is created by combining n measurements. The average number of inequalities goes down from 106.72 to 5.57 which corresponds roughly to five percent if ten measurements are used to build each set of access points. This means that on average during 110 signal strength measurements, six fingerprint data queries are performed. From the highest number of observed values we see that if $n = 2$, positioning spots exist where the theoretical maximum number of inequalities[1] is reached. The theoretical maximum number is never reached again for values greater than two, however, it is often pretty close (e.g., only one or two occurrences are left).

	Number of inequalities			Frequency		
n	average	max	min	average	max	worst case
1	106.72	109	99	0.98	1.0	1.0
2	48.13	54	34	0.44	0.50	0.50
3	27.54	35	12	0.25	0.32	0.33
4	18.11	25	3	0.17	0.23	0.25
5	13.37	20	3	0.12	0.18	0.20
10	5.57	10	1	0.05	0.09	0.10

Table 1. The number of inequalities and the fingerprint data query frequency for different values of n.

We calculate the fingerprint data query frequency based on the number of inequalities. The worst case column corresponds to the theoretical maximum number of inequality. While the average frequency is pretty close to the worst case frequency for $n = 2$, it still shows an average frequency of 0.05 which is only one half of the worst case value if a group size of ten is applied.

Grouping together access points of n measurements before checking if a fingerprint update is required also delays this decision. Especially, for the IAP algorithm we expect stale fingerprint data to negatively impact the position accuracy. A delay for this algorithm implies that the region for which fingerprints are available is not perfectly shaped according to the access points the mobile device is aware of. If the currently used region is larger than the intersection that would be build by the access points being part of the latest measurement, the algorithm has to choose from more reference spots and therefore can erroneously select a far away one. In case the currently used region is smaller than it should be according to the latest measurements, the algorithm can not select a nearby reference spot because it might not be part of the region. To analyze how n-G impacts IAP and UAP, we extended the basic experiment. Our results show that n-G+UAP is only slightly influenced — the average position error varies between 2.86 and 2.89 meters for $n = 1$ and 10, respectively — and hence we do not show its figures. However, n-G+IAP shows interesting results, and therefore the cdfs for different values of n are depicted in Figure 4.

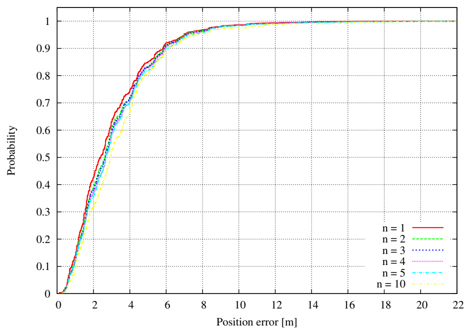


Figure 4. n-G+IAP: Position error for different values of n.

The graphs shout that our expectations are met: The position error increases with the increase of n. The curve representing $n = 1$ is always on top of the other graphs. Furthermore, the graph for $n = 10$ is always below. This yields an average position error of 2.97 and 3.43 meters for $n = 1$ and 10, respectively. Or in other words, by grouping together ten measurements before deciding if a new fingerprint data query is required, the position accuracy degrades by nearly 0.5 meters.

The results show that for n-G+IAP small values of n should be used whereas for n-G+UAP larger values can be applied without decreasing the positioning accuracy.

4.3 Size of Regions

Our fingerprint selection algorithms shape the region of a mobile device by differently combining coverage areas of access points in proximity of the device. The region of a mobile device defines the area in which the device can move around and position itself by utilizing the fingerprints it stores. In this section, we analyze the size of the regions generated by our algorithms. During the execution of our basic experiment, we counted the number of reference spots comprised by a region. Table 2 lists the average number, the maximum number and the minimum number of reference spots we observed.

	Size of regions		
Algorithm	average	max	min
UAP	129.82	130	120
IAP	11.47	23	3

Table 2. Size of regions.

The figures show that UAP covers nearly all reference spots of our operation area whereas IAP always selects only a small subset. On the average, 11.47 reference spots are selected and the maximum number is 23. These

[1] $\text{inequality}^{theoretical}_{max} = \lfloor \frac{\#measurements-1}{n} \rfloor$, whereas the number of measurements is abbreviated by #measurements.

numbers show that storage-restricted devices can benefit from the IAP approach.

The space requirements of our on-demand fingerprint selection algorithms depend on the size of the regions they produce. To get a deeper insight into this topic, the following section provides an analysis of the space requirements.

4.4 Space Requirements

In this section, the space requirements of fingerprint updates for our on-demand fingerprint selection algorithms are analyzed. Our algorithms replace the fingerprints stored on the mobile device with data requested from the repository. So, by analyzing the space requirements of updates we also get the storage requirements needed to store fingerprints on the mobile device.

The analysis is based on the following parameters: I is the set of access points visible to the mobile device; R_{AP} is the set of reference spots covered by access point AP; F_R is the total number of fingerprints at reference spot R; $F_{R,AP} \in \{0, 1\}$ tells whether there is a fingerprint for access point AP at reference spot R. For each of the on-demand fingerprint selection algorithms the space assessments are listed in Table 3.

Algorithm	Space
UAP	$\sum_{r\in(\bigcup_{i\in I} R_i)} F_r$
IAP	$\sum_{i\in I}\sum_{r\in(\bigcap_{i\in I} R_i)} F_{r,i}$

Table 3. Theoretical space assessment.

The worst case space requirements are equal for both fingerprint selection algorithms, however, this only occurs in pathological network conditions: All access points cover the same reference spots. This is why we are more interested in the average space requirements.

The positioning algorithm we selected for our experiments requires fingerprints that contain two histograms: A signal strength histogram and a frequency histogram [16]. To store such a signal strength histogram, 102 integers are required because our 802.11 network card is able to sample signal strength values in the range of 0 to 102 dBm. For the frequency histogram, ten integer are sufficient because our data shows that no access points is measured more than ten times during one sample. Together with two floating-point numbers for the two-dimensional coordinate and the MAC address of the access point, such a fingerprint can be represented by using 126 bytes $(= 102 + 10 + (4 * 2) + 6)$.

To get an impression how much data has to be transferred and stored on a mobile device, we utilize the basic experiment and count the number of fingerprints requested. The average number, the maximum number and the minimum number of fingerprints for each of the two on-demand fingerprint selection algorithms are given in Table 4. The digits in brackets list the storage utilization in terms of kilobytes round up to the next integer.

	Space		
Algorithm	average	max	min
UAP	1987.95 (251)	1993 (252)	1940 (245)
IAP	94.73 (12)	168 (22)	20 (3)

Table 4. Empirical measured space requirements.

These figures show that IAP requires at most 21 kilobytes of data which makes this algorithm highly attractive for memory-restricted devices such as sensor nodes containing only a few hundred kilobytes of memory. As IAP covers on average only 11.47 reference spots an increase of the operation area would not result in higher storage demands. In contrast, in many cases UAP selects the 1993 fingerprints covering our operation area. The reason for this is that at any location on our operation area, access points are reachable that jointly cover it completely. If we had a larger operation area, UAP would select far more data.

4.5 Discussion

In this paper, we show results for our novel on-demand fingerprint selection algorithms by combining them with one positioning algorithm. We have also repeated our experiments with the K-Nearest-Neighbors-P-Unknown algorithm [5] and the outcome shows the same trend as what we present here. So, due to page restrictions we omit an in-depth discussion of the second positioning algorithm results.

Our fingerprint selection algorithms automatically update the fingerprints stored on a mobile device if it moves around and triggers a fingerprint data query. To keep fingerprints on motionless devices up-to-date, we suggest to apply a timer to the data. Each time a fingerprint query is asked the timer is restarted. If the timer expires, a query is executed to get the latest version of the data.

To speed-up replies to fingerprint queries, the central repository could pre-calculate the intersections and unions of coverage areas of access points contained in the fingerprint data. This only needs to be done once and can be stored in a look-up table. To find the fingerprints for a given list of access points, it is just a matter of searching the look-up table.

5 Related Work

In this paper, we consider only terminal-based positioning systems, however, network-based approaches such as described in [1] and [20] also face the problem that a fingerprint database has to be commonly created and shared between different owners of access points. The problem arises from the fact that network-based approaches sample mobile devices at the access points. So, in case an operation area such as a shopping mall is covered by overlapping access points that are owned by different parties, access to these access points and the corresponding fingerprint databases has to be granted to the owners. However, the authors simplify the scenario to avoid the problem of fingerprint data distribution. They just assume that all access points covering an operation area are under the control of a single entity. Our fingerprint selection algorithms would need a few modifications to be usable for network-based positioning systems. We are not going to describe these changes here because they are out of scope of this paper.

The basic idea behind the PlaceLab research project [17] is beacon location to achieve a pervasive positioning system. The researchers of PlaceLab exploit the fact that many public available wireless network technologies regularly emit beacons. For instance, 802.11 access points broadcast beacons to maintain the network and GSM towers broadcast their existence to make it easier for cell-phones to find them. A large portion of their research work deals with 802.11-based positioning systems. In [8], the authors slightly touch the question how the fingerprint database can be distributed to a large set of mobile devices. However, they do not provide a solution for this problem and proceed with other research topics. In another paper [18], the authors discuss how the initial effort of collecting fingerprints can be reduced by bootstrapping a fingerprint database from seed data and keep it valid while fingerprints are added and removed. In contrast to our work, we do not try to reduce the initial workload, instead we are providing mechanism to keep the fingerprint data on mobile devices up-to-date and the footprint of this data small.

In [22], the authors propose clustering of the fingerprint database in order to reduce the computational requirements of the positioning algorithm. The traditional way was to compute a result for each reference spot being part of the fingerprint database, even it was not covered by any of the access points sampled by the mobile device. The approach suggested by Youssef et al. clusters the fingerprint database in such a way that only reference spots are selected that are covered by the access points the sample of the mobile device contains. As their results show, this reduces the computational requirements a lot. However, this approach does not consider solutions to the administrator's occupational therapy. In particular, Youssef et al. do not consider different algorithms for selecting variable parts of the fingerprint database in order to reflect storage capabilities of different device classes. A positive side-effect of our system is that the computational requirements are also reduced in a similar way as proposed by Youssef et al.

Compared to our approach, Lorincz et al. [19] came up with the idea of an infrastructure-less positioning system for sensor networks. In their approach, there is no infrastructure that stores the fingerprint database. Instead, this task is accomplished by so-called beacon nodes. These nodes carry the fingerprint data of their proximity. Further, the authors describe two methods how position estimates can be computed: A terminal-based and a network-based approach. In the former approach, the mobile device requests, comparable to our UAP algorithm, fingerprint data from beacon nodes within its communication range. Based on this data the device estimates its position. If the latter approach is applied, the mobile device send its signal strength measurements to surrounding beacon nodes and asks them to compute a position estimate based on their local fingerprint data. While the system proposed by Lorincz et al. bears a resemblance to our approach, they only consider infrastructure-less scenarios. Further, they provide only one way of getting the fingerprint data on the mobile device neglecting to take into account the restricted storage capabilities of sensor nodes. In addition, they do not consider updates and removals of fingerprints in their system.

6 Conclusions and Future Work

We presented our novel on-demand fingerprint selection algorithms to avoid the cumbersome and time-consuming task of manually copying fingerprint data from a single repository to all mobile devices. Our algorithms dynamically select only those fingerprints from the repository that are required to compute a position estimate; this also makes sure that the latest fingerprints are always available. To take the different storage capabilities of two classes of mobile devices into account, we presented two fingerprint selection algorithms that use different shapes for the region wherein the mobile device is located. Our empirical study shows that the position accuracy is not harmed by our fingerprint selection algorithms.

Furthermore, we investigated the frequency of fingerprint data queries and presented improvements to our basic algorithms to reduce the frequency of queries. Our investigation shows that the region in which the mobile device is located are differently shaped. We also showed that the space requirements of our fingerprint selection algorithms meet the constraints of the corresponding class of mobile devices.

In a direct extension of this work, we plan to apply our fingerprint selection approaches to tracking systems. The difference with tracking is that users move while signal strength measurements are taken. So, we want to investigate the impact of our fingerprint selection algorithms on the position accuracy of tracking systems. Furthermore, we are working on a scalable system architecture, comparable to the DNS system, to get rid of the single point of failure of the central repository. Finally, we plan to test the general applicability of our fingerprint selection algorithms for other sensor technologies such as Bluetooth.

Acknowledgments

We would like to thank Alexander Hanschke and Philip Mildner for their help collecting data for the experiments. Furthermore, the authors acknowledge the financial support granted by the *Deutschen Forschungsgemeinschaft* (DFG).

References

[1] P. Bahl and V. N. Padmanabhan. RADAR: An In-Building RF-Based User Location and Tracking System. In *Proc. of the IEEE InfoCom*, pages 775–784, 2000.

[2] 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 ACM MobiCom*, pages 50–61, 2006.

[3] P. Castro, P. Chiu, T. Kremenek, and R. R. Muntz. A Probabilistic Room Location Service for Wireless Networked Environments. In *Proc. of the UbiComp*, pages 18–34, 2001.

[4] P. Castro and R. Muntz. Managing Context Data for Smart Spaces. *IEEE Personal Communications*, pages 44–46, 2000.

[5] Y.-C. Cheng, Y. Chawathe, A. LaMarca, and J. Krumm. Accuracy Characterization for Metropolitan-scale Wi-Fi Localization. In *Proc. of the ACM MobiSys*, pages 233–245, 2005.

[6] C. Fretzagias and M. Papadopouli. Cooperative Location-Sensing for Wireless Networks. In *Proc. of the IEEE PerCom*, pages 121–131, 2004.

[7] J. Hightower and G. Borriello. A Survey and Taxonomy of Location Systems for Ubiquitous Computing. UW CSE 01-08-03, University of Washington, Department of Computer Science and Engineering, 2001.

[8] J. Hightower, A. LaMarca, and I. Smith. Practical Lessons from Place Lab. *IEEE Pervasive Computing*, 5(3):32–39, 2006.

[9] E. Kaplan and C. Hegarty, editors. *Understanding GPS: Principles and Applications*. Artech House Incorporated, second edition, 2005.

[10] T. King, T. Butter, M. Brantner, S. Kopf, T. Haenselmann, A. Biskop, A. Färber, and W. Effelsberg. Distribution of Fingerprints for 802.11-based Positioning Systems. In *Proc. of the MDM*, pages 224–227, 2007.

[11] 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 ACM WiNTECH*, 2007.

[12] 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 LoCA*, 2007.

[13] T. King, T. Haenselmann, S. Kopf, and W. Effelsberg. Overhearing the Wireless Interface for 802.11-based Positioning Systems. In *Proc. of the IEEE PerCom*, pages 145–150, 2007.

[14] M. B. Kjærgaard. A Taxonomy for Radio Location Fingerprinting. In *Proc. of the LoCA*, 2007.

[15] J. Krumm and E. Horvitz. LOCADIO: Inferring Motion and Location from Wi-Fi Signal Strengths. In *Proc. of the Conference on Mobile and Ubiquitous Systems: Networking and Services*, 2004.

[16] 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 *Proc. of the ACM MobiCom*, pages 227–238, 2002.

[17] 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 IEEE PerCom*, pages 116–133, 2005.

[18] A. LaMarca, J. Hightower, I. Smith, and S. Consolvo. Self-Mapping in 802.11 Location Systems. In *Proc. of the UbiComp*, pages 87–104, 2005.

[19] K. Lorincz and M. Welsh. MoteTrack: A Robust, Decentralized Approach to RF-Based Location Tracking. In *Proc. of the LoCA*, pages 63–82, 2005.

[20] J. Yin, Q. Yang, and L. Ni. Adaptive Temporal Radio Maps for Indoor Location Estimation. In *Proc. of the IEEE PerCom*, pages 85–94, 2005.

[21] M. Youssef. *Horus: A WLAN-Based Indoor Location Determination System*. PhD thesis, University of Maryland at College Park, 2004.

[22] M. Youssef, A. Agrawala, and A. U. Shankar. WLAN Location Determination via Clustering and Probability Distributions. In *Proc. of the IEEE PerCom*, pages 143–150, 2003.