Pre-processing of Fingerprints to Improve the Positioning Accuracy of 802.11-based Positioning Systems

Hendrik Lemelson, Thomas King, and Wolfgang Effelsberg
Dept. of Computer Science, University of Mannheim
Mannheim, Germany
{lemelson,king,effelsberg}@informatik.uni-mannheim.de

ABSTRACT

Since the first days of location-based services, the reliable estimation of a user's position has been seen as one of the key prerequisites for their success. While satellite-based systems can deliver suitable results when the user is outdoors, another solution is needed in case that he is located inside a building. Here, the position estimation with the help of 802.11 and fingerprinting algorithms promises to fill the gap. One major drawback though is the high initial effort that is needed to deploy such a fingerprinting-based 802.11 positioning system. Therefore, in this paper, novel algorithms that pre-process the data collected for the fingerprints are presented. By this pre-processing step, the amount of data that needs to be collected and therefore also the amount of time that needs to be spent collecting the data can be reduced by about 80%. Additionally, despite the reduced amount of data, the positioning accuracy cannot only be retained but even slightly improved by about 15% depending on the system's setup.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless communication*; I.6.7 [**Simulation and Modeling**]: Simulation Support Systems—*Environments*

General Terms

Algorithms, Design, Experimentation, Measurement

Keywords

IEEE 802.11, positioning systems, fingerprinting, context-aware applications, location-based services

1. INTRODUCTION

During the last few years, there has been an eager development in the area of mobile electronic devices. Not only have these devices gained in processing power, the amount of volatile and non-volatile memory, and battery runtime, but also their feature set regarding sensors and communication interfaces has grown remarkably. Among these interfaces, we find advanced wireless networking technologies like UMTS [1], 802.11 [6] or Bluetooth [7] nowadays. Besides their use for communication, these interfaces also make new and promising mobile applications possible.

One of the most exciting areas for such mobile applications are the so-called *context-aware services* (CAS) and, among these, especially the *location-based services* (LBS).

Applications belonging to the category of CASs try to improve their service by taking not only the user's input into consideration, but also available information about the user's context to be able to deliver a more user adapted service experience. Applications belonging to the category of LBSs essentially do the same as CASs. The major difference though is, that they only rely on information about the user's current location and leave the other context information aside. This approach, while remarkably reducing the development complexity for such applications, still offers enough information to highly leverage the usefulness of the offered services.

As long as the user is outdoors, LBSs can rely on satellite-based positioning systems like the *Global Positioning System* (GPS) [8] or the upcoming Galileo [3] to estimate a user's position. These – in most cases – offer a reasonable precision to the CAS and LBS applications. This though is not given for the inside of buildings or even for highly metropolitan areas where large buildings shade the user's device from the satellite signals. In these – in reality quite frequent – cases, satellite-based systems tend to be either very imprecise or even fail to deliver a position estimation. This is a major drawback for both CASs and LBSs.

Positioning based on 802.11 can offer a solution in such cases. Not only in urban, but also in residential areas, in office buildings and shopping malls, one can almost always receive the signals of at least a few 802.11 access points nowadays. These signals can be used to reliably estimate a user's position. To accomplish this task, in the past few years, several different approaches and algorithms have been proposed. Of these, *fingerprinting* has emerged as one of the more promising ones.

Generally, fingerprinting-algorithms are split into two phases. In the first, so-called *training phase*, a set of reference positions has to be selected in the area the system shall cover. Afterwards, for each reference position, a dataset is collected and stored in a database. This dataset shall reflect the unique properties of the signal space at the reference position it was collected at. It therefore is also called the *fingerprint* of that reference position. The second phase is called *position determination phase*. In this phase, the system is already operational and in use. If a user wants to determine his location, he as well creates a fingerprint of the signal space at his – yet unknown – position. The positioning systems then compares this live-fingerprint to all the fingerprints in the database. For this com-

MELT'08, September 19, 2008, San Francisco, California, USA.

parison, several different metrics, like e.g., an Euclidian distance measure, are possible depending on the underlying sensors of the system and the system itself. If the comparisons are finished, the system selects the position of the best-matching fingerprint and returns this position as a position estimate to the user. Despite the already quite promising results regarding the positioning accuracy of such systems, one major problem that the fingerprinting algorithms still have in common is the need for the very time-consuming training phase.

This high initial effort is one of the major reasons why 802.11-based positioning systems are still struggling to make their step from a scientific to a convenience product. In this paper, we therefore propose novel algorithms to setup a fingerprinting system. Our algorithms reduce the time needed to collect the necessary data during the training phase tremendously without losing positioning accuracy. To accomplish this task, our algorithms pre-process the data that has been collected during the training phase. To create a fingerprint, not only the samples that have been collected at a single reference position are considered. Instead, also samples that have been collected at adjacent positions are taken into consideration. By doing so, we can reduce both the amount of data that needs to be collected at each reference position and the time needed to collect the data by about 80% without losing positioning accuracy. In fact, our algorithms even increase the positioning accuracy in most cases.

The remainder of this paper is structured as follows. In Section 2 we give a short overview of past and current developments in the area of indoor positioning. Section 3 describes the theoretical background and the novel ideas of our algorithms. This is followed by an overview on our evaluation environment in Section 4. The results of our novel algorithms' evaluation are presented in Section 5 just before we conclude the paper in Section 6.

2. RELATED WORK

When we talk about position estimation, we have to distinguish between systems for indoor usage and those for outdoor usage. The area of the latter ones is generally covered by satellite-based systems like GPS [8] that can deliver precise position estimations, have a good coverage, and are convenient to use. Even though there exist other approaches that use e.g., GSM for positioning outdoors [15, 12], these have less practical relevance [14]. Exceptions though are situations in which the signals of the satellites are heavily attenuated or even totally blocked by large buildings or other structures. In these cases, alternative systems can offer some advantages.

Compared to the outdoor sector, lots of development was and still is done in the area of indoor positioning. Early systems like *Active Badge* [19], *Active Bat* [5], or *Cricket* [16] use infrared light or ultrasonic pulses to estimate a user's position. In contrast to these, most current systems try to use radio signals for position estimation [20, 13, 18]. The advantage of this approach is that the signals are not stopped by walls and, especially when using 802.11, there is already infrastructure available in many cases.

The main problem of such systems though is the very bad propagation behaviour that radio signals have inside buildings. Generally, here we have to cope with diffraction, scattering, shading and multipath propagation [17] which make it very hard to create proper propagation models and to anticipate the signal's properties at a certain position in space and over time. Therefore, many systems nowadays use the fingerprinting approach.

An example for a system using the fingerprinting approach is the *RADAR* positioning system [2]. RADAR uses 802.11 and was the first system to introduce the aforementioned fingerprinting approach together with 802.11 to determine a user's position. To compare live-measured and stored signal strength measurements, the Euclidian distance in signal space between the two is computed. Afterwards, the algorithm selects the position of the fingerprint with the closest distance (in signal space) to the live data as a position estimate. With this approach, RADAR achieves an average positioning accuracy of about three meters.

Another representative of the class of fingerprinting algorithms is described by A. Haeberlen in [4]. In difference to RADAR, this algorithm, to which we will from now on refer using the term *Rice Gaussian*, uses a probabilistic approach based on Gaussian distributions to compute the probability of a user being at a certain reference position. The average positioning accuracy of the Rice Gaussian algorithm is about two meters.

While by using only the signals of 802.11 base stations, an average positioning error of less than two meters is hard to achieve in our experience [10], this can be accomplished by using additional sensors. One example for a system following this approach of *sensor fusion* is the *COMPASS* positioning system [11]. This system uses not only signal strength measurements, but takes also the user's orientation into consideration while determining his position. With the help of this additional information, the system can better handle the signal attenuation caused by the human body and therefore deliver improved positioning results. This lowers the average positioning error to a value of about 1.65 meters.

Also many other approaches have been introduced during the last years in order to estimate a user's position indoors with the help of 802.11 and fingerprinting. To our best knowledge though, none of these algorithms pre-processes the data collected during the training phase to reduce the amount of data needed while still retaining the positioning accuracy.

3. ALGORITHMS

In this section, we introduce the basic techniques used by our algorithms as well as by the Rice Gaussian algorithm which was taken as benchmark during our evaluation. Further, an overview of the ideas and the advantages of our algorithms is given.

3.1 Probabilistic Fingerprinting in General

Our novel algorithms as well as the Rice Gaussian algorithm use a probabilistic metric based on signal strength observations to estimate a user's position. This approach is sketched in the following. For further in-deep details, please refer to [4].

We decided to use the Rice Gaussian algorithm as benchmark, because according to Youssef et al. [20] it seems that probabilistic algorithms generally are better suited for positioning than deterministic ones.

During the training phase, at each reference position, samples of the signal strengths of all receivable access points are collected. Then, for each access point contained in the samples of one reference position, the average signal strength and the standard deviation of the signal strength measurements are computed. These values are stored in the fingerprint of the corresponding reference position. As a result, each fingerprint contains a pair of average signal strength and standard deviation for each access point of which signals have been received at the reference position the fingerprint belongs to. During the position determination phase, samples of the signal strength of each receivable access point are collected. These samples then are compared to the values stored in the fingerprints. This is done by computing a normalized probability for each reference position and selecting the position with the highest probability as a position estimate.

In the following, we describe the computation of the single probabilities in a more detailed fashion. This is necessary to understand what our novel algorithms essentially do.

For each fingerprint and each access point ap contained in the live samples, we compute a probability $P(s_{ap})$. This probability states how probable it is to receive a signal of access point ap with a signal strength s_{ap} at the reference position r to which the fingerprint belongs to.

To compute the probability, the live measured signal strength for access point ap is matched with the density function of a normal distribution with the parameters for access point ap taken from the current fingerprint. As the probability for a single value and a given continuous probability distribution is zero per definition, the probability for an interval of $s \pm 0.5$ is computed instead.

While for the width of the interval another value could also be used generally, we decided to use the value introduced by [4] to map the discrete signal strength values the hardware delivers (-102dB to 0dB) onto the continuous numberspace of the normal distribution.

$$P_{r,ap}(s_{ap}) = \int_{s-0.5}^{s+0.5} df(\mu_{r,ap}, \sigma_{r,ap}) \tag{1}$$

For the access point ap, df is the density function of the normal distribution with an average of $\mu_{r,ap}$ and a standard deviation of $\sigma_{r,ap}$. These values are taken from the fingerprint of reference position r.

This step is repeated for all the access points that are contained in the live samples, which finally, for each reference position, leads to one overall probability P_r of the user being at reference position r:

$$P_r(s) = \Pi_{i=1}^{n} P_{r,ap_i}(s_{ap_i}) \tag{2}$$

Here, n is the number of access points that are found in the collected live sample, s_{ap_i} is the signal strength collected for the access point ap_i and r is the reference position for which the overall probability is currently computed.

After the computations are finished for all the fingerprints, the algorithm selects the reference position with the highest overall probability of the user being there as a position estimate.

3.2 Neighbour Considerations

During our earlier research [10], we have seen that several important factors exist that have an influence on the results of probabilistic fingerprinting algorithms. For example, the number of received access points and the quality of the hardware have a major influence on how good or bad a fingerprinting algorithm performs in terms of the average positioning error. The same is true for the number of samples collected to create a fingerprint and the distances between the reference positions. For example, earlier publications [4] as well as our own research indicate that at least about 20 to 30 training samples are needed for each fingerprint to achieve satisfying results during a later positioning with common fingerprinting algorithms.

For a fixed position, the collected signal strength samples have a natural variation we are not able to overcome. The reason for this variation is the bad propagation behaviour of radio signals in indoor environments where we have to face signal attenuation, signal defraction, scattering, and multipath propagation. To compensate these variations when creating a fingerprint, a sufficiently large number of samples is generally needed to allow the average signal strength value to stabilize.

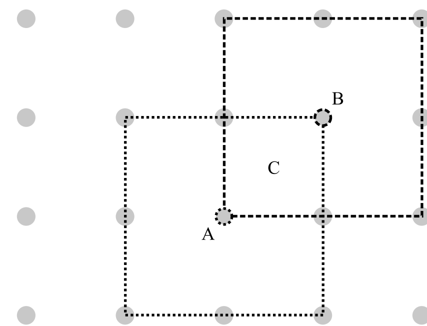

Figure 1: The area C is – amongst others – covered by both the fingerprints for reference positions A and B

Having in mind the time consuming task of collecting these samples for each single reference position, the goal of our work is to reduce this required effort in a transparent and easy implementable way.

A user who is located somewhere in a building during the position determination phase generally does not want to stand still and wait for the collection of enough samples to let the averages stabilize. As a result, the signal strength measurements taken during the position determination phase will scatter around the values stored in the fingerprints that were collected at the reference positions close to the user's current whereabout.

Additionally, the user will also almost never be located exactly at a reference position. Instead, he will stand somewhere in between several reference positions in most cases. Therefore generally more than one fingerprint of adjacent positions around the user will give a suitable match to the live data collected at the user's real position.

Considering that the user will be located in between several reference positions most of the time and that the signal strength of an access point at a given position has an inevitable fluctuation indoors, we propose new algorithms that take these insights into account in order to increase the positioning accuracy on the one hand while reducing the amount of needed samples to be collected at each reference position on the other hand.

3.2.1 Equal Weighting

We developed an algorithm that considers not only the samples that have been collected at a certain reference position itself but also the samples that have been collected at all the adjacent reference positions in order to create the fingerprint. If the current reference position has m adjacent positions and n samples are available for each position, this leads to $m * n$ samples used to compute the fingerprint. The samples in this case are *equally weighted*, which means that all the samples from the reference position as well as those from the adjacent positions have an equal influence on the resulting fingerprint. The fingerprint itself does no longer represent only a single reference position, but instead represents an area that is defined by the adjacent reference positions (see Figure 1).

At this point, we want to note that even though we use more samples to create the fingerprints, we do not need to collect more samples. The essence of what we do is to re-use the same samples several times for different fingerprints.

3.2.2 Unequal Weighting

The equal weighting algorithm works quite well for smaller gridsizes. Nevertheless, we have to face a decrease in the positioning accuracy of the algorithm in case we increase the distance between the reference positions.

Investigating this behaviour, we realized that the samples collected at the adjacent positions increasingly blur the resulting fingerprint if the distances between the reference positions get too large. To overcome this, we extended our equal weighting algorithm by adding a weight factor w. This factor defines the influence ratio that samples of adjacent positions shall have on the final fingerprint.

Our idea was to combine the samples from the reference position itself with those of the adjacent positions in such a way, that the influence of the latter set of samples can be adjusted by varying w.

To accomplish this, for each set, we at first compute the representing normal distributions for all the access points that are contained in the samples. This gives us two temporal fingerprints. One represents the reference position itself, the other represents the surrounding area.

Afterwards, for each access point contained in at least one of the two temporal fingerprints, we compute a new distribution based on w and the two distributions from the temporal fingerprints. The computation is carried out using the following two formulas (see Equation 3) common for the combination of normal distributions:

$$\begin{aligned} \mu_{new} &= w * \mu_{adj} + (1-w) * \mu_r \\ \sigma_{new} &= \sqrt{w^2 * \sigma_{adj}^2 + (1-w)^2 * \sigma_r^2} \end{aligned} \tag{3}$$

The information about the resulting distribution for that access point then is stored in the new fingerprint for use during the position determination phase. Even though we lose information by reducing the samples to only these two values, this approach is valid. It is essentially how the probabilistic algorithms that we use act when creating the fingerprints.

After the computation is finished for the current reference position, the newly computed fingerprint embodies two values (μ_{new} and σ_{new}) for each access point that was contained in the original set of samples for that position.

4. EVALUATION SETUP

The following section briefly describes the environment in which the data collection for our evaluation took place. Additionally, an overview of the data we collected as well as of the hard- and software used for collecting the data is given.

4.1 Evaluation Environment

The environment where we conducted the data collection for our evaluation is the second floor of the building A5,6 B at the University of Mannheim (Germany) in which the offices of our group are located. The story is split up into two hallways, several offices, and three smaller rooms in the middle of the hallways (see Figure 2). The two hallways are measured 30 times 6 meters and 15 times 4 meters respectively, covering an area of approximately 240 square meters.

4.2 Data Collection

To get a sufficient amount of data for the evaluation, 612 reference positions were laid out in the evaluation environment using a grid of 0.5 meters side length (see the grey dots in Figure 2). The samples collected at these positions are the foundation for the fingerprint databases used during our evaluation.

Additionally, 70 points were spread randomly over the hallways (see the black dots in Figure 2) to which we will refer to as *live positions* from now on. The samples collected at these positions are used to emulate a user requesting a position estimation during the following emulation of the position determination phase.

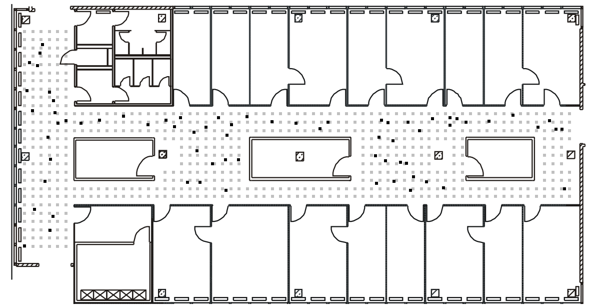

Figure 2: Positions at which we collected training and live data to build the fingerprint databases and to emulate a user requesting a position estimate

110 samples were collected at each reference position and live position during the data collection process. The collected samples contain a timestamp, the MAC address of the collecting 802.11 card, the current position, and for each received access points the MAC address, the channel, and the signal strength. All samples were stored in a logfile for easy reference during the following evaluation.

4.3 Hard- and Software

To collect the samples, we used an IBM Thinkpad R51 laptop computer running Suse Linux 10.1. The laptop computer was equipped with a plug-in Lucent Silver PCMCIA card.

On the software side, the samples were collected with the LocEva framework [9]. The application to collect the samples uses the Java Native Interface and a wrapper written in C to interact with the operating system kernel's wireless extensions interface. This makes it possible to request the communication parameters and connection information directly from the 802.11 card's driver.

5. EVALUATION

With the huge amount of collected data, we performed an intense evaluation of our algorithms. Using our set of emulation tools, each algorithm had to perform 250 runs per setting and the average positioning accuracy as well as the standard deviation of the error and a cumulative error distribution for the runs was recorded. During each run, the algorithm for each reference position was given a number of randomly selected training samples to create the fingerprint. Then, for each live position, we emulated a user in the position determination phase requesting a position estimation. This was done by supplying the algorithm with randomly selected live samples collected at that live position. Afterwards, we compared the position estimated by the algorithm to the real position and computed the physical error distance between those two. The procedure was conducted with the Rice Gaussian algorithm from [4] and with our equal weighting and unequal weighting algorithms using different sets of reference positions. The sets were built by selecting grids of reference positions with grid distances ranging from 0.5 meters to 4.5 meters. We will call this inter-position distance *gridsize* from now on.

Especially for larger gridsizes, the layout and the origin of the reference grid have a major influence on the positioning accuracy. We therefore varied the grid origin for the different gridsizes and selected offsets that maximize the number of available reference positions.

In addition to the varying gridsize, all algorithms had to perform each run with not only 20 but also four training samples given per

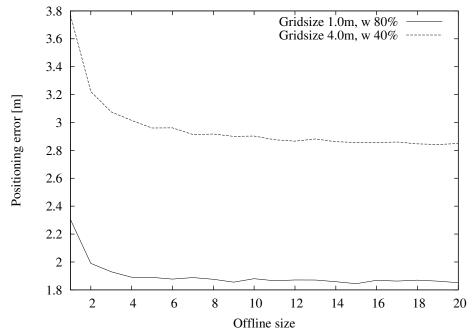

Figure 3: The avg. positioning error of our algorithms for different training set sizes and gridsizes

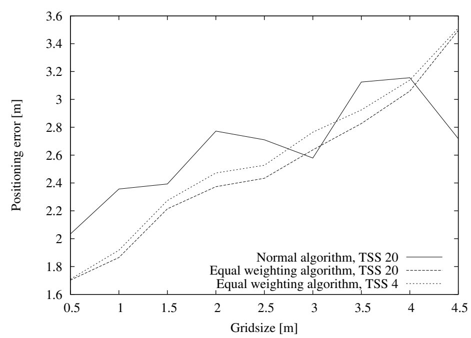

Figure 4: The equal weighting algorithm compared to the Rice Gaussian algorithm

reference position to create the fingerprints. This parameter will be referred to as *training set size* (TSS) for the remainder of this paper. The value 20 was selected based on earlier research [10]. For this value, the gain in positioning accuracy starts to saturate when using the Rice Gaussian algorithm. The value four was selected based on preliminary experiments with our own novel algorithms, for which the average increase of the positioning accuracy already starts to saturate at a value of four training samples (see Figure 3).

5.1 Equal Weighting

Our equal weighting algorithm performs quite well as long as the distance between adjacent reference positions stays below a certain threshold. For gridsizes between 0.5 meters and 2.0 meters, our equal weighting algorithm outperforms the Rice Gaussian algorithm constantly (see Figure 4). With a training set size of 20 samples, the gain of positioning accuracy by our algorithm is about 20% or in absolute numbers about 35 cm. If we reduce the training set size to only four samples, the supremacy of our new algorithm is even higher and we have an average increase of about 67 cm compared to the Rice Gaussian algorithm. An interesting insight also is, that while the accuracy of the Rice Gaussian algorithm decreases noticeably when reducing the training set size to four samples, our equal weighting algorithm still delivers almost as good results as with a training set size of 20 samples (see Figure 4).

The fluctuations in the positioning accuracy of the Rice Gaussian positioning algorithm especially for the larger gridsizes are caused by our selection of the reference positions. Due to the layout of our testbed, for certain gridsizes the positions used to emulate a user in the position determination phase were quite far away from the positions selected for the reference grid. This resulted in a worse

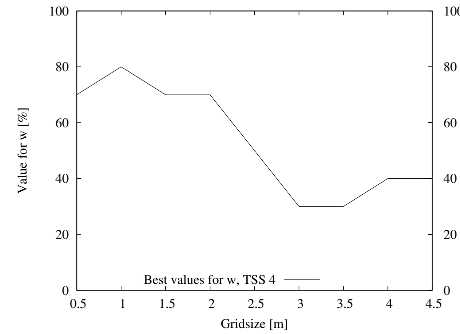

Figure 5: Weights that performed best for different gridsizes

positioning accuracy. Even though we tried to minimize this effect by adapting the grid origin to maximize the number of available reference positions for each gridsize, it still could not be completely eliminated for our local testbed.

5.2 Unequal Weighting

Due to the fact, that our equal weighting algorithm was not able to outperform the Rice Gaussian algorithm for larger gridsizes, we decided to further elaborate the reason for this issue. If for a certain gridsize, the average error of the Rice Gaussian algorithm for that gridsize is smaller than the gridsize itself, then our algorithm is no longer able to deliver satisfying results. In such a case, the samples for the adjacent reference positions are – regarding the metric of the algorithm – different enough for the Rice Gaussian algorithm to properly distinguish between them and to make proper position estimations. For our equal weighting algorithm though, the samples are already too different to get an advantage by taking them into consideration. For larger gridsizes, the merging of the samples even decreases the positioning accuracy below that of the Rice Gaussian algorithm.

We chose to vary the weight in 10%-steps from 0% to 100% during our evaluation and computed the fingerprints for each reference position according to Section 3.2.2. We then checked which value performed best for the different gridsizes. The result of these experiments can be seen in Figure 5.

They indicate that the more the gridsize increases, the smaller should the influence of the surrounding samples be to achieve satisfying results.

If we - for each gridsize - compare the result of the unequal weighting algorithm using the best suited percentage for the current gridsize to those of the Rice Gaussian and the equal weighting algorithms, the concept of adapting the influence clearly shows its advantages. As we can see from Figure 6, for smaller gridsizes, the unequal weighting algorithm delivers results as good as those of the equal weighting one. For gridsizes larger than 3.5 meters though, the former algorithm clearly performs better than the Rice Gaussian and also the equal weighting algorithm.

Even for the gridsizes larger than 4 meters, where the Rice Gaussian algorithm outperforms our equal weighting algorithm, the unequal weighting algorithm still delivers better results.

If we continue to increase the gridsize even beyond the values shown in Figure 6, the results of the Rice Gaussian and the unequal weighting algorithms finally converge. For these large gridsizes, the weight factor w approaches the value zero. This means, that for a value of zero for w, both the unequal weighting and the Rice Gaussian algorithm perform essentially the same steps to compute a position estimate and therefore also deliver the same results.

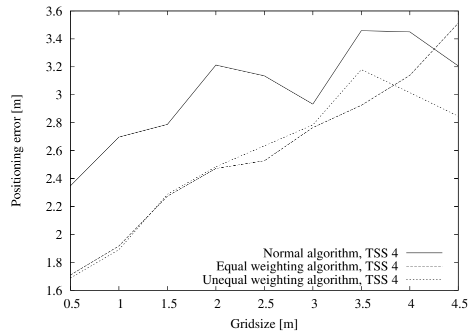

Figure 6: The unequal weighting algorithm compared to the equal weighting and the Rice Gaussian algorithm

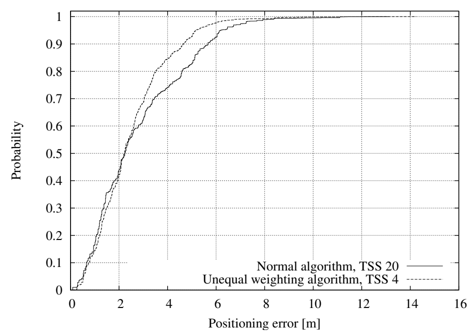

Figure 7: Cumulative error distribution for the Rice Gaussian algorithm compared to that of our novel unequal weighting algorithm

Taking a look at the cumulative error distributions of the Rice Gaussian algorithm and our unequal weighting algorithm, we can see even more advantages of our novel algorithm besides the better average positioning accuracy. In Figure 7, the error distributions of the two algorithms for a gridsize of two meters are depicted. As we can see, the unequal weighting algorithm has an average positioning error of less than 5.18 meters in 95% of all cases. For the Rice Gaussian algorithm, this value is 6.18 meters, which means a difference of about 15% or 100 cm.

While these improvements do not look too exciting at a first glance, it is very important to understand that our novel unequal weighting algorithm achieves these results using only four training measurements per reference position. In comparison, the Rice Gaussian algorithm needs 20 training measurements to achieve its results. Taking the time needed to collect these samples into consideration, our algorithm therefore offers the possibility to decrease the needed time by about 80%.

6. CONCLUSIONS

In this paper, we presented two novel 802.11-based positioning algorithms. Our algorithms pre-process the available data used to create the fingerprints and by such increase the positioning accuracy. At the same time, they decrease the amount of time needed to collect the training data. This is achieved by reducing the required samples per reference position from 20 to no more than four samples. Taking the time needed to collect the samples into consideration, this is an improvement of 80%.

During the pre-processing, the samples collected at adjacent reference positions are combined into one fingerprint. This fingerprint then no longer represents only one single position, but rather an area surrounded by the adjacent reference positions. This takes into account, that the user of an indoor positioning system based on fingerprints is located in between the reference positions most of the time. As we also re-use the samples collected at one reference position several times, we can reduce the overall amount of samples needed per reference position tremendously.

7. REFERENCES

[1] 3GPP. Work programme for the standarization of Universal Mobile Telecommunications System (UMTS). TR 00.01U, 3rd Generation Partnership Project (3GPP), 2000.

[2] P. Bahl and V. N. Padmanabhan. RADAR: An In-Building RF-Based User Location and Tracking System. In *Proc. of the 19th IEEE Conf. on Computer Communications (InfoCom)*, Tel Aviv, Israel, March 2000.

[3] European Union - Directorate-General for Energy and Transport. Galileo: European satellite navigation system. Website: `http://ec.europa.eu/dgs/energy_transport/galileo/index_en.htm`, 2005.

[4] 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 ACM Conf. on Mobile Computing and Networking (MobiCom)*, New York, NY, USA, Sept. 2004.

[5] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster. The Anatomy of a Context-Aware Application. *Wireless Networks*, 8(2-3):187–197, March 2002.

[6] Institute for Electrical and Electronics Engineers, Inc. ANSI/IEEE Standard 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. Website: `http://standards.ieee.org/getieee802/`, 1999.

[7] Institute for Electrical and Electronics Engineers, Inc. Core Specification of the Bluetooth System Specification Volume 1. Website: `http://ieee802.org/15/Bluetooth/`, 2001.

[8] E. Kaplan and C. Hegarty, editors. *Understanding GPS: Principles and Applications*. Artech House Incorporated, 2nd edition, Dec. 2005.

[9] 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 ACM Int. Workshop on Wireless Network Testbeds, Experimental evaluation and CHaracterization (WiNTECH)*, Montreal, Quebec, Canada, Sept. 2007.

[10] 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. Symp. on Location- and Context-Awareness (LoCA)*, Oberpfaffenhofen, Germany, Sept. 2007.

[11] 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 *Proc. of the 1st ACM International Workshop on Wireless Network Testbeds, Experimental evaluation and CHaracterization (WiNTECH)*, Los Angeles, CA, USA, Sept. 2006. ACM Press.

[12] 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 (Pervasive)*, Munich, Germany, May 2005. IEEE Computer Society.

[13] K. Lorincz and M. Welsh. MoteTrack: A Robust, Decentralized Approach to RF-Based Location Tracking. *Personal and Ubiquitous Computing*, Oct. 2006.

[14] C. D. Malcolm Macnaughtan, Craig Scott. *Next Generation Wireless Networks*, volume 598, chapter A Series of GSM Positioning Trials, pages 195–229. 2002.

[15] M. Mouly and M.-B. Pautet. *The GSM System for Mobile Communications*. Telecom Publishing, 1992.

[16] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The Cricket Location-Support System. In *Proc. of the 6th ACM Conf. on Mobile Computing and Networking (MobiCom)*, Boston, MA, USA, Aug. 2000. ACM Press.

[17] T. S. Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall PTR, 2nd edition, Dec. 2001.

[18] A. Varshavsky, A. LaMarca, J. Hightower, and E. de Lara. The SkyLoc Floor Localization System. In *Proc. of the 5th Int. Conference on Pervasive Computing and Communications (PerCom)*, pages 125–134, White Plain, NY, USA, March 2007.

[19] R. Want, A. Hopper, V. Falcao, and J. Gibbons. The Active Badge Location System. *ACM Transactions on Information Systems*, 10(1):91–102, January 1992.

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