Automatic Calibration of Large-scale WiFi-Positioning Systems

Florian Gschwandtner
Mobile and Distributed Systems Group
Ludwig-Maximilians-Universität
Munich, Germany
Email: florian.gschwandtner@ifi.lmu.de

Peter Ruppel
Mobile and Distributed Systems Group
Ludwig-Maximilians-Universität
Munich, Germany
Email: peter.ruppel@ifi.lmu.de

***Abstract*—The growing market for location-based services causes a great demand for positioning systems with high accuracy. In addition to GPS, there are several WiFi positioning systems which allow a cost-effective indoor and outdoor positioning. But despite the fact that they provide accurate measurements in indoor environments, their outdoor performance is not as good as with GPS. This is due to the fact that outside buildings the access point density is typically lower than indoors and radio signals propagate in a different way. In order to enhance accuracy, modern WiFi positioning systems allow calibrating the radio maps by the use of empirical measurements. But in large-scale environments it is nearly impossible to calibrate radio maps manually because of the high expenditure of human labor.**

In this paper we present an approach for the automatic calibration of WiFi positioning systems assisted by another reference positioning system. By constantly observing all measurements, e.g. on a mobile calibration unit or by a calibration server, it is possible to decide at any moment and place if a calibration is possible and necessary. In that case the calibration process is automatically started and afterwards the accuracy improvements achieved are evaluated. By its modular design the described framework is able to use different reference systems for calibration. Also a combination of multiple systems is possible. A prototype has been implemented and tested at Munich Airport in Germany. The test results prove the applicability of the approach and indicate good accuracy improvements.

I. INTRODUCTION

The use of stable and widely spread standards allow the production of cost-effective WiFi components and nowadays WiFi systems are not only used in small home and business offices but also in large scaled environments. They e.g. allow access to a company's network or provide Internet access to an entire city. With increasing regularity these systems are applied not only for communication but also to locate mobile network components by using network or terminal based WiFi-positioning systems (WPS). These systems mostly measure received signal-strength (RSS) or signal-to-noise-ratio (SNR) between a mobile device and multiple access points to obtain the device's position.

Small battery powered WiFi tags can be attached to important assets to allow the tracking of those objects. The position can then be used e.g. to improve the company's disposition system. The accuracy of many WPS can be improved by calibration. During calibration, sample measurements are taken to allow a better accuracy on position calculation. This procedure requires big effort, especially in large-scale environments as the calibration has to be done manually until now.

In this paper we describe a framework which allows the automatic calibration of large-scale WiFi-positioning systems. Therefore a reference positioning system (RPS) with a higher accuracy assists in getting the real position of a mobile terminal, which then is used to calibrate the WPS. By applying multiple constraints to the position, the framework is able to decide for each location if a calibration is needed. This makes it possible to consider information about the environment, which is useful to adapt the framework to diverse use cases. The modular architecture allows to use different reference systems as source for the calibration and makes it also independent of the WPS. To demonstrate the framework's capabilities we implemented a prototype at Munich Airport in Germany.

The remainder of the paper is organized as follows: First we describe general problems for WPS calibration. Then we discuss the framework's architecture and its capabilities in Section 2. In Section 3 the experimental setup at Munich Airport is shown and evaluated. Section 4 covers related work done in the area of automatic calibration of wireless positioning systems and finally we conclude our work in Section 5.

II. GENERAL CALIBRATION PROBLEMS

WiFi positioning is mostly achieved by the measurement of the RSS or SNR between access points and mobile terminals. Based on these measurements the position of a mobile terminal can be computed by estimating the distance between the terminal and each access point in range. Due to the fact that the RSS is always dependent on several factors like multi path propagation and fading, it is not easy to derive the distance from the signal strength. While stationary objects like walls and buildings can be considered, it is almost impossible to take into account the error caused by other mobile objects.

Because of that most positioning systems allow a calibration by using empirical measurements to refine the system's theoretical models. During calibration a mobile terminal must be located within the WiFi coverage area and must not move. After the user has specified the real position of the mobile terminal, the system measures the RSS and accordingly modifies its radio map. That radio map describes the signal propagation of the coverage area.

This method works for indoor areas quite well, but is hard to deploy in outdoor and large-scale areas because of the high expenditure of human labor. In addition to that, wide areas are too dynamic for just one initial calibration. They require regular updates to the radio map.

Normally, WiFi positioning systems have their own coordinate system to calculate and display an estimated position P_{est} of a mobile terminal. This is acceptable for manual calibration because a human can specify the real position just by clicking on a map or by manually entering the correct coordinates. But if this procedure should be done automatically, the calibration system (CS) must get the correct position of a terminal by a RPS and convert it to the target's coordinate system. This is achieved by using a suitable translation between any participated system and a common coordinate system (CCS) which is used by the framework.

III. An Approach for automatic calibration of WiFi Positioning systems

During calibration the terminal must not move, so that suitable results can be achieved. Like mentioned before this is no problem in case of manual calibration because a person who is performing the procedure is able to ensure that the mobile terminal does not move during the calibration progress. A framework which tries to automatically calibrate a positioning system encounters the problem that it is not able to guarantee that. That's because it can not predict or influence actions done by the terminal as it is only observing positions. For that reason it has no information about when the terminal will start moving again. Actually, depending on the used RPS, it's difficult to ensure that it will be detected when a mobile device has stopped moving.

The amount of time needed by a calibration process depends not only on the positioning system but also on the number of calibrations that have been made before. This is due to the fact that for an update of the radio map all measured calibration points need to be considered. It is even more important to know how long a mobile terminal will keep motionless, because with the increasing amount of calibrations the probability will increase that a calibration progress will fail due to a moving terminal.

In this section we describe the modules of the framework as shown in Figure 1. The communication of these modules is not dependent on a specific protocol or underlying infrastructure. Therefore we will not address this in detail.

The interaction between the modules is as follows: At first the sensor module detects that the mobile terminal is not moving. Then it transmits its position to the calibration core which compares it with the position received by the WPS as mentioned above. If this comparison shows that a calibration could optimize the system, the calibration core polls every available constraint to check if they also approve the calibration. These polls result in a value which describes the probability that a calibration is executed. After the execution,

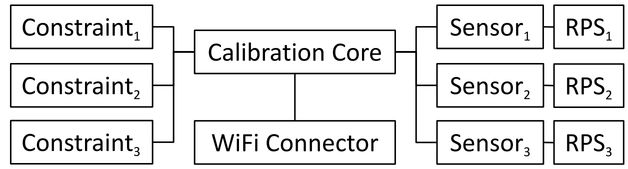

Fig. 1. Framework architecture.

the sensor again is polled for the current position of the mobile terminal. If a movement is detected, the calibration will be reversed.

A. Sensor module

A sensor module is the interface to a reference positioning system S_i which can provide a good accuracy to determine the real position of a mobile terminal. Since positioning systems with really high accuracy like differential GPS are too expensive to be used in mobile terminals, our reference systems must only meet a lower condition: Their accuracy needs to be better than the one of the WPS. One can even use a system which provides a better accuracy only in parts of the covered area. In that case constraint modules have to be specified (Section III-B) which ensure that this sensor is only used in those sub areas.

The sensor module has to fulfill multiple tasks. First it must be able to detect that a mobile terminal has stopped moving. Given the fact that S_i is not a perfect positioning system, multiple position queries will result in multiple positions even if the object is not moving. So slow movements and stops must be distinguished. In Section IV-A the method employed in our prototype is explained.

Additionally it is important to consider specific characteristics of the underlying positioning system. Knowing that the mobile terminal is not moving is only relevant if the framework can trust the measurements. Therefore the sensor module has to combine all information obtained from S_i to ensure that the values used for movement detection are reliable. After receiving the result, the sensor module must translate them to the CCS. This is necessary because it is the only coordinate system supported by the calibration core (Section III-C).

The sensor module requires permanent position updates which is the reason why it should be implemented as close to the RPS as possible to prevent unnecessary load to communication systems. The split into two modules for gathering position information (sensor module) and performing a calibration (calibration core) enables the system to send position information only if it is relevant to the calibration core.

B. Constraint module

The optional constraint modules allow adjusting the framework to a given environment. They offer the capability to affect the probability that a calibration is performed at a given position. Therefore the calibration core polls every available constraint module and applies them as described in section III-C.

Every constraint returns a value between 0 and 1 expressing the probability that a calibration should be executed at a given position. A value of 0 means that calibration should be denied whereas values close to 1 indicate that the position is convenient for calibration. In addition every constraint module has a weight $\omega \in N$ which allows to priorize one constraint over another.

The constraint modules can be used to allow the usage of additional information within the framework, e.g. they can restrict the calibration in selected areas by returning 0 for points lying inside. Further examples are described in Section IV-A.

C. Calibration core

This is the central part of the framework where all information from sensors and constraints are merged together to decide whether a calibration should be performed for a given position. A position from a sensor S_i is received if the sensor detects that the corresponding mobile terminal is not moving. The calibration core now knows the reference position P_{ref} of that mobile terminal. Using the WiFi Connector module (Section III-D), which acts as an interface to the WPS the calibration core can also obtain P_{est} returned by the WiFi system. The quality of the WiFi positioning is derived by calculating the Euclidean distance d between P_{ref} and P_{est}. Since every positioning system S_i is subject to a positioning variation Δ_{S_i} the framework must ensure that the distance between the positions is greater than the variation sum of the involved systems.

$$d_{var} = \Delta_{S_i} + \Delta_{WPS} \tag{1}$$

If d is greater than d_{var} we can expect an accuracy improvement by performing a calibration.

Some positioning systems do not allow querying the current position because they use a position list of all recognized terminals which is updated at a given interval. In that case the additional time between query and actual measurement has to be considered, too.

If Equation 1 is fulfilled the constraint modules have to be applied. As mentioned in Section III-B every constraint C_i returns a value $C_{i_{val}}$ between 0 and 1 and a weight ω_i. For all available constraints their returned values and weights are taken into account using the formula

$$prop(pos) = \begin{cases} \frac{\sum_{i=1}^{n} C_{i_{val}} \omega_i}{\Omega} & \text{if } \forall C_{i_{val}} \neq 0, \\ 0 & \text{else} \end{cases} \tag{2}$$

with $\Omega = \sum_{i=1}^{n} \omega_i$. The value of $prop(pos)$ represents the probability that the calibration will be executed for a given position pos.

As mentioned above a calibration process can take a while so the framework must also poll the current position of the mobile terminal after the execution. If that poll indicates that the mobile terminal was moved during the calibration, the modification to the positioning system must be reversed. If the WiFi positioning system doesn't support this procedure, the invalid calibration can affect the position accuracy negatively. Again this shows how important it is to detect movements and to specify constraints which decrease the possibility of failed calibrations.

D. WiFi Connector module

This module acts as an interface between the framework and the WiFi positioning system. It has to support the polling of terminals' current positions. Additionally it has to provide a function to start a calibration and must be capable of translating between the coordinate system of the WPS and CSS. If supported by the WPS, the module should be able to remove unwanted calibrations.

Besides these functions the WiFi Connector is required to compute and store the expected duration $T_{expected}$ of one calibration process. As already mentioned in Section II, it depends both on the used positioning system and on the number of calibrations executed so far. Therefore this value must be refreshed after each calibration. We recommend using the equation

$$T_{expected} = T_{last} + \Delta_T \tag{3}$$

with T_{last} as the last observed duration. The parameter Δ_T depends on the WPS and should be derived from manual test calibrations.

IV. Experimental setup

To ensure that the framework is working properly, we implemented it at Munich Airport in Germany. There, over 60 outdoor access points provide 802.11 WiFi access to the company's network, covering the whole airport campus. The WiFi system is extended by a WPS which enables positioning of all assets equipped with a WiFi interface. Because the system is infrastructure-based no WiFi client has to be equipped with additional software or hardware to be located. The positioning system offers a web service interface to access the main functions like polling an asset's position or its position history. A calibration can be started by using the web interface.

Prior to the framework's implementation the airport has not used the calibration functions to optimize the outdoor positioning accuracy. This is due to the fact that calibrating an area like the airport campus manually is way too much effort for the expected accuracy improvements. The coverage area of the airport WiFi system is about $6,35\ km^2$ including multiple ramps and buildings. On the ramps it is not allowed to walk around making it nearly impossible to use the normal calibration functions which require a person to stand there and wait until the calibration is done.

Before implementing the framework, measurements have been taken to determine the position accuracy provided by the WPS. Therefore a car was equipped with WiFi antennas on the roof. This was necessary because the outdoor WiFi coverage at the airport is not strong enough to allow network access from inside a car.

The measurements resulted in a average position accuracy of about 240 meters. More important than the position accuracy is the fact that the accuracy fluctuates which is caused by the dynamic environment given by rolling airplanes, cars etc.

A. Implementation

Although the airport offers multiple systems suitable for the given requirements, the Global Positioning System (GPS) was chosen as the reference system. This decision was made to show that the system is functional with widely spread hardware.

Two cars had been equipped with hardware which was able to use GPS for positioning and could be detected by the WiFi system. A smartphone and a laptop with a bluetooth GPS receiver was used to show that the sensor module can be implemented on different systems. As already described, using GPS to receive the real position will result in the problem of detecting if the terminal is moving. The GPS receiver is not good enough to detect that a terminal is not moving. It rather delivers different positions lying within a perimeter. Its size depends on multiple factors like GPS signal strength or the quality of the GPS receiver.

Therefore a simple approach for movement detection was developed which was reliable enough for the experimental setup. First the sensor module only observes the speed of the mobile terminal. If that speed drops below a certain threshold $v_{threshold}$ the positions get logged. After at least 12 positions were logged, the latest 10 are used to calculate the minimum enclosing circle (MEC, [1]) around these positions. If the radius of the MEC is below a given value r_{max} the terminal is detected as not moving. To allow an easy calculation of Euclidean distances the Universal Transverse Mercator (UTM) was used as CCS within the framework. Therefore a translation between WGS84 and UTM had to be implemented in the sensor module.

The WiFi Connector was implemented in the way that it can access the web service interface to poll positions from the WPS. As mentioned above the WPS has no real interface to start a calibration so the connector had to use the web interface. First some manual calibrations were made to get initial values for T_{last} and Δ_T as mentioned in Section III-D. For the translation between the selected CCS and the coordinate system used within the WPS, three points where specified on the airport campus for which the coordinates in both systems were known.

The calibration core was implemented as described in Section III-C. The CCS enabled the use of a simple algorithm to calculate distances between different positions. To adapt the framework to the airport environment, three different constraint modules were specified:

- The first one, C_{onRamp}, assured that only the area of a special ramp was calibrated in order to limit the test area.
- The constraint C_{onRoad} limited the calibration to roads within the ramp. This constraint should avoid a calibration of places where normally no car is allowed to stop.
- The previous two constraints only returned either the value 0 to prevent a calibration or the value 1 to allow it. The third constraint, $C_{carPark}$, controlled whether the mobile terminal was standing at a parking area. In that case it returned the value 1 because it's very likely that the car will stay there for a longer time. In any other case, the constrain module returned a value of 0.8, so that the remaining areas will be calibrated with a probability of 80 percent.

All constraint modules had a weight of 1. While the sensor modules were deployed on the mobile terminals, the rest of the modules were installed on a server within the airport network.

B. Evaluation

By using two cars on the ramp of the Munich Airport it was possible to demonstrate that the concept of the framework is correct and that it can be used to improve the position accuracy of the WiFi positioning system. After the calibration, measurements were taken again on the ramp to detect possible changes within the accuracy.

The improvements depend not only on the framework but also on the WiFi system and the environment. More important than the accuracy measurement results (which improved to an average accuracy of 180 meters) is the number of calibrations that has to be reversed.

On the first run (one day) 287 calibrations were performed from which 9 percent had to be reversed. Given the dynamic nature of an airport ramp this value can be interpreted as a successful result. It shows that the detection algorithm for "no movement" is working and that the three constraint modules have done a good job to prevent invalid calibrations. An analysis of the positions where invalid calibrations were performed has shown that they occurred most often at crossings. This observation is easy to explain: An airplane always has priority in traffic, so a car has to wait until an airplane rolled by. This can take more time than the 12 measurements taken when a stop is detected. The car will be recognized as not moving. Additionally it is within the ramp and it is on a road, but not on a parking area. This means, that the calibration will be executed with an 93 percent probability.

There were two options to correct that obvious error: A solution would have been to increase the number of positions which have to be polled before movement detection will be started. This would have prevented the system to calibrate on crossings if the duration of these measurements takes longer than the crossing airplane. But this approach can also prevent calibrations which otherwise would have been succeeded successful. Additionally, by waiting longer before starting a calibration the probability that the car will move on before the calibration is finished is also increased. Because of that a new constraint was added to the framework to make sure that no calibration is started at crossings. After that the percentage of failed calibrations dropped to 6 percent on the second run.

Although the comparison of the position accuracy before and after the automatic calibration only illustrates a slight improvement on the position accuracy, it is obvious that the framework has accomplished its goals. The system was calibrated without any human influence. The framework also makes it possible that other systems can be used to influence the calibration process, e.g. a system which knows the current positions of all airplanes and thus can prevent a calibration if an airplane is too close and could disturb the calibration.

V. Related Work

WiFi positioning has been the topic of active research since RADAR [2] was published. Many location technologies have been explored and different methods for location determination like lateration and fingerprinting were developed. Nowadays most WiFi positioning systems [2][3][4][5][6][7] use fingerprinting to locate objects. But fingerprinting always requires a map describing the radio propagation within the coverage area. There are different approaches to create such a map while trying to decrease the amount of manual effort which is caused by calibrations.

In [8] a system is introduced, which creates the propagation model by using measurements between sniffers and access points and no further calibration is needed. A similar approach requiring no calibration is described in [9].

Although these approaches achieve a good accuracy, calibration can be adopted to improve the results. In [10] a learning algorithm is proposed which provides similar accuracy with a reduced number of sampled locations. In [11] interpolation is used between different location samples to compute the positions between, while [12] uses topological models to reduce the calibration time.

Given the characteristics of large-scale enviroments, [13] evaluates different positioning algorithms to find out how to adapt them in such environments. In [14] a probabilistic algorithm based on a Fast Fourier Transform is used to allow a good accuracy although only inaccurate range measurements are possible. In [15] RFID sensors are used to adapt the location system automatically to the changing environmental dynamics.

VI. Conclusion

In this paper we introduced a modular framework for an automatic calibration of WiFi positioning systems. It makes use of a reference system with higher accuracy to receive the correct position of mobile terminals. These terminals are equipped with a WiFi interface which makes it possible to use them for calibration. Due to the modular design of the framework it is easy to replace reference positioning systems or even to apply multiple systems at the same time. This is achieved by a common coordinate system within the framework.

Constraints can be used to decide whether a calibration should be performed at a given position. Therefore it is possible to adapt the framework to every location just by defining constraints which address the important characteristics of the environment. With the assistance of another module the framework is connected to a WiFi positioning system which needs to support position polling and calibration. If required it also allows to reverse calibrations already finished.

To evaluate the framework we implemented a prototype at the Munich Airport in Germany, where we used GPS as a reference system. The test results proved that the approach is applicable and can be used to improve the position accuracy in almost any large-scale environment.

References

[1] E. Welzl, "Smallest enclosing disks (balls and ellipsoids)," in *Results and New Trends in Computer Science*. Springer-Verlag, 1991, pp. 359–370.

[2] P. Bahl and V. N. Padmanabhan, "Radar: an in-building rf-based user location and tracking system," 2000, pp. 775–784.

[3] T. Kitasuka, T. Nakanishi, and A. Fukuda, "Wireless lan based indoor positioning system wips and its simulation," in *Nara Institute of Science*, 2003, pp. 272–275.

[4] P. Castro, P. Chiu, T. Kremenek, and R. R. Muntz, "A probabilistic room location service for wireless networked environments," in *UbiComp '01: Proceedings of the 3rd international conference on Ubiquitous Computing*. London, UK: Springer-Verlag, 2001, pp. 18–34.

[5] M. Youssef and A. Agrawala, "The horus wlan location determination system," in *MobiSys '05: Proceedings of the 3rd international conference on Mobile systems, applications, and services*. New York, NY, USA: ACM, 2005, pp. 205–218.

[6] M. Youssef, A. Youssef, C. Rieger, U. Shankar, and A. Agrawala, "Pinpoint: An asynchronous time-based location determination system," in *MobiSys '06: Proceedings of the 4th international conference on Mobile systems, applications and services*. New York, NY, USA: ACM, 2006, pp. 165–176.

[7] M. Ciurana, F. Barceló, and S. Cugno, "Indoor tracking in wlan location with toa measurements," in *MobiWac '06: Proceedings of the 4th ACM international workshop on Mobility management and wireless access*. New York, NY, USA: ACM, 2006, pp. 121–125.

[8] L. F. M. de Moraes and B. A. A. Nunes, "Calibration-free wlan location system based on dynamic mapping of signal strength," in *MobiWac '06: Proceedings of the 4th ACM international workshop on Mobility management and wireless access*. New York, NY, USA: ACM, 2006, pp. 92–99.

[9] Y. Gwon and R. Jain, "Error characteristics and calibration-free techniques for wireless lan-based location estimation," in *MobiWac '04: Proceedings of the second international workshop on Mobility management & wireless access protocols*. New York, NY, USA: ACM, 2004, pp. 2–9.

[10] X. Chai and Q. Yang, "Reducing the calibration effort for location estimation using unlabeled samples," in *PERCOM '05: Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 95–104.

[11] A. Smailagic, D. P. Siewiorek, J. Anhalt, D. Kogan, and Y. Wang, "Location sensing and privacy in a context-aware computing environment," *IEEE Wireless Communications*, vol. 9, pp. 10–17, 2001.

[12] 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 *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2004, pp. 70–84.

[13] Y.-C. Cheng, Y. Chawathe, A. LaMarca, and J. Krumm, "Accuracy characterization for metropolitan-scale wi-fi localization," in *MobiSys '05: Proceedings of the 3rd international conference on Mobile systems, applications, and services*. New York, NY, USA: ACM, 2005, pp. 233–245.

[14] R. Peng and M. L. Sichitiu, "Probabilistic localization for outdoor wireless sensor networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 11, no. 1, pp. 53–64, 2007.

[15] Y.-C. Chen, J.-R. Chiang, H. hua Chu, P. Huang, and A. W. Tsui, "Sensor-assisted wi-fi indoor location system for adapting to environmental dynamics," in *MSWiM '05: Proceedings of the 8th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*. New York, NY, USA: ACM, 2005, pp. 118–125.