

REIHE INFORMATIK
TR-2006-019

Distribution of Fingerprints for 802.11-based Positioning Systems

Thomas King, Thomas Butter, Matthias Brantner, Stephan Kopf,
Thomas Haenselmann, Alexander Biskop, Andreas Färber,
Wolfgang Effelsberg
University of Mannheim
– Fakultät für Mathematik und Informatik –
Praktische Informatik IV
A5, 6
D-68159 Mannheim, Germany

Distribution of Fingerprints for 802.11-based Positioning Systems

Thomas King, Thomas Butter, Matthias Brantner,
Stephan Kopf, Thomas Haenselmann, Alexander Biskop,
Andreas Färber, Wolfgang Effelsberg
{king,haenselmann,kopf,effelsberg}@informatik.uni-mannheim.de,
butter@bwl.uni-mannheim.de
brantner@db.uni-mannheim.de
{abiskop,afaerber}@rumms.uni-mannheim.de,
Department of Computer Science
University of Mannheim

Abstract

While indoor positioning systems based on 802.11 and fingerprinting work pretty well, it is unknown how to distribute a large amount of fingerprint data to mobile devices. Even worse, many mobile devices are restricted in terms of memory. In this demo proposal, we present two distribution approaches for fingerprints that fill this gap: the *Strongest Access Point (SAP)* and the *Intersection of Access Points (IAP)* algorithms. These approaches utilize the 802.11 infrastructure to download only a subset of the complete fingerprint data to a mobile device. The subset covers the area close to the actual position of the mobile device in such a way that position estimates can be computed.

For the MDM 2007 demo session, we offer to demonstrate how these distribution approaches for fingerprint data work. The demo will include live indoor positioning of visitors using mobile devices while concurrently displaying the distribution of fingerprint data.

1 Introduction

In recent years we have seen a considerable amount of research in the area of indoor positioning systems mainly because the well-known *Global Positioning System (GPS)* does not work well in indoor environments. One of the most promising technologies that could be an equivalent to GPS for indoor applications are *802.11-based positioning systems* [1, 2]. Nowadays, 802.11 hardware is readily available and installed nearly everywhere where people live and work. Another important fact is that 802.11 is a wireless local area network technology that is usually used to provide Internet access to mobile users; however, it can be used for positioning purposes at the same time. Even

better, almost all modern PDAs, cellphones and laptops are capable to communicate with 802.11 infrastructure because they are shipped with built-in 802.11 hardware.

The best positioning results can be achieved with 802.11 positioning systems that utilize the so-called *Fingerprint* approach [1]. This technique comprises two stages: an offline training phase and online position determination phase. During the offline phase, the signal strength distributions collected from access points at predefined reference points in the operation area are stored in a table together with their physical coordinates. One dataset is called a Fingerprint. During the position determination phase, mobile devices sample the signal strength of access points in their communication range and search for similar patterns in the fingerprint data. The closest match is selected, and its physical coordinates are returned as a position estimate.

Recent research has focused on algorithms that compute the closest match (e.g., [6, 8, 5]). The authors of these papers assume that the entire fingerprint data is stored on the mobile device. If we think of large deployments of these positioning systems (e.g., covering all buildings on a campus), keeping the entire fingerprint data on the mobile device is not feasible for many reasons: fingerprints change due to structural alterations, are updated because of new deployments or relocation of access points, or they are just too big to be stored on a mobile device. Furthermore, computing position estimates on a central server is not practical for scalability reasons.

In case of positioning systems, mobile devices are restricted in three respects: by processing power, by network access (such as bandwidth and delay), and by storage capacity (main memory as well as fixed-disk storage). Modern mobile devices provide enough processing power to compute the algorithms used by positioning systems. Furthermore, because we are focusing on 802.11-based positioning systems, a network is available to easily transfer large amounts of data. So, from a positioning system point of view the only remaining major restriction of mobile devices is their storage capacity. Mobile devices usually offer only a few megabytes of main memory and only a few dozen megabytes of fixed-disk storage which are easily exceeded by a huge amount of fingerprint data. This said, for the remainder of this work, we only focus on storage-limited mobile devices.

Since we are not aware of any work that covers the distribution of fingerprints, we are going to present two novel approaches in this demo proposal that demonstrate how fingerprint data can be automatically distributed while not overloading the restricted capacities of mobile devices. Our approaches only keep a fraction of the entire data on the mobile device, so that only fingerprints are available that are close to the mobile device's actual position. The fingerprints in close proximity of the mobile device are needed to compute position updates. If users with mobile devices move around, the fingerprints on the device must be replaced with fingerprints that are closer to its actual position. The two distribution approaches differ in the update strategy used, the amount of data stored on a mobile device, and the number of updates required to keep the data up-to-date. For both approaches, we assume that the mobile device is able to use the 802.11 infrastructure to access a server that hosts the complete fingerprint data.

The remainder of this paper is structured as follows: We present our two distribution approaches in the following section. In Section 3, we briefly discuss the implementation of these approaches. The demo setup is described in Section 4. We finally conclude the paper in Section 5.

2 Distribution Algorithms

In this section, we first present our distribution approaches.

2.1 The Algorithms

We have developed two distribution approaches: the *Strongest Access Point (SAP)* and the *Intersection of Access Points (IAP)* algorithm. For both approaches, we assume that the mobile device scans regularly for access points in communication range. Additionally, the first approach requires that the mobile device is able to measure the reception power of frames transmitted by access points. These assumptions are valid, because the IEEE 802.11 standard [3] defines means such as active and passive scanning that provide this information. Furthermore, both approaches assume that all the fingerprint data is stored on a server that is accessible through the 802.11 infrastructure.

2.2 Strongest Access Point

If a mobile device scans for access points in communication range and sorts the results by the reception power, the access point that shows the best reception power is the access point that is closest to the mobile device. We call this particular access point the *strongest access point of a mobile device*. The basic idea behind the *Strongest Access Point (SAP)* algorithm is the fact that the coverage area of the strongest access point of a mobile device defines a small natural area wherein the mobile device is located. An abstract definition of this area can be accomplished without any further computation or any additional information of the actual position of the mobile device. Furthermore, we have observed that the strongest access point of a mobile device tends to be a long-running stable value even if the user moves around indoors. The reason for this is that access points are usually deployed in such a way that they cover a complete building floor or at least a major part of a floor. Additionally, users tend to move between floors only occasionally.

The SAP algorithm works as follows: A mobile device scans for access points in communication range and sorts the result by reception power. The strongest access point is picked and reported to the server. The server selects all reference points that are covered by this access point. Then, based on these reference points, all fingerprints of access points that cover one of these reference points are selected and transferred to the mobile device. Each time the strongest access point changes, the procedure is repeated.

2.3 Intersection of Access Points

We came up with the *Intersection of Access Points (IAP)* algorithm while we considered mobile devices that are extremely limited in terms of storage. In this case, the amount of fingerprint data on the 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 define the smallest area wherein the mobile device can be located.

The IAP algorithm utilizes this fact: A mobile device scans for access points in communication range and reports all access points to the server. The server computes the intersection of the access points' coverage areas. Only for reference points inside this intersection, the fingerprints for access points in communication range 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.

3 Implementation

We implemented the SAP and IAP distribution approaches as part of our Loclib [4] framework. Loclib serves to test and evaluate novel location determination algorithms and approaches. It is a layered framework (see Figure 1). For instance, on the lowest layer different sensors such as Bluetooth, Wireless LAN, GPS, Zigbee and digital compasses are supported. On a higher layer, position estimates are offered to applications through the so-called Location API [7]. In between is a set of algorithms to determine the position as well as distribution mechanisms for the fingerprint data.

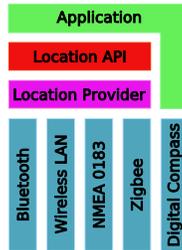


Figure 1: Architecture of Loclib

The two distribution approaches are realized as a client-server application where the server hosts the complete fingerprint data and returns a subset of this data on request. A mobile device runs the client and requests data regarding the selected distribution approach. The client functionality is part of Loclib and resides inside the Location Provider layer.

4 The Demo

For the demo, we will deploy a set of access points and take fingerprint measurements at a reference point grid in the room where the demo session will be held. To show how the positioning system works and to make sure that enough data is available we will define at least ten reference points. We will mark these reference points by sticking numbered markers on the floor, so that visitors are able to recognize the difference between the actual position and the position estimate computed by the mobile device. We will place at least two access points in such a way that only a subset of the reference points will be covered.

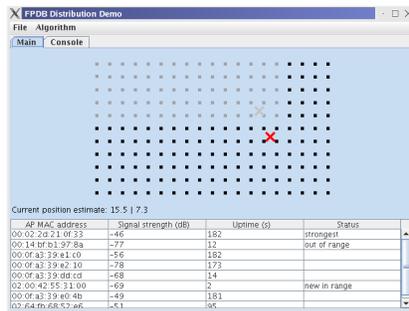


Figure 2: Screenshot of the demo application

4.1 Poster

A descriptive poster explains the purpose of the two distribution approaches for fingerprint data and provides the context in which such algorithms are needed. Additionally, a tutor will provide assistance if necessary.

4.2 Client

We use a Fujitsu Siemens Lifebook T4010 as mobile device that runs the client application. The client application consists of the Loclib framework that computes the position estimates and handles the fingerprint distribution, and a graphical user interface on top of Loclib that displays what is going on inside Loclib. Figure 2 shows the application and what is displayed to the user:

- if an access point is discovered while moving,
- if a known access point is out of communication range,
- the query sent from the client to the server to request fingerprints,
- the response from the server containing a subset of the complete fingerprint data.

Additionally, the client displays the actual position estimate calculated by the positioning algorithm based on the fingerprint data stored on the mobile device. For the positioning, we selected an algorithm as presented in [6, 8, 5].

4.3 Server

The server application will run on a host containing the entire fingerprint data. Each time a client requests a subset of the fingerprints, the server answers with the requested data.

5 Conclusions

In this demo proposal, we first explained why distribution techniques for fingerprints are required in the area of 802.11-based positioning systems. After that we presented two distribution approaches for fingerprint data, namely *Strongest Access Point* and *Intersection of Access Points*. Third, we briefly discussed the implementation of the 802.11-based positioning and how our two novel fingerprint distribution approaches are realized to seamlessly update the fingerprints available on a mobile device. Finally, we described a possible demo setup and the presentation for potential visitors.

We think that this demo will be very beneficial for the MDM 2007 attendants since most will probably have read about or even worked with 802.11-based positioning systems. Our contribution is another step in building easy-to-use indoor positioning systems.

Acknowledgments

The authors acknowledge the financial support granted by the *Deutschen Forschungsgemeinschaft* (DFG).

Availability

The tools presented in this demo proposal will be released under the terms of the GPL on our website [4].

References

- [1] P. Bahl and V. N. Padmanabhan. RADAR: An In-Building RF-Based User Location and Tracking System. In *Proceedings of the 19th International Conference on Computer Communications (InfoCom)*, volume 2, pages 775–784, Tel Aviv, Israel, March 2000. IEEE.
- [2] P. Castro and R. Muntz. Managing Context Data for Smart Spaces. *IEEE Personal Communications*, pages 44–46, October 2000.
- [3] 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.
- [4] T. King and S. Kopf. Loclub - A Location Library. Website: <http://www.informatik.uni-mannheim.de/pi4/lib/projects/loclub/>, November 2005.
- [5] T. King, S. Kopf, T. Haenselmann, C. Lubberger, and W. Effelsberg. COMPASS: A Probabilistic Indoor Positioning System Based on 802.11 and Digital Compasses. In *Proceedings of the First ACM International Workshop on Wireless Network Testbeds, Experimental evaluation and Characterization (WiNTECH)*, pages 34–40, Los Angeles, CA, USA, September 2006. ACM Press.
- [6] A. M. Ladd, K. E. Bekris, A. Rudys, G. Marceau, L. E. Kavradi, and D. S. Wallach. Robotics-Based Location Sensing using Wireless Ethernet. In *Proceedings of the Eight ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 227–238, Atlanta, GE, September 2002. ACM Press.

- [7] K. Loytana. JSR 179: Location API for J2ME - Final Release 2. Website: <http://www.jcp.org/en/jsr/detail?id=179>, March 2006.
- [8] M. Youssef and A. Agrawala. The Horus WLAN Location Determination System. In *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 205–218, 2005.