This is a preliminary version of an article published in
Proc. of the 8th International Conference on Mobile Data Management (MDM) (Demo Session), pp. 224 – 226, Mannheim, Germany, May 2007
by Thomas King, Thomas Butter, Matthias Brantner, Stephan Kopf, Thomas Haenselmann, Alexander Biskop, Andreas Färber, Wolfgang Effelsberg

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,kopf,haenselmann,effelsberg}@informatik.uni-mannheim.de
butter@uni-mannheim.de
brantner@db.informatik.uni-mannheim.de
{abiskop,afaerber}@rumms.uni-mannheim.de
University of Mannheim, Germany

Abstract

While indoor positioning systems based on 802.11 and fingerprinting work pretty well, it is unknown how to distribute large amounts of fingerprint data to mobile devices. Even worse, many mobile devices are restricted in terms of memory capabilites. We identified three classes of mobile devices representing different levels of storage capabilities. For each of these classes, we present a distribution approach for fingerprint data: the Union of Access Points (UAP)*, the* Strongest Access Point (SAP)*, and the* Intersection of Access Points (IAP) *algorithm. These approaches utilize the 802.11 network to download only a subset of the 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. The size of the subset is different for each of the three algorithms.*

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 real-world coordinates. One dataset is called a Fingerprint. During the position determination phase, mobile devices sample the signal strengths of access points in their communication range and search for similar patterns in the fingerprint data. The best 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, 5, 3]). 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, in [1] the authors propose a central server that stores all fingerprints and computes position estimates of mobile devices. While this approach works pretty well in a small testbed, it does not scale with hundreds or thousands of mobile devices.

Nowadays, 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). However, modern mobile devices provide enough processing power to execute algorithms used by positioning systems to calculate a position estimate. 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 capacities. If we compare storage capabilities of mobile devices, we see a wide range of settings: laptop computers offer dozens of gigabytes of memory, high-end PDAs and smartphones provide a few gigabytes of fixed-disk storage and a few hundred megabytes of main memory, simple cellphones and PDAs provide only a few dozens of megabytes of main memory and no additional fixed-disk storage, and sensor nodes also provide no fixed-disk storage and contain only a few hundred kilobytes of main memory. We group the last two items together, because both commonly face the danger of running out of memory if users execute meaningful applications on them.

Since we are not aware of any work that covers the distribution of fingerprints, we are going to present three novel approaches that demonstrate how fingerprint data can be automatically distributed. For each previously described group of mobile devices, we present an adequate fingerprint distribution algorithm. 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 in the near future. If the user of a mobile device moves around, the fingerprints stored on the device must be updated with fingerprints that are closer to its actual position. The 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. However, due to page restrictions we do not discuss these metrics here.

2 Distribution Algorithms

In this section, we first discuss the assumptions we make and then present our three novel distribution approaches.

2.1 The Algorithms

We have developed three distribution approaches: the *Union of Access Points (UAP)*, the *Strongest Access Point (SAP)*, and the *Intersection of Access Points (IAP)* algorithm. For these approaches, we assume that the mobile device scans regularly for access points in communication range. Additionally, the SAP 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 [4] defines means such as active and passive scanning that provide this information. Furthermore, our approaches require the complete fingerprint data to be stored on a server that is accessible through the 802.11 network.

2.2 Union of Access Points

The advances in miniaturization of memory technology allow to build mobile devices, such as laptops or fancy PDAs, that offer plenty of memory storage. For these devices, a fingerprint distribution approach should not try to minimize the amount of data used to store fingerprints, because these devices can easily handle large sets of fingerprint data. For this scenario, the prime reason for a fingerprint data distribution algorithm is to keep fingerprint data on mobile devices up-to-date. To achieve this goal, the fingerprint data of all access points in communication range are stored on the device and each time an unrecognized access point comes into communication range the fingerprint data on the mobile device is updated. We call this algorithm the *Union of Access Points (UAP)* approach.

To be more precise, we describe the algorithm in more detail: Each time the mobile device is started, it scans for access points within communication range and requests the fingerprints for these access points by sending a request to the fingerprint server. The server omits access points it is not aware of. If a user carrying the device moves around a fingerprint data update is only requested if an unrecognized access point comes into communication range. In case the user leaves the coverage area of a known access point, no update request is sent to the server. Instead, the fingerprints corresponding to this particular access point are marked as stale on the mobile device. This procedure requires only an fingerprint data update from the server if the user moves around widely.

2.3 Strongest Access Point

The subsequent algorithm is designed to support middle class mobile devices. If such a mobile device scans for access points in communication range and sorts the results by reception power, the access point that shows the best reception power is typically the access point that is closest to the mobile device. We call this particular access point the *strongest access point of the 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 carrying the device moves around. 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 fingerprint 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 transfered to the mobile device. Each time the strongest access point changes, the procedure is repeated. If the strongest access point of a mobile device is unknown by the server (e.g., it has lately been deployed) the second strongest access point will be used, and so on.

2.4 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 the reference points inside this intersection, the fingerprints of access points in communication range are transfered 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 server, its presence is ignored.

3 Conclusions

In this paper, we first explained why distribution techniques for fingerprints are required in the area of 802.11-based positioning systems. After that, we compare the storage capabilities of mobile devices and define three different classes of storage-restricted devices. This classification in mind, we presented three distribution approaches for fingerprint data, namely *Union of Access Points*, *Intersection of Access Points*, and *Strongest Access Point*. Each of these algorithms is designed for one of the storage-restricted mobile device classes.

The distribution algorithms presented in this paper are another step on the way to build easy-to-use 802.11-based positioning systems. As a next step, we are working on a theoretical analysis of these algorithms to get a deeper understanding of their performance.

Acknowledgments

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

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] 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 *Proceedings of the Tenth ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 70–84, New York, NY, USA, September 2004. ACM Press.

[4] 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.

[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] 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.