2012 32nd International Conference on Distributed Computing Systems Workshops

Towards a practical, scalable self-localization system for Android phones based on WLAN fingerprinting

Peter Riedl
University of Applied Sciences Upper Austria
Hagenberg, Austria
peter.riedl@fh-hagenberg.at

Rene Mayrhofer
University of Applied Sciences Upper Austria
Hagenberg, Austria
rene.mayrhofer@fh-hagenberg.at

***Abstract*—Indoor localization is becoming increasingly important for mobile applications. WLAN fingerprinting is a compelling technique because it builds upon existing infrastructure and client hardware available in off-the-shelf mobile devices. We evaluate different methods for WLAN fingerprint classification with a focus on on-device localization. The main scientific contribution of this approach is that any Android based device can localize itself (without any server being able to determine the current location) using existing WLAN infrastructure (no additional access points have to be installed, the firmware of existing access points doesn't have to be changed). This approach was chosen to make indoor localization feasible in non-academic use cases.**

With a functional implementation and a simple procedure for collecting WLAN fingerprints, we currently achieve an accuracy of 4 m in 90% of all cases with a mean error of only 2.2 m when the same device is used for training and testing. Next steps are calibration between different mobile devices, post-processing in terms of movement, and automatic downloading of the required WLAN fingerprint databases on a global scale.

***Keywords*-indoor localization; fingerprinting; client based localization**

I. Introduction

Context in general and location specifically are very important aspects of many pervasive computing applications. Especially for applications in home, office, school, and industrial settings, location of users and devices is one of the most significant aspects for inferring context information. For outdoor localization there is GPS, but until now there is no widely deployed system for indoor localization. In Section II some current indoor localization projects are presented, but none of the currently existing systems are ready to be deployed by non-expert personnel in a real world setting.

We study systems that:

- support localization for arbitrary Android devices;
- require no additional infrastructure;
- provide self-localization for client devices for privacy reasons; and
- are simple enough for end users (such as home owners or building personnel) to set up, calibrate, and deploy for their own spaces.

Especially the last requirement proves difficult to fulfill with existing systems, and we therefore investigate the whole work-flow from set-up to live usage.

Based on WLAN fingerprinting, we present an analysis of the effects of different classifiers, device heterogeneity, and resolution on the achieved accuracy. We show that, when using the same device for training, an accuracy of better than 4 m can be achieved with on-device self-localization. That is, client devices only need to install a small application and download the fingerprint database to localize themselves within already deployed WLAN infrastructure. Calibration methods for improving cross-device accuracy are subject to future work.

II. Related Work

Numerous different approaches for indoor localization exist. One differentiation criterion is the underlying (hardware) technology. There are systems based on GPS [1], GSM cells [2], Bluetooth [3], Ultra Wide Band (UWB) [4], RFID [5], Ultrasound [6] and infrared light [7].

Our localization system uses WLAN (Wireless Local Area Network) IEEE 802.11 for position estimation. One of the first systems that also used WLAN to determine the location is RADAR [8]. The most obvious similarity between our localization system and RADAR is the usage of WLAN. Another aspect that is common to both systems is that fingerprinting (see Section III) is used to estimate the user's position. However, in our localization system no propagation model is used. One of the differences between the systems is that RADAR uses PCs as base stations and notebooks as hosts while our localization System uses commodity wireless access points as base stations and Android phones as hosts. The biggest difference is that in our system hosts do not send any data to base stations, they only receive data and position themselves passively with obvious benefits for user privacy.

Our localization system and the system in [9] have in common that WLAN fingerprinting is used to determine the location of mobile hosts. The application paradigm on the other hand is diametrical. While our system completely avoids changing existing infrastructure components to reduce costs and maximize privacy, the system in [9]

1545-0678/12 $26.00 © 2012 IEEE
DOI 10.1109/ICDCSW.2012.26

IEEE computer society

relies heavily on location estimation in the infrastructure to overcome device heterogeneity.

Placelab [10] was one of the first projects where on-device-localization was implemented for outdoor localization. We follow the same approach for indoor scenarios, but with improvements to localization accuracy.

All aforementioned projects have in common that the whole system installation demands expert personnel for configuring/installing hardware or performing site surveys. Our research ambition is to overcome this issue and provide a system which can easily be set-up by non-expert personnel.

III. Method

The aforementioned term fingerprinting constitutes the process of collecting received signal strength information (RSSI), associating them with real world locations and finally comparing the previously recorded RSSI with the current RSSI of a device that should be located. That is the operational principle of our localization system.

Fingerprinting consists of two phases. In the first phase (off-line phase) a grid of measurement points (MPs) is laid out on a floor plan. Then RSSI is recorded at every MP (substituting -100 dBm when BSSIDs are not visible, based on practical experience). For every MP tuples of the form (n, t, RSSI_0, RSSI_1, …, RSSI_k) are stored, where n stands for the name of the MP (e.g. MP1), t is the time stamp of the record and $\mathrm{RSSI}_0 \ldots \mathrm{RSSI}_\mathrm{k}$ stands for the RSSI of all access points in range. The records of all MPs are called fingerprint database.

The second phase is called on-line phase. Here the currently recorded RSSI of a device that should be located are compared to the entries in the fingerprint database. The position of the most similar MP is then assumed to be the position of the device and therefore, of the user.

For comparison of fingerprint database and current RSSI several classifiers from the Waikato Environment for Knowledge Analysis (WEKA)[1] were investigated. Empirical tests showed that the best results are achieved with ”weighted k nearest neighbours” (WKNN), ”KStar”, ”Random Tree” and ”Random Forrest”. In this context the criteria for ”best results” are: accuracy (percentage of correctly classified MPs/rooms), training time (time needed to ”teach” the algorithm the fingerprint database) and matching time (the time a trained algorithm needs to match the current RSSI to the fingerprint database). Since the feature vector only consists of RSSI of all the access points it can easily be processed by all investigated classifiers.

IV. Implementation

The first step to successfully perform fingerprinting is to lay out MPs on a floor plan. To achieve this, the interior design application ”Sweet Home 3D” (SH3D)[2] was used. In SH3D existing floor plans can be imported as background images, rooms can be defined on that plan and objects can be positioned in those rooms. The main reason to choose SH3D was that custom functionality can be added via plug-ins. This plug-in mechanism was used to automatically lay out all MPs on a two meter grid in every room and export the coordinates and names of all modelled rooms and MPs into an XML (eXtensible Markup Language) file.

The next step after laying out the MPs is to record the fingerprint database. Therefore, an Android application to collect RSSI at every MP was written – the so called ”survey tool”. Of course this tool was designed to be used by non-expert users to keep the whole process as simple as possible. After the fingerprint database was recorded, the gathered data can be exported as ARFF (Attribute-Relation File Format)-file, that can directly be used for training and classification by any WEKA classifier.

The initial selection of suitable classifiers was done with WEKA. After the best classifiers were identified, the WEKA API was used to implement an Android service that uses those classifiers in conjunction with the previously recorded fingerprint database to estimate the current position. Due to the fact that this functionality was implemented as Android service applicability and reusability are greatly enhanced. Regardless of the operating system’s version that service can be installed on any Android device. It then offers an interface to query the user’s current location via one simple method call. This architecture enables arbitrary third party applications to use our indoor localization service. In the current implementation the fingerprint database has to be provided by the application using the service, one of the next steps will be the implementation of a work flow to automatically download all relevant indoor localization data from nearby sources.

V. Results

All data samples were acquired at the University of Applied Sciences Upper Austria, campus Hagenberg in the rooms labelled L2.MC, PL1.MC, L1.MC, PL2.MC, SRMC. All rooms except SRMC are situated on the same corridor. The MPs were laid out on a 2m grid in every room. The devices used for data collection were: Google Nexus S (*nexus*), Google G1 (*g1*) and HTC Desire HD (*desire*).

We investigated two different approaches. The first is matching on MP level, i.e. the attribute for classification is the name of the MP. The second is matching on room level, i.e. the attribute for classification is the room name. Our goal in the preliminary tests was to determine the room the device is located in. Room level matching directly provides that information, but when using MP level matching the room information can easily be inferred since the room a MP is in is known. In order to evaluate how device heterogeneity influences localization results several combinations of training- and test-datasets were analyzed.

[1] http://www.cs.waikato.ac.nz/ml/weka/

[2] http://www.sweethome3d.com/

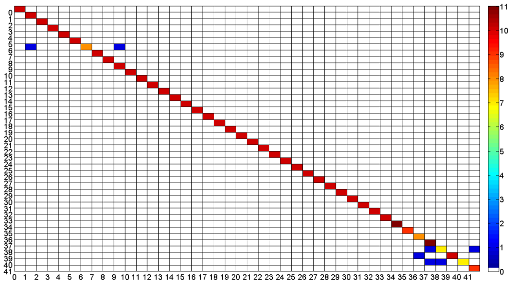

Figure 1. Confusion Matrix for MP-level matching with *nexus* dataset for training and testing.

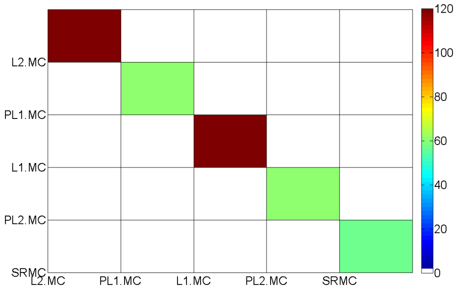

Figure 2. Confusion Matrix for room-level matching with *nexus* dataset for training and testing.

Figure 1 shows a confusion matrix for MP-level matching. The *nexus* dataset was used for both training and testing. In this case 410 out of 417 instances were matched correctly (98,32%). One explanation for the incorrect matches at MP38, MP39, MP40 is that MP36 to MP41 are situated in SRMC where less access points are visible than in the other rooms.

Figure 2 shows that 100% of all instances were matched correctly on room level when the *nexus* dataset was used for training and testing. However, when training with the *nexus* and testing with the *desire* dataset, only 397 out of 430 instances were matched correctly (92.33%, cf. Fig. 3). This indicates that even for coarse location estimation, device heterogeneity is a significant issue.

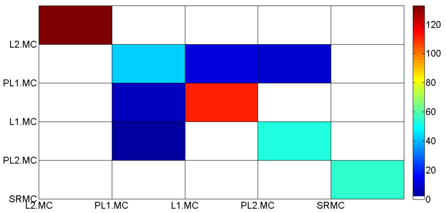

Figure 3. Confusion Matrix for room-level matching with *nexus* dataset for training and *desire* dataset for testing.

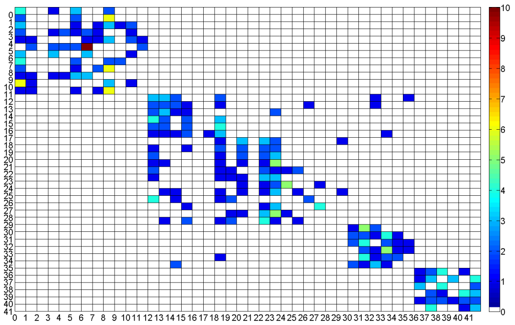

Figure 4. Confusion Matrix for MP-level matching with *nexus* dataset for training and *desire* dataset for testing.

Table I
PERCENTAGES OF CORRECTLY MATCHED MP WITH ALL INVESTIGATED CLASSIFIERS AND ALL DATASET COMBINATIONS.

	1NN	W3NN	W4NN	K*	RC	RF
AA	95,03	95,60	95,60	89,93	95,60	95,28
AD	38,37	39,30	38,60	33,72	43,72	40,70
AN	33,09	35,73	36,69	31,18	35,01	35,49
AG	41,84	41,84	39,33	62,76	63,60	65,27
DA	24,03	24,54	24,09	25,30	25,11	25,24
DD	93,02	93,49	93,49	84,88	93,49	92,79
DN	14,39	13,67	14,39	16,31	14,39	13,67
DG	10,88	13,39	13,81	11,72	9,21	11,72
NA	22,31	21,22	21,48	22,75	22,75	20,84
ND	14,42	16,28	18,14	16,74	15,12	16,05
NN	98,32	98,32	98,32	93,77	98,32	98,32
NG	6,69	7,11	7,11	7,95	11,72	6,69
GA	17,27	17,91	17,97	20,59	18,61	17,08
GD	3,95	4,42	3,95	6,74	6,05	6,98
GN	9,11	7,67	8,63	6,24	6,00	3,60
GG	100	100	100	100	100	100

For fine location estimation (gridsize=2m) the error rate drastically increases; only 72 out of 430 MPs were matched correctly (16.74%, cf. Figure 4).

Table I gives an overview of the device heterogeneity problem. Every column shows the percentage of correctly matched MPs for one classifier (1NN = 1 nearest neighbour, W3NN = weighted 3 nearest neighbors, W4NN = weighted 4 nearest neighbours, K* = KStar, RC = random committee, RF = random forest). The first column shows which combination of training/test dataset was used (A = combined dataset from all devices, D = HTC Desire HD, N = Google Nexus S, G = Google G1).

The cumulative distribution function in Fig. 5 shows that in 90% of all classifications the absolute error is below 4 m with a mean error of 2.2 m. These are results obtained with the *nexus* dataset for training and testing.

The cumulative distribution function shown in Fig. 6 once again depicts the problematic situation of heterogeneous devices. For this CDF, the *nexus* dataset was used for training and the *g1* dataset for testing. Here the errors significantly increase. Only about 15% of all classifications result in an error below 4 m with a mean error of 15.71 m.

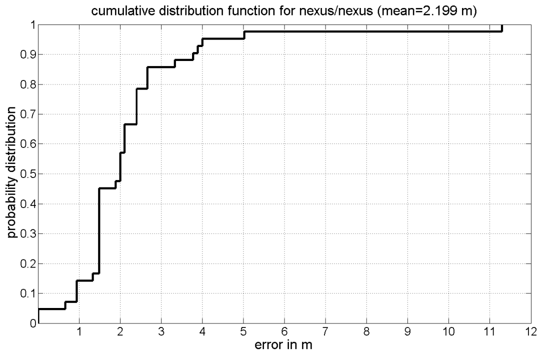

Figure 5. Cumulative distribution function for MP-level matching with *nexus* dataset for training and testing.

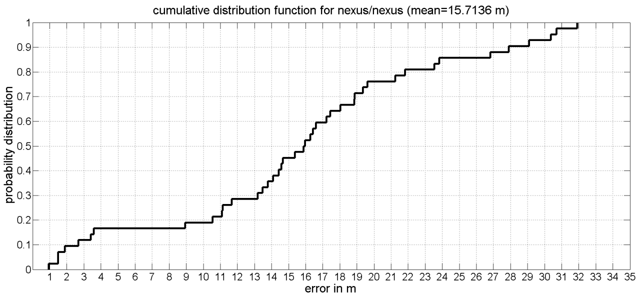

Figure 6. Cumulative distribution function for MP-level matching with *nexus* dataset for training and *g1* dataset for testing.

VI. Discussion

These results are promising, but there still is potential for improvement. The biggest issue is device heterogeneity. Even though all Android devices provide RSSI in dbm[3], factors like antenna position and network interface hardware make it difficult to compare a fingerprint database that was recorded with device A with current RSSI of device B. Manual linear calibration was applied but did not yield the desired effect of overcoming device heterogeneity. The recently suggested method of offset calibration [11] could improve the situation but was not applied yet.

Another option for improving accuracy of the system is post-processing. Until now one very simple algorithm for room level matching was implemented. The average of the three most likely positions is mapped to a room which is then presented as result. This approach slightly improves the room-level accuracy, but not significantly.

We are aware that when it comes to tracking (continuous localization of a moving user/device) new issues arise. Until now no representative tests with moving users/devices were conducted, but we expect Kalman filtering to be beneficial for accuracy. Next steps are an integration with known "anchor points" to compute absolute coordinates in a format compatible with GPS for a fusion of indoor and outdoor localization within the same application, and further analysis on *practical* cross-device calibration that can be performed by end users in their own homes. First results are encouraging, but subject to future work.

All data analyzed in this paper was recorded on a single floor and only on one day. Evaluation of additional datasets of multiple floors at different times is subject to future work.

To complete the process of self localization, we suggest a mechanism to automatically download indoor localization data of nearby buildings. Our approach is to store an index of available databases on a globally available platform (e.g. Open Street Map), but to store the actual databases on a local server at the building where devices can localize themselves. This approach is described further in a paper currently submitted for publication.

Acknowledgment

This work is supported by the Austrian national funding line FFG COIN within project SESAME-S.

References

[1] F. van Diggelen, "Indoor GPS theory implementation," in *Proc. Position Location and Navigation Symposium*. IEEE, 2002.

[2] V. Otsason, A. Varshavsky, A. L. Marca, and E. de Lara, "Accurate GSM indoor localization," in *Proc. UbiComp*. Springer-Verlag, 2005.

[3] F. Subhan, H. Hasbullah, A. Rozyyev, and S. Bakhsh, "Indoor positioning in Bluetooth networks using fingerprinting and lateration approach," in *Proc. ICISA*. IEEE, 2011.

[4] P. Steggles and S. Gschwind, "The ubisense smart space platform," in *Proc. Pervasive*. Springer-Verlag, 2005.

[5] L. Ni, Y. Liu, Y. C. Lau, and A. Patil, "LANDMARC: indoor location sensing using active RFID," in *Proc. PerCom*. IEEE, 2003.

[6] M. Hazas, C. Kray, H. Gellersen, H. Agbota, G. Kortuem, and A. Krohn, "A relative positioning system for co-located mobile devices," in *Proc. MobiSys*. ACM Press, 2005.

[7] R. Want, V. Falcao, and J. Gibbons, "The active badge location system," *ACM Transactions on Information Systems*, vol. 10, 1992.

[8] P. Bahl and V. Padmanabhan, "Radar: an in-building rf-based user location and tracking system," in *Proc. INFOCOM*, vol. 2. IEEE, 2000.

[9] H. Schmitzberger and W. Narzt, "Leveraging WLAN infrastructure for large-scale indoor tracking," in *Proc. ICWMC*. IEEE, 2010.

[10] 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. Pervasive*. Springer-Verlag, 2005.

[11] T. Vaupel, J. Seitz, F. Kiefer, S. Haimerl, and J. Thielecke, "Wi-Fi positioning: System considerations and device calibration," in *Proc. IPIN*, 2010.

[3] Android: ScanResult API documentation http://developer.android.com/reference/android/net/wifi/ScanResult.html (accessed October 2011)