Demo Abstract
Use the Force, Luke
Implementation of RF-based gesture interaction on an android phone

Christoph Rauterberg
Institute of Computer Science
Georg-August-Universität Göttingen
christoph@rauterberg.eu

Stephan Sigg, Xiaoming Fu
Institute of Computer Science
Georg-August-Universität Göttingen
{sigg,fu}@cs.uni-goettingen.de

***Abstract*—Various approaches exist to detect gestures and movements via smartphones. Most of them, however, require that the smartphone is carried on-body. The abscence of reliable ad-hoc on-line gesture detection from environmental sources inspired this project for on-line hand gesture detection on a smartphone using only WiFi RSSI. We highlight our line of work and explain problems at hand to provide information for possible future work. We will furthermore introduce *wifiJedi*, a smartphone application, that is able to detect movement in front of the smartphone by reading the WiFi RSSI and use this information to control a Slideshow.**

I. Introduction

Over the last decade, smartphones have taken up an important role in our daily lives. With this growth in wealth of opportunities, the investigation of control mechanisms for smartphones has become an active research topic. The exploitation of the wide range of sensors that are included in mobile phones has spurred a multitude of interaction principles. Gesture recognition over the wireless interface is, however, yet widely unexplored for mobile phones. Gesture recognition from wireless signals was demonstrated with software defined radio (SDR) devices [3], [5]. Recently, it has also been shown how RSSI information from WiFi packets captured at a mobile phone can be exploited to detect gestures in proximity of the phone [6]. While the former study exploited specialised hardware and the data in the latter work was processed offline, in this paper we present the first implementation of RSSI-based gesture recognition that runs completely on a mobile phone in real-time and with no external interface to the RF-channel. Furthermore we will describe lessons learned and give an outlook on possible future work, that could improve our results.

II. Related Work

When an object reflecting a signal wave is in motion, this causes Doppler Shift. The direction and speed of the movement conditions the strength and nature of this frequency shift. Pu and others showed that simultaneous detection of gestures from multiple individuals is possible by utilising multi-antenna nodes and micro Doppler fluctuations [5], [4]. They utilise a USRP SDR multi antenna receiver and one or more single antenna transmitters distributed in the environment to distinguish between a set of 9 gestures with an average accuracy of 0.94. Their active device-free system exploits a MIMO receiver in order to recognise gestures from different persons present at the same time. By leveraging a preamble gesture pattern, the receiver estimates the MIMO channel that maximises the reflections of the desired user.

A main challenge was for them that the Doppler shift from human movement is several magnitudes smaller than the bandwidth of the signal employed. The authors therefore proposed to transform the received signal into several narrowband pulses which are then analysed for possible Doppler fluctuation. The group discussed application possibilities of their system in [1].

In a related system, Adib and Katabi employ MIMO interference nulling and combine samples taken over time to achieve a similar result while compensating for the missing spatial diversity in a single-antenna receiver system [3]. In their system, they leverage standard WiFi hardware at 2.4GHz.

Later, this work was extended to 3D motion tracking by utlising three or more directional receive antennas in exactly defined relative orientation [2]. In particular, the system is able to track the center of a human body with an error below 21cm in any direction and can also detect movement of body parts and directions of a pointing body part, such as a hand. This localisation is possible through time-of-flight estimation and triangulisation. Higher accuracy of this estimation is granted by utilising frequency modulated carrier waves (sending a signal that changes linearly in frequency with time) over a bandwidth of 1.69GHz. Impact of static objects could be mitigated by subtracting successive sample means whereas noise was filtered by its speed of changes in energy over frequency bands.

Very recently, Sigg et al. investigated the distinction of gestures and situations in a passive device-free system with only one SDR [8] or off-the-shelf smartphone [7], [6] receiver. They observed that 10 RSSI packets per second are sufficient to distinguish between simple classes and also hand gestures in proximity of the receiver. Although the accuracy reached with purely WiFi-RSSI-based features was lower than for the USRP-systems exploiting Doppler fluctuation, they could distinguish 11 gestures performed in close proximity of the phone.

III. Implementation

In order to achieve a reliable on-line detection of movement and nearby gestures trough RSSI signal strengh information and to demonstrate its usage, we developed a show-case application for Android mobile phones. The *apk*[1] and Source code[2] for the proposed application is available online. After setting up *tcpdump*, data is collected, and parsed into a training set for the Machine learning algorithms before the trained classifier is used to control the slideshow.

A. Platform Requirements

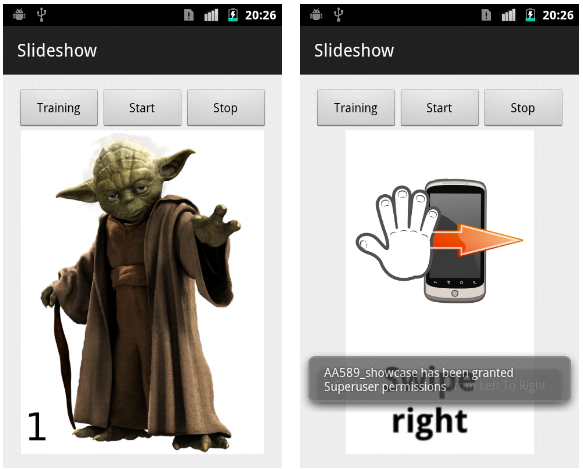
The setup for the proposed *wifiJedi* application requires a *rooted smartphone* for access to system resources. For this purpose, we used *Cyanogenmod 7*[3] on a *Nexus One* phone. Furthermore, we utilized the *BCMon*[4] firmware in order to be able to access the WiFi interface of the phone in *monitor mode*. For this, it is mandatory that the hardware of the phone employs one of two widely instrumented broadcom chipsets (bcm4329 and bcm4330). We also employed the library *libsuperuser*[5] in order to become able to issue terminal-commands as root from within an application.

B. Data processing

1) Recording data: The first choice for collecting RSSI information on Android Phones would be a *Java*-library. Yet, to our knowledge, all available libraries, do not suffice our needs: They do not capture all required data (e.g. sender, receiver, time, packet type) and, most significantly, provide average information rather than individual packet RSSI with an update frequency of not less than 1 Hz. Therefore, we used the *BCMon*-software to set the WiFi chip into *monitoring mode* and then *tcpdump* to capture data. We then parse all relevant information on the recorded packets from the output: *RSSI*-values, source *MAC*-address, packet type and the time stamp of the captured packets. We utilised libsuperuser[5] in combination with the *AsyncTask* class from the Android Developer Tools to run commands with root privileges without blocking the UI.

2) Data analysis: In order to improve recognition rates compared to previous work in the literature, we employ further features from the data.

1) The user will be asked to perform movements for a period of time t. This data will be recorded and stored into a file.
2) The recorded data will be divided into n slots. Each packet is sorted into a slot according to its *time stamp*.
3) We then use statistical analysis to compute the *features* for Machine Learning. Those features are for one timeslot the *statistical mean*, *standard derivation*, *highest* recorded RSSI value and *lowest* recorded RSSI value, *total number* of packets as well as *numerical representation* of the source MAC-Addresses.
4) With these features, we use a *k-nearest-neighbour*-Classifier to classify the data.



AA589_showcase has been granted Superuser permissions

Fig. 1: UI of the application (slideshow and training phase)

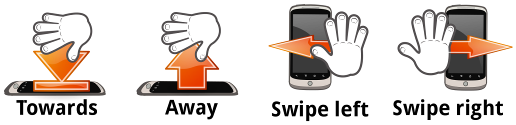


Fig. 2: All gestures considered. The Showcase Slideshow uses swipes over the phone to navigate within the slideshow. The corresponding classes are listed in Fig. I

All mentioned parameters can be set using the *Settings*-Menu of the application. We will later describe the setup we used to achieve the presented results.

C. wifiJedi application

We propose the *wifiJedi* application that is able to detect movement in front of a smartphone using the WiFi RSSI. It is a simple picture slideshow to demonstrate the RSSI-based gesture recognition. The UI of the application is displayed in Fig. 1. Before it is started, the user will be asked to *train* the application: To distinguish between several classes, we need an example of those first. After the training is complete, the user can start the slideshow. The different classes, that each represent a gesture, are displayed in Fig. 2. Within the "settings" menu, the user can optimize several settings, that have proven to be relevant to the performance. With a hand movement *from left to right* or *from right to left* the user then navigates trough the showcase slideshow.

D. Demonstration setup

The application is available online[6] and will be part of any given presentation. Users will be able to interact with a show case application by controlling the application via gestures. For all purposes, we consider the *GNU Free Documentation License* as on open source license.

[1] user.informatik.uni-goettingen.de/∼c.rauterberg

[2] https://github.com/crauterb/

[3] http://wiki.cyanogenmod.org/w/Install_CM_for_passion

[4] http://bcmon.blogspot.de/2012/09/working-monitor-mode-on-nexus-one.html

[5] http://su.chainfire.eu/

[6] user.informatik.uni-goettingen.de/∼c.rauterberg

TABLE I: The confusion matrix of a standard training phase. The movements corresponding to the classes are listed with the actual classes.

		Predicted Class				
		C_1	C_2	C_3	C_4	Recall
Actual Class	$C_1 = Away$	0.88	0.02	0.04	0.06	0.88
	$C_2 = Towards$	0.01	0.91	0.04	0.04	0.91
	$C_3 = Swipe_{right}$	0.00	0.02	0.95	0.03	0.95
	$C_4 = Swipe_{left}$	0.00	0.01	0.05	0.94	0.94
	Prec.	$0,98$	$0,94$	$0,87$	$0,87$	

IV. Evaluation of the Proposed Approach

First, we will evaluate the default usage of the showcase slideshow: After a training phase, while holding the phone in the hand, we record test samples to evaluate the accuracy of the classification. We will refer to this scenario as the *standard scenario*. In addition, with two phones we tested the classification in different scenarios:

- Training:We train the classifier whilst holding the phone *on a certain spot in a certain room* and then *walk about the room* and evaluate the classifier on different positions in the room.
- Ad-hoc: We train the classifier whilst holding the phone *on a certain spot in a certain room* and then *completely leave the context* and go to other spaces within the city and evaluate the performance there.

A. Choice of parameters

In all scenarios, we recorded $t = 10\,\text{sec}$ of training data for each of the classes. We divided this data into $n = 40$ slots, resulting in $t_i = 0.25\,\text{sec}$ duration for each time slot. To avoid the affects of fluctuations in the networks traffic, we did not record $t = 10\,\text{sec}$ of packets for each class at once, but iteratively recorded only $2\,\text{sec}$ for each class. For Machine Learning, we used *k-nearest-neighbour* with $k = 5$ neighbours. The *standard scenario* and the *training scenario* were each tested with 125 test runs, the *ad-hoc scenario* was tested with 60 test runs.

V. Results

A. Results within the Showcase Slideshow

Table I shows the confusion matrix of a validation in the *Standard Scenario*. We observe that the classes C_3 and C_4 are correctly identified with about 95% and that there is only a small number of *false-positives* to these classes. The classes C_1 and C_2 are correctly identified with about 90%. In Fig. 3 we can see the *statistical mean* over all relative *true-positives* for all three scenarios. Here we observe that on average more than 90% of the classes C_3 and C_4 are predicted correctly. This means, that in both scenarios we are able to control the proposed showcase slide show via gestures.

B. Results for the alternative scenarios

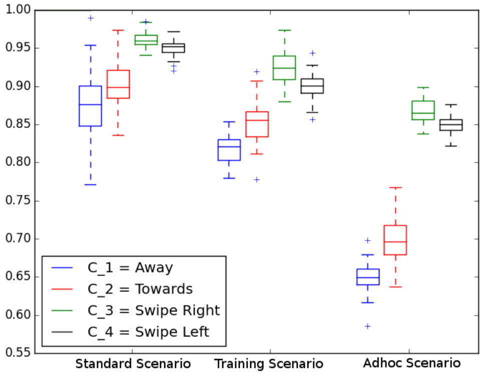


Fig. 3: An illustration off the averages of all experiments we did in all three described scenarios.

Fig. 3 shows, that in the *Ad-hoc Scenario* the average amount of correct classifications suffers greatly, especially for the classes C_1 and C_2. This is due to the fact, that this approach is still *very dependent on the current fluctuation within the network*: The change in context, i.e. number of access points and fluctuation leads to a low accuracy of the learned classifier.

VI. Summary & Outlook on Future Work

We have proposed a showcase application for online RF-based gesture interaction on Android devices. After initial training we are able to correctly classify certain gestures with an accuracy of about 95%. In future work, we want to circumvent the long initial training by providing several *scenarios* to the application. Upon starting, the application will *analyze* the current network and provide a likelihood of the provided scenarios.

References

[1] Bringing gesture recognition to all devices. In *Presented as part of the 11th USENIX Symposium on Networked Systems Design and Implementation*, Berkeley, CA, 2014. USENIX.

[2] F. Adib, Z. Kabelac, D. Katabi, and R. C. Miller. 3d tracking via body radio reflections. available online: http://witrack.csail.mit.edu/witrack-paper.pdf, 2014. Usenix NSDI14 - to appear.

[3] Fadel Adib and Dina Katabi. See through walls with wi-fi. In *ACM SIGCOMM'13*, 2013.

[4] Youngwook Kim and Hao Ling. Human activity classification based on micro-doppler signatures using a support vector machine. *IEEE Transactions on Geoscience and Remote Sensing*, 47(5):1328–1337, 2009.

[5] Qifan Pu, Sidhant Gupta, Shyam Gollakota, and Shwetak Patel. Whole-home gesture recognition using wireless signals. In *The 19th Annual International Conference on Mobile Computing and Networking (Mobicom'13)*, 2013.

[6] Stephan Sigg, Ulf Blanke, and Gerhard Troester. The telepathic phone: Frictionless activity recognition from wifi-rssi. In *IEEE International Conference on Pervasive Computing and Communications (PerCom)*, PerCom '14, 2014.

[7] Stephan Sigg, Mario Hock, Markus Scholz, Gerhard Troester, Lars Wolf, Yusheng Ji, and Michael Beigl. Passive, device-free recognition on your mobile phone: tools, features and a case study. In *Proceedings of the 10th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, 2013.

[8] Stephan Sigg, Shuyu Shi, and Yusheng Ji. Teach your wifi-device: Recognise gestures and simultaneous activities from time-domain rf-features. *International Journal on Ambient Computing and Intelligence (IJACI)*, 6(1), 2014.