Karlsruhe Reports in Informatics 2016,13
Edited by Karlsruhe Institute of Technology,
Faculty of Informatics
ISSN 2190-4782

Ubiquitäre Systeme (Seminar) und Mobile Computing (Proseminar) SS 2016

Mobile und Verteilte Systeme
Ubiquitous Computing

Teil XIV

Herausgeber:
Martin Alexander Neumann, Anja Exler, Andrei Miclaus,
Antonios Karatzoglou, Long Wang, Michael Beigl

2016

KIT – University of the State of Baden-Wuerttemberg and National Research Center of the Helmholtz Association

Karlsruhe Institute of Technology

Ubiquitäre Systeme (Seminar) und Mobile Computing (Proseminar) SS 2016

Mobile und Verteilte Systeme Ubiquitous Computing

Teil XIV

Herausgeber

Martin Alexander Neumann
Anja Exler, Andrei Miclaus
Antonios Karatzoglou
Long Wang, Michael Beigl

Karlsruhe Institute of Technology (KIT)
Fakultät für Informatik
Lehrstuhl für Pervasive Computing Systems (PCS) und TECO

Interner Bericht 2016-13
ISSN 2190-4782

Vorwort

Die Seminarreihe Mobile Computing und Ubiquitäre Systeme existiert seit dem Wintersemester 2013/2014. Seit diesem Semester findet das Proseminar Mobile Computing am Lehrstuhl für Pervasive Computing System statt. Die Arbeiten des Proseminars werden seit dem mit den Arbeiten des zweiten Seminars des Lehrstuhls, dem Seminar Ubiquitäre Systeme, zusammengefasst und gemeinsam veröffentlicht.

Die Seminarreihe Ubiquitäre Systeme hat eine lange Tradition in der Forschungsgruppe TECO. Im Wintersemester 2010/2011 wurde die Gruppe Teil des Lehrstuhls für Pervasive Computing Systems. Seit dem findet das Seminar Ubiquitäre Systeme in jedem Semester statt. Ebenso wird das Proseminar Mobile Computing seit dem Wintersemester 2013/2014 in jedem Semester durchgeführt. Seit dem Wintersemester 2003/2004 werden die Seminararbeiten als KIT-Berichte veröffentlicht. Ziel der gemeinsamen Seminarreihe ist die Aufarbeitung und Diskussion aktueller Forschungsfragen in den Bereichen Mobile und Ubiquitous Computing.

Dieser Seminarband fasst die Arbeiten der Seminare des Sommersemesters 2016 zusammen. Die Themen der hier zusammengefassten Aufsätze umfasst Benutzermodellierung und Privatsphäre in ubiquitären Systemen, Sicherheits- und Geschäftsmodelle im Internet der Dinge sowie Mobile Sensing. Wir danken den Studierenden für ihren besonderen Einsatz, sowohl während des Seminars als auch bei der Fertigstellung dieses Bandes.

Karlsruhe, den 01. Oktober 2016

Martin Alexander Neumann
Anja Exler
Andrei Miclaus
Antonios Karatzoglou
Long Wang
Michael Beigl

Inhaltsverzeichnis

Health monitoring based on smart devices

Lennart Gilbert*
Advisor: Long Wang†

Karlsruhe Institute of Technology (KIT)
Pervasive Computing Systems – TECO
*lennart.gilbert@student.kit.edu
†wanglong@teco.edu

Abstract. A collection of state of the art approaches to monitor health related vital signs and behaviours based on smart devices are presented. To make informed decisions and an accurate diagnosis, medical experts need to monitor the patients symptoms and health. This requires repeated clinical sessions for statistical significant ratings, as traditional measuring techniques have to be carried out by an expert (e.g. to operate the monitoring devices). The available mobile measuring devices are expensive, bulky and usually obtrusive to use. Considered an annoyance self monitoring suffers from low adherence, preventing the early detection of declining health. The ubiquity of smart devices is used to raise the frequency of self measurements, thereby lowering the rate of clinical checkups by the possibility of remote assessment. Furthermore they are used to raise the users awareness of their state of health. Which helps them to make changes to their unhealthy behaviours and supports them during their exercise. Techniques presented monitor amongst others respiration function, heart rate, eating behaviour. As well as offering support for food logging and exercise feedback.

Keywords: health monitoring, smartdevice, breathing, food logging, heart rate, exercise

1 Introduction

Monitoring the health of a person or different indicators thereof is an essential part of medical studies, but also during and after medical procedures to ensure a proper rehabilitation process. Outside the clinical area it is an integral part in battling widespread diseases like obesity and their likes. Offering easy, low-cost ways to monitor ones own health helps to prevent these diseases by tracking the own health development and adjusting the lifestyle accordingly.
A major obstacle in achieving good adherence and widespread use of health monitoring are the monitoring devices themselves. Traditionally to track the wanted vital signs specialized, expensive hardware is needed, which has to be operated and evaluated by a trained medical technician. For long term monitoring a patient has to regularly visit the clinic to either perform monitoring tests or drop of a mobile device for its evaluation. As such devices are expensive, only few people

can be supplied or afford them. Their measuring methods are intrusive and take time and training to properly set up. Although they are considered mobile, they are an annoyance to carry around due to size and form. In some cases people are ashamed to be seen with them in public, exposing their health problems. Outside these clinical settings with acute medical reasons, where these regular checkups are not a necessity, their price, ease of use and intrusiveness into the everyday life are a major hindrance for more people to start using them.
Therefore a new group of devices is needed, capable of health monitoring while being easy to use and adapt to. For proper tracking sensors are needed capable of measuring vital signs. Evaluation requires either a data storage, as the saved data has to make its way to the medical staff. Or on-device processing, evaluation and presentation of the results.
Such a group has been found in smart devices. They offer processing power and the needed storage, as well as ways to connect to other devices not bound to a location. The built-in multitude of sensors, programmability and ubiquity allows the implementation of diverse monitoring techniques. Additionally they are in most cases socially accepted and a large portion of the population already owns such a device, mitigating both the price point and intrusiveness. Therefore monitoring could achieve higher adherence, since no new advice is needed and it can run in the background without further interaction.
The challenge is to utilize the available sensors to create systems comparable in monitoring quality to dedicated devices or manual monitoring. In the following we present a diverse selection of state of the art proposals and systems covering vital signs including heart rate and breathing as well as secondary health aspects of eating habits and exercise monitoring.

2 Respiration

Breathing can be considered as the most obvious vital sign, offering a lot of information about a subjects health. Primarily related to the respiratory system, but also fitness level and exercise intensity. A major problem for assessing lung functionality is the sparsity of symptoms during short in-lab monitoring sessions. Long term studies are needed to capture statistically significant estimates. As self reported symptoms vary wildly from the ground truth, automated system are preferred to capture relevant events. Those can be realised with commodity smart devices in an unobtrusive manner.

2.1 Cough sensing

In [9] Larson et al. propose a mobilephone based system that accurately tracks cough frequency, while preserving the privacy of people in system range. Since coughing is a clear symptom to consider seeking medical advice, it suffers from a lack of accurate description of severity and frequency. The existing guidelines for cough monitoring systems led to advances in its automation. As Larson et al. discovered current systems don't fulfil all requirements, especially audio bases

methods regarding privacy of recorded speech. For example if extracted sound features are transmitted to a server for classification, conversations can be reconstructed from them. Sending non-invertible features does protect the privacy, but prevents experts from listening to cough sounds to remove false positives.
The proposed system is able to prevent a breach of privacy, while keeping the option to manually remove false positives. They achieve this goal by transmitting component weights instead of extracted features, from which only the cough audio can be reconstructed intelligible. The algorithm consists of four steps. First a cough model is generated from a manually annotated training fold. To do this a principle component analysis (PCA) is run on 40 randomly selected cough sounds, specifically on the first 150ms as they are generally consistent across subjects. The N components with the largest eigenvalue then build the model $\widehat{X}_N$. The spectrograms are reconstructed with the model resulting in N projection weights and a residual error as features. Additionally three energy means over the FFT are calculated, therefore $N + 4$ features are available for classification.
Following the model creation is a preprocessing step for event extraction. For each feature the threshold is set to retain 98% of all coughs in the trainings set. All audio that has to be classified is pruned by the five thresholds with the lowest false positive rates. This retains 96% of coughs and lets through $5\% - 16\%$ of other audio. Once the events are extracted the cough classification begins. A random forest classifier is trained on all features from the extracted training events putting more weight into cough errors. The classes are cough- and non-cough sound with the majority votes balanced $1 : 3$. The tree parameters weren't completely investigated, only a few variations were tested on a small data subset.
The last optional step is the reconstruction of cough sounds. An optimal PCA reconstruction is used to minimize the mean-square error between original and reconstruction. The resulting spectrogram estimation is then transformed back to the time domain. For this to work three additional values besides the the component weights have to be transmitted to the server. Fortunately they don't increase intelligibility, as they have little use in speech recognition.
The classification performance is evaluated on a five-fold cross validation, each fold containing all audio from 3-4 participants totalling in 72 hours and 2500 annotated coughs. A true positive is defined as any cough sound containing two consecutive classifier identifications in a 300ms window. A false positive are any consecutive identifications outside a 10 second window around an actual cough sound. The best performance is considered to be the highest recall with false positive rate below 1% . Across all folds they achieve am mean recall of 92% and a 0.5% false positive rate. Due to different sizes of sampling windows for the audio data, false alarms per hour are preferred as comparison metric over false positive rate. The analysis also showed that the most common false positives were either noise or speech (56%, respectively 43% of all false positives).
To evaluate the privacy aspect, volunteers had to write down the words they understood from reconstructed speech signals at $N \in \{5, 10, 15, 25, 50\}$ components. The word error rate was calculated compared to the baseline from the

original signal. With 15 components or less the WER is greater than 95% , it starts dropping with more components reaching 76% at $N = 50$ as only keywords can be spotted. In order to balance this with the fidelity of reconstructed coughs, the volunteers were tasked to rate two cough sounds based on similarity from dissimilar to indistinguishable. The ratings show that $N = 15$ is on the on the border between dissimilar and similar, with the fidelity growing if more components are used. All in all the results reveal that an automated cough detection with recall greater 90% is possible, while keeping coughs at a good fidelity but almost all speech unintelligible. Incorporated into a telemedicine system it allows medical personal to assess the symptoms without the patient having to visit the clinic.

2.2 Low cost mobile spirometry

With SpiroSmart [8] Larson et al. developed a smartphone application capable of reliable, low cost home spirometry, reaching a mean error of 5.1% compared to a clinical spirometer. Spirometry is an objective measurement of lung function, used to diagnose and monitor chronic lung diseases. During a test session flow and the cumulative volume of a patients exhalation are measured. Traditionally performed at clinics or with expensive portable home devices about the size of a laptop. These home devices offer the possibility for more frequent testing to detect changes early on which may need medical attention. But they lack a trained technician for coaching the patient in the device usage, as well as evaluating the test results. To raise the test quality and patient compliance SpiroSmart is run on a smartphone utilizing the built-in microphone, while guiding the patient through the test visually on the screen. The patient has to exhale their full lung volume at the phones screen while holding it at arms length. The exhalation is recorded and sent to a server to calculate the flow rate and graphs from which the airflow limitation can be quantified.
To transform the digital audio samples from the microphone to measures of airflow at the patients lips, a processing pipeline is introduced. The first stage has to compensate for the pressure drop between mouth and microphone and reflections that occur. A transfer function based on arm length and head circumference, both inferred from patients height, is applied to the audio resulting in an estimate for the pressure at the lips. These pressure values are converted to a flow rate with an estimated mouth opening. Since the estimations for arm length and head circumference weren't tested on their performance compared to real values, all three measures raw, lip pressure and flow rate are sent to the feature extraction. To calculate the volumetric flow rate from these high frequency signals, three different transformations are applied each resulting in an approximation of the flow rate. The first one calculates the signal envelope, which measures the overall signal power corresponding to the flow rate. The next transformation detect resonances in the spectrogram. As they can be assumed to be reflections in the vocal tract, the flow rate should be proportional to their occurrences. The last transformation uses the concept of linear prediction, meaning a signal can be split into source and shaping filter. In this case the filter is

a vocal tract estimation and the source the white noise exciting the filter equal to the flow rate power.
In the machine learning stage of the pipeline, the flow rate approximations are used as features in two regressions. One uses the values to calculate the lung function quantities and directly regress on the results with a bagged decision tree. The second regression creates the *flow vs. volume*-curve, which needs to integrate the flow. Flow and volume are estimated in two separate decision trees. The dataset for training and testing was gathered from 52 volunteers with self reported no or only mild lung conditions. As gold standard for comparison a clinical spirometer was used. After the volunteers were trained how to use the devices, three baseline efforts were recorded for each session . With SpiroSmart four usage configurations were evaluated: a sling to keep the arm length steady, a mouthpiece to keep the mouth opening steady, both and neither extra utensil. 10 participants were selected for subsequent sessions after several days to measure test consistency. Totalling in 248 gold standard and 864 SpiroSmart efforts. Because both devices can't be used at the same time the explicit ground truth for each effort is unknown. Therefore each effort was associated with an acceptable gold standard. The data was folded into several training subsets, each used to create a different regression model. These models form a global model for decision making with k-mean clustering. Larson et al. evaluate based on the error between actual and predicted value of the lung function quantities. For all 4 measures they reach without personalization of models a mean of less than 7%, performing best on subjects with normal lung functionality. Abnormal subjects profit greatly from personalized models lowering the mean error to less than 5%. Each lung function quantity has an allowed limit of variability when measured. SpiroSmart has an accuracy of at least 80% to stay in those limits for normal subjects and 60% for abnormal. With these accuracies an effective diagnose for obstructed lungs is possible. It was also found that neither fixed arm length nor a mouthpiece decrease the error enough for them to be necessary. 10 generated *flow-vs.-volume* curves were evaluated in an online survey for pulmonologists. Regardless which device created the curves, a general agreement was found regarding lung obstructiveness. Thus showing the feasibility of low-cost home spirometry without a dedicated device and the possibility of spirometry via telemedicine.

2.3 Respiratory symptom tracking

SymDetector [15] is a lightweight, pervasive monitoring app to track respiratory symptoms. A smartphone is used to continuously sample its environmental audio, recognize potential symptoms, log and categorize them. classification is done on the fly to ensure privacy by not saving or transmitting raw audio data. It shows high accuracy at detecting the four symptoms sneeze, cough, sniffle and throat clearing in various indoor settings. To detect symptoms unobtrusively the system adapts to the phones location, tested for desk, pocket and backpack placement.
The audio is sampled at 16kHz and segmented into non overlapping 50ms frames.

A preliminary data set of symptom lengths showed that symptoms last for several frames. Therefore 80 frames are grouped together as a window for the detection step with 1 second overlap. The detector selects windows with potential respiratory symptoms by filtering out as many non-symptom windows as possible. Specifically ambient noise and continuous acoustic events, the two most common non-symptoms. 3 time domain features are calculated to do so. Root mean square (RMS) of a frame to measure the energy in a frame, which is used for the other two window based features. Above α-mean ratio (AMR) denotes the ratio of frames with an energy above α, to distinguish between discrete, continuous and noise acoustic events. Average of top k RMSs (ATR) is the RMS average of the k highest energy frames, to distinguish high and low energy events. Based on these features AMR filters out continuous events like background noise. The AMR values don't change much depending on context, therefore it is robust to distinguish categories regardless of the phones context. Since ATR reflects the event energy, it is used in the next step to distinguish events caused by the user and more distant events (e.g. bystander coughing). A dynamic ATR threshold γ is used, which adapts according to the phones context due to energy loss when recording in a pocket/backpack. γ is calculated and updated from every window classified as talking and saved as intensity level.
For classification additional features are extracted to better distinguish the remaining acoustic events. Symptom length (SL) is the largest frame set covered by the event, built by region growing from the maximum RMS frame f_m. Left to right ratio is the length ratio of frames before and after f_m to distinguish sneeze/sniffle and cough/throat clearing. Relative maximum RMS (RMR) uses normalized RMS and intensity γ to compare event energy regardless of the context. Thus splitting the 4 symptom events into two groups sneeze/cough and sniffle/throat clearing. The zero crossing rate (ZCR) of a frame to detect percussive sounds. From the fast Fourier transformation of frame f the spectral centroid (SC) is calculated measuring the spectral energy distribution and the bandwidth to evaluate spectrum flatness, as well as λ-percentile spectral roll off (SR). SR tells up to which frequency bin λ % of the spectral energy is contained. At window level mean and variance from these frame spectral features are calculated. To prevent strong class imbalance a pre-classifier is used to split events by RMR into the mentioned categories. Then SL and ZCR are used to remove non-symptom sounds from both categories and SR to remove high frequency events from group sneeze/cough. After this coarse classification SVMs are trained for each category with one-against-one to also discern the non-symptom class. Classified symptoms are logged with the start and end time from SL. The preliminary study for feature engineering collected sounds from 5 users over 7 days in total 105 hours audio. The participants labelled their symptoms themselves to preserve privacy.
For the evaluation study with 16 participants, a data collection scheme was used. With the AMR threshold only discrete audio events were recorded, reducing the labelling work to 15 hours for 204 days monitored. The classification was evaluated with cross validation and leave one participant out, reaching a recall for

symptoms of at least 82% and 99% for non-symptoms. Precision was also over 82%, meaning few events were misclassified. Since sniffles are short and have low energy, they produce the lowest TPR and PPR. These results confirm that reliable and accurate automated detection of respiratory symptoms is possible with smart devices, extending the state of the art from only cough sensing by three symptoms (sniffle, sneeze, throat clearing). Whether the severity of the symptoms can be derived or the symptom can be reconstructed from extracted features wasn't elaborated. Which would be useful for automated long distance assessment.

2.4 Discussion

	Aim	Capture method	Sensors	Processing
2.1	cough sensing	ambient sound	microphone	feature based threshold
2.2	mobile low cost spirometry	active spirometry	microphone	lung function quantities from flow rate
2.3	respiratory symptom sensing	ambient sound	microphone	filter non-symptoms and pre-classification

	Features	Prediction	Results
2.1	PCA, projection weights, FFT energy	random forest	recall 92%, false positive rate 0.5%
2.2	flow rate, raw audio, lip pressure	bagged decision trees	accuracy 80% (60%) (ab-) normal lung functionality
2.3	7time-domain[1], 3frequency-domain[2]	SVM for each category	recall and precision over 82%

Table 1. Comparison of approaches

With CoughSense and SymDetector we have two systems for automatic symptom tracking with similar detection rates. Their main difference is that CoughSense (CS) can only recognize coughs, whereas SymDetector (SD) is able to discern three symptoms besides coughing. For classification SD uses on-device SVMs. CS however, being an older approach, couldn't as the processing power of mobile phones at the time of writing wasn't sufficient for on-device classification. Therefore a random forest on a cloud server is used, which lead to the need of privacy preserving design decisions. With state of the art smartphones

[1] Root Mean Square, Above α-Mean Ratio, Average of Top k RMSs, Symptom Length, Left to Right Ratio, Relative Maximum RMS, Zero Crossing Rate

[2] Spectral Centroid, Bandwidth, Spectral Rolloff

it should be possible to realize the CS classification on the device. But since SD performs better, the only advantage of using CS is the possibility to reconstruct symptom audio. A reasonable consideration is using the CS component extraction on symptoms recognized by SD, thus they can be transmitted and reconstructed for remote assessment as intended by CS. As SD has to buffer a few seconds of raw audio, the PCA can be run after symptom recognition on extracted events by only slightly increasing the overall buffered raw audio. It is to evaluate how many components are needed per symptom to reach good audio fidelity and whether speech is intelligible.

3 Heart rate

The heart rate reveals plenty about health and fitness, but also about the emotional state and stress level. It is used to estimate exercise intensity to ensure effective training and prevent early exhaustion. The heart rate is an integral part of assessing the overall patient health and monitoring heart diseases. Making the measuring easy to use and available on commodity devices, lowers the effort and cost needed.

3.1 Smartphone based PPG

Grimaldi et al. proposed a system [3] capable of detecting photoplethysmography (PPG) signals with a smartphone. PPG monitors the changes in intensity of light scattered in tissue, either by transmission or reflection. Common devices are either attached to a finger or earlobe, emitting red and infrared light into the skin. A photodiode in close proximity can infer the heart rate based on the properties of the reflected light. Such special devices are useful in a clinical or stationary setting, but are cumbersome to carry around. Since smartphones contain everything needed for PPG and offer the wanted mobility, they are the perfect platform for this approach.
The system utilizes the phones camera and the neighbouring LEDs. A finger is placed firmly on both, covering them completely. The LEDs are the light source illuminating the finger tissue. The camera records changes in illumination caused by the blood flow. To ensure wrong usage won't deteriorate the result, each frame is evaluated for proper finger placement. A frame is considered proper if the average intensity of red pixels is high enough and the intensity of green and blue pixels low enough. As their evaluation showed, red pixels carry the best PPG signal across different smartphone models. The PPG signal is then calculated by counting the amount of pixels higher than a threshold. The threshold can't be constant due to the fact that cameras and the finger placement differ between uses, users and phones. Instead it is calculated from the average minimum and maximum of red pixels over a 5 second calibration window. During the measurement the center of pulsations is calculated. If its locations or width-height ratio changes, the PPG computation should be restarted. This prevents wrong readings due to movements of the finger.

They evaluated their system with different smartphone models compared to a commercial finger pulse oximeter. Video data from one finger was recorded on the phone while the oximeter was on a different finger of the same hand at rest and after 1 minute of squatting. For processing and better comparison the video data was transferred to a pc. They present two calculated heart rate curves which show high correlation between actual and estimated pulse. Unfortunately not enough data is given to actually draw conclusions on precision and accuracy.

3.2 Workload estimation during walking

In [14] Sumida et al. manage to reliably estimate the physical load of a walking person from the accelerometer and GPS sensor of a smartphone. Keeping an exercising person at an appropriate load, prevents injuries and loss of motivation due to a non-effective training. The physical load is expressed as a variation of the normalized heart rate, as the absolute heart rate varies depending on the subject. Wearing a dedicated heart rate sensor lowers the convenience of a walking workout and smart device applications that need interaction with the device are complicated to use during walking. Therefore a system is proposed that monitors the acceleration and GPS sensor during the exercise, with the smartphone somewhere on the person.
The main goals are to measure the physical load and its variation during the walk, as well as making the system adaptable to a user. The adaptation is necessary, because the heart rate at rest and its change over time depend on user factors like age and their exercise habits. The estimation therefore depends on a shared database to predict the heart rate from uploaded walking data. This walking data is used during training for model creation. The acceleration amplitude is used to derive the walking speed from distance and time walked. The distance is estimated from a relative location with stride length and step count. Additionally the variation of oxygen uptake is estimated periodically from walking speed and gradient of the route, calculated by sampling the GPS height value. The oxygen uptake is used in the model for the estimation of workload change, as it correlates better with the variation of exercise intensity.
The prediction model is based on a neural network with one hidden layer and the described walking data as parameters for the input layer. Given the data from a sampling window the output layer delivers the estimated heart rate for the start of the next sampling window. The recorded heart rate data was normalized, to train the network only on heart rate variation not on subject depending heart rate levels. For adaptation to a subjects specific fitness, characteristic parameters are added to the oxygen uptake formula. They are used to minimize the average error between estimation and real oxygen uptake, for different subject groups without the need for a new model.
The evaluation dataset was gathered from 18 subjects on 5 different routes. The subjects categorized themselves in groups of either frequent exercise, almost no exercise or neither ($5, 2, 11$ subjects each). The models were trained with the folds *leave one subjects out*, *leave one route out*, *leave one subject and route out* and *leave one subject-route combination out*. The results for the mean absolute

error between predicted and actual heart rate never exceeds 7bpm, even for the worst case of subject and route fully unknown. Surprisingly leaving only one combination out is only 0.37 bpm better. Deeper analysis shows most error is introduced at a high heart rate, likely due to GPS errors or low granularity of subject categories. Furthermore training a subject(-category) is more important than training a route, as a route training also suffers from the low granularity of categories. Overall the estimation is accurate enough to keep the needed workload for an effective training. The application relies on GPS and a cloud based prediction server, both increasing drain on the battery. Furthermore the physical load estimation and resulting exercise advice are only given visually on the phones screen. Adapting an eyes-free feedback system would increase the exercise benefit during the walk, as the user doesn't need to turn their attention to the phone to check the physical load.

3.3 Heart and breathing rate measured at the wrist

With Biowatch [6] Hernandez et al. created a reliable system to recover heart and breathing rate from motion sensors worn at the wrist. Their research goals was to determine how precise built-in motion sensors of smart watches can measure heart rate (HR) and breathing rate (BR) compared to traditional approaches. Furthermore they tested performance of HR measurement in real-life sleep settings. The method used is based on ballistocardiography (BCG). The blood flow induced by the pumping heart causes minor movements due to the shifting mass. Those can be captured by motion sensors and depending on their location even lung activity is picked up. The presented approach uses 3-axis accelerometer and gyroscope from a smart watch to extract HR and BR from the sensor data. Each component is normalized to z-scores to even their relevance. HR and BR are estimated from pulse and respiratory waves. The pulse wave is estimated by first applying a averaging filter with window length 1/7 second. This removes signal shifts and trends caused by body motion, but preserves BCG signals. Then a bandpass Butterworth filter with range 4-11Hz is applied. Sensor components x_i are aggregated by $x = \sqrt{\sum(x_i)^2}$ lowering the influence of different body postures. A second band pass BW filter with cutoff frequencies 0.66Hz and 2.5Hz limits the resulting pulse wave to a reasonable range of 40 to 150bpm. For the respiratory wave an averaging filter is used with window length equal to one breathing cycle at 40 breaths per minute, thus eliminating cardiac motions. The most periodic component signal is selected as the wave, here the signal with maximal amplitude within 0.13Hz and 0.66Hz (8-40 breaths per minute).
For comparison a HR gold standard was recorded with a chest ECG sensor. Additionally a wrist worn PPG sensor representing state of the art devices as baseline was used. The BR gold standard was recorded with a commercial chest belt sensor. Different sensor were combined for analysis whether it would yield better results than single sensors. Two validation studies were conducted. The first in a lab with 12 participants in 3 different body positions (standing, sitting, lying) each at rest and after 1 minute of exercise. Each recording lasting 1 minute. The second was a in-situ study of HR during sleep. 3 participants wore

the smart watch and the ECG sensor for two nights of which 6 hours each were recorded. To not alter sleeping behaviour, no BR gold standard was recorded due to the cables of the needed chest belt. Participants got two non-recorded nights to adapt to sleeping with the applied sensors.
The lab data was split into 20 second windows at 75% overlap, with average HR 76.7 (standard deviation 14.26) and BR 16.6 (std dev 4). The mean absolute error (ME) of the combination accelerometer-gyroscope was with 1.27 less than each sensor on its own (1.39/2.01) but worse than PPG (0.95). The combination of all 3 performed best with a ME of 0.88bpm, meaning the body motions contain info about the HR the PPG can't capture and each (gyroscope/accelerometer) contain parts of it. For BR gyro outperformed accelerometer and the combination of both with 0.38 compared to 0.97/0.55. As the estimation extracts the most periodic signal, adding more data isn't always beneficial, especially when estimates are already that good. The same BR results showed for different postures, gyro outperformed both other options. The HR while sitting and lying was best with the PPG sensor, but almost the worst for standing since the device loses its needed tight skin contact when the arms are hanging down. Combined with the accelerometer it improves to the best estimate. Overall the gyro alone performed worst, followed by the accelerometer. But the combination of all three yielded the best result. Of 21 mean error values 17 were less than 1.5bmp compared to a precise ECG sensor using electrode gel. Therefore the estimation can be considered accurate.
The sleep experiment delivered 31.5 hours of data, which was segmented in 20 second windows at 95% overlap. Additionally segments containing non BCG body movements were removed by applying a threshold to the accelerometer data, retaining 85.9% of the windows. The mean absolute errors are comparable to the lab results. Again the combination of all sensors yielded the best estimate, closely followed by gyro/PPG combo. The gyro ME improved by 1bmp compared to the lab setting. This shows that motion sensors are a viable source for heart and breathing rate, compared to PPG sensors. Furthermore they can be used to increase the precision of PPG based heart rate sensors, while enabling such devices to capture breathing rate as an additional vital sign.

3.4 Intermittent smartphone based PPG

BayesHeart [2] is a blood volume pulse monitor based on video from camera phones. Able to measure heart rate and phases of the cardiac cycle accurately even for intermittent and noisy signals, its main advantage compared to other commodity camera based techniques. They suffer from illumination differences, motion noise from the region of interest and the cameras low sampling rate (30Hz), which prevents the use of common noise cancelling algorithms. The observable PPG signal contains all 4 phases of the cardiac cycle, therefore BayesHeart uses a 4-state hidden Markov model to segment the signal and estimate HR from the phase segments. Due to extrinsic noise the third phase can't always be segmented, leading to a two phase cycle. This problem is eliminated

by using a 2/4 state model which chooses either 4- or 2-state HMM for segmentation.
Instead of absolute observations, which are noise sensitive, *local trends* for each sample are used as features (in/de-creasing observation, local min/maxima) to encode phase specific regularities. The model is trained offline using Baum Welch for parameter estimation. The online HR estimation is done in 4 stages. First the Bayesian information criterion is used with the first 5 seconds of observations to select the HMM to use (2 or 4 states). The Viterbi algorithm generates the most likely state sequence to create the observed signal for both HMMs to reduce latency from model selection.
In the selected sequence a new cycle is marked as the state transition from last to first state. The duration between two adjacent marks estimates the instant HR. Outliers are reduced by dropping HR outside the 30-300bpm range and changes between two estimates greater 5bpm. To correctly estimate HR from intermittent signals (Region of interest appears not continuously in front of the lens) three problems have to be dealt with. To detect a covered lens they've used a linear classification model with global mean and standard deviation of frame pixels, resulting in sequences of covering actions. The next problem is noise caused by these covering actions, primarily finger movements and pressure changes. Therefore covering sequences less than 2 seconds are fully discarded and the first second of sequences longer than 2 seconds. The last problem is how to estimate HR from the now fragmented signal. To utilize as much signal as possible, they concatenate adjacent sequences by joining the last valid peak in the tail second of a sequence with the first peak in the first second of the following sequence. As valid peak chosen is the local maximum in each second long window.
They investigated their design choices with a study featuring 20 subjects and two 10 minute monitoring sessions (static and intermittent lens covering). Patients covered the lens fully with their fingertip, for intermittent every 5-10 seconds they removed their finger for 1-3 seconds. The phone was operated with one hand, while a pulse oximeter measured the gold standard on the other hand. Each sessions data was split evenly on training and test set. Reported metrics are mean error rate (MER), estimation latency which is the time an algorithm needs for the first accurate HR estimation ($\pm 5\%$). As well as utilization rate, ratio between samples used for estimation and total samples. The HMM selection chose 4-state for 14 subjects and 2-state for the other 6. The MER was better overall with 3.66% for normal covering (5.34% intermittent) compared to only 4-state 4.08% (5.84%) and only 2-state 4.84% (6.9%). This shows both state models improve the MER. The impact of the used post-processing was even bigger, lowering the MER from 10.86% to 5.34% for intermittent covering, while reducing the utilization slightly by 11.2percentage points to 68%.
A second study was conducted to compare state of the art noise reduction and pulse counting techniques. The noise reduction methods are red channel only(R), brightness (Y) only, independent component analysis (ICA) of RGB and PCA of RGB. Pulse counting was temporal domain counting by heuristic, FFT frequency

domain counting and BayesHeart, resulting in 12 tested combinations. Compared to R all noise reduction techniques improved the estimation. BayesHeart resulted in a lower error rate than the other pulse counting measures. For intermittent covering the combination Y and BayesHeart worked best with an MER of 5.23% and for normal covering ICA with BayesHeart 3.44%. R with FFT performed worst in both cases (normal 6%, intermittent 12%). With the achieved accuracy it is possible to use smartphone based PPG from intermittent camera covering actions during normal use. The estimated heart rate can then be applied to other applications for context sensitive usages.

3.5 Ballistocardiography from wrist motion

Haescher et al. [4] used a wrist worn accelerometer for ballistocardiography (BCG) at rest, to infer heart rate (HR), breathing rate (BR) and muscle vibration (MV). The accelerometer captures motions caused by breathing, blood flow and muscle fibres grinding against each other. These motions happen at a frequency equal to their respective vital sign (e.g. blood flow $\sim$ heart rate). The 3 axis raw sensor signal is converted into its magnitude signal, reducing dimensionality and influence of the devices orientation.
BR is extracted by low pass filtering the magnitude signal. For MV a high pass is applied to the raw magnitude signal, removing gross body motions like breathing. The resulting signal is low pass filtered and squared and the HR can be estimated. On each feature signal a FFT is performed to extract the actual frequencies. The frequency with the highest amplitude in the spectrum is selected. The approach was evaluated with data from 15 participants. A pulse oximeter was attached to the right hand and a IMU to the wrist to capture accelerations. On the left arm a blood pressure monitor and a smart watch with PPG sensor were attached. BR was recorded by chest belt and microphone on the head. During the recording participants lay as calm as possible on a blanket on the ground. Gold standard for HR was the oximeter and for BR the recorded audio. The IMU performed best for HR with a mean error of 1.63% followed by the blood pressure monitor at 2.32%, both showing no statistical significant difference to the gold standard. The smart watch performed worse than both devices at 5.58% mean error. The chest belt delivered the exact same BR as the gold standard. Oximeter and IMU performed significantly worse (mean 12.47%/16.6%). In actual values the biggest percentage errors form the IMU were 4bpm for the HR and 5bpm for BR. This shows acceleration data can be used to measure HR at rest accurately and BR *good enough*. In [5] they present a one participant study on IMU estimation quality of the three vital signs, based on IMU location on the body. Measuring at 12 positions they created a heat map for each vital sign depicting the peak signal to noise ration of the feature signal. Overall the results show that most smart devices with a built-in accelerometer can reliably estimate these vital signs.

3.6 Discussion

Both PPG and BCG allow accurate estimations for heart rate using smart devices. Their main difference lies in the intrusiveness. As PPG needs at least intermittent lens-covering actions, it is necessary for the user to interact with the device. This makes phone based PPG unusable during high motion activities like running or workouts. BCG on the other hand suffers during these exercises from motion noise obscuring the HR/BR motions. Therefore both approaches are more or less only suitable for after workout or rest heart rate estimations. Physical load estimation as presented in 3.2 was only test during outside walking exercises. As the oxygen uptake is based on route gradient measured via gps and walking speed estimated from accelerometer, it is unclear if it would work in stationary situations like a treadmill. Since the route would have zero gradient the uptake estimation has to rely solely on the walking speed, whether the model is correct in this situation is to be tested. Furthermore the gps sensor is only used for the height value, which could also be measured with a barometric altimeter at a lower energy usage, if one is available in the device.

	Aim	Capture method	Sensors	Processing
3.1	HR estimation	PPG	Camera	intensity threshold
3.2	walking Physical load/HR	oxygen uptake	Accelerometer, GPS	step counting, distance/gradient est.
3.3	HR,BR est. at wrist	BCG	Accelerometer, gyroscope	normalization, bandpass filter
3.4	Pulse monitor	PPG	Camera	HMM selection
3.5	HR,BR,MV est. at wrist	BCG	Accelerometer	low/high pass filters

	Features	Prediction	Results (ME)
3.1	Pixel intensity	Peak counting	unclear
3.2	Walking speed, route gradient	Neural network	$<$ 7bpm
3.3	Pulse wave	Peak counting	HR: 1.27bpm, with PPG 0.88bpm BR: 0.38bpm
3.4	Pixel intensity trends	HMM	2.5bpm/3.6bpm
3.5	FFT spectrum	Frequency amplitude	HR: 1.11bpm, BR: 16.6%

Table 2. Comparison of estimations. Mean error rate converted to ME at 68bpm

4 Eating

As obesity is a massive health problem in many countries, people usually are not aware how many calories they consume during their normal meals. Food logging applications can be used to raise the users awareness of how much they've eaten during the day. This allows them to counteract and not exceed their recommended daily calorie limit. Furthermore eating behaviour and speed has influence on how meals are processed in the body, when satiation kicks in and how much is eaten. These can be monitored to identify possible starting points to change the eating behaviour.

4.1 Chew counting from in-ear sounds

Nishimura and Kurroda [11] turned a bluetooth earpiece into an in-ear microphone for accurate counting of chewing. As monitoring eating habits can be achieved with a multitude of sensors in different ways, many of them are invasive and need long setup time, causing low adherence. An earpiece is small enough to be mobile, easy to use and in socially accepted to wear publicly. For testing a commercial bluetooth headset was used, with the speaker internally replaced by the microphone. The headset is worn as usual, but now recording and transmitting the sounds from inside the ear canal. Besides chewing and other mouth sounds, head scratching and swallowing can be observed in the audio signal. Whereas the earplug lowers environmental noise, hereby keeping it from influencing the signal.
A single chewing sound consists of two characteristic parts, the bite when upper and lower jaw meet and the jaw opening in preparation for the bite. The series of chewing sounds after the first bite start at high energy as the food is crushed into smaller chunks. The energy gradually falls as the chunks are ground into a fine paste for swallowing. The average occurrence rate of chewing sounds is between 1 and 2Hz.
The chewing recognition algorithm first detects *chew like* events in the audio signal. The signal is windowed and filtered with a low pass Butterworth filter. With adjacent windows the regression coefficient for one window is calculated. A chew like sound is detected when the regression gradient crosses zero and the LP filter output exceeds a threshold. For verification a chewing sound model was trained with MFCCs extracted from 100 chews from 10 different food samples. Other detected chew-like sounds like humming or coughing were used to train the non-chewing class.
Only *characteristic* chews for each food are used, o raise classification quality, by filtering out chews that only yield common feature vectors. At 90% true positive rate (Sensitivity) the use of only characteristic sounds drastically lowers the false positive rate from 65% to 2%. Detection performance was evaluated on the chewing sounds from 5 different food types, at average 516 chews per type. The average error rate was 1.93%, with 0% the lowest for salad and 3.45% the highest for wafers. In conclusion in-ear sounds are an accurate detection method for chewing counts, that can be recorded with minimal intrusion. At the

need of a special in ear microphone, the benefits of high accuracy and minimal intrusiveness are reduced by low adherence to extra single-purpose devices.

4.2 Eating activities from throat sounds

Bi et al. present a system [1] for recognizing eating related events like chewing. Their aim was to create a low cost device for eating monitoring that is convenient to use and not interfering. The system consists of a throat mounted microphone and audio processing unit, that sends audio signals to a smartphone for recognition.
A sound event is detected if a sample value exceeds a predefined threshold. Starting with the current one, 100 samples are grouped to a single complete event. 10 statistical features are extracted each from the time and frequency domain of an event. All features are normalized to zero mean and Relief-F is used for feature selection. As recognition model K-nearest neighbours and support vector machines were used. The training data was recorded in a quiet lab from a single subject. 30 annotated events were registered for breathing, chewing, swallowing and no-event (microphone hanging in the air) each. Folds are created using the first $k \in \{1, ..., 20\}$ sorted features and for each one KNN and SVM model is trained. For KNN the optimum is at $k = 11$ features with a recognition accuracy of 87%. With precision and recall of chewing the lowest at 70% respectively 80%. The SVM jumps to over 95% accuracy if more than 10 features are used, with the maximum 98% at 17 to 19 features. All single recognitions reach with that many features at least 95% recall and precision. Even though these results indicate good detection rates and an improvement to the multi class eating activity detection problem, their training and test data was too small for statistical significant evaluation.

4.3 Eating activities from head movements

Rahmann et al. used head movements to recognize eating activities [13]. Utilizing the built in sensors and display of google glass, they've created an unintrusive recognition and feedback system for monitoring of eating activities. The head movements consist of sensor data from accelerometer, gyroscope, magnetometer and processed values for gravity, linear acceleration and rotation (device orientation). The raw data was split into minute long windows for feature extraction. The window length was chosen to capture periodic head movements at a granularity that allows onset activity detection. From each sensor value and axis 5 statistical, 3 spectral and 2 temporal measures are calculated, resulting in a 180 element feature vector per window labelled with the ground truth activity. Cross validation folds were created using leave one participant out (LOPO), leave half participant out (LHPO) and leave one window out (LOWO). For LHPO the labelled activities of one test person were halved, one part was the test data the other was added to the training data.
As classification models Gaussian naive Bayes (NB), k-nearest neighbours (kNN) with k=1, C4.5 decision tree and Random Forest were evaluated. NB and kNN

were trained only on the 10 most informative features, chosen by a not specified information gain criterion. Besides area under curve (AUC) a $F_{0.5}$ score is reported as performance metric. The F score weights precision higher than recall, because in a feedback system false positives cause falsely logged events and high precision is preferred.
The dataset was collected from 38 participants over a 2 hours period each. During which start-, end time and label of each performed activity was marked independently by two researchers. With eating detection as main goal, participants ate a meal wit a short break in between and performed other daily activities like walking stairs, reading, drinking or doing homework. This resulted in a class imbalance of only 12% eating but 72% homework/free time samples. Tree based classifiers performed better than the other two and improved with more data, LOWO RF performing best (F-score 67.55%, AUC 0.922). For 9 participants (23% of all part.) no eating was detected of which 2 often adjusted their glasses, influencing the sensors and one moved abruptly in their chair. On the other hand for 11 participants 100% precision was achieved. For detected events the average recall was 30%. The completely undetected meals are most likely due to the massive class and in-class imbalance, as participants were allowed to bring their own food the meals varied in length and food type. In a food logging scenario the precision is more important than recall. In case of a false positive a wrong eating event would be logged or the user annoyed by a wrongly timed reminder. A false negative on the other hand is only relevant for the given sampling window, one of many during an actual meal. It is to assume the detection rate will rise with more meals in the training set. Given the uncertain future of the Google glass, any broad use of this system in the foreseeable future is unlikely. But the findings on eating recognition from head movements can be applied to other head worn devices with accelerometer.

4.4 Unintrusive food intake and habit monitoring

Zhou et al. present a non obtrusive sensing platform [19] to monitor food intake and recognize various dining motions. They've combined a fabric based pressure matrix from their previous work on pressure sensitive surfaces with tactile force sensors and a dining tablet, into a mobile non invasive monitoring platform. They are able to detect low level eating actions without body worn sensors or special dinnerware outside of special lab settings. Furthermore they can assess food weight and arbitrary content.
The used smart-tablecloth is a resistive force distribution sensor matrix with a spatial resolution of $1cm^2$ and 24bit real time accuracy. Externally applied force changes the resistance at intersections between column and row electrodes, transforming the force into an electric signal. The sensor matrix is attached to a normal dining tablet and covered with a layer of cloth and plastic to protect it against heat and dirt. A force sensitive resistor (FSR) is attached at each tablet corner and connected with the matrix to the sampling IC underneath the tablet. The IC transmits the sensor data via USB to a computer for further processing. Matrix and FSR are sampled at 40Hz. Both sensor systems track pressure

change of different spatial resolution. The matrix is used to differentiate between plates and pressure distribution on them and the FSR measures the weight at high accuracy.
During the experimental stage the focus was on actions performed with common western cutlery, like cutting, poking, scooping. 7 action classes for different main-, side dish and drinks were distinguished and a 8th no-action class. 5 participants ate 8 meals, each containing one main and side dish and a drink, with varying food type for the action class. They had free reign over how to eat, how long and whether they would socialize during the meal. The experiments were recorded for manual labelling of the data. The raw sensor data undergoes several steps until a classification model is created.
First the matrix output is conditioned by applying image processing methods. After being up scaled, background nodes are removed from each matrix by a dynamic threshold. Using the most common bin of a 10 bin histograms the frames are converted to binary frames. 40 consecutive frames are summed to a single non binary frame. The biggest circle in that frame is detected (phase coding by Atherton and Kerbyson) as the main dish and removed from the frame. Side dish and drink are the best remaining circles in the left/right half of the frame. This simplified localization is used as dynamic localization isn't focus of their work. With these 3 circles the raw sensor data is segmented into subframe P1,P2,P3 each representing one dish/glass on the tablet. The background is removed again same as before. The FSR data is mean filtered to R4.
For feature extraction the average value of a frame is calculated as weight and its center on x and y axis as center-x respectively center-y. All three values are normalized. 68 statistical features are then calculated, 48 from weight and center, 20 from the 4 channels in R4. The classification model was created with confidence based AdaBoost and decision trees using *10 fold cross validation* for participant dependant and *leave one participant out* for independent evaluation.
To extract actions for classification from raw data a spotting algorithm is evaluated. The normalized data for feature calculation is used to create two binary frame masks. One based on the standard deviation threshold extracts continuous activities like cutting from the data. Abrupt activities like removing a glass for drinking are difficult to spot and group together. Therefore the second mask tracks the overall *trend* of the data. The masks are combined by bitwise *or*. A sequence of ones is considered a spotted action.
This algorithm reaches a F_1 score of 87% for participant dependent folds. In this case recall and precision look at matched and extracted actions. An action is considered matched, if it overlaps by 40% with a ground truth label of the same class. For patient independent classification the accuracy drops below 80% as eating habits vary between patients. Most miss classification happens for classes with the same location or softness. Scooping soup is sometimes classified as no event, when the spoon didn't expel force onto the dish. Using action class and spotted activities for majority voting, the contents of 30 out of 40 meals could be successfully estimated. Even if food classes are much bigger in reality, the dining actions aren't as diverse. Therefore good estimations should be possible

with adequately trained classificator. The R4 values showed a non random correlation to the food weight (error ratio of 17%), hence good estimates can be given for the actual weight. The tablet offers a good recognition rate, but is limited in its application area due to cost and mobility. Furthermore the fine grained evaluation capabilities are only needed in some special cases, making the system ideal for stationary rehabilitation of patients with eating disorders.

4.5 Eating detection from ambient sounds

Thomaz et al. conducted a feasibility study [16] to recognize eating activities from sounds recorded with smart devices. Instead of fully automated food tracking, which takes away the benefit of self reflection from tracking meals, they aim at semi-automated food logging. The system recognizes eating activities and reminds the user to note the contents of the consumed meal. Thereby reducing the effort needed, but preserving the reflection aspect of food logging. Their main goal was a system capable of doing so without any additional hardware besides a smart-phone or -watch and minimal intrusiveness. Additionally the system is supposed to infer eating activities not only from eating sounds, but also from environmental sounds depicting the context of the meal.
As the focus lies in the feasibility of recognition from ambient sounds they rely on proven previous works for their implementation. Audio is sampled at 11025Hz and 50ms long frames are extracted with a Hanning filter at 50% overlap. For each frame 50 features are calculated, amongst others Zero Crossing rate, loudness, energy, envelope shape statistics, linear predictive coding(LPC), line spectral pairs as well as spectral-flatness, -flux,-roll-off, -shape, -statistics and variation. As frames are shorter than most characteristic ambient sounds, 400 consecutive ones are clustered together by applying a sliding window with 50% overlap. Mean and variance of all 50 features from the clustered frames are used in a Random Forest for classification. The length of a cluster is 10 seconds to capture both short and long sounds of interest at an adequate granularity.
For the study 20 participants wore a wrist mounted audio recorder between 4 and 7 hours on a single day, capturing at least one meal (lunch or dinner). The participants were students, scientists, designer and other professionals, to cover a multitude of environments for ambient sounds. They were allowed to review their captured audio and cut segments for privacy. To label ground truth for eating activities, participants had to remember their activities start and end time while going through the audio with a researcher. Then two authors independently reviewed these times. There was a strong class imbalance towards the non-eating activity class. A person dependent 10-fold cross validation achieved an F-score of 79.8% on the RF classifier, which is comparable to state of the art systems using body worn sensors. With a leave one participant out cross validation the F-score drops to 28.7% most likely due to the diverse ambient situations the participants were in during their meals. As the dataset mostly contained only one meal per participant the classifier didn't have enough similar meal context training data. Furthermore self reported eating activity ground truth depends on the participants memory and varies for correct start/end times recollection. Besides that

the different eating habits for participants made it difficult to precisely label eating activities (some meals lasted an hour due to long conversations). From all eating detection systems listed here it is the least physical intrusive one. But due to difficult to obtain ground truth and overall highly diverse ambient sounds of dining locations, performing worst for participant independent testing. With a know location the detection rate is high enough for actual use of the system, meaning more and better training data is necessary to capture most ambient dining sounds.

4.6 Eating detection with hand to mouth gestures

Ye et al. used head and wrist worn smart devices to detect eating from motions [17]. They used two different approaches depending on sensor location, both utilizing the built-in accelerometer. As head mounted device Google glass was used to detect chewing sessions from the head movement. A session starts when a bite of food was taken until the first swallow. As a session consists of bite, chew and swallow, they are repetitive and sequential which provides good features for classification. The sampling rate was set at 50Hz grouped to 4 second windows. The magnitude of the 3-axis accelerometer data was used, as the accelerometer alignment, therefore its orientation, is user specific. A Butterworth low pass filter with cut off frequency 5Hz was applied to remove noise outside the 1-2.5Hz chewing frequency range. From this 17 mostly statistical time and frequency domain features were calculated.
The wrist worn smartwatch was used to detect hand to mouth (HtM) gestures of the dominant hand. Hereby a gesture consists of ascending-, biting- and descending-period. As 30% of the acceleration data is only gravitational during HtM gestures, it is used to calculate the orientation of the hand in roll, pitch and yaw. From the orientation the HtM periods can be inferred (e.g. hand inside faces the body when biting). 12 orientation and 9 magnitude features are then extracted.
Evaluation data was gathered from 10 participants in two sessions. Each split into a 5 minute jogging/rest cycle, 10 minutes walking, followed by a 5 minute reading/rest cycle and eating a meal. The meal was made up of 4 dished from different softness food categories (soft solids easy/hard to chew, semi-solid, crunchy). It was consumed in front of an observer, who recorded start and end time of each HtM gesture and chewing cycle. SVMs were created with 10-fold cross validation and leave one participant out (in-person/cross-person validation). Head motions alone reached an accuracy of 95.1% in- and 89.5% cross-person. For HtM gestures in-person precision and recall greater 94% are reported, cross-person it drops slightly to 90%. Without orientation features only 72% respectively 59% are possible, this shows the importance of the orientation features. The combination of head motions and HtM reached a cross-person accuracy of 97%. Furthermore they use the combined classification result in a HMM to estimate the duration of the eating session, for which they achieve an error of ± 105 seconds.

They used the HtM gesture detection in a second field study [18], to test adherence for a semi automated food logging system. For a two week study period 7 participants were given a smartwatch running a food logging application. If the app would recognize HtM gestures it would show a prompt remembering the user to log their meal. They could decline, ignore or accept the prompt, in the last case a photo and note taking app would open for logging the meal.
As participants may have forgotten to log meals, actual ground truth is unknown and recall can't be calculated. Precision was 31%, calculated from total number of reminders, positive/negatives responses and the lack of a response (8% of prompts were ignored). As detection prompts can happen every 20 seconds, at 37% precision only 7 false detections per user per day occurred. Participants reported false positives as disturbing only during driving, other occurrences were e.g. yawing, washing he face. 3 participants reported FPs during driving and one while running. The low accuracy might stem from the fact that none of the participants where part of the trainings set in the first study and neither were activities like driving. Participants further reported delayed prompts a the end of their meal.
The correlation between reminder and note taking had a R-value of 0.76. For positive responses 0.91 and for negative responses 0.59. Correct detected meals are almost always followed by food logging and in some cases FPs remind the user to log an eating activity. Overall participants accepted the smartwatch as a commodity device and made heavy use of the logging app. With a successful field study the system has been proven useful at raising adherence and overall numbers of food logs.

4.7 Discussion

The results show that most of the presented systems are usable for automated eating detection, with suitable accuracy for meal logging applications. At the cost of different levels of intrusiveness and device requirements. The three stand-alone devices allow more in-depth monitoring compared to the systems using smart devices, which have the main goal of just eating detection. From these 3 the smart tablet is the least intrusive with no privacy issues, opposed to sound based approaches, but also its results are most difficult to evaluate. Hence the best application area for it would be stationary rehab, where it offers a low priced solution for automated eating habit monitoring. This frees up staff from evaluating patients manually. The two other devices don't offer the granularity and need too much setup time, making them less useful in this situation. The other systems can be applied to self evaluation and as part of a meal logging app, due to their physical non-invasive monitoring methods.

5 Exercise

For exercise purposes the use of smart devices has two major goals. One is to motivate the user to exercise and track their progress to a better health and

	Aim	Capture method	Sensors	Processing
4.1	Chew counting	In-ear sound	microphone	low pass filter
4.2	Eating related event recognition	Throat sounds	microphone	normalized features, Relief-F
4.3	Eating detection	Head movements	accelerometer, gyroscope, magnetometer	linear acceleration and rotation values
4.4	Eating habit monitoring	Smart tablet	pressure sensors/matrix	image processing methods, event spotting
4.5	Eating detection	Ambient sound	microphone (at wrist)	clustering, features mean and average
4.6	Eating detection	Head movements, HtM gestures	accelerometer (head and wrist)	low pass filter, hand orientation

	Features	Prediction	Results
4.1	Audio signal	Regression	90% Sensitivity, 2% fp rate
4.2	10 statistical (time/ frequency)	SVMs	unclear
4.3	5 statistical, 3 spectral, 2 temporal	Random forest	$F_{0.5}$-score 67.55% LOWO/ 49.7 LOPO
4.4	68 statistical	Decision trees	F1-score 87% (71.4%) user (in-)dependent
4.5	50 time and frequency	Random forest	F1-score 79.8% (28.7%) user(in-)dependent
4.6	17 statistical time and frequency, 12 orientation	SVMs	90% accuracy / precision / recall user independent

Table 3. Comparison of eating detection systems

fitness. For example by keeping score of their efforts or through gamification of the exercises. The other goal aims to optimize the exercise execution. A correct performed exercise raises training efficiency and prevents injuries from wrong training methods.

5.1 Unobtrusive gait assessment

Postolache et al. present two approaches [12] for gait assessment based on wireless sensor units. Their aim is to create low cost smart sensing devices to track the gait rehabilitation progress of patients undergoing physiotherapy. Traditionally movement is monitored in *gait laboratories* with camera systems, pressure sensitive floors and EMG to track muscle activity. Those set-ups aren't applicable for physiotherapists in their normal rehabilitation sessions. To evaluate gait patterns for therapeutic measures, they are interested in forces, acceleration and velocities acting on joints and muscles. They present two non invasive hardware set-ups to gather this data. One is a so called inertial measurement unit (IMU), a body worn network of smart wireless sensors. The second one is a smart walking aid with built in Doppler radar to monitor the gait of its user.
An IMU consists of a tri-axis gyroscope, an accelerometer and magnetometer as sensors. The IMU is connected to a micro controller which reads the sensor vales periodically, packs them and transmits the package wireless to a gateway coordinator. Such a compact end-node is attached to each foot to monitor them separately. The network coordinator is connected via USB to a computer, where the sensor data will be evaluated by an expert. It is also used to configure end-nodes remotely, without firmware updates. Their prototype uses the ZigBee IEEE 802.15.4 wireless protocol, but their aim is to use WiFi or bluetooth in the future to make the hardware gateway unnecessary.
The smart walker encapsulates a Doppler radar array, processing power and communication hardware to WiFi devices. An array of Doppler radar antennas is mounted on the walker, oriented at the patients legs. Using frequency modulated continuous wave (FMCW), a know continuous wave with in- or decreasing frequency is sent out from the transmission antenna. The receiving antenna picks up the signal reflected by the patient and it is demodulated into in-phase I and quadrature component Q. The resulting Doppler shift signal is proportional to the distance between sensor and patient. A short time Fourier transformation is applied to the in-phase components I1 and I2. The resulting frequency spectrograms are transmitted to a tablet running visualization software used by a physiotherapist for evaluation. The spectral analysis shows the differences between healthy and abnormal gait in power and frequency band. Using these systems basic gait analysis can be reduced in cost and effort, making it more accessible and raising the frequency of it being used in day-to-day practice.

5.2 Live walking feedback

Komninos et al. present their findings [7] on eyes free live feedback for walking exercise. For optimal exercise effect walking should be performed at an in-

creases pace to raise the metabolic rate. Target rates are expressed in multiples of Metabolic Equivalent of a resting Task (MET), for walking 3 or higher is advised. Their prototype test system aims at people wanting to exercise but fail at finding the right pace.
They implemented a music player app for smartphones, that reduces the audio quality if the users walking speed drops below a steps per minute threshold. Instead of using pure audio feedback to teach the correct walking pace, it allows the user to listen to their own music for added motivation, while the quality reduction is less intrusive than adding sound cues to the music. Their goal for the app was to provide fast, eyes-free feedback with minimal intrusiveness. to measure the walking speed, they utilized the accelerometer as step counter by applying a peak detection algorithm to the sensor data.
They've tested different effects to degrade the music quality, such that the reduction can be perceived on different music genres and headphones. The prototype uses equalizer effects the Android Media AudioFx SDK offers. Removing frequencies from high, mid and low frequency bands leaves only the main bass line and parts of the melody and was noticeable on all tested genres. But as the effect was only subtle at best, pink noise was added to emulated bad radio reception.
To evaluate the effectiveness of the feedback system, 20 participants performed 4 walks on 2 routes around a pedestrian only zone. Their hypotheses were whether maintaining correct pace is easier with the feedback player and will there be a learning effect for correct cadence. The shorter route (300meter) was used for two training walks. The first one to record the normal walking pace while listening to music of each participant. Which was used to calculate the individual MET rate for exercising, capped at 5 MET. During the second walk the participants listened to a metronome beat of their target pace and were tasked to remember and match their walking pace to the beat. For the two walks on the long route (800m) participants were split into two equal groups. Group P used a normal music player first, then on the second walk a player with degrading audio quality. The other group D started with the degrading player and used the normal player on the second walk. For both walks the metronome player the first 16 seconds as a reminder of the correct pace. After each walk participants completed a questionnaire to asses their perceived workload. Overall it took group P more time to complete the route with the normal player than with the degrading, the time of group D was almost the same in both walks. Looking at the steps taken, group P and D had the same amount with the normal player. Group P raised their step count with the degrading player, Ds count stayed the same. With the normal player both groups stayed below their MET rates for around 38% of their steps, group P more than D. With the audio feedback group P reduced that ratio from 41% to 15%. As group D started with the feedback, their rate rose slightly from 28% to 35% when walking without it. These results indicate that group D learned to some extend their correct pace and that the system helped group P to better achieve their correct pace, reassuring both hypotheses. The questionnaire results imply that participants felt no difference in their performance or frustration using either player. But they worked harder

and felt under time pressure using the degrading player and felt it was easier to walk at the right pace.

5.3 Muscle activation detection with accelerometers

MyoVibe [10] is a wearable sensor network to measure muscle activation from micro vibrations. Common techniques to track muscle activations like electromyography (EMG) suffer from motion induced noise, therefore aren't reliable in real-world settings to monitor heavy exercises. A reliable monitoring device is wanted to ensure proper muscle activations during training which increases sport performance and reduces injury risk.
MyoVibe uses several small accelerometer attached to the to be monitored muscle region. When the muscles contract small vibrations are produced by friction between the individual muscle fibres. These vibrations have low amplitude and are in the frequency band of 5-100Hz, the exact range depending on the particular muscle. The accelerometers are able to capture these vibrations and with some processing they are used to measure and identify muscle activity, so called mechanomyography (MMG). Due to the low amplitude, it is difficult to find them during heavy exercise because of high impact/motion noise. Lightweight 3-axis accelerometers are attached to the body either with flexible texture straps or sewn in form-fitting pants, ensuring consistent orientation of the sensors. Via a common data bus several sensors are connected to an aggregator node, which saves and transmits the sensor data to a back-end computing device for processing and evaluation.
To detect muscle activation, two steps for reduction of motion artefacts are applied to the raw sensor data. A high pass filter with cut off frequency 5Hz removes low frequency noise like gross body motion. To remove the remaining high amplitude, high frequency noise they propose the k-EVA algorithm. Using extreme value analysis (EVA) the distribution of extreme sensor values is modelled to selectively remove this high motion noise. Before each exercise 20 seconds low-motion (Mot_{low}) sensor data calibrates the distribution model. Extreme values in that time frame estimate the scale parameter $\delta_{Mot-Low}$ of the type I Gumble distribution. $k \cdot \delta$ is used as threshold to separate low and high motion distributions, effectively identifying muscle vibration and motion noise. Every data window exceeding this threshold is discarded. The parameter k has to be chosen cautiously as it balances the trade off between false alarm rate and recognition rate, by including more or less muscle activation data with motion noise.
Afterwards five frequency domain features are extracted with a DFT and 500ms sliding window. $\{25, 50, 75, 90\}^{th}$ frequency percentile to localize frequency shifts into specific bands. Area under curve of the cumulative distribution function (CDF), which shows how frequencies are distributed in a given window. A decision tree is used to classify whether a muscle is active or inactive for a given accelerometer feature matrix ($N \times 15$, N the amount of used accelerometers on the muscle). The DT is trained on labelled feature matrices, labels are obtained

from a time synced state of the art sEMG system and the help of an sEMG-expert who manually preprocessed the ground truth data.
6 *generally fit* participants were fitted with MyoVibe and sEMG sensors on their legs. They performed 4 two minute exercises from the categories isometric, repetitive motion and high mobility/impact. Isometric leg extension to measure MA without motion noise. Squats to evaluate performance during full body free form exercise. Cycling as repetitive, motion intensive exercise and jumping to test resilience against high impact noise. The ground truth was generated from the sEMG signal, by first manually removing areas of high motion artefacts. Then an algorithm to detect muscle activation (here AGLR) was applied. For isometric exercises the HP alone achieved recall, precision and accuracy above 97%, which k-EVA didn't improve. Repetitive motion exercises benefit greatly from k-EVA. For both squats and cycling k-EVA increased precision by 20% to above 90%. High impact exercises also improved precision, recall and accuracy from the 47-82% range to 75-88%. The results are promising that the system can be used to improve training performance on an almost daily basis. With the network sewn into form-fitting pants it is easy and fast to correctly put on, which allows to perform monitored session more often due to the reduced effort.

5.4 Discussion

The three presented approaches are just a small portion of feasible exercise support systems. But they show a glimpse at the possibilities smart devices offer in this area of health monitoring.

	Aim	Capture method	Sensors	Processing
5.1	unobtrusive gait assessment	IMUs, Doppler radar	accelerometer, gyroscope, magnetometer, radar array	-
5.2	live walking feedback	Step counter	accelerometer	peak counting
5.3	muscle activation detection	MMG	accelerometer	high pass filter, k-EVA

	Features	Prediction	Results
5.1	raw sensor data	manual by expert	-
5.2	steps per minute	threshold	learning effect noticeable
5.3	5 frequency	decision tree	precision/ recall/ accuracy $> 75\%$

Table 4. Comparison of estimations.

6 Conclusion

The presented techniques showed that state of the art smart devices can accurately track vital signs and the results are accurate enough to make the similar assessments as from clinical devices. Furthermore they can be used to reduce health risks or utilize them in context sensitive approaches, like recognizing stress and change workload accordingly.
In chapter 2 systems to track lung functionality and respiration symptoms were presented. Used for automated unintrusive assessment they lower the risk of unnoticed decline in health. Monitoring heart rate is examined in chapter 3, explaining different sensing modalities like Ballistocardiography that can be realised with smart devices. Chapter 4 covers methods for semi-automated food-logging and eating behaviour analysis, like eating detection and chew counting. Last but not least a small selection of exercise support systems are discussed in 5. Representing the methods for live exercise feedback and in-depth exercise evaluation.
As the results from ambient sounds eating detection and food logging in the wild have shown, some conducted studies could benefit from bigger sample sizes especially when machine learning is used. The logical next step would be to develop applications that make use of these monitoring and supporting abilities to raise the overall health and lower its cost.

References

1. Bi, Y., Xu, W., Guan, N., Wei, Y., Yi, W.: Pervasive eating habits monitoring and recognition through a wearable acoustic sensor. In: Proceedings of the 8th International Conference on Pervasive Computing Technologies for Healthcare. pp. 174–177. PervasiveHealth '14, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Belgium (2014), http://dx.doi.org/10.4108/icst.pervasivehealth.2014.255423
2. Fan, X., Wang, J.: Bayesheart: A probabilistic approach for robust, low-latency heart rate monitoring on camera phones. In: Proceedings of the 20th International Conference on Intelligent User Interfaces. pp. 405–416. IUI '15, ACM, New York, NY, USA (2015), http://doi.acm.org/10.1145/2678025.2701364
3. Grimaldi, D., Kurylyak, Y., Lamonaca, F., Nastro, A.: Photoplethysmography detection by smartphone's videocamera. In: Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), 2011 IEEE 6th International Conference on. vol. 1, pp. 488–491 (Sept 2011)
4. Haescher, M., Matthies, D.J.C., Trimpop, J., Urban, B.: A study on measuring heart- and respiration-rate via wrist-worn accelerometer-based seismocardiography (scg) in comparison to commonly applied technologies. In: Proceedings of the 2Nd International Workshop on Sensor-based Activity Recognition and Interaction. pp. 2:1–2:6. WOAR '15, ACM, New York, NY, USA (2015), http://doi.acm.org/10.1145/2790044.2790054
5. Haescher, M., Matthies, D.J., Trimpop, J., Urban, B.: Seismotracker: Upgrade any smart wearable to enable a sensing of heart rate, respiration rate, and microvibrations. In: Proceedings of the 2016 CHI Conference Extended Abstracts on Human

Factors in Computing Systems. pp. 2209–2216. CHI EA '16, ACM, New York, NY, USA (2016), http://doi.acm.org/10.1145/2851581.2892279
6. Hernandez, J., McDuff, D., Picard, R.W.: Biowatch: Estimation of heart and breathing rates from wrist motions. In: Proceedings of the 9th International Conference on Pervasive Computing Technologies for Healthcare. pp. 169–176. PervasiveHealth '15, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Belgium (2015), http://dl.acm.org/citation.cfm?id=2826165.2826190
7. Komninos, A., Dunlop, M.D., Rowe, D., Hewitt, A., Coull, S.: Using degraded music quality to encourage a health improving walking pace: Beatclearwalker. In: Proceedings of the 9th International Conference on Pervasive Computing Technologies for Healthcare. pp. 57–64. PervasiveHealth '15, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Belgium (2015), http://dl.acm.org/citation.cfm?id=2826165.2826174
8. Larson, E.C., Goel, M., Boriello, G., Heltshe, S., Rosenfeld, M., Patel, S.N.: Spirosmart: Using a microphone to measure lung function on a mobile phone. In: Proceedings of the 2012 ACM Conference on Ubiquitous Computing. pp. 280–289. UbiComp '12, ACM, New York, NY, USA (2012), http://doi.acm.org/10.1145/2370216.2370261
9. Larson, E.C., Lee, T., Liu, S., Rosenfeld, M., Patel, S.N.: Accurate and privacy preserving cough sensing using a low-cost microphone. In: Proceedings of the 13th International Conference on Ubiquitous Computing. pp. 375–384. UbiComp '11, ACM, New York, NY, USA (2011), http://doi.acm.org/10.1145/2030112.2030163
10. Mokaya, F., Lucas, R., Noh, H.Y., Zhang, P.: Myovibe: Vibration based wearable muscle activation detection in high mobility exercises. In: Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing. pp. 27–38. UbiComp '15, ACM, New York, NY, USA (2015), http://doi.acm.org/10.1145/2750858.2804258
11. Nishimura, J., Kuroda, T.: Eating habits monitoring using wireless wearable in-ear microphone. In: Wireless Pervasive Computing, 2008. ISWPC 2008. 3rd International Symposium on. pp. 130–132 (May 2008)
12. Postolache, O., Ribeiro, M., Girão, P.S., Dias Pereira, J.M., Postolache, G.: Unobtrusive sensing for gait rehabilitation assessment. In: Proceedings of the 8th International Conference on Pervasive Computing Technologies for Healthcare. pp. 386–389. PervasiveHealth '14, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Belgium (2014), http://dx.doi.org/10.4108/icst.pervasivehealth.2014.255364
13. Rahman, S.A., Merck, C., Huang, Y., Kleinberg, S.: Unintrusive eating recognition using google glass. In: Proceedings of the 9th International Conference on Pervasive Computing Technologies for Healthcare. pp. 108–111. PervasiveHealth '15, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Belgium (2015), http://dl.acm.org/citation.cfm?id=2826165.2826181
14. Sumida, M., Mizumoto, T., Yasumoto, K.: Estimating heart rate variation during walking with smartphone. In: Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing. pp. 245–254. UbiComp '13, ACM, New York, NY, USA (2013), http://doi.acm.org/10.1145/2493432.2493491

15. Sun, X., Lu, Z., Hu, W., Cao, G.: Symdetector: Detecting sound-related respiratory symptoms using smartphones. In: Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing. pp. 97–108. UbiComp '15, ACM, New York, NY, USA (2015), http://doi.acm.org/10.1145/2750858.2805826
16. Thomaz, E., Zhang, C., Essa, I., Abowd, G.D.: Inferring meal eating activities in real world settings from ambient sounds: A feasibility study. In: Proceedings of the 20th International Conference on Intelligent User Interfaces. pp. 427–431. IUI '15, ACM, New York, NY, USA (2015), http://doi.acm.org/10.1145/2678025.2701405
17. Ye, X., Chen, G., Cao, Y.: Automatic eating detection using head-mount and wrist-worn accelerometers. In: 2015 17th International Conference on E-health Networking, Application Services (HealthCom). pp. 578–581 (Oct 2015)
18. Ye, X., Chen, G., Gao, Y., Wang, H., Cao, Y.: Assisting food journaling with automatic eating detection. In: Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems. pp. 3255–3262. CHI EA '16, ACM, New York, NY, USA (2016), http://doi.acm.org/10.1145/2851581.2892426
19. Zhou, B., Cheng, J., Sundholm, M., Reiss, A., Huang, W., Amft, O., Lukowicz, P.: Smart table surface: A novel approach to pervasive dining monitoring. In: Pervasive Computing and Communications (PerCom), 2015 IEEE International Conference on. pp. 155–162 (March 2015)

Health Monitoring based on Smart Devices

Niklas Sänger*
Advisor: Long Wang†

Karlsruhe Institute of Technology (KIT)
Pervasive Computing Systems – TECO
*niklas.saenger@student.kit.edu
†wanglong@teco.edu

Abstract. With gaining popularity of smart devices and the increased amount of sensors, there arise new possibilies to help the user to monitor his health using a smart device. This research describes the approaches and their performance on exercise monitoring, heartrate monitoring and lung function monitoring. It will also give an overview of basic sensors and algorithms which are used for analyzing and classifying the input data. Exercise monitoring is able to detect the users repetitions, execution and the exercise itself. It can deliver the user feedback in order to improve his performance. Heartrate monitoring detects the heartrate utilizing sensors provided by a smart device. Lung function monitoring is able detect features which are used to assess a patients lung function.

Keywords: Health Monitoring, Smart Devices, Exercise Monitoring

1 Introduction

With more than 1.4 billion Smartphones sold worldwide and smart devices like smart watches or smart wristbands becomming more popular, there is a large potential for new applications usings sensors provided by smart devices[9]. Even entry level smart devices inlcude a basic amount of sensors, this is especially important since they are often sold in emerging countries with the strongest growing smartphone markets of up to 18% growth in sales. (Hier fehlt die Quelle)

Another important business sector is the eFitness market. It is a strong growing market creating a revenue of up to $3.495B in 2015 [8] and is expected to grow further over the next years. This demonstrates how interested users are in tracking their sports exercises with their smart devices. Smart devices can not only bring a benefit to users by tracking their activities, but also by analyzing their performance and give them feedback right on the spot. This can lead to an improved exercise performance creating a higher training effect. Hence the user's motivation rises and potentially results in more frequent exercise sessions.

The above mentiond factors can also raise the awareness of the user's health situation and improve it. Especially in emerging countries with low medical supply and greate distances to the next hospital. Smart devices can not just deliver a first diagnosis (e.g. detecting the lung function of a user[4]) but also

give the user exercises to treat his condition (e.g. [5]). This does not only save costs and effort, but gives people who can not afford meddical care a slight insight on their wellbeing and even a way to treat it. Additionally, smart devices can automatically send information gatherd to the users doctor for review, which saves time and costs. Since medical care using smart devices is in the early stages, it is far from being perfect. Results are not as reliable as professional machines because the user has to execute the task without any professional advice and can thereby create falsified data.

This research has been done to give the reader an insight on exercise monitoring and medical healthcare based on smart devices. For a better understanding, the following chapter will descirbe the fundamental terms and definitions and the basic data analysis methods used in the considered papers. The chapter *Monitoring* will consider the different methods to gather data and compare their results with each other. At last the reader will receive a short overview and conclusion about exercise monitoring using smart devices.

2 Fundamentals

This chapter will define the basic terms and common data analyzing algorithms used in this research.

2.1 Sensors

Smart devices, especially smartphones, have gained sensors over the past years. In comparision to the current iPhone 6s using 8 Sensors, the first generation iPhone used only 3 different types of sensors, excluding camera and microphone. Sensors have also become more accurate not only by improving the technologie but also by combining them (e.g. GPS).

Accelerometer An Accelerometer is a mechanical device that will meassure acceleration forces. The force meassured by an accelerometer can either be static or dynamic. If the force is static the device is not moving. Otherwise dynamic forces are caused by vibration or movement. By analyzing the amount of static acceleration the angle of the accelerometer with respect to the earth can be calculated. Analyzing dynamic acceleration results in information about the movement of the device.

Camera Cameras used in smart devices are digital. Light falling through their lenses are filtered by a mosaic filter. This creates red, green or blue signals which are then captured by an electronic pickup device and saved to memory.

GPS GPS is a synonym for Global Positioning System. It is a naviagtion system providing location and time information based on satelite positioning. In order to calculate the position of the GPS sensor, at least four satelites must be visible

from the sensors location. The more satelites are visible, the higher the accuracy. The position is calculated by processing the signals received from the satelites containing the exact time and position.

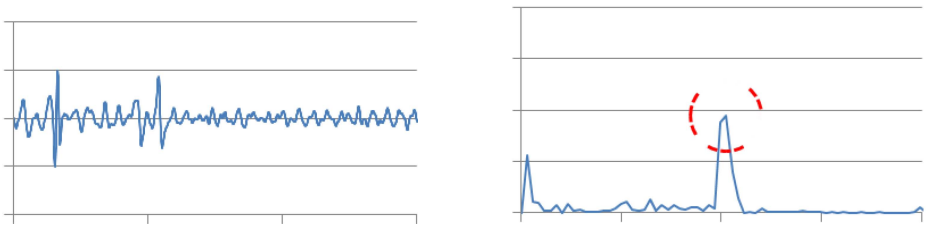
Magnetometer A magnetometer can calculate the direction of the magnetic north. It uses a hall-efect sensor to detect the Earth's magnetic field and converts it into a digital signal.

Microphone A microphone is a sensor that converts an acoustic signal into an electronic signal. Most smart devices use MEMS(MicroElectrical-Mechanical System) microphones containing a pressure-sensitive diaphragm.

Pressure Sensing Matrix A pressure sensing matrix is able to detect the position and amount of force effecting the matrix. It consists of two electrically conductive sheets which create a digital signal when force is applied.

2.2 Common Algorithms

This chapter describes some common algorithms used by the papers in this research.



(a) Before FFT, x-Axis = Time, y-Axis = Intensity

(b) After FFT, x-Axis = Frequency, y-Axis = Power

Fig. 1: Input Data before and after a FFT

Source: [2]

FFT The *Fast Fourier Transformation* (FFT) transforms input data from its original domain into a frequency domain and vice versa. For this research, the original domain is mostly time releated. Figure 1 shows the difference before and after a FFT. 1a used time and intensity as the domain, applying a FFT the domain changes to frequency and power. Therby different types of information can be gathered from the same set of data. In this example the red circle in 1b describes a pulsation peak frequency.

Peak Detection Algorithms Peek detection algorithms try to find peaks and remove noise in a given set of data. A peak is larger than both elements next to it. In Fig 1b the red circle marks one of many peeks. Another part of peak detection algorithms is meassuring the distance between two peeks. For example, the amount of peaks can equal the amount of repetions of an exercise. The same can be done with the distance between peaks, those might equal the time between repetitions.

Filter A filter is applied to a set of data and removes elements which are not fulfilling the criteria used. There are several filters used by the papers in this research. The *Savitzky-Golay filter*, is a filter used to smooth data points. This can be used to smoothen data gathered from a gps sensor by removing noise created during location calculation. A *low pass filter* removes elements which are below a certain limit, this can be used vice versa or in other different variations.

3 Monitoring

This chapter will consider the different types of monitoring and compare them with each other. It will look into the techniques used to gather information and how they are evalutaed. An important aspect for the evaluation is the ground truth,it has to be manually protocolled and evaluated. The comparison between the ground truth and the results gained by the teqchniques described, determines the accuray and the success. In the following the nine papers used in this researched are categorized into three different groups, exercise monitoring, heartrate monitoring and lung function diagnosing.

3.1 Exercise Monitoring

This section will regard exercise monitoring for free weight lifting or resistance training and running rythm monitoring. In order to monitor free weights or resistance training, there are three different aspects which can be monitored. The ground truth and testing environment can be found in Table 3.

- *Repetition Detection* to detect valid repetitions
- *Exercise Execution* to detect the execution of an exercise and give feedback to the user.
- *Exercise Recognition* to detect the exercise and process the incomming data correct.

Beginning with exercise recognition, Pernek et al. [6] use a smartphone and its sensors to detect repetitions on different exercises. Their work should provide a simple user interface that can suggest training exercises and give feedback during the exercise [6]. They use resistance exercises that can be done in an closed environment like a gym (e.g. constraind exercises) or in an outdoor environment

Table 1: Exercises Monitorung : Ground Truth

Paper	Subects	Age	Exercises	Sets	Repetitions	Comment
[7]	11	60-70, ±2.5	26 6	3	10	7 COPD patiens and 4 young health subjects
[6]	10	10-15	6	5	10	–
[1]	13	20-30	1	39	–	Only running; 39 runs
[11]	7	23-28	10	2	10	first set fast exectued, second set more relaxed

with the own body weight or resistance bands (e.g. unconstraind exercises). These exercises train the legs, arms and the lower or upper body. When using lifting machines, the smartphone is attached or put on top of the weights, otherwhise they placed the smartphone to the wrist or an ankle. Pernek et al. [6] only used the accelerometer of the smartphone for their work and detected similar data when performing the exercises in with or without machines.

In order to compare the input data, repetition patterns were used which where either prerecorded and downloaded from an external source or calibrated once for every user[6]. These patterns then where evaluated by using dynamic time warping (DTW), which can compare sequences temporarly not aligned, by producing a mapping with minimized distance between the input sequences[6]. Since DTW is intensive to process they used various procedures to recude workload. First, the sampling frequency is reduced to one sample every 100ms (10Hz). Then a Savitzky-Golay filter is applied to smoothen the data stream. At last, the major direction of movment is detected and peaks selected which are within $1/3$ of the maxium peak of the compared pattern. For the classification a set of eight different features is calculated (DTW distance, minimum, maximum, arithmetic mean, standard deviation, root mean square and duration) and used for a logistic regression with coefficients determined by machine learning. A sequence can either be accepted or rejected.

For evaluation of the algorithm, a total of 3598 samples was collected by 10 subjects (see table 3). The ground truth was established by using voice annotations which where utilized to assess the results of the algorithm. A F-score metric, calculated as the harmonic mean of precision and recall was used to describe the effectiveness of the classification[6]. The results for both pattern generation methods were similar. The average F-score for the prerecorded pattern was 0.993 ± 0.034 with an error rate of less then 1%. The error rate was slightly higher in unconstrained environments, but the overall rate is not significant. This leads to the conclusion that repetition detection can be done accuratly for a limited amount of constraind and unconstrained exercises. Since there ex-

ist a lot of similar exercises, the approach should be able to work for a greater variety of exercises with only small modifications.

For exercise execution, Spina et al. [7] describe an application called COPDTrainer in their paper. This application is developed for COPD (chronic obstructive pulmonary disease) patients to help them execute exercises and give them feedback in a non-clinical environment. It should be used in coordination with a therapist. The feedback consists of information about exercise speed and correct execution (e.g. move the arm higher/lower). They attach a smartphone with a holster to the wrist for upper body exercises and to an ankle for lower body exercises. The application has a teach and a train mode. During the teach mode, any exercise is monitored under a supervising therapist with a specified amount of repetitions. Thus no audio feedback is provided. The train mode is unsupervised but specifies an amount of repetitions for the user and gives him audio feedback. The phone collects data using an accelerometer, a magnetometer and the Android orientation API.

After the train mode, COPDTrainer processes the gathered data to receive model parameters. At first a moving average filter is applied, using the window size proportional to the amount of data[7]. Then the exercise repetitions were counted by estimating the position of positive and negative peaks using a hill climbing algorithm with a peak threshold. A repetition was counted if both sides (uphill and downhill from the minimum/maximum) were greater then the peak threshold. Since the peaks can be lost during the peak detection, a peak correction algorithm was applied to add missing peaks or to remove addtional peaks. It works, by first removing redundant peaks and then insert peaks that were missed during the first iteration [7]. After the analysis, five parameters (number of repetitions, mean and standard deviation of repetition duration, mean and standard deviation of range of motion) were estimated. In trainmode, the incomming data stream was compared to the parameters estimated in teach mode. A sliding window was used to cover two average repetitions with an overlap of 75% [7]. The hill-climbing and peak correction algorithm was again applied like in the teach mode. To give the user in-time feedback, the first half of the repetition was evaluated for feedback. The difference between the first half and the total repetition length was insignificant. Repetitions can be classified into nine categories including too fast, too slow, too small, too large, the combination from the previous four and correct. By this classification, a simple feedback instruction can be given.

For the validation of the system, Spina et al. [7] used four healthy subjects and eleven COPD patients performing six exercises a five sets with ten repetitions each (see table 3). Every test subject did one exercise supervised in teach mode and afterwards unsupervised in train mode. A therapist judged each repetition during train mode according to the defined categories for comparision. The ground truth was established by manually categorizing the raw sensor data. Among healthy subjects, the average classification accuracy is 96.7%. The lowest accuracy is in the categorie *too slow and too small* with 92% which is

mostly generated by *Leg Lifts* with an average accuracy of 77.5%. In order to determine the accuracy for COPD patients, six sets of different exercises have been removed because of recording issues. Another exercise repetition had to be removed because the subject felt pain, leaving 1176 exercise repetitions. The overall average counting accuracy was 96.7% and average classification accuracy was 87.5%. In 119 cases the audio feedback caused a change, in 79 cases the feedback was ignored and in only 8 cases the feedback caused no change. This leads to the conclusion that COPDTrainer can improve the exercise execution and provide an accurate system to perform a special set of exercises at home after a supervised instruction.

Sundholm et al. [11] describe a pressure sensing matrix in their paper. This sensing matrix, called *Smartmat*, is utilized for counting repetitions and detecting different exercises. This matrix can be implemented easily in different objects (e.g. gym mat). They focus on exercises which can be performed anywhere with free weights or the bodyweight, since there is no practical application with exercise machines. For their work, they only used the pressure sensing matrix.

Before analyzing the incomming data, they preprocess the data stream into frames. In those frames, DC (direct current) and calibration noise are removed with a median filter and a compensation term. In order to identify the activity and count the repetitions, several features are needed. Therefore the area, the weight, the pressure and *Hu's Seven Moments* were calculated as the feature set for each frame. Hu's seven moments are invariants with respect to translation, scale and rotation. The area is the sum of all pixel above a treshold. Weight is the sum of all pixels within an area. Finally pressure is weight divieded by area. Hu's seven moments are used to describe the shape of the contact area. To classify the exercise categorie of a frame, mean, standard deviation, maximum, minimum, maximum - minimum, were determined. Then a k-nearest-neighbour classifier was applied to the feature set to determine an exercise. To evaluate the incomming data stream, templates are needed for each exercise. Sundholm et al. [11] calculate templates for each exercise by recording different instances of an exercise, up and downsample it, to create equal length and calculate the mean between all instances. When an exercise is recorded, it might be time shifted to allign better with the exercise template. The recording is shifted to the position where the cross-correlation between the recording and the template is maximized. Recordings are also weighted in respect to their alignment with the different templates. To compare the incoming data stream with the created templates, DTW is used to determine the Euclidian distance norm for all ten dimensions. The exercise template with the smallest euclidian distance is selected. Exercises are counted when they are above a static threshold by using a peak detection algorithm.

Sundholm et al. [11] evaluated two different performances. First they used seven subjects, performing ten predefined exercises with two sets a ten repetitions each, to evaluate the classification (see table 3. When they used the first set of each subject for training and the second for testing, they reached an average

of 88.7% for all exercises. Then they used one subject for testing and six subjects for training and reached an average accuracy of 82.5%, which showed that the system works with different users. The slightly worse performance for multiple users is connected to the different execution styles for each user. For repetition detection, [11] also used a F1-score. This revealed that biceps curl has the lowest recognition with 66% on average while lunges have 100% average detection. The low recognition rate of biceps curls can be explained through the subtle change in pressure since the exercise is performed standing. On the oterside, lunges have only a low variation in execution. The average F1 score for all exercises is 82.8%. In order to achieve better results for many user types, a better template for each user is needed. In conclusion, Smartmat can detect changes is pressure and thereby identify exercises and repetitions. Although some exercises (e.g. biceps curl) were hard to determine, the average rate of identification makes it usable for day to day use.

Another approach to monitor exercise execution is Runbuddy. Hao et al. [1] try to improve a users running rythm by giving him usefull feedback (e.g. run faster/slower, play a special song, etc.) and thereby increase his motivation. RunBuddy measures running rythm using a physiological metric called Locomotor Respiratory Coupling (LRC)[1]. It can be calculated as strides within each breath cycle and therefore Runbuddy needs to access a smartphones accelerometer and microphone. A humans LRC is two strides for each breath. To protect privacy and provide real time feedback, the incomming data has to be processed in real time and should not be stored.

In order to detect breaths, the large incomming data from the microphone with a samplerate of 16kHz is filterd through a low pass filter because the sound frequency of a breath is in the range of 500 to 3500 Hz. Then the accustic signal is framed using a moving window which contains 40ms of acoustic signal with 30ms overlapp. A feature set of the first seven mel-frequency cepstral coefficants are used for detection. At first use, training data has to be gathered in a quite environment to ensure breathing can be captured with little noise. For each frame, the likelihood of containing a breath will be determined in comparison to the training data by calculating the cosine similarity. The cosine similarity reaches from -1 to 1 representing the minimum and the maximum degree of similarity. Stride detection utilizes the incomming datastream from the accelerometer. Hao et al. [1] project the incomming data onto a global coordinate system. A peak detection algorithm with a short time window is applied to find peaks in vertical acceleraction. Since one stride contains one left step and one right step, a stride has two peaks. The degree of correlation between breath and stride is calculated by using generated LRC ratios and the detected strides to provide a more reliable indicator of rythm changes. It is needed since breath detection is influenced by envrionmental noise which might have similar features to the users breath. The LRC ratio with the highest degree of correlation is chosen to represent the LRC ratio. Since runners tend to switch their running rythm more frequently, a 10-second moving window was used.

For evaluation 13 subjects completed 39 runs which were about 526.1 minutes. They used different Bluetooth headset models to investigate the difference of breath detection. The different models performed similar. For the ground truth, an inline mic was used to capture better recordings. Then a threadhold-based peak detection algorithm was applied. This was used in 34 of 39 runs. For the rest, the ground truth was established by manually labeling the recording gathered by the bluetooth mic. Runners were classified into *Non-runner, Occasional runner and Regular runner*. Test subjects were allowed to wear the phone whereever they liked. The detected stride accuracy for all subjects was over 99%. The overall correct detection rate for indoor runs was 93.3% with the lowest of 87.3% and the highest of 95.2%. The indoor testing resulted in different trends. Female subjects tend to worse detection run through their lower breathing intensity. Also, subjects, who are non-runners are below the average because of their inconsistent breathing. Indoor running has a slightly higher average accuracy of 93.5% in comparision to outdoor running with 91.4%. Even though the best result was achieved outdoors in a quite neighborhood with 95.2% because there was only little ambient sound and no operating treadmills. In conclusion, Hao et al. [1] created a system that can robustly detect exercise execution and help the user to improve his performance by giving him valuable feedback.

The presented papers all leveraged a smartphone and its very basic sensors in order to monitor exercise recognition, execution and repetition. The first three papers all detected repetitions with similar techniques, the difference is what happend after recognizing repetitions. While Pernek et al. [6] only recongnized repetitions, Sundholm et al. [11] also detected the exercise and Spina et al. [7] monitored the exercise execution. It is worth to mention, that all papers approached their problem in the same way. At first they filtered the incomming data stream to a level they were comfortable with. Then they searched for features that they could use in order to classify the data and applied different algorithms. Amongst others, all of them used peak detection algorithms with time-windows and filters. In conclusion, they all used the same type smart device and they all used similar techniques for evaluation but the results and the applications were different.

3.2 Heartrate Monitoring

Heart rate is the speed of the heartbeat. It is measured in beats per minute (bpm). The normal resting heartrate is inbetween 60-100bpm, although it depends on various factors like age, gender, weight, etc. Hearbeats lead to a slight increase of the chest. This section will regard estimating the heart rate without the help of traditional devices. In order to estimate the heart rate, GPS, an accelerometer and a camera are used. Three different approaches will be intruduced. The ground truth and testing environment can be found in Table 2

- *During Walking* by utilizing incomming data and utilizing the oxygen uptake and the physical load

– *Infront of a Camera* by determining properties of the skin

Table 2: Heartrate Monitoring : Ground Truth

Paper	Subects	Activity	Ground Truth	Comment
[10]	5	each subject walk on 5 routes	wrist wirless heart monitor	subjects divided in 3 groups
[2]	10	perform cyclic motion infront of camera	belt band heart rate monitor	–
[3]	43	–	ECG measurements	measurements performed by professinal

Sumida et al. [10] describe a system that can estimate heart rate vartitions during walking. They only use an accelerometer and GPS sensors. In order to calculate the heartrate, they estimate the physical load, the oxygen uptake and the gradient since changes in those factors generate a change in heartrate. To calculate the heart rate, they focus on three requirements. Frist they measure the physical load. Second they estimate the temopral changes of physical load. Lastly they need to adapt to difference depending on the user[10].

As feature set, they used the input data of the accelerometer (X-, Y-, Z-axes and the composition of all axes), oxygen uptake, walking speed and the gradient. Acceleration data is first filtered using a FFT to remove DC compontents and then split into fixed sized windows which contain the average amplitude. Walking speed is calculated by using dead reckoning which utilizes the accelerometers and gyro sensor. It is walking distance divided by walking speed. The distances between two points are selected in a fixed time window. The gradient is the difference in elevation between two points detected by the gps sensors. The two points are selected in a fixed time window. The oxygen demand is calculated as the sum of

– *Uptake at rest:* set to 3.5,
– *Horizontal component of walking speed:* speed and
– *Vertical component of walking speed:* difference of elevation

in a fixed time window because walking speed and the gradient can change unexpectedly. Then the trend of oxygen demand is determined in order to calculate the oxygen uptake. . The feature set is then used as input layer for a multilayered neuronal network which has an input layer, a middle layer and an output layer (see Figure 2). The network uses supervised learning and therby learns the correlation by using training data. The computation is not done on the smartphone, the user sends the data needed to a server which calculates

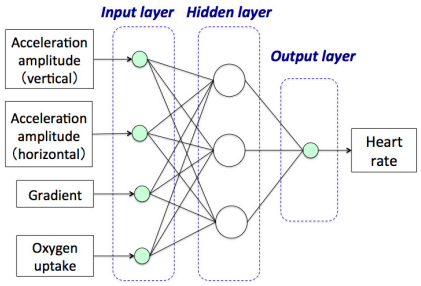


Fig. 2: Neuronal Network with given input Parameters outputting the Heartrate

Source: [10]

the heartrate and sends it back to the phone. To adapt better to individual differences they assumed that users who have similar profiles change similarly. Therefore they introduced new parameters to represent the reaction rate in increasing and decresing periods of oxygen uptake and coefficients to represent individual differences in oxygen uptake for walking speed and gradient. They also normalize the heartrate data for better learning. Since the normalized hearrate reflects a change pattern, it must be converted to the heartrate by using the initial heartrate[10].

In order to evaluate their system Sumida et al. [10] used a wristwatch wirless heart rate monitor to establish the ground truth. In total 18 subjects collected data on five different routes by mounting the smartphone on the body. The routes had similar lengths (form 1000m to 1700m) but varied in altitude. The sampling period was set to 20 ms for the accelerometer, two seconds for heart rate and three seconds for the gps. The total window size was set to 25s. The users were categorized into three groups reaching from users who have taken 30 min or more exercises a week to users who have taken almost none exercise a week. To define the accuracy of the heart rate they used the *Mean Absolute Error* between the actual heart rate and the estimated one which is meassured in bpm. They also differentiate wheter the model was trained with or without data in respect to the user and the route. If the user and the route were not trained, the average error was the worst with 6.78% while the best result was achieved when the model was trained with the route and the user which achieved a result of only 6.41%. When the model was trained with the user but not with the route it achieved a result of 6.49% and vice versa 6.68%. This shows that the information about the route leads to a better result than only the user information. The difference between the best and the worst result were only 0.37 which shows that a good result can be delivered even without any data for training. For some subjects, the error is a bit larger, which might be introduced by a GPS and inaccurate information of altitude which leads to an error in estimation of oxygen uptake. In conclusion, Sumida et al. [10] describe a model which can estimate the heartrate of a user by attaching a smartphone to his body. They

do this by utilizing various sensors which create input data for their neuronal network which estimates the heartrate with an average error of 6.78 to 6.49 bpm.

Another approach to estimate the heartrate was done by Lagido et al. [3]. They use a smartphone camera and an accelerometer to determine the users heartrate. Thereby a finger has to be put on the camera lense with the flashlight turned on. They try to obtain a Photoplethysmogram signal (PPG), which is an optical signal that meassures changes within an organ. In this case, changes in light absorption over time need to be detected because each cardiac cycle pumps blood through the body resulting in a peak in the PPG.

In order to meassure the PPG, a frame equals an image. The PPG is measured by summing up each pixel of a frame where the red component of a pixelis higher than a treshold. The treshold is established by calculating the mean of the 90% percentile of the red color over five seconds. If the treshold is too high, signals might be lost and if it is too low there might be false peaks. Therefore, the treshold is continously validated by saving the ratio between treshold and the average of red pixels for every frame in a time window. Then the mean value of those ratios must be within a fixed range. To check if the finger is correctly aligned, a simple analysis of a frame is done. When a finger is on the camera sensor, the amout of colors for each pixel, ranging from 0 to 255, must be inbetween a certain range. Otherwhise there might be colors, which are not similar to the human skin (e.g. blue, yellow). Sudden movements can distort the monitoring. Therefore the accelerometer is used to detect movements by comparing the detected value to a pre defined maximum and to stop the measurement if the maximum is reached. To detect the heartrate, the PPG signals are filtered with a rectangular smoothing function before applying a peak detection algorithm with a window size of five seconds. To detect heart rate variability, five criteria were extracted and interpreted.

For evaluation purposes, 43 heart failure patients performed heart rate measurement over a period of 60 seconds. Simultaneously a ECG measurement was taken in a hospital to establish the ground truth. The average error rate among the patients was 4.75% with an average standard deviation of 2.61 beats per minute. Lagido et al. [3] describe relaible method to detect the heart rate by only applying the finger onto a camera lense.

Kitajima et al. [2] also try to estimate the heartrate by utilizing a camera. However, they do not need the user to put his finger onto a camera. Instead they can estimate the heartrate from a distance of up to six meters by analyzing the face of a user. They focus on the blood oxygenated hemoglobin which detects the intensity of green light. The intensity changes within each cardiac cycle which creates detectable peaks.

Before the intensity of green light can be calculated, the face of the user needs to be detected. When the face is detected, an area with small amount of motion (the area between eyes and mouth) is selected to calculate the green light. To detect motion, the change in brightness is detected between consecutive images.

If there was a large amount of movement, the difference in brightness will be high and vice versa. They applied a moving average filter to the sections with high amount of motion and subtracted the average green light intensity. This resulted in the variation corresponding to the heartrate (See Figure 1a). Then a FFT was applied where the peak frequency multiplied with 60 equals the heart rate in beats per minute (Figure 1b).

To evaluate the heartrate, they had ten subjects which performed a cyclic motion infront of a camera. The distance between the camera and the subject was set between two meters up to six meters. The light intensity also varied from 30 lux up to 2000lux. To establish the ground truh, a belt band heart rate monitor was used. In a normally lit room (∼1000lux), the average error upto four meter distance to the camera was two bpm. Between four and six meters, the error increased to three bpm. When the luminance in the room was less then 500 lux, the average error rate was arround four bpm. In between 500 and 1000 lux the average error was arround three bpm and beyond that it was only two bpm. In case of lateral motion the average error was within three bpm while vertical motion had an average error of five bpm which is connected to the greater effects in lighting.

The presented papers measure the heart rate with different approaches. The last one is performing the best with an average error of only two to three bpm even though it uses a similar technuiqe as presented by Lagido et al. [3]. Besides detecting red or green pixels, the main difference between the last two papers was the handling of motion. While the last paper tried to determine the heartrate while sensing motion, the second one stopped calculating until the phone was below a certain limit of movement. In comparison to the last two papers, the first one also utilizes certain correlations regarding the heartrate, but estimates the correlations based on different input data. This shows that different combination of sensors and analysing techniques can lead to almost the same results even though their area of application is completley different.

3.3 Lung Function Diagnosing

This section will regard the detection of the lung function. Lung function is most commonly tested with medical devices like a spirometer. These devices measure instantaneous flow and cumulative volume while the user forcefully exhales. All the papers presented will detect the following lung features :

- *Forced Vital Capacity (FVC)* the total expelled volume,
- *Forced Expiratory Volume in one Second (FEV1)* the volume exhaled in the first second,
- *FEV1/FVC* and
- *Peak Expiratory Flow (PEF)* the maximum flow during the test.

. Therefore they all will utilize a microphone which is provided by a smartphone.

Table 3: Lung Function Diagnosing : Ground Truth

Paper	Subects	Positioning	Ground Truth	Comment
[4]	52	–	standard spirometer	clinical 248 clinal spirometer uses, 864 Spirosmart uses
[5]	40	phone fixed in front of mouth piece	standard spirometer (fully electronic)	clinical measurements performed by professinal

SpiroSmart, is an application developed by Larson et al. [4]. It is used to detect lung funciton. The users has to hold the smartphone infront of his mouth with a distance of approximatly one arm length. Then he has to forcefully exhale his full lung volume. The recorded audio is send to a server in order to estimate the previous mentioned features. Throughout the recording, the user gets a real-time visualization of the airflow and the duration of the test.

To analyse the incomming audio signals, recoreded by the microphone, and to estimate the flow pressure, three steps have to be applied to the data. First, the pressure loss, which accurs during the distance the sound travels, has to be compensated. Then the data has to be converted into a approximation of flow. At last, the effect of AC-coupling is removed. In order to compensate the pressure, inverse radiation modeling is applied. The inverse function utilizes arm length and head circumference which is approximated by the users height. It is applied to the incomming data which is filtered by a FIR filter. The output of the inverse radiation model is the approximate pressure at the lips. This approximate pressure is then converted into flow rate through the lips. To extract features, a regression is used which utilizes the *audio samples*, the *approximate pressure at the lips* and the *approximate flow rate through the lips*. To approximate the volumetric flow rate, they first perform a Hilbert transformation for envelope detection. Then they performed spectogram processing, since each test lasts four to seven seconds, it has 250 to 500 frames. Multiple frames are windowed and then a FFT is performed in a sliding window. The local maxima in each frame represents resonances, which are saved when they are longer than 300ms. At last they perfrom linear prediction coding, which results in an approximation of flow rate for each frame when it is applied. The approximated flow rates are denoised by applying a Savitsky-Golay filter. Then the features were used in two different regressions. The first one was used to attain the lung function meassures. The second one to attain the shape of the curve. The regressions were calculated by using machine learning. The system was trained twice, with all the participants datasets and for each participant with his particular datasets.

To evaluate the results, Larson et al. [4] tested their system with 52 subjects. The ground truth was established with a clinical spirometer (248 uses). SpiroSmart and the clinical spirometer could not be used at the same time. The distribution of error in lung function measures are 5.2% (5.0%) for FVC, 4.8% (3.5%) for FEV1, 6.3% (4.6%) for PEF and 4.0% (3.5%) for FEV1/FVC

when the system was trained with all (personalized) datasets. Accuracy can improve when a personalized test ist used, especially for patients with abnormal lung function. SprioSmart tends to slightly over-estimate the actual value for abnormal patients, which is due to two patients where inconsistent results were recorded. Those patients can be considerd outliers. They also looked at some trends. Thereby they found out that using a mouthpiece can decrese the error rate for FEV1 about 0.5% and for PEF about 1.0% on average. Also, a fixed distance to the microphone lead to no significant changes. Larson et al. [4] also created a survey for pulmonologists to evaluate cruves created by SpiroSmart. Thereby they could compare their results with professional results. Eight out of ten graphs were determined the same as by SpiroSmart. In the other two cases, the result was either classified as restrictive or as false negative which would lead to a more detailed professional analysis. In conclusion, SpiroSmart is not able to replace a clinical spirometer but it can compete to handheld spirometers which deliver similar results. This makes SpiroSmart a usefull tool for home monitoring in addition to clinical diagnosis.

Laure and Paramonov [5] describe a system called mCOPD. It is made for COPD evaluation and can determine the previously mentioned lung features. The calculation is completley done on the smartphone and needs no communication to a server. It also gives the patient guidance to perform breathing exercises which are playfully supported by two small games.

Before the incomming audio signal, which is recorded by the smartphone microphone, can be preprocessed, natural flow turbulances are eliminated by applying generated wind noise. Then the data is preprocessed in a frequency domain model by first applying a low-pass filter and then a FFT. This results in the airflow speed. After that, a machine learning regression is applied in the frequency response domain. It uses training samples and historic research. This outputs the wind speed. In post-processing the wind speed will be used to calculate the lung features. Laure and Paramonov [5] provide two games to perform breathing exercises. Both games have a ball that needs to be hold in an special area in order to *win* the game. The position of the ball varries with the strength of the patients breath. The first game requires the user to breath slow and evenly for as long as possible through their pursed lips. The second game requires the user to train diaphgramatic breathing, which strengthens the diphagram and abdominal muscles by breathing without using the chest muscles[5].

Laure and Paramonov [5] used 40 subjects in order to evaluate their results. The ground truth was established by using a clinal spirometer. When comparing the results of mCOPD to a clinical spriometer, the smartphone was behind the spriometer, so both devices mesaured the lung function during the same test run. The distance to the device was very close and there was no resistance in the mouth piece which would have resulted in decrease of airflow. This resulted in a average deviation of 6.5% in FVC, 3.6% in FEV1 and 3.9% in FEV1/FVC. In conclusion. mCOPD provides a simple solution to determine the LUNG features and can help the user to perform breathing exercises. It does that with the

average deviation for all features of less than 6.5%.

When comparing the previously presented approaches to meassure and to estimate lung function, both deliver relativly similar error rates. Although it has to be noted that SpiroSmart has more ground truth data. SpiroSmart uses a far more complicated algorithm to determine the lung function which can be personalized even better that mCOPDs.

4 Conclusion

Smart devices are becomming more popular and the technology is becomming more powerfull which will not only increase the computational power of a smart device but also the accuracy of its sensors. There might even be new sensors which will open completely new fields in monitoring.

In this research almost only smartphone were used to monitor exercises, heartrate or lung functions. Besides the lack of computational power offered by smartwatches or wristbands, most of the researches presented were published over one year ago, but smartwatches and wristbands have just started to gain popularity over the last two years. It is possible to combine multiple smart devices. For example, a smart wristband can send data to the smartphone via bluetooth. The smartphone then processes the data with its more powerfull processor. With that beeing said, combining not only the sensors and devices, but also the techniques used can lead to more powerfull feature exertraction and more accurate results.

The presented techniques in exercise monitoring are able to detect repetitons, execution and the actual exercise. They can perform exercise monitoring with a low error rate while having a reasonably high amount of ground truth data. The feedback they supply can really help the user to improve his fitness level and general training results. Heartrate monitoring can be done in different ways without special hardware. The detected heartrate, wheter it was detected with a camera or by utilizing an accelerometer, is not as accuracte as a professional clinical device, but it has only a small error rate of less then five bpm on average. This makes it usuable for home monitoring and detecting trends. Lung function monitoring is similar to heartrate monitoring. The results have an average error of less than 7% for all features, this is only suitable for home monitoring, performing exercises and especially detect trends which can lead to treatment or a professional review.

Health monitoring on smart devices is obviously not as accurat as on clinical devices. But this is not the point in monitoring. Although smart devices try to receive as good and accurate results as possible, the main idea is to detect trends and give the user a simple tool to check his health condition and improve it. The presented techniques are able to do so for a great amount of people without any professional calibration or help. The short term goal for monitoring with smart devices should be providing accurate information about trends (e.g. lung function) and in the long term they should provide exact information.

Bibliography

[1] Hao, T., Xing, G., Zhou, G.: Runbuddy: A smartphone system for running rhythm monitoring. In: UbiComp'15 (2015)
[2] Kitajima, T., Choi, S., Murakami, E.A.Y.: Heart rate estimation based on camera image. Tech. rep., Samsung (2014)
[3] Lagido, R.B., Lobo, J., Leite, S., Sousa, C., Ferreira, L., Silva-Cardoso, J.: Using the smartphone camera to monitor heart rate and rhythm in heart failure patients. Tech. rep., Associacao Fraunhofer Portugal Research and Faculdade de Medicina da Universidade do Porto (2014)
[4] Larson, E.C., Goel, M., Boriello, G., Heltshe, S., Rosenfeld, M., Patel, S.N.: Spirosmart: Using a microphone to measure lung function on a mobile phone. In: UbiComp'12 (2012)
[5] Laure, D., Paramonov, I.: mcopd: Mobile phone based lung function diagnosis and exercise system for copd. Tech. rep., University of California, Los Angeles (2013)
[6] Pernek, I., Hummel, K.A., Kokol, P.: Exercise repetition detection for resistance training based on smartphones. In: Pers Ubiquit Comput. Springer-Verlag London (2013)
[7] Spina, G., Huang, G., Vaes, A., Spruit, M., Amft, O.: Copdtrainer: A smartphone-based motion rehabilitation training system with real-time acoustic feedback. In: UbiComp'13 (2013)
[8] Statista: Fitness worldwide. Website (2015), online :https://de.statista.com/outlook/313/100/fitness/weltweit#market-revenue
[9] Statista: Smartphones sales worldwide. Website (2016), online :http://www.statista.com/statistics/263437/global-smartphone-sales-to-end-users-since-2007/
[10] Sumida, M., Mizumoto, T., Yasumoto, K.: Estimating heart rate variation during walking with smartphone. In: UbiComp'13 (2013)
[11] Sundholm, M., Cheng, J., Zhou, B., Sethi, A., Lukowicz, P.: Smart-mat: Recognizing and counting gym exercises with low-cost resistive pressure sensing matrix. In: UbiComp'14 (2014)

Semantics support location-aware systems – Semantic trajectory mining

Dominik Köhler*

Betreuer: Antonios Karatzoglou†

Karlsruher Institut für Technologie (KIT)
Pervasive Computing Systems – TECO
*ubeas@student.kit.edu
†antonius@teco.edu

Zusammenfassung. Standortvorhersagen finden häufig Anwendung in mobilen Assistenten, Werbung oder sozialen Netzwerken. Klassische Ansätze zur Standortvorhersage, wie die Anwendung von Markov Modellen, beachten jedoch nur geografische und vernachlässigen dabei semantische Eigenschaften einer Trajektorie. Ein Verfahren, dass durch Anwendung von semantischen Trajektorien eine bessere Standortvorhersage bieten will, wird in Ying [4] vorgestellt. Dieses Verfahren ermöglicht beispielsweise Standortvorhersagen für einen Benutzer in einer bisher unbekannten Stadt.
Zu Beginn der Arbeit wird ein klassischer Ansatz mit Markov Modellen beschrieben. Im weiteren Verlauf wird das neuartige Verfahren von Ying [4] vorgestellt sowie mögliche Probleme mit diesem erläutert. Dabei werden Ähnlichkeiten und Unterschiede beider Verfahren aufgezeigt.

Schlüsselwörter: Standortvorhersage, Trajektorien, Semantische Trajektorien, Markov Modelle, Data mining

1 Einleitung

Mobile Assisten wie Google Now oder Siri haben das Ziel unseren Alltag zu vereinfachen. Solche Systeme verwenden kontextbasierte Informationen, um Vorhersagen über den nächsten Aufenthalt zu tätigen und dabei für den Benutzer relevante Informationen anzuzeigen. Zwei wichtige kontextbasierte Informationen sind Ort und Zeit. Diese Daten werden gesammelt und können für zukünftige kontextbasierte Vorhersagen ausgewertet werden. Beispiele für kontextbasierte Vorhersagen sind der nächste Aufenthalt des Benutzers sowie damit verbundenen Informationen wie Öffnungszeiten, Wetterdaten etc. Es ist also notwendig eine möglichst genaue Standortvorhersage treffen zu können. Eine solche Vorhersage basiert auf kontextbasierten Informationen, die in der Vergangenheit liegen.
Es gibt verschiedene Ansätze um Standortvorhersagen zu treffen. Ein klassisches Verfahren, welches Markov Modelle verwendet um die Wahrscheinlichkeiten eines nächsten Standortes des Benutzers zu bestimmen, wird in Ashbrook [1] beschrieben. Solche Verfahren gehören zur Kategorie der sogenannten "general-pattern-based"Verfahren und basieren meist auf häufiges und ähnliches Verhalten eines

Benutzers. Dabei werden relative Häufigkeiten von Bewegungsmustern eines Benutzers verwendet um Standorte vorherzusagen.
Bewegungsmuster können aus geographischen Trajektorien eines Benutzers gewonnen werden. Diese Trajektorien werden als Sequenzen von GPS-Punkten (Latitude, Longitude) aufgefasst. Alle GPS-Punkte einer Trajektorie sind mit einem Zeitstempel markiert und zeitlich sortiert. Eine Trajektorie beschreibt somit den Verlauf von geographischen Punkten eines Benutzers. Abb. 1 zeigt drei verschiedene Trajektorien eines Benutzers an. $Trajektorie_1$ und $Trajektorie_2$ ähneln sich in diesem Beispiel in ihren Verläufen, jedoch ist deutlich zu erkennen, dass $Trajektorie_2$ und $Trajektorie_3$ die gleiche semantische Trajektorie "*School*" $\rightarrow$ "*Bank*" $\rightarrow$ "*Hospital*" darstellen. Verfahren, die semantische Eigenschaften einer Trajektorie nicht berücksichtigen, können somit für $Trajektorie_2$ zu einer falschen Vorhersage führen. Desweiteren können klassische Ansätze nicht dazu verwendet werden, Standorte in einer unbekannten Stadt vorherzusagen.
Hier knüpft der Ansatz von Ying [4] an. Anhand von semantischen, sowie geographischen Eigenschaften von Trajektorien, werden Standortvorhersagen mittels eines neuartigen Verfahrens getroffen. Dieses Verfahren macht sich die Vorteile beider Eigenschaften zu nutze mit dem Ziel genauere Vorhersagen zu treffen.
In dieser Arbeit sollen Probleme beider Verfahren dargestellt und verglichen werden. Zunächst werden beide Verfahren beschrieben und dabei aufgezeigt welche Probleme in bestimmten Abschnitten auftreten.

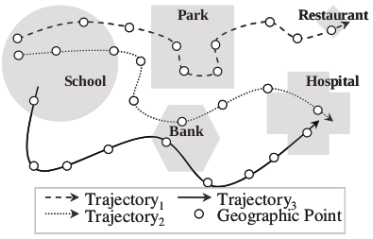


Abb. 1. $Trajektorie_1$ und $Trajektorie_2$ ähneln sich im Verlauf. Semantisch gesehen sind diese jedoch unterschiedlich voneinander. Quelle: [4]

2 Verfahren von Ashbrook

Der Ansatz von Ashbrook [1] beschreibt im ersten Schritt das Auffinden von signifikanten Orten. Man geht davon aus, dass während der Aufzeichnung von GPS-Punkten des Benutzers Lücken entstehen. Diese Lücken entstehen durch den Verlust des GPS-Signals. Ein GPS-Signal geht beispielsweise beim Betreten eines Gebäudes verloren.
Nach der Bestimmung von sogenannten Places werden diese anhand der Dichte im Raum geclustert. Dazu verwendet Ashbrook den k-Means Algorithmus.

Die entstandenen Cluster werden Locations genannt. Einer Location wird dabei der Mittelwert aller enthaltenen GPS-Koordinaten von Places als geographisches Attribut zugewiesen. Um Sublocations zu erhalten wird k-Means wiederholt auf eine Location angewandt. Wurde zum Beispiel eine *University* Location geclustert, so kann diese den ganzen Campus repräsentieren. Möchte man jedoch auch die einzelnen Gebäude der Universität berücksichtigen, können durch Anwendung von k-Means auf diese Location weitere Sublocations geclustert werden. Dies ermöglicht das Erstellen von dichten Locations.
Am Ende des Verfahrens wird für alle gefundenen Locations eines Benutzers ein Markov Modell erstellt. Jeder Knoten im Markov Modell reprästentiert dabei eine Location und jede Kante die Wahrscheinlichkeit eines Übergangs des Benutzers von Location A zu Location B. Verwendet man Markov Modlle n-ter Ordnung so werden die Wahrscheinlickeiten präziser, da schon besuchte Locations mit in die Wahrscheinlichkeiten eines Übergangs einbezogen werden.

2.1 Places

Ashbrook geht davon aus, dass während der Aufzeichnung von GPS-Punkten, Lücken entstehen. Dies wird durch Signalabbrüche des aufzeichnenden GPS-Gerätes begründet. Betritt man zum Beispiel ein Gebäude, geht das Signal verloren. Dadurch sollen sich signifikante Orte bzw. sogenannte Places finden lassen. Das bedeutet wiederum, dass Orte wie Freibäder und Parks nicht als signifikante Orte erkannt werden.
Zur Festlegung eines signifikanten Ortes muss die Differenz der GPS-Punkte, nach und vor dem Signalverlust, mindestens einem Schwellenwert entsprechen. Dieser Parameter darf nicht zu klein ausfallen, da ansonsten bei kurzen Signalverlusten fälschlicherweise ein Place erkannt wird. Kürzere Signalverluste in Städten, sind aufgrund von Störfaktoren nicht zu vermeiden. Außerdem benötigt das Gerät eine gewisse Zeit bis es wieder ein Signal erfassen kann.
Diese Art der Bestimmung von signifikante Orten ist stark auf den von Ashbrook verwendeten Datensatz angepasst. Dieser wurde wiederum anhand eines einzigen Benutzers erstellt. Der genaue Ablauf der Aufzeichnung und die Struktur des Datensatzes waren also im Voraus bekannt. In Datensätzen ohne Vorkenntnis der Vollständigkeit muss jedoch garantiert sein, dass Benutzer kontinuierlich GPS-Daten aufzeichnen. Unter realen Bedingungen kann es vorkommen, dass das aufzeichnende Gerät ausgeschaltet oder die Aufzeichnung gestoppt wird. Dadurch würden Lücken entstehen die fälschlicherweise als Places erkannt werden. Eine mögliche Vorfilterung von "falschen" Lücken ist schwer, da nachträglich nicht nachvollzogen werden kann aus welchem Grund eine Lücke entstanden ist.

2.2 Locations

Die festgelegten Places können noch nicht zur Vorhersage verwendet werden. Aufgezeichnete GPS-Punkte können sich um mehrere Meter unterscheiden, obwohl der Benutzer den gleichen Ort besucht hat. Dadurch würden verschiedene Vorhersagen für gleich Orte getroffen werden.

Zur Behebung des Problems wird ein Clustering Algorithmus verwendet, der Places aufgrund ihrer Dichte clustert. Solche Cluster werden als Location bezeichnet. Dieser Schritt stellt einen wesentlichen Unterschied zum Ansatz von Ying dar. Dieser clustert Places nicht nur anhand ihrer Dichte sondern auch anhand der "Popularität" eines Places. Ein Place wird als populär betrachtet, sobald eine Mindestanzahl verschiedener Personen ihn besucht hat. Places, die nicht als populär erkannt werden können, markiert der von Ying verwendete Algorithmus als Rauschen.
Ashbrook verwendet den k-Means Algorithmus, welcher kein Rauschen erzeugt. Dadurch bleiben persönliche Places erhalten und werden einer Location zugeteilt. Es können jedoch auch Locations enstehen, welche eine geringe Signifkanz besitzen und wenig zur Vorhersage beitragen.
Im nächsten Schritt werden zudem potenzielle Sublocations gefunden. Ein zu groß gewählter Radius des k-Means Algorithmus kann dafür sorgen, dass naheliegende Locations zusammengefasst werden. Ein zu klein gewählter Radius kann hingegen zu vielen Locations mit nur einem Place führen. Doch selbst bei der richtigen Wahl des Radius wird die Dichte von Places teilweise missachtet. So können Locations aus mehreren dichten Bereichen entstehen, welche aufgrund von Ausreißern miteinander in Verbindung gebracht werden. Ziel ist es Sublocations in Locations einzufügen. Dafür wird k-Means auf eine Location angewandt.

2.3 Vorhersage

Zur Vorhersage wird ein Markov Modell der Locations erstellt. Dazu wird der Verlauf der Places eines Benutzers in eine Liste von Locations umgewandelt. Dadurch kann nachvollzogen werden in welcher Reihenfolge bestimmte Locations besucht wurden. Es können relative Häufigkeiten ermittelt und damit die Wahrscheinlichkeit für den nächsten Aufenthalt berechnet werden. Ein Knoten im Markov Modell entspricht dabei einer Location. Eine Kante zwischen Knoten gibt die Wahrscheinlichkeit an, mit der die Person diese Location besuchen wird. Hat sich ein Benutzer noch nie zwischen zwei Locations bewegt, ist die entsprechende Wahrscheinlichkeit null. Markov Modelle n-ter Ordnung ermöglichen an dieser Stelle genauere Wahrscheinlichkeiten und somit bessere Vorhersagen. Der Grund liegt darin, dass schon besuchte Locations in die Wahrscheinlichkeiten miteinbezogen werden. Dabei sollte die Ordnung entsprechend der Größe des Datensatzes gewählt werden.
Zur Vorhersage werden die aktuellen Übergänge eines Benutzers zwischen Locations betrachtet und mit denen im Markov Modell abgeglichen. Stimmen die Übergänge kann die nächste Location mit der höchsten relativen Wahrscheinlichkeit vorhergesagt werden.
Desweiteren können nur Locations vorhergesagt werden die bereits einmal besucht wurden. Unbekannte Orte können nicht zur Vorhersage verwendet werden. Genau aus diesen Gründen beschreibt Ying ein neuartiges Verfahren, dass diese Probleme lösen soll.

3 Verfahren von Ying

Ying teilt sein Framework "SemanPredict" in ein Offline und Online Modul auf. Ersteres dient zur Analyse der Daten und lässt sich in drei wesentliche Schritte aufteilen: 1) Aufbereitung der Daten, 2) Semantisches Mining, 3) Geographisches Mining. Das Online Modul dient zur Vorhersage von Standorten und greift auf die zuvor im Offline Modul gewonnenen Informationen zu. Dabei werden nicht nur Verhaltensmuster eines einzigen Benutzers zur Standortvorhersage verwendet sondern auch die von anderen Benutzern welche ähnliche Verhaltensmuster aufweisen.
Das Offline Modul soll im wesentlichen häufige Bewegegungsmuster einzelner Benutzer anhand von sogenannten semantischen Mustern, sowie mit Hilfe von geographischen Clustern aller Benutzer bestimmen. Diese semantischen Muster geben häufige Muster in semantischen Trajektorien eines Benutzers wieder. Eine semantische Trajektorie beschreibt eine Trajektorie, deren Aufenthalte bestimmt und mit semantischen Labels annotiert wurden.
Geographische Cluster werden anhand von Ähnlichkeiten zwischen semantischen Mustern der Benutzer gebildet. Ying wählt zur Repräsentation von semantischen Mustern und geographischen Mustern eine kompakte Baumstruktur, welche die Prädiktion im Online Modul vereinfachen soll. Durch jeden Schritt im offline Modul werden die Daten immer weiter komprimiert und gleichzeitig der Informationsgehalt weiter erhöht um eine effiziente und schnelle Vorhersage im online Modul treffen zu können.
Abb. 2 zeigt den Verlauf der einzelnen Schritte, die jeweils im offline sowie online Modul angewandt werden. Im weiteren Verlauf der Arbeit wird auf jeden der Schritte eingegangen und Probleme mit diesen aufgezeigt, sowie mögliche Lösungsvorschläge erläutert.

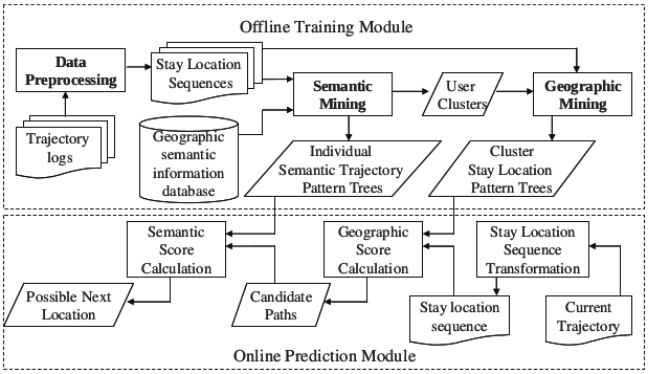


Abb. 2. Offline und Online Modul von SemanPredict. Quelle: [4]

3.1 Aufbereitung der Daten

Zu Beginn stehen GPS-Trajektorien aus einem gewählten Datensatz zur Verfügung. Eine Trajektorie besteht aus einer Sequenz von geographischen Punkten, die aus Latitude und Longitude bestehen. Zudem sind diese Punkte mit einem Timestamp versehen. Durch die Informationen von Ort und Zeit kann der geographische Verlauf einer Person nachvollzogen werden. Eine Trajektorie entspricht einem Tagesablauf einer Person. Diese Bedingung muss nicht im verwendeten Datensatz gegeben sein. Deshalb ist eine Vorfilterung des Datensatzes notwendig. Dabei werden GPS-Punkte eines einzigen Tages und Benutzers zu einer Trajektorie zusammengefasst.
Als weiteren Schritt werden Stay Points [6] erstellt. Diese stellen den Aufenthalt einer Person an einem Ort über eine bestimmte Zeit hinweg dar. Um Stay Points zu finden wird der in Zheng [6] beschriebene Algorithmus angewandt. Der Algorithmus basiert auf zwei wesentlichen Parametern: der Distanz, die mindestens hinterlegt werden muss, sowie einem zeitlichen Schwellenwert, in der die hinterlegte Strecke liegen muss. In Abb. 3 ist eine GPS-Trajektorie $[p_1 \rightarrow p_2 \rightarrow ... \rightarrow p_n]$ zu sehen. $\{p_1, .., p_n\}$ stehen für GPS-Punkte die in einer GPS-Trajektorie als zeitlich aufeinanderfolgende Sequenz dargestellt wird.
Aus der in Abb. 3 zu sehende GPS-Trajektorie wurde ein Stay Point bestimmt, der aus einer Menge von GPS-Punkten $\{p_3, p_4, p_5, p_6\}$ besteht. Um als Stay Point erkannt zu werden, muss die Distanz von p_3 zu p_6 mindestens dem festgelegten Schwellenwert betragen. Zudem muss die zeitliche Differenz von p_3 und p_6 mindestens dem zeitlichen Schwellenwert entsprechen.

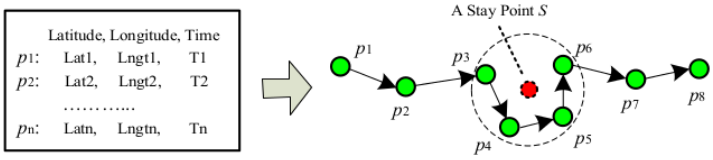


Abb. 3. Beispiel eines Stay Points einer GPS Trajektorie. Quelle: [6]

Zheng beschreibt, dass Stay Points typischerweise in folgenden zwei Situationen auftreten können:
1) Eine Person betritt ein Gebäude, weswegen das GPS Signal des aufzeichnenden Gerätes über eine bestimmte Zeit verloren geht. Wird das Gebäude am anderen Ende verlassen, so wird ein Stay Point erkannt.
2) Die Person läuft besipielsweise im Park oder am Strand eine Strecke auf und ab.
Diese zwei Fälle verdeutlichen, dass die Wahl beider Parameter zur Auffindung von Stay Points eine wichtige Rolle spielen. Zur Erkennung eines Stay Points sei beispielsweise eine maximale Distanz von 50m und ein Mindestaufenthalt von 30min erforderlich. Betritt eine Person ein 100m langes Gebäude auf der einen Seite und verlässt es nach 30min auf der gegenüberliegenden Seite, so würde kein

Stay Point erkannt, da die Distanz den erlaubten Wert überschreitet. Außerdem kann eine zu klein gewählte maximal erlaubte Distanz zu falschen Stay Points führen. In Fall 2) würden so fälschlicherweise mehrere Stay Points erkannt werden.
Da sich die Aufenthalte von Benutzern innerhalb von Städten auf kleinem Raum bei kurzem zeitlichem Intervall abspielen, sollte ein zu großer Schwellenwert für die Distanz vermieden werden. Somit empfehlen sich für Städte nicht zu große aber auch nicht zu kleine Schwellenwerte für die Distanz. Welche Schwellenwerte letztenendlich gewählt werden hängt jedoch individuell vom vorliegenden Datensatz ab. Da sich Yings Vorhersagen innerhalb einer Stadt bewegen, verwendet er die in Zheng [6] verwendeten Schwellenwerte von 200 Meter und 30 Minuten. Ashbrooks [1] Vorhersagen gehen jedoch über die Grenzen einer Stadt hinaus, weshalb er Schwellenwerte von 300 Meter und 10 Minuten benutzt.
Die Wahl des richtigen Schwellenwertes für die Mindestdauer eines Aufenthaltes innerhalb der Distanz ist auch kritisch zu betrachten. Wählt man diesen Schwellenwert zu klein so kann es dazu führen, dass zuviele unwichtige Stay Points erkannt werden. Bleibt eine Person zum Beispiel im Stadtverkehr an einer Ampel zu lange stehen, wird dort ein Stay Point erkannt. Demnach sollte ein zu kleiner Schwellenwert vermieden werden. Umgekehrt werden zu wenige Stay Points erkannt, falls die Wahl des Wertes zu groß ausfällt.

3.2 Stay Locations und Stay Location Sequenz

Nach Bestimmung aller Stay Points folgt die Extraktion von Stay Locations. Eine Stay Location ist definiert als Cluster von Stay Points, die sich in einem bestimmten Radius zueinander befinden. Zusätzlich muss ein gefundenes Cluster mindesten Stay Points von k verschiedenen Personen enthalten. Zum Clustern von Stay Points kann demnach kein Algorithmus wie k-Means verwendet werden, der auf Basis der Dichte im Raum Stay Point Cluster bildet. Es muss ein Algorithmus verwendet werden, der es ermöglicht Stay Points denen kein Cluster zugewiesen werden konnte, als Rauschen zu markieren.

3.2.1 P-DBSCAN Ying verwendet hierfür P-DBSCAN [3] als Algorithmus zum clustern von signifikanten Orten. Dabei wird ein Ort als signifikant erachtet, falls von mindestens k verschiedenen Personen Stay Points im Cluster vorhanden sind. Bei $k > 1$ werden demnach Orte die von nur einer Person besucht wurden, vom Algorithmus als nicht signifikant erachtet und markiert. Das zuvor genannte Problem ist die Generalisierung von persönlichen Orten. Entsprechende Stay Points persönlicher Orte wurden zwar zuvor berechnet, werden jedoch durch P-DBSCAN als unrelevant aufgefasst. Dadurch gehen wichtige persönliche Stay Locations verloren und man erhält eine stark generalisierte Menge von Stay Locations. Es können also nur Vorhersagen über Orte getätigt werden, die als signifikant erachtet wurden. Selbst wenn ein Benutzer regelmäßig einen relevanten Ort besucht, dieser jedoch von keiner anderen Person besucht wurde, wird dieser

Ort in keine Vorhersage einbezogen. Dies kann auf viele wichtige Orte zutreffen wie den Arbeitsplatz, die eigene Unterkunft oder ähnliche persöhnliche Orte, die in einer kontextbasierten Standortvorhersage nicht fehlen sollten.

3.2.2 Parameterwahl P-DBSCAN benötigt zwei verschiedene Paramter um Cluster bilden zu können. Zum einen wird die Mindestanzahl k an verschiedenen Personen benötigt, welche Stay Points zu einem potenziellen Cluster beigetragen haben. Zum anderen wird ein Radius r benötigt, der die Größe der zu betrachtenden Nachbarschaft [3] eines Stay Points angibt. Alle Stay Points die innerhalb einer Nachbarschaft liegen werden zum aktuellen Cluster hinzugefügt. Die Nachbarschaft aller neu hinzugefügten Stay Points wird wiederum betrachtet. Dies wird wiederholt bis keine neuen Stay Points in allen Nachbarschaften des Clusters zu finden sind. Die Wahl der Parameter sollte hierbei wieder gut überlegt sein und hängt stark vom zugrunde liegenden Datensatz ab. So sollten innerhalb einer Stadt andere Parameter als Überregional gewählt werden. In Abb. 5 ist ein Fall zu sehen, der bei einem zu groß gewähltem Radius eintritt. Dabei gehen wir von einem Datensatz aus, der sich auf eine Stadt begrenzt. Statt zwei seperate Stay Locations zu erkennen wird eine große erkannt. Grund dafür sind die beiden Stay Points p_4 und p_5, die jeweils am Rand ihres eigenen Clusters sitzen. Ein zu großer Radius führt dazu, dass p_5 in der Nachbarschaft von p_4 erkannt wird. Da sich alle restlichen grünen Stay Points, in Nachbarschaft zu p_5 befinden und alle roten Stay Points in direkter Nachbarschat zu p_4 stehen, wird eine große Stay Location gebildet. Offensichtlich ist dies jedoch falsch, da zwei verschiedene Orte, “Fitness Club” und “Hotel”, als Stay Location erkannt werden sollten.
Zu kleine Radii könnten wiederum zum Verlust von Stay Locations führen. Dies bedeutet, dass der komplette Cluster verworfen wird, da nicht ausreichend viele verschiedene Personen zum Cluster beigetragen haben.

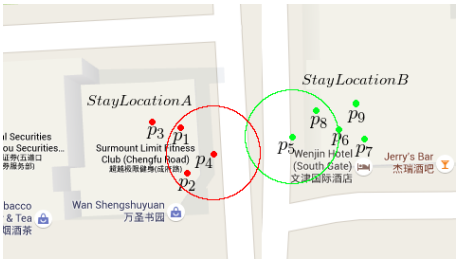


Abb. 4. Cluster Rot und Grün werden als seperate Stay Locations erkannt.

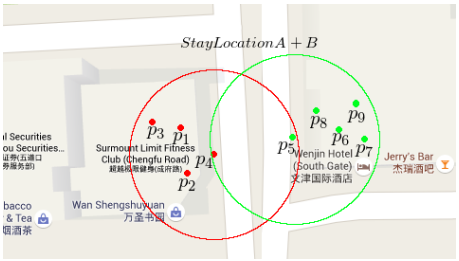


Abb. 5. Beide Cluster wurden als nur eine Stay Location erkannt.

Bisher wurde nur aufgezeigt wie sich der gewählte Radius auf die Ergebnisse auswirkt. Die Wahl von k hat einen genauso starken Einfluss auf die Anzahl der gefundenen Stay Locations wie die Wahl des Radius. Wählt man einen Da-

tensatz, welcher viele räumlich dichte Stay Points liefert, so kann die Anzahl k größer gewählt werden. In Kisilevich [3] werden beispielsweise Werte von $k > 10$ genommen, was in anbetracht von räumlich verteilten Stay Points sehr hoch ist. Kisilevich [3] verwendet zur Optimierung der Suche von dichten Points und der schnelleren Konvergenz des Algorithmus hin zu Clustern einen sich anpassenden Schwellenwert der Dichte. Diese Optimierung wird von Ying jedoch nicht verwendet. Ohne Optimierung wird die Nachbarschaft von Stay Points geprüft, die sich am Rande eines Clusters aufhalten. Da die Dichte am Rande eines Clusters für gewöhnlich abnimmt und dieser Bereich für das gesamte Cluster eher uninteressant ist, sollten diese nicht weiter betrachtet werden. Ohne diese Optimierung können Fälle auftreten wie in Abbildung 5. Stay Points am Rande eines Clusters sorgen dafür, dass das Cluster immer weiter wächst, obwohl die Dichte abnimmt. Es entstehen somit große und verteilte, anstatt kleine und dichte Cluster.

3.2.3 Stay Location Sequenz Nach Auffinden aller Stay Locations werden sogenannte Stay Location Sequenzen erstellt. Eine solche Sequenz spiegelt den Tagesablauf einer Person wieder und gibt Aufschluss darüber an welchen signifikanten Orten sich dieser aufgehalten hat. Aus einer Stay Point Trajektorie wird die Stay Location Sequenz gewonnen. Zunächst werden alle Stay Points, welche durch P-DBSCAN als Rauschen markiert wurden, aus der Trajektorie genommen. Alle verbleibenden Stay Points werden durch ihre zugehörige Stay Location substituiert und man erhält eine Stay Location Sequenz. Dieser Schritt kann jedoch zu unbrauchbaren Sequenzen führen. Hat ein Benutzer eine Stay Point Trajektorie $[StayPoint_1 \rightarrow StayPoint_2 \rightarrow StayPoint_3 \rightarrow StayPoint_4]$ und gehören $\{StayPoint_1, StayPoint_3, StayPoint_4\}$ keiner Stay Location an, so wird die Stay Location Sequenz $[StayLocation_1]$ erzeugt.
$StayLocation_1$ besteht dabei aus $StayPoint_2$ und weiteren ihr zugewiesenen Stay Points. Eine solche Stay Location Sequenz spiegelt jedoch nicht den tatsächlichen Tagesablauf einer Person wieder. Es wurde nur der Aufenthalt an einem für den Algorithmus signifikanten Ort erkannt. Ein weiteres mal kommt das Generalisierungsproblem zum tragen. Trajektorien einer Person werden hierdurch auf stark generalisierte Tagesabläufe heruntergebrochen.

3.3 Semantic Trajectory Pattern Tree

In diesem Abschnitt wird erklärt wie aus den zuvor bestimmten Stay Location Sequenzen ein Semantic Trajectory Pattern Tree entsteht. Dieser Teil besteht im wesentlichen aus zwei Schritten: zuerst werden Labels für Stay Locations gefunden und alle Stay Location Sequenzen eines Nutzers in semantische Trajektorien umgewandelt. Eine solche semantische Trajektorie kann die Form $[School \rightarrow Park \rightarrow Restaurant]$ wie in Abb. 1 besitzen. Es können auch semantische Trajektorien der Form $[School \rightarrow Bank \rightarrow \{Market, Restaurant\}]$ vorkommen. Da der letzten Stay Location in der Sequenz mehrere semantische Label zugewiesen wurden.
Jede Person besitzt demnach ein Menge an semantischen Trajektorien. Der Un-

terschied zur normalen Trajektorie ist die Unabhängigkeit von der geographischen Eigenschaft. Eine semantische Trajektorie ist somit in einer beliebigen Stadt gültig, da diese den Verlauf von Aufenthalten an allgemein bezeichneten Orten beschreibt. Es ist nun möglich, häufige Verhaltensmuster eines Benutzers aus seinen semantischen Trajektorien zu bestimmen. Hierfür wird der Prefix-Span [2] Algorithmus auf allen semantischen Trajektorien eines Benutzers angewandt. Im zweiten Schritt werden aus den erhaltenen Mustern eines Benutzers, ein Semantic Trajectory Pattern Tree (kurz STP-Tree) erstellt. Dieser STP-Tree ist eine kompakte Schreibweise aller semantischen Muster. Jeder Pfad zu einem Knoten im Baum kann als Entscheidungsregel aufgefasst werden. Durch die kompakte Form des Baumes können die Verhaltensmuster in der Prädiktionphase effizent ausgewertet werden.

3.3.1 Semantische Trajektorie Über Reverse Geocoding ist es möglich, eine semantische Annotation zu einem gegebenen geographischen Punkt zu erhalten. Es gibt verschiedene APIs wie die Google Maps API oder Foursquare API, welche Reverse Geocoding unterstützen. Solche APIs geben Informationen über die umliegende Umgebung und falls vorhanden konkrete Informationen, wie Name, Straße und Beschreibungen über den angegeben Ort, preis. Die einzige Information die jedoch von Interesse ist, ist die Kategorie eines Ortes wie z.B. $Park, Bar, Restaurant, Krankenhaus$. Demnach wird eine möglichst allgemeine und übergeordnete Beschreibung eines Ortes verwendet. Konkrete Namen von Orten wie "Oxford Pub" oder "Stövchen" werden nicht benötigt. Diese Art des Annotieren von Stay Locations ist von Nöten, falls der verwendete Datensatz keine Label angibt. Aufgrund von Ungenauigkeiten der erfassten GPS-Punkten kann nicht garantiert werden, dass richtige Labels gefunden werden. Vielmehr ist dieser Schritt abhängig von der Korrektheit der verwendeten Reverse Geocoding API. Idealerweise sollte ein Benutzer manuell labeln, um oben genannte Fehler bei der Annotation zu vermeiden.
Ein semantisches Label wird vorerst einem Stay Point zugewiesen. Da eine Stay Location nichts weiter als eine Menge von Stay Points ist und diese möglicherweise Stay Points mit unterschiedlichen Labels enthält, können Ihr mehrere Labels zugewiesen werden. In Abbildung 4 sieht man rechts, in unmittelbarer Nähe zum grünen Cluster, eine Bar. Der Stay Point p_7 könnte das Label "Bar" erhalten wohingegen das Label "Hotel" den Punkten p_5, p_6, p_8 und p_9 zugewiesen wird. Die Stay Location würde somit das Label $\{Hotel, Bar\}$ erhalten.

3.3.2 *Home* Stay Location Aufgrund der in 3.2.1 beschriebenen Generalisierung von siginifikanten Orten, sind persönliche Stay Locations wie die eigene Unterkunft verloren gegangen und somit auch deren Label *Home*. Ein ideales Verfahren sollte jedoch einen derart wichtigen Ort in seine Standortvorhersage miteinbeziehen. Eine mögliche Lösung wäre, Stay Points nach Uhrzeiten zu durchsuchen und nachträglich das Label anzupassen. Wir nehmen an, dass sich die meisten Menschen zwischen 0 und 6 Uhr morgens in ihrer Wohnung aufhalten. Durch diese Annahme können wir Stay Points finden, die zwischen diesen

Zeiten liegen. Es sollten nur Stay Points beachtet werden, die keiner Stay Location angehören. Mit den gefundenen Stay Points kann für jede Person eine Stay Location mit dem Label *Home* erstellt werden. Das ist möglich, da *Home* eine Stay Location darstellt, die zugleich als persönlich und signifikant angesehen werden kann. Man kann davon ausgehen, dass jede Person eine *Home* Stay Location haben sollte und deshalb ist es erforderlich diese in die Vorhersagen mit einzubeziehen.

3.3.3 Semantic Trajectory Pattern Nach Erzeugung von semantischen Trajektorien, werden diese nach häufigen Mustern durchsucht. Dazu wird PrefixSpan [2] auf die Menge aller semantischen Trajektorien eines Benutzers aufgerufen. Als Parameter muss dem Algorithmus ein sogenannter Support übergeben werden. Dieser gibt an, wie häufig ein einzelnes Label mindestens in der Menge aller semantischen Trajektorien eines Benutzer vorkommen muss, um betrachtet zu werden. Nach Ausführung von PrefixSpan werden alle Muster, sowie deren Teilfolgen der semantischen Trajektorien, ausgegeben. Diese Muster werden semantic trajectory patterns genannt und bieten uns mehrere Entscheidungsregeln. So kann man anhand der Muster das nächste semantische Label vorhersagen, indem man die aktuelle Bewegung eines Benutzers mit seinen semantischen Mustern vergleicht. Bei längeren Mustern kann dies jedoch ineffizient sein. Aus diesem Grund verwendet Ying einen sogenannten semantic trajectory pattern tree, wie in Abb. 8 zu sehen. Dieser Präfixbaum enthält alle Muster der semantischen Trajektorien eines Benutzers. Ein Pfad kann als Eintscheidungsregel aufgefasst werden. Desweiteren kann ein Pfad mehrere semantic trajectory patterns gruppieren. Hat man beispielsweise den Pfad $(\{Park, Bank\}) \rightarrow (Hospital)$, so repräsentiert dieser die Muster $< Park, Hospital >, < Park >, < Bank >, < \{Park, Bank\} >$ und $< Hospital >$. Die in Abb. 6 dargestellten Support-Werte geben an, wie häufig ein Muster in allen semantischen Trajektorien vorkommt. Dieser Wert wird dazu verwendet, die Entscheidung im Knoten eines STP-Trees zu treffen.

3.4 Cluster von Benutzer

Im nächsten Schritt werden Cluster von Benutzer anhand der zuvor gebildeten semantischen Muster gebildet. Dadurch soll ermöglicht werden, dass nicht nur aufgrund der eigenen semantischen Bewegungsmuster Vorhersagen getroffen werden können, sondern auch auf Basis von ähnlichen Benutzern. Die Ähnlichkeit zwischen zwei Benutzern wird über die MSTP-Similarity [5] ermittelt. Das Verfahren berechnet den gewichteten Durchschnit der anteiligen Verhältnisse an der Longest Common Sequence zweier semantischen Muster. Die Anzahl aller enthaltenen Labels eines Musters wird dabei als Gewicht verwendet.
Danach wird ein gewichteter Durchschnitt aller MSTP-Similarities berechnet. Dieser Wert gibt die Ähnlichkeit zwischen zwei Benutzern an. Wiederholt man diesen Prozess für alle möglichen Benutzerpaare, erhält man eine Matrix, welche Ähnlichkeiten in Bezug auf semantische Bewegungsmuster der Benutzer liefert.

Semantic Trajectory Pattern	Support
<Unknown>	2/3 = 0.667
<School>	3/3 = 1.0
<Park>	3/3 = 1.0
<Hospital>	2/3 = 0.667
<Bank>	2/3 = 0.667
<{Park, Bank}>	2/3 = 0.667
<Unknown, School>	2/3 = 0.667
<Unknown, Park>	2/3 = 0.667
<Unknown, Hospital>	2/3 = 0.667
<School, Park>	3/3 = 1.0
<School, Bank>	2/3 = 0.667
<School, Hospital>	2/3 = 0.667
<School, {Park, Bank}>	2/3 = 0.667
<Park, Hospital>	2/3 = 0.667
<Unknown, School, Park>	2/3 = 0.667
<Unknown, School, Hospital>	2/3 = 0.667
<Unknown, Park, Hospital>	2/3 = 0.667
<School, Park, Hospital>	2/3 = 0.667
<Unknown, School, Park, Hospital>	2/3 = 0.667

Abb. 6. Durch PrefixSpan berechnete Muster.

Trajectory	Semantic trajectory
$Trajectory_1$	< School, {Bank, Park}, Restaurant >
$Trajectory_2$	< Unknown, School, {Bank, Park}, Hospital>
$Trajectory_3$	<Unknown, School, Park, Hospital>

Abb. 7. Semantische Trajektorien eines Benutzers.

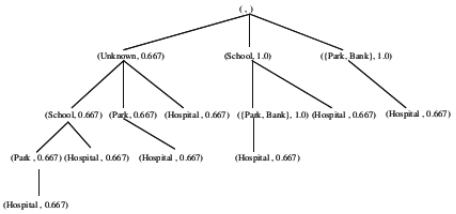


Abb. 8. Resultierender STP-Tree

Nach Erhalt der Ähnlichkeiten können Benutzer geclustert werden. Ein Cluster beinhaltet somit alle Benutzer mit ähnlichen Bewegungsmustern.

3.5 Geographisches Mining

Da man aus den gefundenen semantischen Muster keine direkte geographische Vorhersage treffen kann, muss noch ein weiterer Schritt vor der eigentlichen Vorhersage folgen. Dafür werden Muster aus den Stay Location Sequenzen gewonnen. Diese sogenannten Stay Location Muster werden durch den bereits bekannten PrefixSpan [2] Algorithmus berechnet.
Diesmal werden also häufige geographische Bewegungsmuster einer Person gewonnen. Analog zur Bestimmung der semantischen Muster wird ein Support angegeben. Es werden die zuvor erhaltenen Cluster von Benutzern mit ähnlichen semantischen Bewegunsmustern verwendet, um die Stay Location Sequenzen zu aggregieren. Auf diese aggregierten Cluster von Stay Location Sequenzen wird dann PrefixSpan angewandt. Dadurch werden Muster und alle zugehörigen Teilfolgen gewonnen. Diese werden wiederum in einer Baumstruktur dargestellt, die aus der Berechnung von Semantic Trajectory Pattern Trees bekannt ist. Solche Stay Location Pattern Trees (SLP-Tree) repräsentieren somit eine Menge von Stay Location Muster auf Basis von Stay Location Sequenzen aller Benutzer eines in 3.4 bestimmten Clusters.

3.6 Online Modul

Das Online Modul dient zur eigentlichen Vorhersage. Dazu wird der Semantic Trajectory Pattern Tree sowie der Stay Location Pattern Tree des zugehörigen

Clusters eines Benutzers verwendet. Die verwendeten Cluster wurden in 3.4 bestimmt. Sind aktuelle Trajektorien eines Benutzers gegeben, werden mögliche Pfade im Stay Location Pattern Tree des Clusters bestimmt. Zur Bestimmung eines möglichen Pfades wird ein geographischer Score berechnet, der größer null sein muss. Der geographische Score gibt an, wie sehr sich das aktuelle geographische Verhalten eines Benutzers mit den Mustern im Stay Location Pattern Tree seines Clusters ähneln. Wurde ein möglicher Pfad identifiziert, wird dieser durch eine semantische Sequenz substituiert. Danach wird der Semantic Score mit dem Semantic Trajectory Pattern Tree des Benutzers berechnet. Der Semantic Score misst, wie sehr sich die gefundene semantische Sequenz und die semantischen Bewegungsmustern des Benutzers ähneln.
Der endgültige Score berechnet sich aus dem gewichteten Mittel von semantischen Score sowie geographischen Score.

3.6.1 Geographischer Score Zu Beginn werden alle aktuell aufgezeichneten Trajektorien eines Benutzers in Stay Location Sequenzen umgewandelt. Da es im Vergleich zu den Pfaden des SLP-Trees zu komplex wäre, alle Teilfolgen einer Sequenz zu beachten, wird nur partiell verglichen. Beim Vergleich werden drei einfache Entscheidungsregeln beachtet: 1) Veraltete Bewegungen können die Genauigkeit der Vorhersage verschlechtern, 2) Aktuellere Bewegungen haben einen größeren Einfluss auf die Vorhersage, 3) Ein übereinstimmender Pfad mit einem hohen Support und einer höheren Länge macht die Vorhersage sicherer.
Zur effizienten Bestimmung des geographischen Scores wird einem angepassten DFS-Algortihmus [4] der Semantic Trajectory Pattern Tree des Clusters und die aktuelle Stay Location Sequenz des Benutzers übergeben. Dieser findet alle möglichen Pfade, die in Frage kommen und berechnet währendessen deren geographischen Score.

3.6.2 Semantischer Score Da bisher nur geographische Informationen verwendet wurden und diese für eine Vorhersage nach Ying nicht ausreichend sind, werden in diesem Abschnitt die semantischen Eigenschaften der Pfade mit einbezogen. Dafür werden alle zuvor gefundenen Pfade in semantische Pfade umgewandelt. Danach wird jeder semantische Pfad sowie der Semantic Trajectory Pattern Tree des Benutzers ein weiteres mal dem angepassten DFS-Algorithmus übergeben. Dieser berechntet den Semantic Score für einen semantischen Pfad.
Letzlich wird der Score eines Pfades über das gewichtete Mittel von geographischem sowie semantischem Score berechnet. Der mögliche Pfad mit dem höchsten Score wird betrachtet und es wird der Kindsknoten dieses Pfades im Semantic Trajectory Pattern Tree als nächster Standort vorhergesagt. Falls der gewählte Pfad kein Kind hat, wird der Pfad mit dem zweit höchsten Score gewählt und dessen Kind vorhergesagt. Das wird wiederholt bis ein Pfad gefunden wurde, der ein Kindsknoten besitzt.

4 Fazit

In dieser Arbeit wurden zwei unterschiedliche Ansätze zu Standortvorhersagen durch Markov Modelle sowie semantischen Trajektorien vorgestellt, wobei der Fokus auf mögliche Probleme in den jeweiligen Verfahren lag. Hierbei wurde aufgezeigt, dass ein komplexeres Verfahren wie das von Ying zwar Probleme eines rein geographischen Verfahrens wie das von Ashbrook löst, jedoch auch weitere verursacht. Beispielhaft wurde das Generalisierungsproblem in Yings Ansatz beschrieben, welches durch das Clustern mit P-DBSCAN entsteht. Des Weiteren wurde auf mögliche Probleme bei der Wahl von Paramatern hingewiesen. Die Wahl dieser musste häufig auf den verwendeten Datensatz des beschriebenen Verfahrens angepasst sein. Das durch Ashbrook beschriebene Verfahren basiert auf Datensätze, welche durch ihre Art der Aufzeichnung bestimmte Eigenschaften erfüllen. Diese sind jedoch unter realen Bedingungen nicht notwendigerweise gegeben, da die Art der Aufzeichnung nicht garantiert ist.
Yings Ansatz arbeitet hingegen auf Datensätze von denen keine Vorkenntnisse existieren müssen. Zudem ist das Verfahren aufgrund der Aufteilung in Offline und Online Modul praktikabler, da effiziente Auswertungen und damit Vorhersagen möglich sind.

Literatur

1. Ashbrook, D., Starner, T.: Using gps to learn significant locations and predict movement across multiple users. Personal and Ubiquitous Computing 7(5), 275–286 (2003)
2. Han, J., Pei, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., Hsu, M.: Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In: proceedings of the 17th international conference on data engineering. pp. 215–224 (2001)
3. Kisilevich, S., Mansmann, F., Keim, D.: P-dbscan: a density based clustering algorithm for exploration and analysis of attractive areas using collections of geo-tagged photos. In: Proceedings of the 1st international conference and exhibition on computing for geospatial research & application. p. 38. ACM (2010)
4. Ying, J.J.C., Lee, W.C., Weng, T.C., Tseng, V.S.: Semantic trajectory mining for location prediction. In: Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. pp. 34–43. ACM (2011)
5. Ying, J.J.C., Lu, E.H.C., Lee, W.C., Weng, T.C., Tseng, V.S.: Mining user similarity from semantic trajectories. In: Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks. pp. 19–26. ACM (2010)
6. Zheng, Y., Zhang, L., Ma, Z., Xie, X., Ma, W.Y.: Recommending friends and locations based on individual location history. ACM Transactions on the Web (TWEB) 5(1), 5 (2011)

Vergleich der Standortprognose auf Basis von geografischen Trajektorien und semantisch annotierten Trajektorien

Jasper Zimmer*

Betreuer: Antonios Karatzoglou†

Karlsruher Institut für Technologie (KIT)
Pervasive Computing Systems – TECO
*uqdev@student.kit.edu
†antonius@teco.edu

Zusammenfassung. Der Fokus dieser Seminararbeit liegt auf dem Vergleich von zwei unterschiedlichen Ansätzen zur Standortprognose. Dabei verwendet das erste Verfahren ausschließlich geografische Trajektorien und der Zweite zusätzlich semantische Annotationen und persönliche Informationen über den Benutzer zum erstellen einer Prognose. Um den Vergleich durchführen zu können, werden beide Verfahren beschrieben und implementiert. Anschließend findet ein Vergleich statt, welcher die gewonnenen Erkenntnisse veranschaulicht.

Schlüsselwörter: geografische Standortprognose, semantische Standortprognose, Trajektorie, Evidenztheorie, Dempster und Shafer

1 Einleitung

Die Verbreitung mobiler Endgeräte ist in den vergangenen Jahren stark angestiegen. Damit einhergehend gehören Technologien wie WLAN, Bluetooth, UMTS und GPS heute zur Standardausstattung. Mit Hilfe dieser Drahtlosen Infrastruktur ist es möglich die Position von Personen und anderen Objekten wie z.B. Fahrzeugen auf wenige Meter genau zu bestimmen. Durch das Sammeln von Standortdaten können sogenannte *geografische Trajektorien* (Abb. 1) erstellt werden, welche den individuellen Standortverlauf einer Person oder eines Objektes repräsentieren.
Aus der Möglichkeit der Standortbestimmung sind Standortbezogene Dienste entstanden, welche dem Benutzer oder dem Betreiber einen Mehrwert bieten. Beispielsweise Dienste zur Stauumfahrung, für Ortsbezogene Werbung oder auch Spiele wie Geocaching und Ingress. Dabei informiert das mobile Endgerät die entsprechende Gegenstelle regelmäßig über den aktuellen Standort. Mit Hilfe der Standortprognose kann die Effizienz dieser Anwendungen gesteigert oder Störungen wie z.B. bei der Standortübertragung vorgebeugt werden.
In dieser Arbeit werden zwei verschiedene Ansätze zur Standortprognose miteinander verglichen. Der Erste [2] verwendet alle geografische Trajektorien die

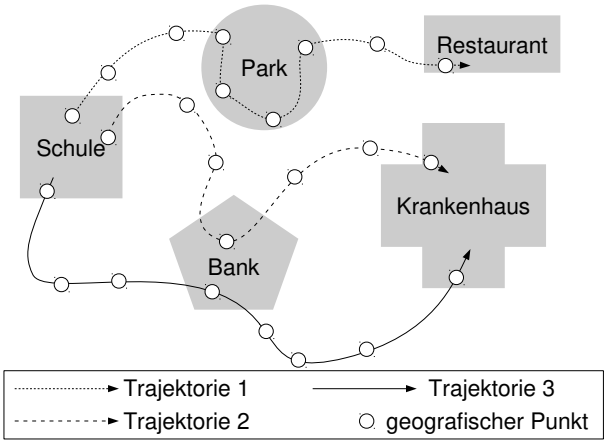


Abb. 1. Beispiel für geografische Trajektorien

zur Verfügung stehen, und nicht nur die des betreffenden Objektes, um daraus Klassifikatoren für frequentierte Gebiete zu erlernen. Dabei wird die Annahme gemacht, dass Menschen dazu neigen ähnliche oder gleiche Gewohnheiten zu haben und somit die selben Wege nehmen um Ziele zu erreichen. Ein Beispiel für dieses Verhalten ist der tägliche Weg zur Arbeit, um die gleiche Zeit mit einem bestimmten Verkehrsmittel. Stehen genug Daten in Form von geografischen Trajektorien zur Verfügung, so können daraus typische Verhaltensweisen ermittelt werden. Diese werden dann bei der Prognose verwendet, um den nächsten bzw. wahrscheinlichsten Standort zu bestimmen.
Der Zweite Ansatz [4] verfolgt die Idee, Prognosen anhand von *semantisch annotierten Trajektorien* (Abb. 2) zu erstellen. Eine solche Trajektorie besteht aus einer Folge von Orten. Diese sind mit sogenannten *semantischen Tags* versehen, welche mit dem Ort verbundene Tätigkeiten implizieren. In Abb. 2 entsprechen Schule, Park, Bank, Restaurant und Krankenhaus diesen semantischen Tags. Die beiden Trajektorien *Trajektorie 2* und *Trajektorie 3* können beide durch die Folge <Schule, Bank, Krankenhaus> beschrieben werden. Dies zeigt dass das semantische Verhalten zweier Personen das gleiche sein kann, auch wenn sich die zugehörigen geografischen Trajektorien voneinander unterscheiden. Somit kann eine Standortprognose an Stelle von einer geografischen Trajektorie auch mit einer semantischen Trajektorie durchgeführt werden. Somit kann bei der Prognose eine geografischen Unabhängigkeit erreicht werden.

2 Verwendeter Datensatz und seine Aufbereitung

Als gemeinsame Datenbasis für den Vergleich zwischen den beiden Ansätzen wurde der Reality Mining Datensatz [3] vom MIT gewählt. Dieser enthält Trajektorien von 94 Personen, welche über einen Zeitraum von neun Monaten in

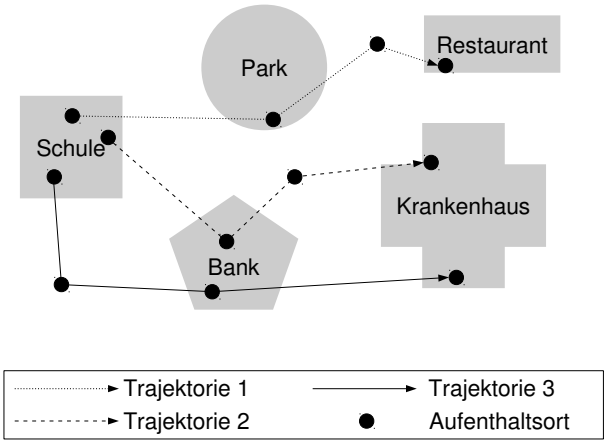


Abb. 2. Beispiel für semantische Trajektorien

den Jahren 2004 und 2005 aufgezeichnet wurden. Dabei variiert die Länge sowie die Vollständigkeit zwischen den einzelnen Teilnehmern der Studie stark. Eine Aufbereitung der vorliegenden Daten ist notwendig, um einen Vergleich durchführen zu können.
Die enthaltenen Trajektorien sind *geografische Trajektorien* welche anstatt GPS-Koordinaten die Position als Aufenthaltsbereich (engl. Location Area Code kurz LAC) und Mobilfunkzellenidentifikation (engl. Cell ID kurz CID) innerhalb des Mobilfunknetzes beinhalten. Diese müssen im Verlauf der Aufbereitung in GPS-Koordinaten umgewandelt werden. Zusätzlich beinhaltet der Datensatz Annotationen zu Orten in der Trajektorie, welche die Teilnehmer im Rahmen der Studie angefertigt haben. Diese sind allerdings sehr inkonsequent gewählt und unterscheiden sich für die selben Orte zwischen den Teilnehmern stark. Deshalb müssen neue Annotationen erzeugt bzw. gefunden werden.

2.1 Datensatz Konvertierung

Da der Datensatz als MATLAB-Datei vorliegt wurde zunächst ein Skript [1] erstellt, um die Daten in das CSV-Dateiformat zu überführen. Dabei wird für jeden Teilnehmer aus dem Datensatz ein Ordner mit der zugehörigen ID als Bezeichnung angelegt, welcher die Personenspezifischen Daten enthält.

2.2 Teilnehmer Profile erstellen

Im nächsten Schritt wurden Profile für alle Teilnehmer der Studie erstellt, welche die Trajektorie, Annotationen der Teilnehmer zu Orten, die Forschungsgruppe

[1] `https://github.com/JasperZ/semantic-location-prediction/blob/master/reality_mining_to_csv.m`

und Informationen über die Nachbarschaft enthalten. Diese Profile werden im JSON-Dateiformat, wie in Abb. 3 zu sehen, gespeichert.

Abb. 3. Beispiel für Teilnehmer-Profil

2.3 Aufenthaltsorte erkennen

Ein Aufenthaltsort ist ein Ort an welchem sich ein Teilnehmer für eine Dauer länger als Θ_t aufhält, ohne ihn zwischenzeitlich zu verlassen. Hierfür wurde Algorithmus 1 implementiert, welcher eine angepasste Variante von [5](Fig. 7) ist. Dieser ermittelt anhand des *LAC* und der *CID* ob der Abdeckungsbereich der Zelle für eine Dauer von mindestens Θ_t nicht verlassen wurde. Tritt dieser Fall ein, wird der Folge von Aufenthaltsorten ein entsprechender Eintrag hinzugefügt, welcher neben des *LAC* und der *CID* auch die zwei Zeitstempel *startTimestamp* und *endTimestamp* besitzt um die Verweildauer innerhalb der Zelle festzuhalten. Anschließend wird diese Folge von Aufenthaltsorten dem Teilnehmer-Profil als Attribut *stayLocs* hinzugefügt (Abb. 4). In diesem Schritt reduziert sich die Anzahl der voneinander verschiedenen Mobilfunkzellen von ca. 32000 auf 3000!


```
"stayLocs": [
  {
    "startTimestamp": 1090854852000,
    "endTimestamp": 1090862489000,
    "locationAreaCode": 24127,
    "cellId": 111
  },
  {
    "startTimestamp": 1090867560000,
    "endTimestamp": 1090870702000,
    "locationAreaCode": 5123,
    "cellId": 48731
  },
  ...
```


Abb. 4. Beispiel für Trajektorie von Aufenthaltsorten

2.4 Ermitteln der GPS-Koordinaten zu den Aufenthaltsorten

Bis zu diesem Schritt beinhalten die Trajektorien noch keine geografische Informationen in Form von GPS-Koordinaten. Diese können mit Hilfe der Google Maps Geolocation API [2] oder OpenCellID[3] ermittelt werden. Dabei ist zu beachten dass für die Google API ein Entwicklerkonto erforderlich ist, und die Anzahl der Anfragen pro Tag auf 2500 limitiert ist. Bei der Verwendung von OpenCellID ist ein solches Limit nicht gegeben. Außerdem gibt es die Möglichkeit den aktuellen Stand der Datenbank für den offline Gebrauch herunterzuladen. Jedoch ist es aufgrund des Alters des Reality Mining Datensatzes nicht möglich

[2] `https://developers.google.com/maps/documentation/geolocation/intro`
[3] `http://opencellid.org/`

Algorithm 1 Erkennen der Aufenthaltsorte in einer Trajektorie

Eingabe: Eine Trajektorie $Traj$ und ein Zeitspannen Schwellwert Θ_t
Ausgabe: Eine Liste von Aufenthaltsorten $S = \{s\}$

```
1: i ← 0;
2: j ← 0;
3: a ← null;
4: b ← null;
5:
6: while i < |Traj| and j < |Traj| do
7:     j ← i + 1;
8:     a ← Traj[i];
9:
10:    while j < |Traj| do
11:        b ← Traj[j];
12:
13:        if a.LAC ≠ b.LAC or a.CID ≠ b.CID then
14:            Δt ← b.Timestamp − a.Timestamp;
15:
16:            if Δt ≥ Θ_t then
17:                s ← StayLocation;
18:                s.startTimestamp ← a.Timestamp;
19:                s.endTimestamp ← b.Timestamp;
20:                s.LAC ← a.LAC;
21:                s.CID ← a.CID;
22:                S.insert(s);
23:            end if
24:
25:            i ← j;
26:            break;
27:        end if
28:
29:        j ← j + 1;
30:    end while
31: end while
32:
33: if a ≠ null and b ≠ null and a.LAC = b.LAC and a.CID = b.CID then
34:    Δt ← b.Timestamp − a.Timestamp;
35:
36:    if Δt ≥ Θ_t then
37:        s ← StayLocation;
38:        s.startTimestamp ← a.Timestamp;
39:        s.endTimestamp ← b.Timestamp;
40:        s.LAC ← a.LAC;
41:        s.CID ← a.CID;
42:        S.insert(s);
43:    end if
44: end if
45:
46: return S;
```

für alle Aufenthaltsorte die korrekten GPS-Koordinaten zu ermitteln, da sich die Netzstruktur der Mobilfunkbetreiber in diesem Zeitraum (ca. 11 Jahre) zu stark gewandelt hat. Dies hat zur Folge, das von den ca. 3000 verschiedenen Zellen nur die Hälfte mit Koordinaten versehen werden konnte. Außerdem ist anzumerken, dass die ermittelten Koordinaten näherungsweise dem Zellmittelpunkt entsprechen, und nicht unbedingt der tatsächlichen Position des Teilnehmers. Wurden für einen Aufenthaltsort Koordinaten gefunden, so werden diese dem Teilnehmer-Profil als die Attribute *lat* und *lng* in der Trajektorie von Aufenthaltsorten (*stayLocs*) hinzugefügt (Abb. 5).


```
"stayLocs": [
  {
    "startTimestamp": 1096582220000,
    "endTimestamp": 1096584570000,
    "locationAreaCode": 32146,
    "cellId": 10833,
    "lat": 40.756982,
    "lng": -73.995733,
    "accuracy": 105.0,
    "userLabel": "ny"
  },
  {
    "startTimestamp": 1096591659000,
    "endTimestamp": 1096598860000,
    "locationAreaCode": 32146,
    "cellId": 10623,
    "lat": 40.755565,
    "lng": -73.995598,
    "accuracy": 75.0,
    "userLabel": "Ny"
  },
  ...
```


Abb. 5. Beispiel für Trajektorie von Aufenthaltsorten mit GPS-Koordinaten

2.5 Aufenthaltsorte mit semantischen Informationen anreichern

Alle Aufenthaltsorte, welchen GPS-Koordinaten zugeordnet werden konnten, werden nun mit semantischen Informationen versehen. Diese stammen aus der Venue Service API von Foursquare[4]. Auch hier ist die Anzahl der Anfragen limitiert, allerdings ist das Limit mit 5000 Anfragen pro Stunde deutlich großzügiger ausgelegt als bei der zuvor verwendeten Google-API. Die API ermöglicht es Anfragen für Koordinaten zu machen, welche mit einer Liste von Orten in deren

[4] `https://developer.foursquare.com/overview/venues.html`

Umgebung beantwortet wird. Teil dieser Antwort ist die Kategorie in der ein Ort eingeordnet werden kann, wie z.B. Hotel, Food, Movie Theater und viele mehr. Diese Kategorien werden als semantische Annotation für die Aufenthaltsorte verwendet.

2.6 Profile der Teilnehmer in Tagesprofile aufteilen

Der Letzte Schritt in der Datenaufbereitung besteht darin, die Profile der Teilnehmer zu unterteilen, sodass diese jeweils den Zeitraum von einem Tag abdecken. Somit sind anschließend ca. 1450 Tagesprofile vorhanden, welche eine Trajektorie von Aufenthaltsorten beinhalten. Wobei jeder Aufenthaltsort die Mobilfunkzelle, GPS-Koordinaten, Kategorie und den Zeitpunkt der Ankunft sowie des Verlassens beinhaltet.

3 Standortprognose auf Basis von T-Pattern Mining

Der erste Ansatz zur Standortprognose welcher implementiert wurde, stammt von Monreale u. a. [2]. Dabei wird die Annahme gemacht, das verschiedene Menschen die gleichen oder zumindest ähnlichen Bewegungsabläufe haben. Dadurch wird es möglich, die Trajektorien aller Personen zu betrachten und daraus gemeinsamem Bewegungsmuster (T-Pattern) zu bestimmen. Das Verfahren umfasst drei Schritte um zu einer Prognose zu gelangen.

1. Bewegungsmuster extrahieren → T-Pattern
2. Entscheidungsbaum aufbauen → T-Pattern Tree
3. Prognose unter Zuhilfenahme des Entscheidungsbaums

3.1 Bewegungsmuster extrahieren - T-Pattern Mining

Die Bewegungsmuster werden aus allen Trajektorien extrahiert, welche zeitlich vor der Prognose aufgezeichnet wurden. Dabei werden aufgrund der Annahme, das Menschen sich gleich oder ähnlich verhalten, nicht nur die Standortverläufe einer einzelnen Person betrachtet, sondern auch all diejenigen, welche nicht Gegenstand der aktuellen Prognose sind. Dafür wird sogenanntes T-Pattern Mining betrieben, welches von Giannotti u. a. [1] beschrieben wird. Dabei dienen zeitlich annotierte Sequenzen der Form:

$$T = s_0 \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_n} s_n \tag{1}$$

als Eingabe für das Verfahren. Diese können auch als $T = (S, A)$ mit der Sequenz $S = \langle s_0, s_1, \ldots, s_n \rangle$ und den zeitlichen Annotationen $A = \langle \alpha_1, \alpha_2, \ldots, \alpha_n \rangle$ geschrieben werden. Im Fall von Trajektorien entspricht die Sequenz S den Aufenthaltsorten und die zeitlichen Annotationen A den Übergangszeiten zwischen diesen. Ein Beispiel dafür ist in Abb. 6 zu sehen. Im Verlauf des T-Pattern Mining werden alle Sequenzen extrahiert, welche Teilsequenzen der Eingabe sind. Dabei muss die Zeitliche Ordnung der auftretenden Orte übereinstimmen. Allerdings

$$T_1 = O_1 \xrightarrow{10min} O_2 \xrightarrow{5min} O_3$$
$$T_2 = O_1 \xrightarrow{8min} O_2 \xrightarrow{12min} O_1$$
$$T_3 = O_2 \xrightarrow{3min} O_1$$
$$T_4 = O_1 \xrightarrow{7min} O_2 \xrightarrow{16min} O_1$$

Abb. 6. Beispiel für zeitlich annotierte Sequenzen mit den Orten O_1, O_2, O_3

#	T-Pattern	Support
1	O_1	1.0
2	O_2	1.0
3	O_3	0.25
4	$O_1 \xrightarrow{[7,10]} O_2$	0.75
5	$O_1 \xrightarrow{[15,15]} O_3$	0.25
6	$O_2 \xrightarrow{[3,16]} O_1$	0.75
7	$O_2 \xrightarrow{[5,5]} O_3$	0.25
8	$O_1 \xrightarrow{[7,10]} O_2 \xrightarrow{[3,12]} O_1$	0.5
9	$O_1 \xrightarrow{[7,10]} O_2 \xrightarrow{[5,5]} O_3$	0.25

Abb. 7. Beispiel für zeitlich annotierte Sequenzen mit den Orten O_1, O_2, O_3

kann eine Sequenz Orte enthalten welche in der Teilsequenz nicht vorkommen. So sind die Bewegungsmuster welche aus Abb. 6 resultieren in Abb. 7 zu sehen. Aus den Übergangszeiten sind nun Intervallangaben geworden und der Support eines T-Patterns gibt an in wie vielen Eingabesequenzen das Muster auftaucht. Das T-Pattern O_1 kommt also in den Sequenzen T_1 bis T_4 ($Support = \frac{4}{4} = 1.0$) vor und $O_1 \xrightarrow{[15,15]} O_3$ dementsprechend nur in T_1 ($Support = \frac{4}{1} = 0.25$).

3.2 Entscheidungsbaum aufbauen - T-Pattern Tree

Alle zuvor ermittelten T-Pattern werden einem sogenannten T-Pattern Tree $PT = (N, E, Root(PT))$ hinzugefügt. Dabei handelt es sich um einen Präfixbaum. Er stellt eine effiziente Möglichkeit dar um zu überprüfen ob eine gegebene Sequenz Teil der gefundenen T-Pattern ist oder nicht. Dabei handelt es sich bei N um ein endliche Menge von Knoten, E ist eine Menge von Kanten und $Root(PT) \in N$ ist ein fiktiver Knoten, welcher die Wurzel des Baumes repräsentiert. Jede Kante aus E ist mit einem Intervall *int* versehen, welches den Übergangszeitraum zwischen zwei Knoten angibt. Die Kante $(u, v, int) \in E$ enthält also das Intervall für den Übergang von u nach v. Das Intervall *int* wird in der Form $[time_{min}, time_{max}]$ angegeben. Alle Knoten $v \in N$ beinhalten die Informationen $\langle id, location, support, children \rangle$. Dabei entspricht *location* einem Aufenthaltsort(Tupel bestehend aus LAC und CID), *support* dem Support des kürzesten T-Patterns welches v erreicht und *children* einer Liste von Knoten

welche von v aus erreichbar sind. Setzt man das Beispiel aus dem Abschnitt T-Pattern Mining fort entsteht der T-Pattern Tree, welcher in Abb. 8 zu sehen ist.

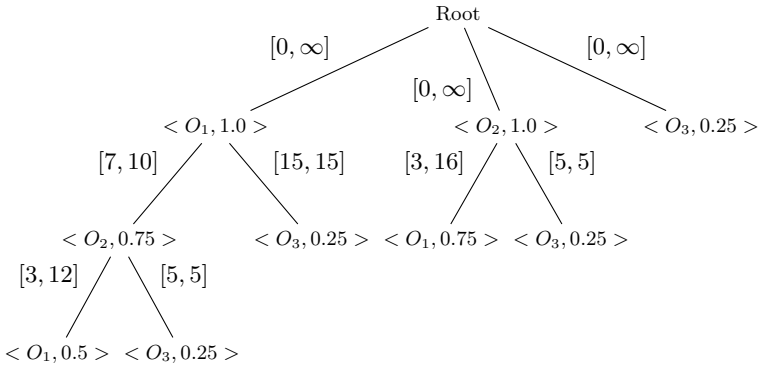


Abb. 8. Beispiel für einen T-Pattern Tree

3.3 Prognose

Eine Prognose wird anhand des aktuellen Aufenthaltsort $O_{current}$ und dessen Vorhergehenden $O_0 - O_{current-1}$ gemacht. Somit erhält man die zeitlich annotierte Sequenz $T_p = O_0 \xrightarrow{\alpha_1} O_1 \xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_{current}} O_{current}$. Im T-Pattern Tree wird nun der Pfad bestimmt welcher sich mit T_p überdeckt. Dabei müssen sowohl die Knoten mit den Aufenthaltsorten aus T_p übereinstimmen sowie die jeweiligen Übergangszeiten in den entsprechenden Intervallen enthalten sein. Existiert ein solcher Pfad, dann werden alle Kinder des Letzten Knotens betrachtet. Sie kommen als Kandidaten für den nächsten Aufenthaltsort infrage. Bei n Kindern ergeben sich somit n potentielle Pfade, wobei sich diese nur im letzten Knoten durch den Aufenthaltsort $O_{next_i} \in O_{current}.children, 1 \leq i \leq n$ unterscheiden.

$$P_1 =< O_0, S_0 >, \ldots, < O_{current}, S_{current} >, < O_{next_1}, S_0 >$$
$$\vdots$$
$$P_n =< O_0, S_0 >, \ldots, < O_{current}, S_{current} >, < O_{next_n}, S_0 >$$

Um unter diesen Kandidaten einen Geeigneten zu bestimmen, wird für jeden potentiellen Pfad $P_1, \ldots, P_n$ ein sogenannter Score errechnet. Dieser errechnet sich für den jeweiligen Pfad aus den Support-Werten, der enthaltenen Knoten. Dabei wurden drei verschiedene Funktionen betrachtet, welche sich bei der Vorhersage unterschiedlich verhalten:

- $avgScore(P) = \frac{\sum_{i=1}^{n} p_i.sup}{n}$

- $sumScore(P) = \sum_{i=1}^{n} p_i.sup$
- $maxScore(P) = max\{p_1.sup, ..., p_n.sup\}$

Unabhängig von der verwendeten Score-Funktion wird der Pfad mit dem höchsten Score $P_m = max\{score(P_1), \ldots, score(P_n)\}$ für die Prognose verwendet, sodass O_{next_m} der nächste vorhergesagte Aufenthaltsort ist. Zusätzlich wird ein Schwellwert th_{score} eingeführt, welcher nur Pfade zulässt, deren *Score* höher als th_{score} sind. Dadurch wird die Prognose robuster gegenüber Fehleinschätzungen. Gleichzeitig werden jedoch auch weniger Vorhersagen durchgeführt, da es Fälle geben kann, in welchen keiner der möglichen Pfade diesen Wert überschreitet. Weiter ist zu beachten, dass für die Score-Funktion *sumScore* kein vernünftiger Wert für th_{score} festgelegt werden kann, da dieser maßgeblich von der Länge des Pfades abhängt.

4 Standortprognose auf Basis von semantisch annotierten Trajektorien

Die Idee, Standortprognosen unter Zuhilfenahme semantischer Informationen zu verbessern wurde erstmals 2005 von Samaan und Karmouch [4] beschrieben. Dabei werden Informationen über Interessen und bevorstehende Aufgaben, sowie Wissen um bevorstehende Termine aus Kalendereinträge vereint, mit dem Ziel die Prognose in ihrer Genauigkeit zu verbessern. Um das Wissen aus den verschiedenen Quellen zu kombinieren und zu einer Gesamtaussage zusammenzusetzen, kommt die Evidenztheorie von Dempster und Shafer zum Einsatz.

4.1 Evidenztheorie von Dempster und Shafer

Die Theorie besagt, wenn es für den Fall A eine Evidenz e vorliegt und es keine andere Evidenz gibt, die dagegen spricht, so wird der Intervall $[1 - e, 1]$ als Unsicherheitsintervall betrachtet und nicht als Evidenz gegen A.

4.2 Semantische Informationen

Aufgrund der Tatsache, dass der verwendete Datensatz keine semantischen Informationen enthält, werden ersatzweise die Kategorien der Aufenthaltsorte als mögliche Interessen gedeutet. Um Aussagen über die persönlichen Interessen der Teilnehmer für die Prognose zu erhalten, wurde statistisch ermittelt wie oft Aufenthaltsorte in den verfügbaren Kategorien besucht wurden.

4.3 Adjazenzmatrix

In der Adjazenzmatrix wird festgehalten, welche Orte direkt erreichbar sind. Diese wurde ebenfalls aus dem Datensatz gewonnen, indem aus jeder verfügbaren Trajektorie jeweils alle aufeinander folgenden Aufenthaltsorte als direkt erreichbar in der Matrix eingetragen wurden.

4.4 Prognose

Für die eigentliche Prognose wird zunächst ein Frame of Discrement Θ gebildet. Dieser enthält alle Orte, welche sich direkt vom aktuellen Aufenthaltsort erreichen lassen. Dafür wird die Adjazenzmatrix verwendet. Danach folgt die Evidenz Gewinnung, wobei für jede verfügbare Evidenz eine Hypothese aufgestellt wird. Diese enthält alle Orte aus Θ, die mit der jeweiligen Evidenz in Verbindung gebracht werden können. Somit entstehen je nach verfügbarem Wissen verschiedene Hypothesen, welche sich gegenseitig unterstützen oder auch widersprechen (Beispiel Abb. 9). Um daraus eine Gesamtaussage zusammenzusetzen, werden

Frame of discrement: $\Theta = \{O_1, O_2, O_3, O_4, O_5\}$

Evidenzen und Hypothesen:
(1) **E1: Benutzer besucht den Kurs Informatik**
$\{O_1, O_2, O_3\}$ 0.2
(2) **E2: Aufgabe: Hausaufgaben bis 15 Uhr erledigen**
$\{O_3\}$ 0.5
(3) **E3: Aufgabe: Buch vor 17 Uhr in Bibliothek zurückbringen**
$\{O_4, O_5\}$ 0.3
(4) **E4: Termin: Kollege um 13:30 Uhr im Labor O_3 treffen**
$\{O_2, O_3\}$ 0.1

Abb. 9. Beispiel Evidenzen E1 bis E4 und die dazugehörigen Hypothesen

diese mit der *Dempster Rule of Combination*(2) kombiniert.

$$m_{E_i} \oplus m_{E_j}(C) = \frac{\sum_{X \cap Y = C} m_{E_i}(X) m_{E_j}(Y)}{1 - K}, K = \sum_{X \cap Y = \emptyset} m_{E_i}(X) m_{E_j}(Y) \quad (2)$$

Als Ergebnis erhält man für jede Hypothese ein sogenanntes *Belief* Maß, welches das Vertrauen an diese angibt (Beispiel Abb. 10). Somit wird für die Prognose des nächsten Aufenthaltsortes die Hypothese mit dem höchsten $Belief$-Wert ausgewählt. Um die Prognose in ihrer Genauigkeit zu verbessern kann ein Schwellwert th_{belief} verwendet werden, welcher nur Hypothesen zulässt deren Belief th_{belief} überschreiten.

Hypothese	Belief
$\{O_3\}$	0.433 ⇐
$\{O_2, O_3\}$	0.043
$\{O_4, O_5\}$	0.077
$\{O_1, O_2, O_3\}$	0.133
$\{O_1, O_2, O_3, O_4, O_5\}$	0.311

Abb. 10. Beispiel für kombinierte Evidenzen (Fortsetzung von Abb. 9)

5 Ergebnisse

5.1 Vorgehen

Um die Leistung der beiden Verfahren miteinander vergleichen zu können, sind gemeinsame Maße und Indikatoren nötig. Es wurden die folgenden vier Maße gewählt:

- Anzahl der richtigen Prognosen: P_c
- Anzahl der falschen Prognosen: P_w
- Anzahl der gemachten Prognosen: $P_m = P_c + P_w$
- Anzahl der versuchten Prognosen: P_t

Der Unterscheidung zwischen P_m und P_t kommt daher, dass es bei Prognosen zu Fällen kommen kann, in denen mehrere verschiedene Orte die gleiche Wahrscheinlichkeit besitzen oder diese so gering ist, dass sie den entsprechenden Threshold nicht überschreitet.
Um die Leistung zu bestimmen, wurden die Metriken *Prediction Rate* und *Accuracy* verwendet:

- $Prediction\ Rate = \frac{P_m}{P_t}$
- $Accuracy = \frac{P_c}{P_m}$

Die *Prediction Rate* gibt das Verhältnis der tatsächlich gemachten zu den versuchten Prognosen an und *Accuracy* das Verhältnis zwischen den richtigen und falschen Vorhersagen. Der aufbereitete Datensatz wurde in die Gruppen Trainingsdaten und Testdaten aufgeteilt. Die Testdaten umfassen 90% des Datensatzes und die Testdaten die restlichen 10%.

5.2 Ansatz: T-Pattern Mining

Die Implementierung erreicht eine maximale *Accuracy* von etwas mehr als 70%. Dabei ist zu beachten dass dieser Wert nur erreicht wird, indem der Threshold th_{score} entsprechend gewählt $(0.25 - 3.0)$ wird. Wie in Abb. 12 zu sehen ist steigt die *Accuracy* bei ebenfalls steigendem th_{score}. Eine Folge davon ist, dass die *Prediction Rate* im Gegenzug sinkt (Abb. 12). Das kommt daher, dass durch die Erhöhung des Schwellwerts alle Prognosen verworfen werden, welche diesen unterschreiten. Die gemachten Beobachtungen decken sich mit denen von Monreale u. a. [2], welche das gleiche Verhalten aufweisen.

5.3 Ansatz: Semantisch annotierte Trajektorien

Mit der Implementierung auf Basis des semantischen Ansatzes wird eine *Accuracy* von bis zu 90% erreicht. Wie schon zuvor bei der Prognose mittels T-Pattern Mining, kann ein solch hoher Wert nur dann erreicht werden, wenn auch der Schwellwert th_{belief} entsprechend hoch gewählt wird. Auch das Verhalten im Bezug auf die *Prediction Rate* ist gleich. Allerdings muss beachtet werden, das

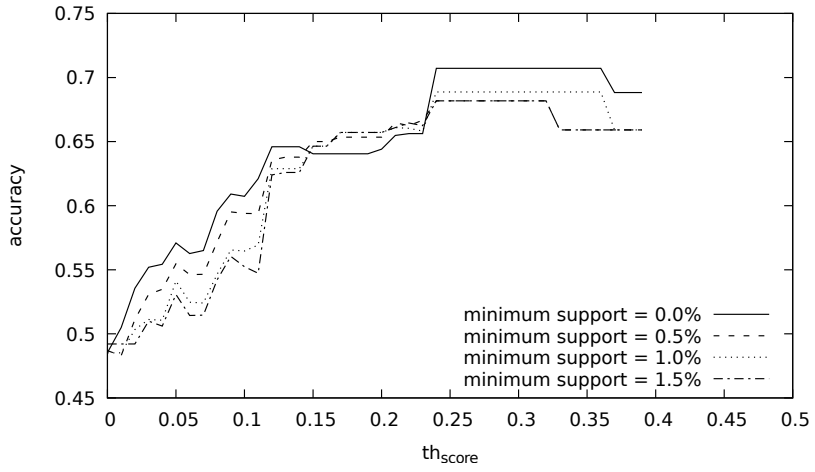


Abb. 11. T-Pattern: Accuracy vs. th_{score}

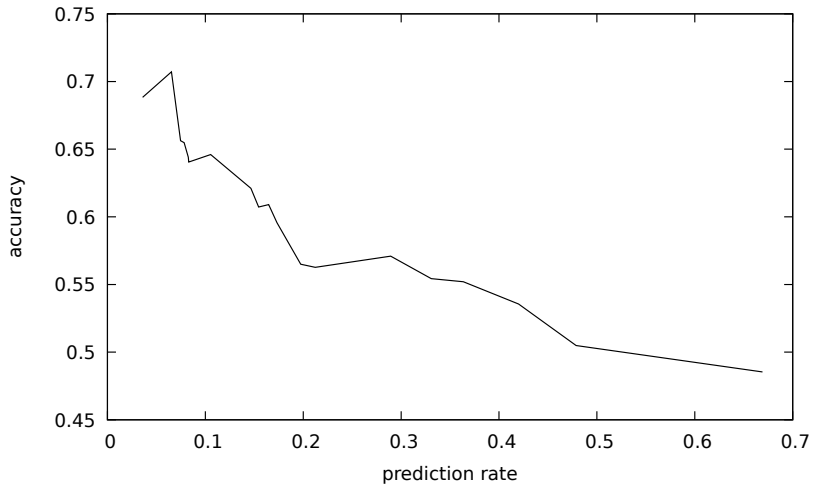


Abb. 12. T-Pattern: Accuracy vs. Prediction Rate

aufgrund des verwendeten Datensatzes sehr wenige persönliche Informationen über die Teilnehmer bekannt sind. Währen z.B. ein Kalender oder der Stundenplan verfügbar, würden diese vermutlich die Genauigkeit steigern, da sie weitere Evidenzen liefern welche mit Hilfe der Evidenztheorie von Dempster und Shafer ausgewertet werden können.

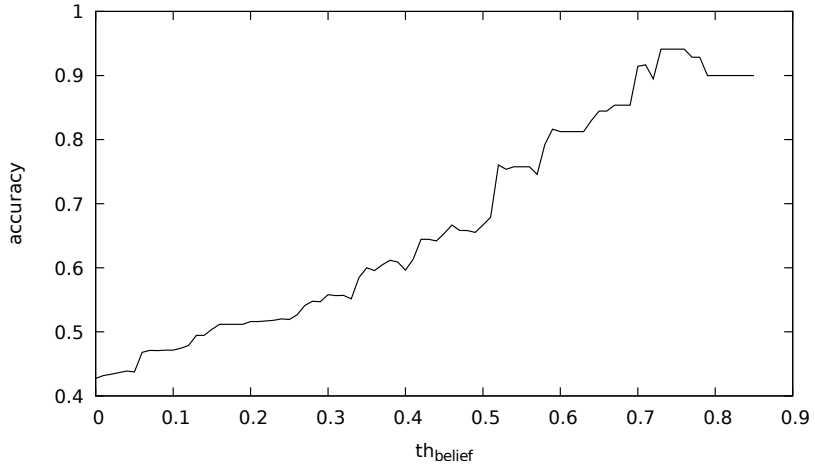


Abb. 13. Semantik: Accuracy vs. th_{belief}

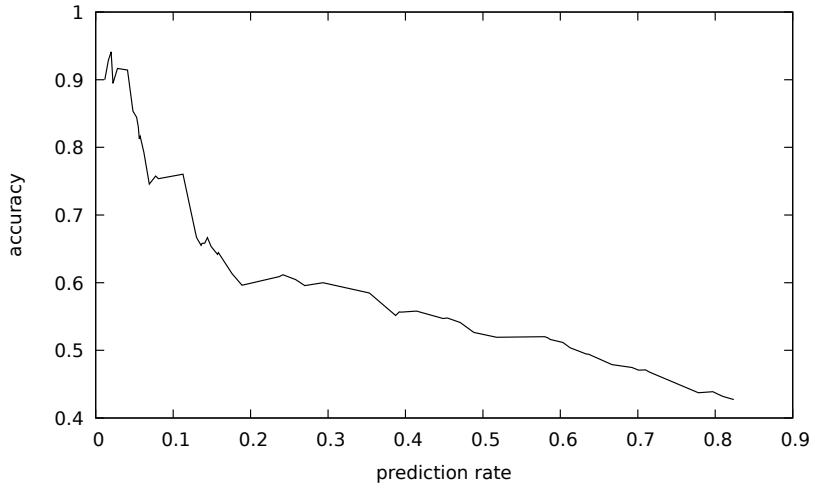


Abb. 14. Semantik: Accuracy vs. Prediction Rate

6 Bemerkungen

Abschließend kann man sagen dass die Prognose auf Basis semantisch annotierten Trajektorien im durchgeführten Vergleich bessere Ergebnisse liefert, als die Vorhersage auf Basis von T-Pattern Mining. Dies wird gut sichtbar, wenn man die *Accuracy*-Verläufe der beiden Ansätze vergleicht (Abb. 15). Außerdem muss berücksichtigt werden dass bei der Implementierung an manchen Stellen Kompromisse geschlossen werden mussten, sodass die Vorgehensweise, wie sie in

dieser Arbeit beschrieben wird, zum Teil von der aus den ursprünglichen Papern abweicht. Die Implementierung ist unter folgender Adresse zu finden:
https://github.com/JasperZ/semantic-location-prediction

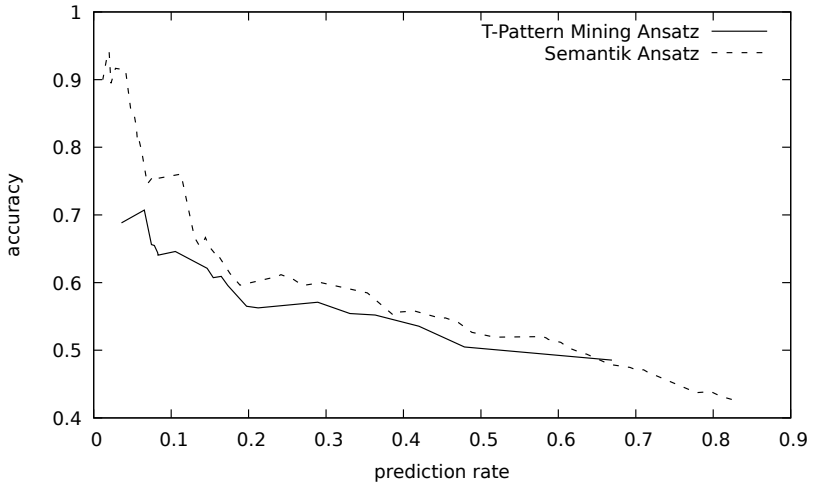


Abb. 15. Vergleich der *Accuracy* der beiden implementierten Ansätze

Literatur

1. Giannotti, F., Nanni, M., Pinelli, F., Pedreschi, D.: Trajectory pattern mining. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 330–339. KDD '07, ACM, New York, NY, USA (2007), http://doi.acm.org/10.1145/1281192.1281230
2. Monreale, A., Pinelli, F., Trasarti, R., Giannotti, F.: Wherenext: A location predictor on trajectory pattern mining. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 637–646. KDD '09, ACM, New York, NY, USA (2009), http://doi.acm.org/10.1145/1557019.1557091
3. Pentland, A., Eagle, N., Lazer, D.: Inferring social network structure using mobile phone data. Proceedings of the National Academy of Sciences (PNAS) 106(36), 15274–15278 (2009)
4. Samaan, N., Karmouch, A.: A mobility prediction architecture based on contextual knowledge and spatial conceptual maps. IEEE Transactions on Mobile Computing 4(6), 537–551 (Nov 2005), http://dx.doi.org/10.1109/TMC.2005.74
5. Zheng, Y., Zhang, L., Ma, Z., Xie, X., Ma, W.Y.: Recommending friends and locations based on individual location history. ACM Trans. Web 5(1), 5:1–5:44 (Feb 2011), http://doi.acm.org/10.1145/1921591.1921596

Interruptibility-Detektierung auf Smartphones

David Krenz*

Betreuer: Anja Bachmann†

Karlsruher Institut für Technologie (KIT)
Pervasive Computing Systems – TECO
*uldzx@student.kit.edu
†bachmann@teco.edu

Zusammenfassung. Der Fokus dieser Seminararbeit liegt auf der Thematik der Unterbrechbarkeit eines Smartphone Benutzers. Wann sollte man ihn auf neue Mitteilungen aufmerksam machen? Wie soll man ihn darauf aufmerksam machen? Es soll ein Eindruck und eine gewisse Übersicht in diesem Bereich geschaffen werden. Es werden Aspekte besprochen die Einfluss in das Konzept von solch Software haben. Es ist dabei wichtig die Psyche des Benutzers zu verstehen und mit einzubringen. Außerdem werden einige Ideen beziehungsweise Ansätze genannt. Die Thematik bewegt sich vor allem, immer in Richtung besserer Möglichkeiten zur Erfassung des Umfelds eines Benutzers, das eine ausschlaggebende Rolle besitzt.

Schlüsselwörter: Interruptibility, Aspekte, Probleme, Strategien, Verfahren, Attention-Awareness

1 Einleitung

Aufmerksamkeit. In unserer heutigen Gesellschaft dreht sich alles um Aufmerksamkeit. Unser Umfeld bestimmt oder gestaltet unseren Grad an Aufmerksamkeit, den wir Dingen widmen, ausschlaggebend mit. Zwischen Arbeit und Privatleben halten uns Mitteilungen auf dem laufenden und zwingen uns gleichzeitig Tätigkeiten zu unterbrechen oder nieder zu legen. In Zeiten des mobilen digitalen Zeitalters ist die Thematik von Aufmerksamkeit im Bezug auf Mitteilungen (engl. „notifications“) empfangen durch Smartphones aktueller denn je. Es stellt sich durch den immer größer werdenden kognitiven Einfluss, der auf uns lastet, hervorgerufen durch eine immer größer werdende Menge an Mitteilungen beziehungsweise Unterbrechungen durch unsere mobilen Begleiter, die Fragen, wann, wo und ob wir diese Mitteilungen eigentlich entgegennehmen müssen beziehungsweise entgegennehmen wollen. Die folgende Arbeit versucht, diese Thematik zu erfassen. Dabei ist bei Unterbrechungen durch Smartphone Nachrichten vor allem der richtige Zeitpunkt des Eintreffens einer Mitteilung der entscheidende Faktor und das Ziel der Forschung im Bereich Interruptibility-Detektion. Durch Analyse diverser Sensordaten, die von Nutzern durch Smartphones gesammelt werden können, ist es möglich, mit neuen Verfahren und Algorithmen präzisere Zeitpunkte für eine geeignete Unterbrechung für die Benutzer ausfindig zu

machen. Oft werden diese Verfahren weiter durch gesammelte Daten und maschinellen Lernens in der Praxis verbessert. Dementsprechend ist es wichtig das Umfeld eines Benutzers so genau wie möglich zu erfassen, um geeignete Zeitpunkte zu filtern. Ist dieser gerade beschäftigt? In einem Meeting? Hier spielen unter anderem Sensordaten, die aus Ort, Zeit und Bewegungsaktivitäten gewonnen werden, eine Rolle. Nur das Nutzen von Sensordaten reicht meist alleine nicht aus. Auch psychologische Aspekte wie die momentane Stimmung eines Benutzers während des Eintreffens einer Benachrichtigung spielen eine gravierende Rolle und können meist nur durch zeitnahe Fragebögen, auch sogenannten „Surveys“, ermittelt werden. In dieser Arbeit wird dem Leser zunächst versucht, eine allgemeine Definition und Erklärung des Begriffs „Interruptibility“ klar zu machen. Des Weiteren folgt ein Abschnitt, der einfließende Aspekte und Probleme in dieser Thematik erfassen soll. Aspekte, die eine Rolle in der Forschung und Entwicklung von Smartphone Interruption spielen. Anbindend die damit verbundenen Probleme. Darauf folgend soll eine Übersicht von Verfahren und Methoden derzeitiger Interruptibility-Detektierung aufgezeigt werden. Letztendlich soll eine Diskussion im Bereich dieser Thematik geführt werden, mit einer abschließenden Zusammenfassung.

2 Definition

2.1 Interruptibility - Was ist das?

Aus dem engl. für „Unterbrechbarkeit“ beschreibt der Begriff der Unterbrechung eine Störung eines gewissen Prozesses oder einer Tätigkeit. Im Rahmen dieser Thematik, die Störung eines bestimmten Prozesses oder einer Tätigkeit einer Person. Der Zustand, an dem diese Person am empfänglichsten für eine neu eintreffende Störung ist. Diese Person ist im Besitz beziehungsweise in Benutzung eines Smartphones, von dem auch die eigentliche Störquelle ausgeht. Das Ziel der Forschung, die in den folgenden Abschnitten vorgestellt wird, ist für jede neue Störung den geeignetsten Zustand des Benutzers zu finden, um ihn so weit wie möglich nicht bei seiner momentanen Tätigkeit oder gar in seiner momentanen Stress- oder Gefühlslage zu stören. Die Intention ist eine so genannte „Attention-Awareness“ bereitzustellen. Optimale Momente, oftmals auch bestimmte Anhaltspunkte, sogenannte „Breakpoints“, sind Unterbrechungen des Benutzers selbst (z.B. ein Wechsel zwischen Tätigkeiten), sind das Ziel. Gerade die Fähigkeit physisch das Augenmerk von einer Tätigkeit auf eine andere Tätigkeit zu wechseln und der eigentlich folgende Effekt auf die Tätigkeit, beschreiben was es bedeutet unterbrechbar zu sein.[16] Diese Arten von Störungen, die von der Störquelle Smartphone ausgehen können, sind unter anderem Mitteilungen in Form von neuen E-Mails, Chat-Nachrichten, App-Meldungen, Systemnachrichten oder auch Update-Nachrichten (etc.)[17]. Des Weiteren können diese Störungen in Form von „Push-Notifications“(„aufploppen“ auf dem Bildschirm), Ruftönen, Led-Benachrichtigungsleuchten oder Vibrationen hervorgerufen werden[17]. Um Berechnungen für geeignete Momente zu gewährleisten, wird der Kontext eines Benutzers zur Analyse hinzugezogen.

2.2 Kontext - Was ist das?

Turner et al.[16] nennt Einflüsse eines Kontext:

1. Charakteristiken des Benutzers(Fühlt sich der Benutzer zurzeit belastet?)
2. Eigenschaften der Unterbrechung(Worüber handelt die Nachricht? Von welcher Quelle geht sie aus?)
3. Das nahe Umfeld(Wo befindet sich der Benutzer und ist er aktiv?)

Der Kontext eines Benutzers beschreibt sein Umfeld. Der Kontext selbst sind Informationen die durch das Smartphone gesammelt werden. Durch Sensoren können Informationen wie der momentane Ort (GPS), ob die Person sich gerade in Bewegung befinden (Beschleunigungssensor) oder Geräuschkulissen (Mikrofon) gesammelt werden. Diese sensorbasierenden Daten werden meisten in bestimmten Vektoren bestehend aus den entsprechenden Variablen gespeichert[15]. Diese können dann zur weiteren Verarbeitung für die Interruptibility-Software genutzt werden. Dabei werden diese oftmals als Trainingsdatensätze verwendet die weitere Anwendung in Modellen des maschinellen Lernens finden. Mit maschinellen Lernmethoden können Beziehungen in Datensätze entdeckt werden, um Prognosen für geeignete Momente zu berechnen. Dabei unterscheidet man zwischen Offline-Machine-Learning, der ganze gesammelte Datensatz wird einmalig zu Berechnung genutzt und Online-Machine-Learning, welches ein immer weiteres Aktualisieren mit neuen Datensätzen zulässt. Doch Informationen eines Kontextes bestehen in diesem Feld meist nicht nur aus sensorbasierenden Daten, sondern auch aus Daten die aus Experience-Sampling gewonnen werden. Die Experience-Sampling-Methode (ESM) ist ein Verfahren das den Benutzer an Ort und Stelle - in situ - dazu auffordert, ein Feedback zu geben[9]. Das heißt, zum Zeitpunkt der Unterbrechung können zum Beispiel Informationen der momentanen Verfassung des Benutzers, in Form von kurzen Fragebögen oder App-Reglern[12] in Erfahrung gebracht werden. ESM ermöglicht das Sammeln von Daten der momentanen Gedanken-, Gefühls- oder Stresslage eines Benutzers(Pejovic et al.[12]) und gibt somit auch Aufschluss auf sein Verhalten.

2.3 Kognitive Belastung - Was ist das?

Im Englischen auch als „cognitive overload“ oder „cognitive load“ bezeichnet, ist die Belastung die der Benutzer bei der Aufnahme von Informationen erfährt. Aufmerksamkeit wird hierbei als eine begrenzte Ressource angesehen[5][11]. Ein Mensch ist bei der Aufnahme von und zu vielen Informationen, in unserem Fall durch den Nachrichteneingang eines Smartphones, nicht in der Lage den vollen Umfang entgegen zu nehmen. Im Nachhinein müssen diese gedanklich verarbeitet werden und lösen einen Lerneffekt auf das Gedächtnis aus[5]. Der wachsende Informationsfluss zwingt den Benutzer zu einem Multi-Tasking verhalten das in Stress resultieren kann[10][11]. Damit ist eine solche Belastung in Pervasive-Computing-Bereichen allgegenwärtig.

3 Aspekte & Probleme

Dieser Abschnitt behandelt einfließende Aspekte im Bereich der Interruptibility Forschung. Was ist zu beachten, wenn man sich mit dem Benutzer und seinen Smartphone Störungen beschäftigt? Womöglich handelt der Benutzer nicht immer wie zu erwarten, was zur Beeinflussung eines verfälschten Resultats in der Analyse der Daten führen kann. Viele Studien setzen sich auch mit solchen Punkten auseinander und versuchen diese zu kategorisieren. Es sind neben den technischen Aspekten, deren Möglichkeiten und Anpassungen, also auch die psychologischen Komponenten zu betrachten. Auf beiden Seiten herrschen auch gewisse Problemaspekte. Da diese Punkte meist eng verflochten beziehungsweise miteinander einhergehen, liegt eine Schwierigkeit vor hier eine klare Gliederung aufzuweisen, wir widmen uns zunächst dem psychologischen Aspekt und dem Verhalten des Benutzers und betrachten dann den technischen Standpunkt.

3.1 Psychologische Perspektive

Das Hauptziel beziehungsweise der Hauptaspekt ist hierbei den Stress, den ein Benutzer erfährt, möglichst zu vermindern oder wenn möglich zu vermeiden. Sahami Shirazi et al.[14] behaupten, dass es in der Natur von Mitteilungen liegt zu stören. Durch jegliche Art einer Mitteilung des Smartphones wie SMS, Instant Messages, Anrufe, E-Mails, soziale Netzwerke, Spiele, Werbung, System- oder Updatemitteilungen, kann eine Störung aufkommen und beim Benutzer Stress auslösen. Dabei kann die Ursache schon in der Einschätzung seiner Handlungsfähigkeit selbst liegen. Funktionen, mit denen man in einen Modus wechselt, die keine Störung mehr zulassen oder gar nur personalisierte Störungen durchlassen von bestimmten Smartphoneapplikationen, sind inzwischen schon lange etabliert für Android und iOS. Nach Yoon et al. [17] sind es nämlich schon die Benachrichtigungseinstellungen, die zu einem höheren Stresslevel bei den Nutzern führen können. Dabei kristallisierten sich auch bestimmte Typen von Benutzern heraus die wenig Stress erfuhren, durch die Kenntnis und das Einstellen der Benachrichtigungsfunktionen. Oft fehlt den Nutzern auch einfach die Bereitschaft dazu, eine Personalisierung ihrer Einstellungen durchzuführen. Dennoch gab es eine Gruppe die sich den Funktionen bewusst war und diese genutzt hat aber dennoch ein erhöhtes Stresslevel aufzeigte. Gründe hierfür sind [17]:

1. Unter anderem das Nachholen von langen Chatverläufen vor allem ausgelöst durch Gruppenchats in Messenger Diensten.
2. Bei den Benutzern kommt die Angst auf etwas zu verpassen und führt nahezu zu einem zwangsmäßigen Verhalten das Smartphone zu überprüfen. Bei Befragung der Teilnehmer der Studie, zeigte sich ein Verhalten von halluzinierenden Zuständen und sie hatten das Gefühl sie bekämen Nachrichten.
3. Ein Missverständnis unter zwei kommunikativen Parteien, die sich miteinander über Chatdienste austauschten. Die einen nutzten den Dienst wie ein Echtzeitdienst (sofortige Antwort) andere wie ein E-Mail Programm, dass einem erlaubt zeitversetzt zu antworten.

4. Weitere Punkte die zu Stress führen, sind unter anderem auch Formen von Spam die als Werbung und Angebote, als Nachrichten auftreten.
5. Des Weiteren falls der Benutzer gerade nicht seine eingehenden Nachrichten überprüfen kann oder will und diese durch Vibration oder Töne auf sich trotzdem aufmerksam machen. Es kommt Stress auf, da er sich des Inhaltes dennoch bewusst sein möchte.

Unter anderem Sahami Shirazi et al.[14] fanden heraus, dass die Relevanz von Mitteilungen variiert, je nach Inhalt. Das heißt mehr Aufmerksamkeit wird Nachrichten gewidmet die einen sozialen Bezug haben (Instant Messenger Dienste, soziale Netzwerke) gegenüber Systemmeldungen. Natürlich kann auch der Drang immer den neuesten Stand in einer Sache zu erzielen eine Ursache sein. Aber genau diese Zurechnung von einer höheren Priorität hat Folgen in Stress, ausgelöst durch „Social Pressure“, d.h. Druck ausgeübt durch Erwartungen von sozialen Kontakten. Denn bei solchen Nachrichten wird mehr Verantwortung von seinen Kommunikationspartnern erwartet. Punkte 2 und 3 finden sich hier wieder, wobei Pielot et al.[13] eine mögliche Ursache für das zwanghafte Verhalten damit erklären, dass genau diese Erwartungen die man seinem Gegenüber zukommen lässt, schnell zu antworten, im Endeffekt, im nicht eintreffenden Fall zu ständiger Kontrolle des Smartphones führt. Pielot et al.[13] beschreiben aber auch, Auslöser für sozialen Druck sind Dienste die die Verfügbarkeit seines Chatpartners preisgeben die Unsicherheiten bei den Benutzern, sowie Probleme bei der Beibehaltung ihrer Privatsphäre auslösen. Der Benutzer wird dadurch unter Druck gesetzt auf eine Nachricht schnell zu Antworten, da er seinen Chatpartner im Glauben lässt, ignoriert zu werden[13].

Wo auch immer wir mit unserem Smartphone agieren, es kann zu einem möglichen Störfaktor werden. Durch mehr Apps, die wir zu benötigen glauben, entsteht auch eine wachsende kognitive Belastung. Okoshi et al.[10][11] beschreiben hierbei, dass das steigende Informationsangebot und konstante Aufbringen von Aufmerksamkeit einen Flaschenhals seitens Mensch erzeugt. Physische und psychische Belastungen denen der Benutzer nicht nachkommt.

3.2 Technologische Perspektive

Aus vorherigem Abschnitt kennen wir nun einige Probleme, die durch die Psyche eines Smartphone-Besitzers auftreten können und die bei dem Design von Interruptibility-Software zu berücksichtigen sind. Dementsprechenden gibt es auch Punkte, die auf technischer Seite anzusprechen sind.

Wie zu Beginn erwähnt, liegt schon eine Problematik in den Einstellungen der Apps die Mitteilungen an den Benutzer senden. Yoon et al.[17] kategorisieren hier noch mal drei Faktoren, die softwaretechnischer Sicht berücksichtigt werden sollten: Inhalt, Updatezeiten und Art der Benachrichtigung. Dabei beschreibt ersterer Punkt die Tendenz zu mehr Eingrenzung der Nachrichten in Abhängigkeit des Inhalts. Es ist schwer auf semantischer Ebene den eigentlichen Inhalt einer z.B. Chatnachricht zu nutzen, aber in dieser Hinsicht zumindest automatisch wichtige Nachrichten von Spam zu trennen oder eine Möglichkeit

wie am Beispiel Google Mail, dass E-Mails aufteilt nach Werbung, sozialem Inhalt und weiteren Themen. Des Weiteren die Punkte periodisch Mitteilungen stumm stellen zu können, dem Nutzer flexiblere Möglichkeiten zu geben sich temporär vor Störungen nach seinem belieben schützen zu lassen. Letztlich ist auch noch die Art, wie eine Nachricht auf sich aufmerksam macht, ein wichtiger Faktor, je nach Umstand des Benutzers möchte der Nutzer keinerlei Anzeigen erhalten in den Phasen, in denen er nicht unterbrochen werden möchte oder zumindest keine akustischen Benachrichtigungselemente. Durch LED-Anzeigen kann er immer noch nach seiner Phase erinnert werden zu antworten und durch Farbauswahl dieser sogar sich Prioritäten legen, ob noch eine wichtige Nachricht vorherrscht[7][17].

Auf Ebene der Verarbeitung von Kontextdaten gibt es auch Dinge zu beachten, denn diese sind womöglich nicht immer zuverlässig. Einzelne Sensordaten spiegeln nicht immer konsistente Verhaltensweisen wieder[6]. Es ist wichtig zu beachten, welche Datensätze zusammen einen zuverlässigen Kontext repräsentieren können, um Verhaltensweisen bewertbar zu machen. Des Weiteren können Probleme auch bei einer genauen Bewertung von Kontextinformationen auftreten. Zur Analyse der Interruptibility des Benutzers werden auch Informationen wie Reaktionszeit, Antwortzeit und entsprechende Verhaltensweisen zu Reaktion, Antwort und etc. genutzt[12]. Was ist also, wenn der Benutzer auf eine Nachricht reagiert aber den Antwortprozess abbricht? Turner, Allen und Whitaker[16] weisen darauf hin, dass „Black-Box“ Methoden (z.B. ein Event das unterbricht, folgt nur zu dem resultierenden Zustand ob eine Antwort folgte oder nicht, ohne zu wissen „wie“) keinen Unterschied feststellen können, ob im gesamten Unterbrechungsprozess ein Erfolg erzielt wurde. Dementsprechend teilten sie diesen in mehrere Phasen ein, die zur Bewertung beitragen sollten. Daraus folgend auch ein genaueres Resultat im Entscheidungskontext.

Ein weiterer und sehr wichtiger Faktor in der Interruptibility-Entwicklung ist die Nutzung von Smartphone Ressourcen. Welche Kapazitäten herrschen vor und wie viel stehen mir zur Verfügung? Es ist zu beachten das ein kontinuierlicher Sensorbetrieb höhere Leistungsansprüche und somit Akkuverbrauch besitzt[2][12]. Da diese Systeme auf Smartphones agieren und Kapazitäten eingeschränkter als z.B. auf PCs sind, ist ein schonender Betrieb ein wichtiger Aspekt, um einen nicht zu hohen Nachteil aufzuwerfen. Eine Möglichkeit wäre hierbei abzuwägen welche Sensoren und wann welche Sensoren genutzt werden. Wie Schonende öfters einzusetzen, um dafür Verbrauchende nur in seltenen Fällen zu nutzen[1].

3.3 Sicherheitstechnische Perspektive

Letztendlich gibt es auch noch den Aspekt beim Design von Interruptibility-Software, die Privatsphäre des Benutzers zu wahren. Gerade zu dem angesprochenen Punkt aus dem psychologischen Abschnitt, die Verfügbarkeit seinem Kommunikationspartner preiszugeben. Nehmen wir den „last-seen“-Status von WhatsApp als Beispiel, welches den Zeitpunkt anzeigt, wann der Chatpartner

zuletzt verfügbar war (des Weiteren auch ob eine Nachricht schon gesehen wurde und den momentanen Online-Status). Der Empfänger fühlt sich womöglich in seinem Verhalten beobachtet. Daraus können falsche Annahmen und Interpretationen im Verhalten des Empfängers geschaffen werden[3]. Zeitpunkte, an denen er mit dem Smartphone interagiert und nicht interagiert geben anderen Sendern womöglich Aufschluss darauf. Der Trend von Apps die Dienste anbieten diese Funktionen zu unterdrücken zeigen auch die Abneigung und die Absichten des Benutzers gegenüber solchen Diensten[13]. Pielot et al.[13] zeigen außerdem, dass solche Dienste keine guten Indikatoren für eine bessere Interruptibility Voraussage darstellen. Es ist zu beachten, ob solche Funktionen im Resultat Vorteile bieten.

4 Methoden, Verfahren und Möglichkeiten

Im Folgenden wollen wir uns nun einer Übersicht von Methoden und Verfahren im Bereich der Interruptibility-Detektierung widmen. Welche Verfahren wurden bisher realisiert und mit welchen Methoden? Was für Möglichkeiten stehen zur Verfügung? Dies soll weitgehendst erfasst und an einigen Beispielen erläutert werden.

4.1 Methoden

Der Hauptbestandteil, einen geeigneten Moment zu erfassen oder im Allgemeinen zur Reduzierung von Störungen und kognitiver Belastung beizutragen, besteht darin, den Kontext des Benutzers zu ermitteln. Wie schon in vorherigen Abschnitten erwähnt sind die dafür gängigen Methoden, das Sensorsampling (sammeln von Sensordaten) sowie die Experience-Sampling-Method (sammeln von Benutzerfeedback, z.B. momentane Stimmung). Es gibt Verfahren, die diese Methoden unabhängig voneinander nutzen, aber auch welche die ein Hybridverfahren darstellen und beide verwenden. Letztendlich werden gesammelte Daten in maschinellen Lernmethoden für ein intelligentes Unterbrechungsverfahren weiterverarbeitet.

4.2 Strategien

Während die Erfassung selbst eine große Rolle spielt, sind Strategien zum Ansetzen eines Verfahren auch wichtig. Wie gelangt man zu einer zuverlässigen Prognose eines geeigneten Moments? Wie reagiert das System? Wann spricht man von einer erfolgreichen Unterbrechung? Vor allem woran orientiert man sich, um einen geeigneten Moment zu erkennen? Pejovic und Musolesi[12] nennen 3 Punkte, die zur Bewertung eines erfolgreichen Moments entscheidend sein können:

1. Die Reaktionsfreudigkeit
2. Die Gefühlslage

3. Die Reaktionszeit

Ersteres gibt an, ob auf eine Unterbrechung reagiert wurde. Die Gefühlslage gibt Klarheit darüber, ob die Mitteilung keinen negativen Effekt verursachte und angenommen werden wollte. Letzteres gibt eine Einschätzung des Momentes selbst, ob der Benutzer in einem bestimmten Zeitraum reagierte. Diese Punkte können jeweils eine Zielsetzung für eine Prognose sein. In der Interruptibility Thematik haben sich zwei Strategien herauskristallisiert, einen optimalen Moment zu nutzen. Zum einen die Möglichkeit, die Mitteilung zum nächsten bestmöglichen Moment nach Eintreffen und Berechnung weiterleiten. Zum anderen die Möglichkeit, von Anfang an nach natürlichen Breakpoints suchen. Letzteres konzentriert sich darauf, den natürlichen Kontextwechsel eines Benutzers zu finden. Während des Wechsels ist es sehr wahrscheinlich, dass der Benutzer aufgrund des Abwendens einer Tätigkeit für Störungen empfänglicher wird[11].

Wann kann man einen Zeitpunkt als erfolgreich ansehen? Ein Zeitpunkt wurde ermittelt, eine Mitteilung gestartet und nun auf eine Reaktion des Benutzers gewartet. Dafür werden bestimmte Zeiträume festgelegt (z.B. oft Zeiträume von ca. 30 Sekunden)[16], in denen eine Reaktion oder Antwort erwartet wird. Kommt es nicht dazu, wird dieser als falscher Zeitpunkt für eine Unterbrechung bewertet. Wird beim Eintreffen einer Nachricht jedoch ein ungeeigneter Moment ermittelt, kann die Mitteilung verzögert werden.

Des Weiteren wie im Abschnitt zu den technischen Aspekten erwähnt, nutzen viele Verfahren eine Blackbox Strategie[16]. Das ist zum Beispiel der Fall, wenn das erfolgreiche Resultat am Ende des Prozesses, nur vom Feedback des Benutzers ausgeht. Dabei spielt der ganze Entscheidungsprozess während und nach Empfangen einer Nachricht, eine wichtige Rolle. Dies ist ein Strategieansatz, den Turner et al.[16] beschreiben.

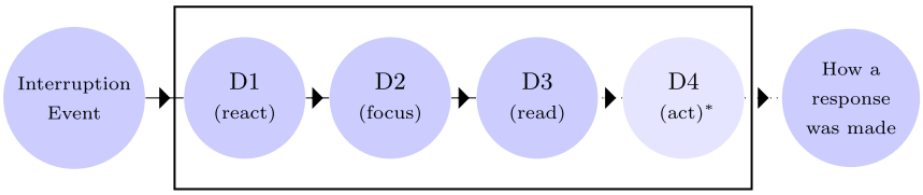


Abb. 1. Zerlegung des Entscheidungsprozesses von Android Nachrichten. *D4 ist optional.[16]

Mit einer To-Do Listen App zu Erinnnerung von Aufgaben lag ihr Ziel darin, tatsächlich unerfolgreiche Momente von nur teilweise unerfolgreichen Momenten im Entscheidungsprozess zu trennen. Dass der Benutzer nicht geantwortet hat, bedeutet noch nicht er habe der Mitteilung keine Aufmerksamkeit gewidmet. Womöglich war der Zeitpunkt passend, nur gerade nicht gegenüber dem

Inhalt dieser Nachricht oder Anwendung. Die Abb. 1 zeigt Zwischenschritte wie das Lesen, Fokussieren und Reagieren, wobei der Benutzer den Prozess vorher schon unterbrechen kann. Durch Überfliegen der Push-Notification, erhält er schon einen Eindruck der Situation. Diese sollten später auch berücksichtigt werden und nicht nur das Endresultat. Das Ergebnis schuf eine Verbesserung der Genauigkeit gegenüber „Null-Responses“[16], also fehlgeschlagenen Nachrichten. Anknüpfend kategorisieren auch Pejovic und Musolesi[12] verschiedene Kontextstadien:

1. Benachrichtigungskontext
2. Antwort-Kontext
3. Benachrichtigungs-Antwort Kontextwechsel
4. Benachrichtigungskontext-Varianz

Erstens legt den Erfolg einer Unterbrechung im Kontext einer gesendeten Nachricht fest. Zweitens den Erfolg einer Unterbrechung im Kontext einer Reaktion. Im dritten Punkt wechselt der Kontext bei einer erfolgreichen Unterbrechung von der Benachrichtigung zur Reaktion. Der letzte Punkt beschreibt den Erfolg einer Unterbrechung in der Variation des Kontextes (verschiedene Kontexte, verschieden zu bewerten). Auch bei reinen ESM Vorgehen, sind Strategien zu berücksichtigen. Wird die Befragung unterbrochen oder falsch beantwortet? Auch hier kann es dazu führen, dass es zu unehrlichen Angaben kommt[9][16]. Um sich nicht nur auf einseitiges Feedback zu verlassen und Möglichkeiten für Studien und bessere Datensätze zu schaffen, kategorisieren Mehrotra et al.[9] hier die zwei Strategien für Momente ein Survey zu veranlassen, zum Erfassen von:

1. täglicher Erfahrung
2. Erfahrung zu einem Event

Ersteres befragt den Benutzer periodisch in seinem Alltag und knüpft die gewonnenen Informationen zu seinem momentanen Kontext. Der zweite Punkt konzentriert sich auf das Eintreffen eines Events, dafür muss aber der Kontext schon vorher beobachtet werden um zum Auftreten des Events das Survey zu starten. Im Allgemeinen ist der Kontext oftmals kategorisiert in physischen Daten wie psychischen Daten mit den gängigen genannten Methoden. Doch es gibt auch anderweitige Ansätze den Kontext eines Benutzers auszumachen. Mehrotra[8] stellt hierbei das Konzept vor, gerade Informationen bezüglich der Stimmung durch Online-Social-Networks (OSN) zu ermitteln. Der Benutzer generiert durch Aktivitäten in sozialen Medien (Posts, Kommentare, teilen von Inhalten) Informationen selbst, die der Kontext Erfassung hinzugezogen werden können. Das können Aktivitäten sein wie das Posten und Teilen von Inhalten, Kommentare setzen oder Stimmungen zum Ausdruck (siehe Facebook) bringen. Zusammen mit dem physischen Kontext können Informationen des Benutzers ohne direktes manuelles Feedback gesammelt werden.

4.3 Möglichkeiten

Softwaretechnische Umsetzungen sind oftmals realisiert in Frameworks, APIs oder in Middleware. Möglichkeiten um neue Anwendungen mit entsprechen-

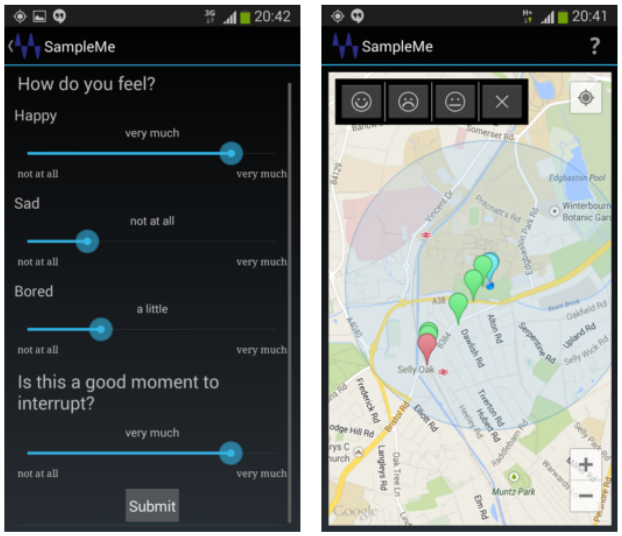
den Funktionen auszustatten oder systemweit miteinzubeziehen werden realisiert. Informationen bezüglich Kontext werden auch zum Teil aus Aktivitäten des Betriebssystems gewonnen. Vor allem Schnittstellen für weitere „Attention-Awareness“ Software.

4.4 Beispiele

Dieser Abschnitt stellt einige Beispiele vor.

– **InterruptMe**

Eine von Pejovic und Musolesi[12] entwickelte Open-Source Android Library für das Management von Unterbrechungen. Es soll intelligente Benutzer Benachrichtigungen ermöglichen. Dafür nutzt es Kontext Aspekte wie Aktivität, Zeit, Ort, Emotionen und Tätigkeit. Es klassifiziert Kontextdaten in maschinellen Lernmethoden für neue Prognosen. Auf Ressourcenschonung wird dabei Wert gelegt. Mit SampleMe (aufbauend auf EmotionSense) realisierten sie eine ESM-Befragung (Studie ausgelegt auf das System). Wird eine Nachricht empfangen, wird der Kontext erfasst, sowie noch mal nach der Befragung. Abbildung 2 zeigt hierbei ein Survey von SampleMe und zeigt auch im Allgemeinen, wie eine ESM gestaltet werden kann. Hierbei handelt es sich um bestimmte Stimmungsregler und eine Karte, die Kontext Aspekte wie Emotion und Ort verknüpft.



(a) Survey. (b) Emotion map.

Abb. 2. SampleMe Sreenshot.[12]

– **Attelia**

Okoshi et al.[11] stellen eine Middleware vor, zur Reduzierung kognitiver Belastung, die natürliche Breakpoints in Echtzeit finden soll und dementsprechend Unterbrechungen starten. Dafür werden die gesammelten Informationen auch in maschinellen Lernmethoden weiterverarbeitet. Sie stellt eine weitreichende Funktion auf, da sie auf viele Applikationen auf dem Smartphone angewendet werden kann. Sie fungiert ohne psychische oder physische Sensoren. Stattdessen wird die Aktivität und Interaktion mit Applikationen beobachtet. Informationen zur Berechnung können gesammelt werden: auf statischer Seite durch ein Stadium einer bestimmten App, wie z.B das Menü eines Spieles (muss von Spielentwickler integriert); auf dynamischer Seite können Informationen zur Laufzeit, wie Events des Betriebssystems (UI-Events: z.B. tap, click, scroll) oder der Anwendung, gesammelt und verarbeitet werden. Offline, online und manuelles Aktivieren von Informationsverarbeitungen sind möglich.

– **SenSocial**

Mehrotra [8] präsentiert eine Middleware die physische Kontextdaten mit psychische Kontextdaten, entstammend aus Sozialen Netzwerk Aktivitäten, verknüpft. Es ist eine Android Library die von einem Server aus fungiert. Informationen, die über soziale Netzwerke entnommen werden können, z.B. wie das Liken und Kommentieren von Dingen, aktivieren das Sensorsampling und knüpfen diese zwei Datenpfade aneinander. Aus sozialen Netzwerken besteht das Potential Informationen zu Gedanken und Gefühle zu sammeln ohne explizite ESM. Durch Einstellen von Filtern bestimmter Daten und Berechtigungsfreigabe soll Privatsphäre ermöglicht werden. Sie stellen hier 2 Prototypen vor: „Facebook Sensor Map“ und „ConWeb“. Es wird die Facebook-Aktivität beobachtet und an den Kontext des Ortes und anderen physischen Daten gebunden. Zweites ist ein Webbrowser der den Inhalt von Webseiten generiert und ändert in Abhängigkeit des Kontextes des Benutzers.

5 Diskussion

Im Folgenden soll eine kurze Diskussion einige Aspekte noch mal aufwerfen. Wir haben uns eine gewisse Übersicht der Thematik geschaffen, doch da diese sehr durchwachsen von neuen Studien und Ergebnissen ist, ist sie im ganzen schwer zu fassen. Der Trend zum Smartphone schafft einen Gegensatz, zum einen schaffen wir es schneller und mehr Aufgaben denn je, z.B. am Arbeitsplatz zu erledigen[4]. Auf der anderen Seite ergibt sich durch ständige Unterbrechungen, kein produktiver Arbeitsfluss[12]. Die Interruptibility-Forschung sucht hier neue Wege unsere Aufmerksamkeit unter Kontrolle zu behalten.

Ein wichtiger Punkt, der noch zu erwähnen wäre, es bestehen auch ausnahmen für Unterbrechungen. Nicht jede Unterbrechung benötigt einen optimalen Moment, es gibt Mitteilungen, deren Wichtigkeit gerade im Moment selbst bestand haben. Anwendungen, die zu Katastrophenwarnung benutzt werden oder

auch einfache Erinnerungen die auf Medikamenteinnahmen aufmerksam machen. Inhalt ist und bleibt der entscheidende Punkt in diesem Bereich.

Des Weiteren, auch wenn Studien sich nicht immer gegenseitig bestätigen, zeigen sie doch einige Probleme auf, die in der ganzen Thematik bestand haben. Diese zeigten oft, dass Benutzer mehr eigene Kontrolle über ihre Mitteilungen haben wollen. D.h. Einstellungen selbst übernehmen. Diesbezüglich aber oft ein Missstand entsteht. Benutzer resignieren, die fehlende Bereitschaft oder kein Know-how führen dazu diese Funktionalitäten dennoch nicht zu nutzen[17]. Dennoch ist gerade die Zielsetzung von Smartphonesystemen, mehr Intuitivität zu schaffen.

Vor- und Nachteile finden sich auch in den Methoden zur Datenerfassung wieder. Während sich einige auf die Zuverlässigkeit von ESM verlassen, aber meistens auch nur unter den richtigen Umständen, distanzieren sich einige auch bewusst davon. Da hier auch Missstände in Form von falschen Ergebnissen aufkommen können. Ist der Feedbackbogen im falschen Moment erschienen, können wiederum Unehrlichkeit und der Mangel an Bereitschaft Probleme aufwerfen. Dennoch gibt es in dieser Hinsicht nicht viele zuverlässige Ansätze Stimmungen des Benutzers mit ein zu berechnen. Ansätze mit schon vorhandenen Daten, wenn sie denn vorhanden sind, aus sozialen Netzwerk Inhalten, scheinen Potenzial diesbezüglich zu haben. Auf der anderen Seite bietet das mobile Assessment bessere Möglichkeiten denn je für Sozialwissenschaftler, überall und einfach Befragungen zu tätigen.

Aufseiten des Sensorsamplings gibt es natürlich auch Kritik. Wie schon erwähnt, können dies Inkonsistenz oder die Frage der Zuverlässigkeit sein. Auch die Frage, ob die benötigten Sensoren immer zu Verfügung stehen. In Verbindung mit all dem steht natürlich die Frage der Privatsphäre. Metadaten können unseren Alltag rekonstruieren. Es ist wichtig, Berechtigungen dem Benutzer zu überlassen oder ihn zumindest zu informieren. Aber das Preisgeben von Informationen hat wahrscheinlich auch Einfluss auf das gesellschaftliche Verhalten selbst, siehe „last-seen“-Status. Sozialer Druck lenkt uns in gewisse Verhaltensschemen. Im Großen und Ganzen bietet diese Thematik noch viel Potenzial vor allem im Bezug auf neu erscheinende Hardware. Sie schafft Schritte in die richtige Richtung, in einem Umfeld, das immer mehr von den negativen Effekten neuer mobiler Technik geprägt ist. Gerade der Trend von mobiler Kontexterfassung will unseren Alltag auch einfacher gestalten.

6 Zusammenfassung

Die Ausweitung von Smartphones und deren mit sich bringende Probleme haben einen großen Einfluss auf unser Leben. Aber nicht nur das, sie bieten auch viele Vorteile und bringen einen großen Raum an Komfort. Die Ziele der Interruptibility Forschung sind recht klar, das Nutzen von Smartphone einfacher zu gestalten und Inhalte aufnahmefähiger zu machen. Einfacher vor allem im Sinne, einer Abnahme von Lasten. Stress und kognitive Belastung sind Probleme, die durch Nutzung von Smartphones aufkommen können. Diese sollen

gerade durch Einfangen und nutzen geeigneter Momente, diese Belastungen minimieren. Psychische Aspekte, die man beim Benutzer ins Spiel kommen sind also zu beachten. Den die Bereitschaft Nachrichten entgegen zu nehmen oder generell unterbrochen zu werden ist abhängig vom Inhalt dieser. Es gibt dabei auch verschieden Verhaltensmuster, die beim Unterbrechen entstehen, zu beachten. Aber auch Technische und vor allem hinsichtlich von Ressourcennutzung, da Interruptibility-Anwendungen auf Dauer kostspielig werden können. Des Weiteren bleibt nichtsdestotrotz dem Potenzial und den Vorteilen dieser Technik der Aspekt der Sicherheit und Privatsphäre eines Nutzers ein kritisch zu betrachtender Punkt. Konzepte, wie das stufenweise Analysieren von Unterbrechungsprozessen, führt in diesem Bereich der Forschung zu genaueren Ergebnissen. Software, die daraus entsteht, will eine Erweiterung für existierende Programme oder auch Ansätze für neue Anwendungen, schaffen. Letzten Endes zeigen auch Trends zu immer mehr „in-the-wild"-Studien, die gerade durch Mobilität und Leistung solcher Geräte profitieren, das Potenzial dieser Thematik enthalten[15].

Literatur

1. Abdesslem, F.B., Phillips, A., Henderson, T.: Less is more: energy-efficient mobile sensing with senseless. MobiHeld pp. 61–62 (2009), http://dl.acm.org/citation.cfm?id=1592621
2. Balan, R.K., Lee, Y., Wee, T.K., Misra, A.: The challenge of continuous mobile context sensing. 2014 6th International Conference on Communication Systems and Networks, COMSNETS 2014 (2014)
3. Church, K., de Oliveira, R.: What's up with whatsapp?: comparing mobile instant messaging behaviors with traditional SMS. 15th international conference on Human-computer interaction with mobile devices and services (MobileHCI'13) pp. 352–361 (2013)
4. Garrett, R.K., Danziger, J.N.: IM = Interruption management? instant messaging and disruption in the workplace. Journal of Computer-Mediated Communication 13(1), 23–42 (2007)
5. Haapalainen, E., Kim, S., Forlizzi, J.F., Dey, A.K.: Psycho-Physiological Measures for Assessing Cognitive Load. Proceedings of the 12th ACM international conference on Ubiquitous computing pp. 301–310 (2010), http://portal.acm.org/citation.cfm?doid=1864349.1864395
6. Lathia, N., Rachuri, K.K., Mascolo, C., Rentfrow, P.J.: Contextual dissonance: Design bias in sensor-based experience sampling methods. UbiComp '13 Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing pp. 183–192 (2013), https://www.cl.cam.ac.uk/{~}cm542/papers/ubicomp2013.pdf
7. Mashhadi, A., Mathur, A., Kawsar, F.: The myth of subtle notifications. Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing Adjunct Publication - UbiComp '14 Adjunct pp. 111–114 (2014), http://dl.acm.org/citation.cfm?doid=2638728.2638759
8. Mehrotra, A.: SenSocial : A Middleware for Integrating Online Social Networks and Mobile Sensing Data Streams. Middleware 2014 pp. 205–216 (2014)

9. Mehrotra, A., Vermeulen, J., Pejovic, V., Musolesi, M.: Ask, but don't interrupt. Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers - UbiComp '15 pp. 723–732 (2015), http://dl.acm.org/citation.cfm?id=2800835.2804397
10. Okoshi, T., Nozaki, H., Nakazawa, J., Tokuda, H., Ramos, J., Dey, A.K.: Towards attention-aware adaptive notification on smart phones. Pervasive and Mobile Computing 26, 17–34 (2016), http://dx.doi.org/10.1016/j.pmcj.2015.10.004
11. Okoshi, T., Ramos, J., Nozaki, H., Nakazawa, J., Dey, A.K., Tokuda, H.: Attelia: Reducing user's cognitive load due to interruptive notifications on smart phones. 2015 IEEE International Conference on Pervasive Computing and Communications, PerCom 2015 pp. 96–104 (2015)
12. Pejovic, V., Musolesi, M.: InterruptMe: Designing Intelligent Prompting Mechanisms for Pervasive Applications. Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '14) pp. 897–908 (2014), http://doi.acm.org/10.1145/2632048.2632062
13. Pielot, M., de Oliveira, R., Kwak, H., Oliver, N.: Didn't You See My Message? Predicting Attentiveness to Mobile Instant Messages. Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14 pp. 3319–3328 (2014), http://dl.acm.org/citation.cfm?doid=2556288.2556973
14. Sahami Shirazi, A., Henze, N., Dingler, T., Pielot, M., Weber, D., Schmidt, A.: Large-scale assessment of mobile notifications. Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14 pp. 3055–3064 (2014), http://doi.acm.org/10.1145/2556288.2557189$\backslash$nhttp://dl.acm.org/ft{_}gateway.cfm?id=2557189{\&}type=pdf$\backslash$nhttp://dl.acm.org/citation.cfm?doid=2556288.2557189
15. Turner, L.D., Allen, S.M., Whitaker, R.M.: Interruptibility prediction for ubiquitous systems. In: Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing - UbiComp '15. pp. 801–812 (2015), http://dl.acm.org/citation.cfm?doid=2750858.2807514
16. Turner, L.D., Allen, S.M., Whitaker, R.M.: Push or Delay ? Decomposing Smartphone Notification Response Behaviour (2015)
17. Yoon, S., Lee, S.s., Lee, J.m., Lee, K.: Understanding notification stress of smartphone messenger app. Proceedings of the extended abstracts of the 32nd annual ACM conference on Human factors in computing systems - CHI EA '14 pp. 1735–1740 (2014), http://dl.acm.org/citation.cfm?doid=2559206.2581167

Usable Security Models for the Internet of Things and Hybrid Cloud Solutions

Maximilian Dietz*

Advisor: Andrei Miclaus†

Karlsruhe Institute of Technology (KIT)
Pervasive Computing Systems – TECO
*Maximilian.Dietz@student.kit.edu
†miclaus@teco.edu

Abstract. Diese Ausarbeitung soll grundlegende Sicherheitsmodelle für den Einsatz im Internet der Dinge oder bei hybriden Cloud-Lösung vorstellen. Nach einer Einführung in die Welt der Dinge und deren Reichweite, sowie dem Phänomen des Cloud Computing sollen konkrete Angriffsszenarien und Ansätze für deren Verhinderung aufgezeigt werden. Auch sollen Beispiel aus der Praxis namhafter Onlineunternehmen verdeutlichen, wie die vorgestellten Sicherheitskonzepte einen Mehrwert für die Sicherheit der noch jungen Technologien bewirken können. Dabei soll auf die Implementierbarkeit und die Vor- und Nachteile der verschiedenen Modelle eingegangen werden, um einen Überblick zu schaffen, welche Technologien sich für den Einsatz im Unternehmen empfehlen würden.

Keywords: Internet der Dinge, Hybrid Cloud, Security, Single Sign-On

1 Einleitung

Viele Unternehmen nutzen bereits Technologien wie das Internet der Dinge und hybride Cloud-Lösungen, da sie deren großes Potential erkannt haben, sind sich allerdings nicht über die Risiken bewusst. Die Nutzung verursacht neue, unbekannte Sicherheitslücken, welche von Angreifern sofort ausgenutzt werden. Durch den gezielten Angriff auf solche ungeschützen Systeme kann es neben dem wirtschaftlichen Schaden auch zu physischem Schaden kommen.

2 Grundlagen

Zunächst sollen die Besonderheiten des Internet der Dinge bzw. von hybriden Cloudlösungen herausgearbeitet werden. Beide Konzepte bieten aus unternehmerischer Sicht große Chancen, eröffnen allerdings auch neue Schlupflöcher für Angreifer. Im Hinblick auf die Sicherheit sollen vor allem die Herausforderungen und mögliche Lösungsansätze für die Umsetzung beider Technologien im Unternehmen herausgearbeitet werden.

2.1 Internet of Things

Der Gedanke des Internets hat sich in den letzten Jahren grundlegenden Veränderungen unterzogen, verglichen mit dem Konzept aus den Anfangszeiten. Eine Zukunftsannahme aus dem Jahr 2006 schätze die Anzahl der Endgeräte bis zum Jahr 2017 auf 7 Billionen, die dann von 7 Milliarden Menschen bedient werden sollten [1]. Wenngleich diese Prognose nur zu Teilen eingetreten ist, erleben wir dennoch Jahr für Jahr einen dramatische Anstieg an Nutzern und Endgeräten. Dieser führt zugleich auch zu einer stärkeren Vernetzung.

Wenn wir von Endgeräten sprechen, gehen wir meist davon aus, dass diese von Menschen bedient werden, um miteinander zu kommunizieren. Genauso sind inzwischen auch andere - alltägliche bis industrielle Dinge mit dem Internet verbunden. Das Internet der Dinge oder auch Industrielle Internet befasst sich mit der Verbindung verschiedenster Dinge miteinander, wobei der Grad der Vernetzung in mehrere Wellen unterteilt worden ist.

1. Welle Diese Geräte aus der Informationstechnik (IT), welche im Geschäfts- und Wirtschaftsleben von Unternehmen häufig im Zusammenhang mit Geschäftprozessen eingesetzt werden, sind meist verdrahtet mit dem Internet verbunden sind. Demnach sind IT-Technologien vorwiegend in der Rubrik "Business" zu verorten.

2. Welle Operationale Technologien (OT), die in den industriellen Produktionumgebungen der Unternehmen zu finden sind und bisher einen abgeschlossenen Bereich innerhalb eines Unternehmens dargestellt haben, sind inzwischen auch oft verdrahtet mit dem Internet verbunden, wie z.B.: Steuerungseinheiten, Prozessleittechnik und Medizinische Apparaturen.

3. Welle Darunter fallen die Verbrauchergeräte aus der IT. Diese werden von Konsumenten bzw. Mitarbeitern gekauft, sind meist durch eine oder mehrere drahtlosen Verbindungsarten mit dem Internet verbunden sind, z.B. Smartphones und Tablets.

4. Welle Zuletzt gibt es noch die Single-Purpose Devices, die sowohl von IT-/OT-Experten sowie Konsumenten gekauft werden und meist durch nur eine drahtlosen Verbindungsform mit dem Internet verbunden ist. Hierzu zählen vor allem die Embedded Systems, wodurch so gut wie Alles die Möglichkeit hat eine Verbindung zum Internet aufzubauen.

Vor allem letztere Kategorie trifft aktuell die herrschende Vorstellung des Internet of Things. Viele Dinge aus diesem Feld waren bereits zuvor schon mit eingebetteter Soft- und Hardware ausgestattet, wobei nun der Faktor der Internetanbindung hinzugekommen ist. Genau hier liegt auch eine der größten Gefahren

des Internet der Dinge. Was vorher nach außen hin abgeschlossene Systeme waren, bieten Angreifer jetzt die Möglichkeit, das Unternehmen über die ergänzte Schnittstelle des Internets anzugreifen. Deshalb ist für viele Unternehmen auch dieser Bereich hinsichtlich der Security-Aspekte von immer größere werdender Bedeutung. Allerdings sind viele Unternehmen noch nicht auf die Sicherheitsanforderungen des Internet der Dinge vorbereitet oder besitzen nicht die fachlichen Kapazitäten, um eine solche Umstellung selbst durchzuführen. Auch der Gedanke, dass ein großer Teil der Verantwortung erst einmal bei den Herstellern der solcher Geräte liegt, wird laut [2].

Die speziellen Herausforderungen für die Sicherheit dieser Dinge verglichen mit herkömmlichen Systemen sind vielseitig. Vieles was in den Kontext des Internet der Dinge fällt, ist nicht zwangsläufig der IT bzw. OT zuzuordnen, sondern fällt eher in den Bereich des individuellen Gebrauchs. Vor dem Hintergrund der verschiedenen Klassen von Geräten innerhalb der Welt des Internet der Dinge, haben sich aufgrund der unterschiedlichen Anforderungen was die Sicherheit betrifft auch zwei Begrifflichkeiten für IT und OT herausgebildet - Security und Safety [3].

Safety Der Ausdruck Safety kommt aus der OT und meint den "Schutz vor physischen Schäden von Mensch und Maschine" in einer industriellen Umgebungen. Die beiden zentralen Aspekte der OT sind hierbei die Verfügbarkeit und die Verlässlichkeit. Hinsichtlich der Produktion soll zunächst garantiert werden, dass diese verfügbar ist. Dementsprechend soll die Fertigung störungsfrei und ohne Ausfälle ablaufen, da Unterbrechungen bzw. ein Stillstand rasch sehr teuer werden können. Ein weiteres Ziel unter dem Begriff Safety ist die Verlässlichkeit. Im Hinblick auf die zu fertigenden Güter soll ein gleichbleibendes qualitatives Niveau sichergestellt werden.

Security In der IT-Sicherheit beschäftigt sich der Begriff Security mit der Überwachung und Kontrolle von Zugriffen und ferner mit der Vermeidung von missbräuchlichen Zugriffsberechtigungen. Neben der Sicherheit, sind die primären Ziele der Security: Die Verfügbarkeit, Vertraulichkeit, Integrität und Authentizität der Datenzugriffe. Ein System ist verfügbar, wenn dessen Funktionen dem Nutzer zur Verfügung stehen, wenn er diese benötigt. Hierbei können Datenverlust und funktionale Problem mit Software bzw. Hardware die Verfügbarkeit während des Zugriffs einschranken. Desweiteren soll die Integrität eine Verfälschung von Daten, Programmen oder Hardware verhindern, um eine Verlässlichkeit der Daten sicherstellen zu können. Bei der Vertraulichkeit geht es in erster Linie um den Zugriff auf Daten, der ausschließlich durch autorisierte Nutzer erfolgen sollte. Deshalb soll hier ein Zugriff durch Unbefugte vermieden werden. Letzlich sorgt sich die Authentizität um den echten Urheber und die korrekten Herkunft der Daten. Diese Echtheitsprüfung muss vorallem bei der elektronischen Übertragung von Dokumenten durchgeführt, um etwaige Zweifel

beseitigen zu können.

Aufgrund der besonderen Eigenschaften waren beide technische Klassen bisher durch die Netzwerkarchitektur getrennt. Betrachtet man die Wartungshäufigkeit der verschiedenen Infrakstrukturen in der Vergangenheit, so lassen sich oben genannte Anforderungen auch hier wiederfinden. Während die IT versucht durch eine hohe Frequenz an Patches sich vor neuen Sicherheitslücken und Angriffsmöglichkeiten zu schützen, laufen Geräte der OT meist mehrere Jahrzente zuverlässig mit der selben Soft- und Hardware. Wegen des Risikos von Ausfällen werden hier selten Änderung am System vorgenommen, was im Zuge der steigenden Vernetzung auf lange Sicht gesehen zu einem Intressenkonflikt führen könnte. Dabei liegt die Entscheidung zwischen Patching auf Kosten der Verlässlichkeit und Systemstabilität mit dem Risiko eines Angriffs wegen einer nicht behobenen Schwachstelle. Denn durch die höhere Erreichbarkeit der OT-Umgebungen lassen sich oben genannte Sicherheitsbegriffe so nicht mehr uneingeschränkt auf das Internet der Dinge anwenden.

Allerdings stellen die Geräte, welche eingebettete Betriebssysteme oder andere Software enthalten, die Betreiber vor die Aufgabe der ständigen Fehlerkorrektur und der Konfiguration der Sicherheitmaßnahmen. Eine Sicherheitslücke im System kann zu Konsequenzen in der realen Welt führen und kann Einfluss auf Bereiche wie Strategie und Finanzen eines Unternehmens haben.

2.2 Hybrid Cloud Solutions

Mit Cloud Computing wird eine serviceorientierte, verteilte und visualisierte IT-Lösung bezeichnet, die sich in den letzten Jahren immer mehr in der Gesellschaft etabliert hat. Der Begriff zielt auf die Auslagerung von Rechenleistung auf Cloud-Infrastrukturen im Internet ab, die als Substitut für das lokale "Computing" dienen sollen. Allerdings umfasst Cloud Computing weit mehr als nur den Faktor Rechenleistung. Diese Form, welche zusätzlich noch die Inanspruchnahme von externem Speicherplatz beinhaltet, wird als Infrastructure-as-a-Service (IaaS) bezeichnet. Weiterhin wird unter Platform-as-a-Service ein Angebot einer Webplattform mit Mehrwert für deren Nutzer verstanden. Außerdem ist mit Software-as-a-Service die Bereitstellung von verschiedenster Anwendungen gemeint. Was als Anything-as-a-Service betitelt wird, meint in erster Linie das Auslagern von Services an externe Anbieter. Cloud-Dienste kennzeichnen sich durch die verschiedene Arten der Bereitstellung.

Public Cloud Die "öffentlichen" Cloud-Dienste sind für jedermann im Internet frei zugänglich. Die Bandbreite reicht von kostenlosen Angeboten, wie beispielsweise der Office-Anwendungen Google Docs bis hin zur kostenpflichtigen Lösung Office365 von Microsoft.

Private Cloud Aus Unternehmenssicht wird diese Form der Cloud wegen ihrer Vorteile hinsichtlich Datenschutz und Sicherheit bevorzugt. Demnach werden die

Dienste weiterhin vom Unternehmen ausgeführt, wobei sich der Personenkreis, der Zugriff darauf hat auf die eigene Belegschaft beschränkt. Allerdings werden ebenso die Vorzüge einer Cloud-Lösung genutzt, was zum Beispiel die Skalierbarkeit betrifft. Jedoch erreicht diese Infrastruktur ihre Ziele häufig nicht in selbem Maße als freizugängliche Cloud-Dienste.

Community Cloud Das gemeinsame Nutzen eines Cloud-Dienstes von mehreren Unternehmen im Rahmen von Projekten nennt man Community Cloud. Diese Cloud-Lösung ist nur dem bestimmten Personenkreis der beteiligten Unternehmen zugänglich.

Hybrid Cloud Als hybride Cloud-Lösung ist es möglich bestimmte Dienste über öffentlichen Cloud-Dienstleister im Internet und Dienste mit strengeren Anforderungen hinsichtlich Datenschutz und Sicherheit über unternehmensinterne Wege laufen zu lassen. Besonders die Entscheidung der Wegewahl der verschiedenen Anwendungen markiert den kritischen Punkt dieser Cloud-Form.

Im Folgenden soll vor allem die Frage geklärt werden, welche Methoden im Rahmen von Cloud-Lösungen angewendet werden können. Hierbei geht es vor allem um die Sensibilität der Daten, in Abhängigkeit derer entschieden werden muss inwiefern ein besonderer Schutz notwendig ist.

3 Threats and Attacks

Größere Angriffe auf Geräte und Benutzer werden nicht erst seit dem Beginn des Internet der Dinge verübt. Allerdings erlaubt das Internet der Dinge durch den Anstieg der Vernetzung bzw. der Steigerung der Anzahl vernetzter Geräte eine völlig neue Dimension von Cyber-Attacken. Größtenteils geht es wohl um die datenschutzrechtlichen Folgen dieser Angriffe, allerdings befindet sich mit dem Eintritt ins Zeitalter des Internet der Dinge auch eine neue Klasse an Geräten online, die auch sicherheitsrelevante Fragen in den Vordergrund stellen könnte.

Die Attacken sind meist nicht von besonders komplexer Struktur und arbeiten häufig nach dem selben Schema. Doch durch die Vielzahl an beteiligten Geräten können sie inzwischen häufig größere Effekte erzielen. Desweiteren werden nun fünf der bekanntesten Angriffsszenarien im Internet der Dinge vorgestellt [5].

3.1 Botnetze

In diesem Szenario wird durch die Summe an verbundenene Geräten ein Netzwerk erzeugt, um das Ausüben größerer Angriffe zu ermöglichen. Den unfreiwilligen Zusammenschluss einer Vielzahl an Systemen, die für bösartige Zwecke missbraucht werden, nennt man Botnetze. Gesteuert werden diese Netze zentral über sogenannte Command-and-Control Server von deren jeweiligen Betreiber.

Diese Netzwerke verüben erheblichen Schaden durch millionenfache Anfragen auf bestimmte Systeme zur selben Zeit, welche von der Last überfordert werden und meist zusammenbrechen. Im Zusammenhang mit dem Internet der Dinge nennt man diese Art von Netzen auch Thingbots, da häufig unabhängige vernetzte Geräte bzw. Dinge unter den Opfern der Botnetzbetreiber zu finden sind. Diese machen sich immer öfter Geräte aus dem alltäglichen Leben wie zum Beispiel aus der Haushalts- und Unterhaltungselektronik zu Eigen. Meist werden dabei deren niedrige Sicherheitsanforderungen ausgenutzt, denn viele der mit dem Internet der Dinge vernetzen Geräte sind sehr schwach oder überhaupt nicht geschützt. Daher sollten auch für dies Klasse ähnliche Sicherheitsmaßstäbe angesetzt werden, wie beispielsweise für Online-Banking Geschäfte und Ähnliches. Den häufig gelingt es den Betreibern von derartigen Botnetzen mithilfe der Vielzahl ungesicherter Geräte an private Information und unter anderem auch an Online-Banking Accounts zu gelangen. Neben Distributed Denial of Service Attacken werden auch das massenweise Versenden von Spam- und Phishing-Mails durch Botnetze verübt [6].

3.2 Man-in-the-Middle

Eine Art von Angriff, bei der ein Angreifer versucht die Kommunikation zwischen zwei Geräten mitzuhören und zu übernehmen, nennt man Man-in-the-Middle. Daraufhin wird auf Angreiferseite meist versucht die Originalnachrichten abzufangen und in veränderter Form an den eigentlichen Empfänger zu übermitteln. So wird diesem System vorgetäuscht, dass er immer noch mit dem ursprünglichen Gesprächspartner kommuniziert. Der Angreifer erlangt so die Gewalt über das Gerät, was für einige Geräte aus dem Spektrum des Internet der Dinge von erheblichem Risiko sein kann. Wenn es einmal nicht um Kühlschränke sondern beispielsweise um gehackte Fahrzeuge geht, kann dieses Szenario für die beteiligten Personen schnell lebensgefährlich werden. Auch sind Steuerungseinheiten in der industriellen Produktion ein beliebtes Ziel für Man-in-the-Middle-Angriffe, da hier schnell enorme finanzielle und materielle Schäden entstehen können [7].

3.3 Diebstahl von Daten und Identitäten

Durch das Aufkommen von sozialen Netzwerken und der Vielzahl an internetfähigen Endgeräten haben es Angreifer einfacher an die persönlichen Daten heranzukommen. Jeder Verwender solcher Geräte bewahrt dort eine große Menge an sehr privaten Informationen auf. Der Identitätsdiebstahl dient zunächst der reinen Ansammlung von benutzerbezogenen Daten. Im weiteren Verlauf wird entweder versucht den Besitzer der Daten zu manipulieren oder dessen Identität komplett zu übernehmen. Dann sind ebenfalls die Personen, die mit dem Opfer des Identitätsdiebstahls in Kontakt stehen durch gefälschte Anfragen des Angreifers bedroht. Häufig kommt es dabei zu großen finanzielle Forderung, die

mit dem Vertrauen zur Identität des Opfers gedeckt werden. Vor diesem Hintergrund sollte die Preisgabe von personenbezogenen Informationen wohl überlegt geschehen [8].

3.4 Social Engineering

Um Benutzern vertrauliche Informationen zu entlocken, werden die Methoden aus dem Social Engineering angewandt. Dabei versuchen die Angreifer ihre Opfer zu täuschen, um Zugang zu dessen privaten Daten zu erlangen. Meist wird zur Manipulation der Benutzer eine vertraute Oberfläche eines Dienstes verwendet, bei dem sie sich registrieren sollen. Fällt dieser darauf herein, erhält der Angreifer Zugang zu dessen Anmeldedaten, kann die Kontrolle über das Benutzerkonto übernehmen und im Namen des Opfers Transaktionen ausführen. Meist können Social Engineering Maßnahmen mit den oben beschriebenen Methoden des Daten- und Identitätsdiebstahls einhergehen, da beide Angriffsarten ähnliche Ziele verfolgen [9].

3.5 Denial Of Service

Was im Zusammenhang von Botnetzen bzw. Thingbots als (Distributed) Denial of Service Attacke schon beschrieben wurde, meint üblicherweise die Belastung eines Dienstes mit so vielen Anfragen, dass dieser für eine gewissen Zeitraum nicht mehr erreichbar ist. Häufigerweise sind dafür eine große Menge an gleichzeitig geschalteten Systemen notwendig, weshalb ein solcher Angriff ausgezeichnet von Botnetzen ausgeübt werden kann. Der Erfolg bzw. der Zweck einer solchen Attacke bedeutet meist ein Verlust von Ansehen und Geld für die Betroffenen. Allerdings werden Denial of Service Angriffe immer häufiger mit idealisierten und ideologischen Zielen begründet, was sie in gewisser Weise auch zu einem politischen Druckmittel macht. Der Anteil an Geräten bei diesem Angriffsszenario, der aus dem Internet der Dinge kommt, liegt bei 21%, was die vorher geäußerten Sicherheitsbedenken noch bekräftigt [10].

4 Security Models

4.1 Authentication/Authorization

Wo auch immer Menschen, Maschinen oder Dinge mit dem System interagieren, sind die beiden zentralen Konzepte Authentifizierung und Autorisierung, was die Sicherheit betrifft unentbehrlich. Beim Versuch auf das System zuzugreifen, übernimmt die Authentifizierung die Prüfung der zugreifenden Seite auf Echtheit. Prinzipiell kann eine Entität d.h. eine Person bzw. ein Gerät online mehrere Identitäten haben, genauso wie die Identität in der Familie und im Beruf in der Realwelt oft nicht dieselbe ist. Es wird also die Identität verifiziert z.B. durch die Abfrage von Benutzernamen und Passwort, ob die Entität, welche eine Anfrage sendet, die ist, welche sie vorgibt zu sein. Ferner gilt die Annahme, dass nur sie

Zugriff auf das System hat, weshalb deren Echtheit garantiert werden muss. Wie schon zuvor angemerkt, muss es sich bei dieser Entität nicht um einen Menschen handeln, denn auch Dinge können auf das System zugreifen [11].

Nach der erfolgreichen Authentifizierung der Identität wird mithilfe der über ihn bekannten Information entschieden, welche Rechte dieser auf dem System hat. In einem Unternehmen kann zum Beispiel je nach Rang bzw. Abteilung unterschieden werden, für welche Handlung die jeweilige Person autorisiert ist. Hier ist abzuwägen, ob wie feingranular die Autorisierung sein muss, um die verschiedenen Rollen im Unternehmen abbilden zu können. Meist ist eine grobere Segmentierung und besser als eine Abbildung für jeden einzelnen Geschäftsprozess. Anders als bei Systemen mit monolithischer Architekur, wo sich meist die Anwendung selbst um Authentifizierung und Autorisierung kümmert, ist in der Welt des Internet der Dinge oder auch bei Cloud-Lösungen ein dediziertes Modell notwendig [12].

4.2 Single Sign-On

Vor allem wegen der gegebenen Dezentralität solcher Systeme ist ein Ansatz nötig, mit dem ein Login nicht für jedes einzelne System notwendig ist. Hier sind unter dem Überbegriff Single Sign-On mehrere Lösungen zusammengefasst. Das Ziel ist hier die Anzahl der notwendigen Logins für alle Systeme zu reduzieren bzw. zusammenzufassen. Das heißt, letztendlich muss sich der Nutzer nur bei einem Dienst anmelden, um auch Zugriff auf alle anderen Dienste zu erhalten.

Die Authentifizierung für einen bestimmten Dienst findet mithilfe eines sogenannten Idenity Providers statt. Nach dessen erfolgreichen Zustimmung, müssen die Daten sicher an den Anbieter des Dienstes übermittelt werden. Letzterer hat dann immer noch die Möglichkeit über die Autorisierung zu entscheiden. Hier gilt es wie im Fall der Hybrid Cloud abzuwägen, wann ein öffentlicher oder ein interner Identity Provider genutzt werden soll. Dabei sind kommerzielle Lösungen wie z.B. die von Google in der Unternehmenswelt vermutlich weniger geeignet, um mit sensiblen Login-Daten umzugehen, als eine hauseigene Lösung. Neben dem Identity Provider wird ein Directory Service benötigt, um die verschiedenen Rollen der einzelnen Benutzer zu verwalten. Beide Dienste werden meist zusammengefasst, können jedoch auch als jeweils eigenständige Lösungen implentiert sein. In der Praxis gibt es eine Vielzahl möglicher Implementierungen, die sich was Komplexität und Benutzerfreundlichkeit angeht, stark unterscheiden [12].

SAML Ein in der Unternehmenswelt weitverbreiteter Ansatz ist die Security Assertion Markup Language (SAML), eine auf XML basierenden Architektur. Diese wurde 2002 von der Organization for Advancement of Structured Information Standards (OASIS) ins Leben gerufen. Es gilt als das Standardprotokoll für den Transport von Identitätsdaten über das Internet. [14]

Verlangt ein Nutzer Zugriff beim Service Provider, also dem Anbieter des jeweiligen Dienstes, wird vom Service Provider eine SAML Authentifierungsanfrage an den Identity Provider des Benutzers gesendet. Diese Anfrage wird im Header einer Simple Object Access Protocol (SOAP) Nachricht via HTTP übermittelt. Nach deren Eingang und Prüfung wird der Nutzer authentifiziert und erhält eine "Assertion" in einem Authentication Token, welche seine Identität und weitere Attribute enthält. Die Authentifizierung findet beim jeweiligen Dienst statt und wird nicht von SAML selbst übernommen, denn das Protokoll kümmert sich nur um den Transport der Information, welche den Nutzer später authentifiziert. Nach deren Verschlüsselung und dem Erhalt einer digitalen Signatur, kann diese über einen sicheren Weg zusammen mit eventuelle weiteren vom Service Provider angeforderten Daten retourniert werden. Dort wird die empfange Information auf Echtheit geprüft und deren Inhalt nach dem Entschlüsseln an die Anwendung weitergereicht. Um Angriffe zu vermeiden, wird empfohlen ein Token nicht mehrfach zu verwenden, da sonst die Gefahr besteht, dass das Token korrumpiert wird [13]. Daraufhin registriert die Anwendung den Nutzer im Sinne des Single-Sign On. Aufgrund der weiten Verbreitung bietet dieser Standard jedoch eine große Auswahl an Möglichkeiten einen internen Identity Provider zu realisieren.

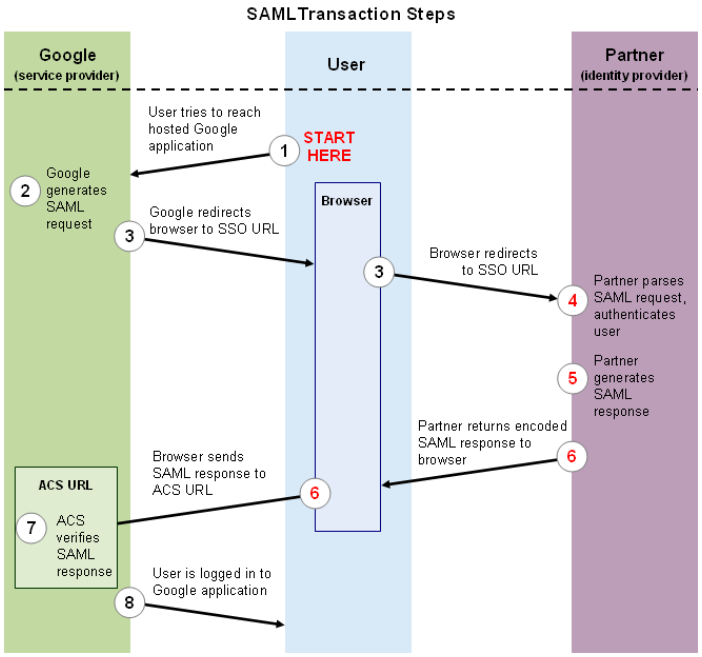


Fig. 1. Single Sign-On für Google Apps mit SAML

Nachfolgend werden zwei mögliche Single-Sign On Lösungen von Google beschrieben. Im Schaubild [A] ist die Variante zu sehen in der Google als Service Provider auftritt. Die Idenitäten werden durch einen Drittanbieter verwaltet. Dazu muss zunächst ein Private und Public Key zusammen mit einem X.509 Zertifikate welches Letzteren enthält generiert werden. Nach dem Erstellen durch einen von Google vorgegebenen Algorithmus für Signaturen, wird dieser Schlüssel durch den Upload bei Google registriert. Falls der Nutzer nun eine Dienst von Google aufruft, wird eine verschlüsselte SAML-Authentifizierungsanfrage zusammen mit der URL des Partnerdienstes an den Browser des Nutzers weitergeleitet. Beim Partner wird der Nutzer so entweder mit den Anmeldedaten des Nutzers oder durch gültige Cookies für die Sitzung authentifiziert. Daraufhin wird eine SAML Antwort mit dem Nutzernamen und dem signierten Schlüsselpaar wieder an den Browser des Nutzers übermittelt. Worauf die Antwort von Google's Assertion Consumer Service anhand des öffentlichen Schlüssels überprüft wird. Mit erfolgreicher Verifizierung des Nutzers wird dieser an die Ziel-URL weitergeleitet und für die Google Apps (G-Suite) angemeldet. [15]

Google bietet für Entwickler auch einen Identitäts- und Zugriffsverwaltungsdienst an, um die Authentifizierung und Autorisierung ihrer Benutzer mithilfe von SAML zu erleichtern (siehe Abb. 1). Das hat den Vorteil einer zentralisierten Benutzerdatenverwaltung der unterschiedlichen Cloudanwendungen. Dabei meldet sich ein Benutzer über einen Single Sign-On mit den Anmeldedaten der G-Suite bei einem der unterstützten Diensten an. In diesem Fall tritt Google als Identiy Provider auf, wobei die Konfiguration einer Vielzahl an Service Providern möglich ist. [16]

Ein großer Nachteil dieses Protokolls liegt in der Flexibilität. SAML ist weder für den Umgang mit mobilen Anwendungen, noch für den Einsatz mit Programmierschnittstellen gedacht. Nur über Umwege ist eine Nutzung von SAML für diese Anwendungsgebiete möglich, wodurch andere Lösungen im Internet der Dinge an Bedeutung gewinnen.

OpenID Connect Im Hinblick auf die Bedienbarkeit ist der Standard OpenID Connect deutlich simpler [17]. Da OpenID Connect auf OAuth 2.0 basiert, nutzt es die dort definierten JSON/REST-Nachrichten um Informationen für die Identitätsprüfung auszutauschen. Es dient zur Authentifizierung für Benutzer auf verschiedenen Webseiten und Diensten, ohne dass die Dienstbetreiber dabei die Passwortdaten ihrer Nutzer verwalten müssen. Trotzdem wird die Identität desjenigen klar, der die Anwendung bzw. den Browser benutzt. Dazu vergibt OpenID Connect sogenannte ID Tokens, die Aussagen über die Identität eines Benutzers treffen. Dieses Konzept wurde mit OpenID Connect eingeführt um die Nachahmung von Nutzern zu vermeiden, da OAuth selbst das Konzept der verschiedenen Identitäten nicht berücksichtigt. Der Gebrauch von OpenID Connect mit dem Browser funktioniert auf allen Mobilgeräten, wobei Androidgeräte

schon Systemschnittstellen für die Nutzung des Standards integriert haben.

Ein bekanntes Beispiel für den Einsatz von OpenID Connect zum Single Sign-On ist der Sign-in with Google. Dabei werden für die Authentifizierung und Weiterreichung der ID Tokens zwei Methoden Server Flow und Implicit Flow verwendet. Der Einsatz hängt vom Kontext der jeweiligen Anwendung ab: Hat die Applikation einen nachgeschalteten Server kommt der Server Flow angewandt, um eine Identitätsprüfung durchzuführen und falls ein direkter Zugriff durch die laufende Browser-Anwendung auf die Schnittstelle geschieht, kommt der Implicit Flow zum tragen. Bei letzterem empfielt Google, aufgrund von Sicherheitsbedenken auf die Implementierung des Google Sign-In zurückzugreifen. Der erste Schritt bei Umsetzung eines Server Flows ist das Erstellen eines sogenannten Anti-Fälschungs-Token um die Nachahmung eines Benutzers zu vermeiden. Daher wird zunächst ein Token benötigt, das den Zustand der Session zwischen dem Benutzer und der Anwendung behält. Anschließend wird eine Authentifizierung durch Formulieren einer Anfrage via HTTPS ersucht. Danach wird zum Beispiel eine ungefähr 30 Zeichen lange Zufallszahl oder ein Hashwert als Token mit der Antwort durch den Google OAuth-Logindienstes auf die Authentifizierungsanfrage abgeglichen, um sicherzugehen, dass sich kein bösartiger Angreifer als User ausgibt. Außerdem wird bei der Anfrage ein Code-Parameter zurückgegeben, den man beim Server gegen ein Access- und ID-Token einlösen kann. Das ID-Token, welches Informationen über die Identität des Nutzers enthält, ist ein JSON Web Token, dass von Google signiert wurde. Nach der Validierung kann das ID-Tokens für die Authentifizierung durch den Dienst verarbeitet werden. Falls der Benutzer bereits in der Datenbank des Dienstanbieters hinterlegt ist, wird der Sessionaufbau für die Anwendung durch Betreiber des Dienstes eingeleitet werden. Ansonsten wird der Benutzer auf ein Registrierungsformular weitergeleitet, welches durch die Informationen des Google-Benutzerkontos ganz oder fast gänzlich ausgefüllt werden kann. Google gehört damit zu einem der größten Identity Provider, die außerdem einen Authentifizierungsdienst mit OpenID Connect implementiert haben.

Facebook benutzt eine erweiterte Form von OAuth, die große Ähnlichkeit mit OpenID Connect hat, jedoch nur Facebook als Identity Provider zulässt. Im Gegensatz dazu bleibt mit OpenID Connect die Wahl des Identity Providers frei. Beim Dateiformat für die Authentifizierung nutzt OpenID Connect den JSON Web Signature der IETF, wohingegen Facebook auf eine proprietäre Lösung setzt. Neben der einfachen Handhabung bei der Authentifizierung bietet OpenID Connect die Möglichkeit benutzerspezifische Daten durch eine Public-Key-Verschlüsselung zu schützen. Inzwischen wird OpenID Connect von namenhaften Firmen wie der Deutschen Telekom, Paypal und Microsoft umgesetzt.

Directory Service Im Unternehmenskontext allerdings ist es nicht üblich auf die Lösung eines Internetgiganten zu setzen, sondern einen hauseigenen Identity

Provider zu implementieren. Neben dem Identity Provider sorgt eine Directory Service zusätzlich für die Verwaltung der Identitäten in einer Art Bibliothek. Hier kommen das Lightweight Directory Access Protocol (LDAP) oder die Active Directory von Microsoft zum Einsatz. Häufig finden sich Identity Provider und Directory Service in einer Lösung sowie zum Beispiel in Microsofts Azure Active Directory (siehe Abb. 2 [B]), die sich als Identity Provider zum einen um die Identitätsprüfung als auch um die Bereitstellung einer Authentifizierungsbibliothek kümmert.

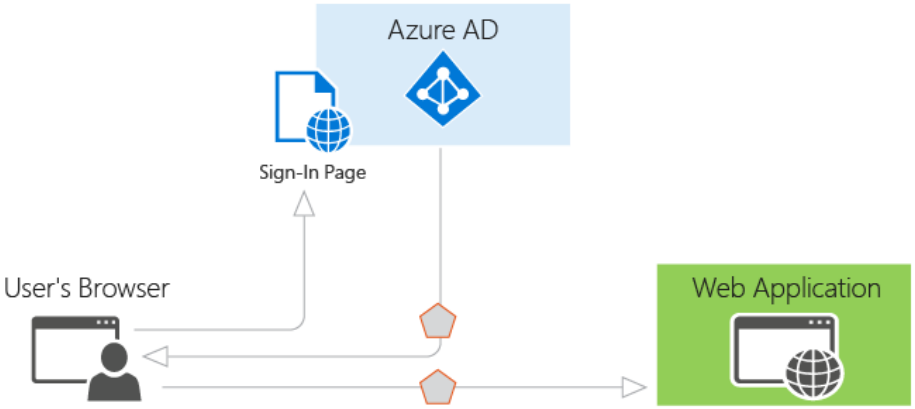


Fig. 2. Authentifizierung mit Microsoft Azure Active Directory

OAuth 2.0 Während sich Protokolle wie SAML oder OpenID Connect mit der Identität beschäftigen, befasst sich OAuth 2.0 mit der Zugriffsverwaltung auf Inhalte, die im Zusammenhang mit einer Identität stehen. Das können beispielsweise bestimmte Dokumente oder Adressdaten sein, welche durch das Protokoll geschützt werden sollen. Zusätzlich soll es Kunden von Internetdiensten ermöglicht werden, ihre Daten auch aus anderen Diensten heraus zu erreichen. Gelöst wird das durch sogenannte HTTP-APIs, mithilfe derer Schnittstelle für Applikationen implementiert werden können [18]. Um eine solche Programmierschnittstellen vor Missbrauch zu schützen, kümmert sich das Protokoll OAuth 2.0 um die Sicherheit ohne zugleich eine Einschränkung für die Performance zu verursachen. Einer der größten Vorteile beim Einsatz von OAuth ist sicherlich der Wegfall des Passwort Anti-Patterns, also die Verarbeitung von Benutzername und Passwort durch einen dritten Dienst. Um Daten des ursprünglichen Dienstes auf dem Dienst eines Drittanbieters zu nutzen, ermöglicht OAuth eine für jeden Dienst spezielle erlaubnisbasierten Prüfung der Anmeldedaten und verzichtet so auf eine vollständige Identitätsprüfung im ursprünglichen Sinn [19]. Während der Entwicklung stand vorallem die einfache Bedienbarkeit und ein Einsatzbereich sowohl für kleine als auch große Systeme im Vordergrund. Unter anderem wurde statt der Verwendung von digitalen Signaturen auf HTTPS, Zu-

fallswerte und Parameter mit nur einmaliger Verwendung gesetzt, was zu einer erheblichen Verbesserungen der Skalierbarkeit im Vergleich zur erste OAuth Implementierung führte. Abbildung 3 [C] soll die Rollenverteilung der bei OAuth 2.0 beteiligten Parteien verdeutlichen:

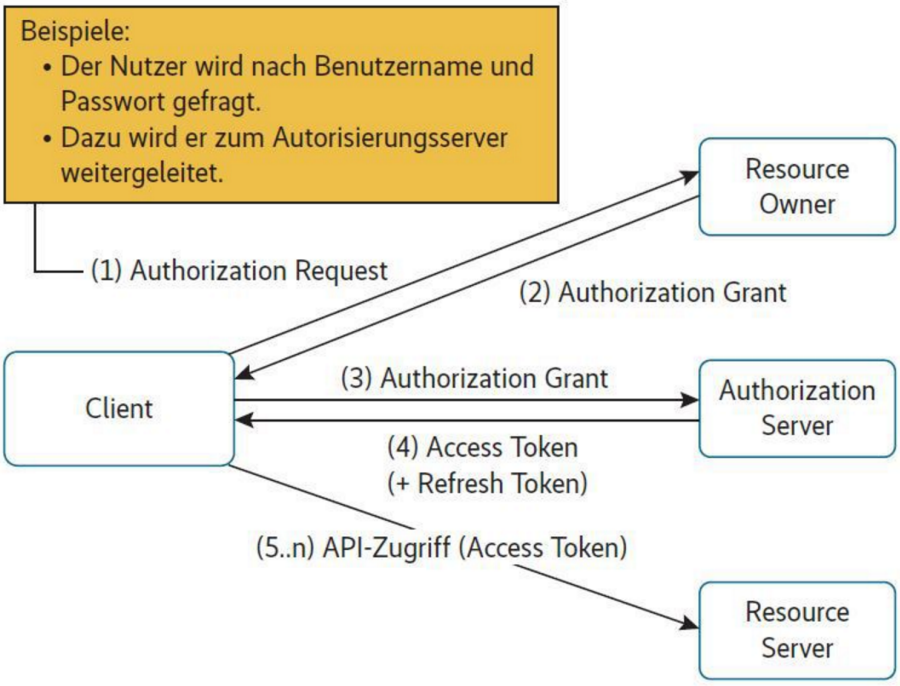
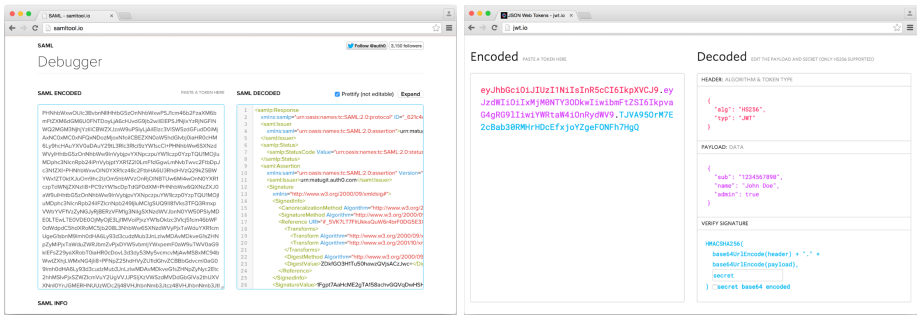


Fig. 3. Rollen der am OAuth-2.0-Protokoll beteiligten Entitaeten

Als Resource Owner gilt der Benutzer, da es seine Inhalte sind, die durch die Benutzerkontrolle von OAuth 2.0 gesichert werden sollen. Eine Anwendung, die auf die von OAuth geschützten Inhalte des Benutzers zugreifen will, hat hier die Rolle des Client. Um die Verwaltung der Inhalte kümmert sich allerdings ein Resource Server, der nur solche Zugriffe auf die Ressourcen des Benutzers zulässt, welche zuvor durch den Benutzer autorisiert wurden. Dabei wird der Benutzer zunächst einer Authentifizierung unterzogen, wobei anschließend noch die auszuführenden Zugriffe autorisiert werden müssen. Der Client erbittet (Authorization Request) zunächst die Zustimmung (Authorization Grant), was beim Benutzer in Form einer herkömmlichen Login-Abfrage bzw. durch eine Weiterleitung des Browsers durch einen vertrauenswürdigen Authorization Server erfolgen kann. Falls der Client die Zustimmung des Benutzers für den Zugriff erhält, erteilt der Authorization Server im Auftrag des Benutzers ein Access Token. Daraufhin erhält der Client Zugriff auf die jeweiligen Inhalte auf dem

Resource Server. Hier kann festgestellt werden von welchem Benutzer und für welche Rechte die Client-Anwendung autorisiert wurde. Da das Token nicht dauerhaft gültig bleibt, hat der Client neben einer erneuten Benutzeranfrage die Möglichkeit ein Refresh Token zu verwenden, dass zwar für längere Zeit Zugriff gewährt, desweiteren aber jederzeit vom Benutzer entzogen werden kann. Diese Form der Authentifizierung nennt man auch token-basiert [19]. Häufig wird OAuth für den Login auf Webseiten verwendet, was die eigentliche Intention des Protkolls verfehlt. Es ist als Autorisierungsprotokoll gedacht und soll nach dem Wunsch der IETF den herkömmlichen Authentifizierungsprozess ergänzen. Als Erweiterung von OAuth 2.0 gilt das JSON Web Token.

JSON Web Tokens Dieser Standard für ein Sicherheitstoken kümmert sich um die sichere Austausch von Javascript Object Notiation (JSON) Objekten, was sogar über ungesicherte Verbindungen wie z.B. HTTP möglich ist [20]. Der JSON-Standard wiederum ist ein leichtgewichtiges Format für den Austausch von Daten, das nicht an eine bestimmte Programmiersprache gebunden ist. Der Header des Tokens besteht aus nur zwei Schlüsseln, nämlich dem Typ also JWT und dem Algorithmus, mit dem es verschlüsselt. Das Token selbst enthält neben der für die Verarbeitung erforderlichen Informationen, wie Herausgeber und Gültigkeitsdauer auch Daten, die vom Benutzer bestimmt werden können. Als Herzstück gilt allerdings die digitale Signatur des JWT. Schließlich werden Header und Inhalt verschlüsselt und zusammen mit einem Secret signiert, das auf Sender- und Empfängerseite bekannt sein muss. Dadurch ist es nur den am Austausch beteiligten Parteien möglich die Identität des Senders und Datenintegrität der Nachricht zu überprüfen. Letztendlich besteht das Token aus drei Base64-kodierte Strings und enthält bis auf das Secret alle relevanten Informationen eines Benutzers in sich selbst, was weitere Datenbankanfragen einspart. Ein großer Vorteil von JWT liegt in der Kompaktheit durch den Einsatz von JSON, was verglichen mit XML-basierten Single Sign-On-Ansätzen wie SAML. Gerade deshalb ist der Standard für das Webumfeld (HTTP und HTML) sehr gut geeignet. Neben einer schnelleren Übertragung durch den relativ geringen Overhead, sind auch die Signatur und anschließende Weiterverarbeitung einfacher zu bewältigen als mit XML-Lösungen (Abbildung 4 [D]). Ein JSON Parser bildet ein Dokument direkt auf ein Objekt ab und ist den meisten Programmiersprachen bekannt. Daraus ergibt sich ein weiterer Vorteil gegenüber SAML, da XML eine solches Mapping nicht benutzt.



(a) SAML-Objekt (b) JSON Web Token

Fig. 4. Vergleich der Länge von verschlüsselten SAML und JWT Daten

Aufgrund dieser Eigenschaften wird JWT vorwiegend für die Authentifizierung insbesondere bei Single Sign-On Lösungen und zum sicheren Informationsaustausch verwendet, der durch die digitalen Signatur gewährleistet werden kann (Abbildung 5 [E]). Im Gegensatz zum traditionellen Sessionaufbau auf dem jeweiligen Server mit anschließender Cookievergabe, wird das Token nach erfolgreicher Anmeldung lokal beim Benutzer gespeichert.

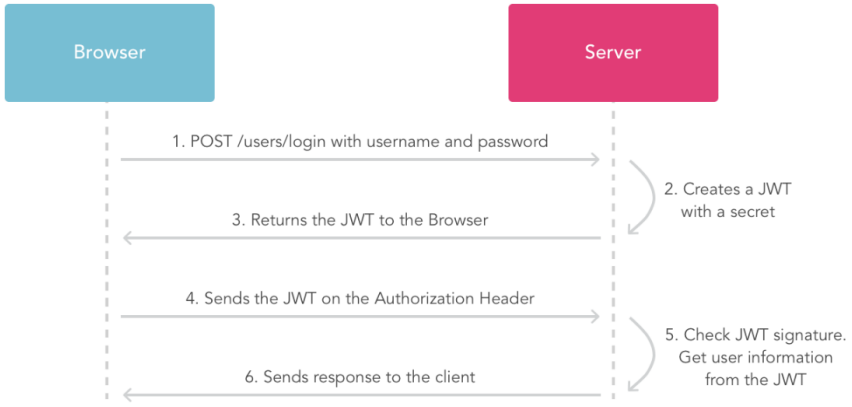


Fig. 5. Authentifizierung mit dem JSON Web Token

Nach dem Login wird das JWT bei jederen weiteren Anfrage mitgereicht und verhindert so die Notwendigkeit einer erneuten Authentifizierung des Benutzers. Der Zustand des Benutzers wird nie auf dem Server gespeichert, son-

dern das JWT wird im Header der Autorisierung vom Server auf Korrektheit überprüft. Woraufhin das Token seinem Benutzer den Zugriff auf die für ihn autorisierten Pfade, Dienste und Ressourcen erlaubt. Was die Sicherheit anbelagt, kann ein Token nur von beiden Seiten mittels eines gemeinsamen Secrets unter Verwendung des HMAC Algorithmus signiert werden. Abgesehen davon besteht auch die Möglichkeit der Signatur mit einem X.509 Zertifikat, welches ein Public und Private Key Paar verwendet. Allerdings besteht hier im Vergleich zur JSON-Signatur ohne vorherige Überlegung die Gefahr massiver Sicherheitslücken. Nicht zuletzt wegen der einfachen Verarbeitbarkeit kommt der Standard auch für mobile Plattformen und das Internet der Dinge in Betracht.

4.3 SCIM

Im Kontext der Identitätsverwaltung bei Cloudanwendungen befindet sich ein Provisioning-Protkoll, das 2011 unter dem Namen Simple Cloud Identity Management Protocol (SCIM) Protokoll entstand. Als die IETF sich der Weiterentwicklung des Prokoll annahm, wurde dieses jedoch umbenannt in System for Cross-domain Identity Management (SCIM). Der Entwurf ist relativ schlank und dient zur Identitäts- und Zugriffsverwaltung für die Menge an Cloudanwendungen für die sich ein bestimmter Benutzer registriert hat. Zur Verwaltung dient eine Schnittstelle mit einer REST-Architektur, welche direkte Operationen zur Benutzerverwaltung an den Anbieter seines Clouddienstes ausführen kann. Die Erstellung der Identitäten erfolgt just-in-time, d.h. im Zuge des ersten Single Sign-Ons wird ein Account für den jeweiligen Benutzer erstellt. Hierfür wird im Zusammenspiel mit SAML festgelegt, welche SCIM-Benutzerinformationen in die SAML-Assertion gepackt werden sollen. Vor allem in der Unternehmenswelt, bei der eine Vielzahl von Aufgaben an SaaS-Provider verlagert wird, kann ein solches Protokoll von erheblicher Bedeutung sein [21] .

4.4 Multi-factor Authentication

Neben dem Single Sign-On lassen sich noch weitere Konzepte für die Sicherheit im Internet der Dinge bzw. in Cloudanwendungen verwenden. Dazu gehört auch die Authentifizierung der Identität mit mehr als nur einem Faktor. Bisher haben wir im Single Sign-On meist von der Authentifizierung mittels Benutzername und Passwort gesprochen. Durch den Ansatz der multifaktoriellen Authefizierung soll der Login um ein oder mehrere Elemente erweitert werden. Neben einem bloßen Gegenstand wie etwa einem speziellen Schlüsselanhänger, kommen wegen der immer größeren Verbreitung Mobilgeräte für eine erweiterte Identitätsprüfung in Frage. Diese ersetzen die alte Hardware wegen der Vorteile des kostengünstigerem Einsatzes, der einfacheren Implementierbarkeit und der Möglichkeit, diese in bestehende Single Sign-On Lösungen einzubinden. Ein weiterer wichtiger Grund ist die Allgegenwärtigkeit von mobilen Endgeräten und die steigenden Erfahrung mit deren Umgang innerhalb der Gesellschaft, was den Einsatz im Kontext der Multi-factor Authentication begünstigen könnte. Denkbar sind zum Beispiel Apps oder Benarichtungen auf dem mobilen Endgerät welche

den Nutzer auffordern eine bestimmte Geste auszuführen oder einen bestimmten Code einzugeben. Nach der erfolgreichen Anwendung des zweiten Faktors ist der Benutzer authentifiziert. In Verbindung mit herkömmlichen Single Sign-On Verfahren könnte dieses Authentifizierungsmethode die Sicherheit um ein Vielfaches steigern [22].

5 Conclusion

Die vorgestellten Angriffsszenarien zeigen, dass sich das Internet der Dinge und Teile des Cloud Computing weiterhin mit dem Thema Sicherheit auseinandersetzen müssen, um einen angemessenen Security-Standard zu erreichen. Weiterhin sollen die aufgezeigten Sicherheitskonzepte darlegen, dass es auch hier Möglichkeiten gibt, Identitätsdaten zu schützen bzw. sicher zu übertragen. Neben der unterschiedlichen Beliebtheit, unterscheiden sich diese Ansätze meist in Komplexität und Architektur. Dennoch werden alle Ansätze von großen Internetfirmen implementiert und existieren Bestrebungen das Thema Sicherheit auch im Umfeld der Cloud bzw. der Dinge voranzutreiben. Vor allem die steigende Vernetzung stellt Entwickler wie Benutzer gleichermaßen vor die Herausforderung eine angemessene Lösungen für den Umgang mit personenbezogenen Daten zu finden.

References

[1] Babar, Sachin, et al. "Proposed security model and threat taxonomy for the Internet of Things (IoT)." International Conference on Network Security and Applications. Springer Berlin Heidelberg, 2010.

[2] Pescatore, John and Shpantzer, G "SANS Institute, January: Securing the internet of things survey" 2014.

[3] Infosec "Operational Technology vs. Information Technology" https://www.infosec.ch/faelle/fall10105.htm.

[4] Fraunhofer-Allianz Cloud Computing "Public, Private und Hybrid Cloud?" http://www.cloud.fraunhofer.de/de/faq/publicprivatehybrid.html.

[5] GlobalSign "5 häufige Cyber-Attacken im IoT – Bedrohungen im großen Stil" https://www.globalsign.com/de-de/blog/fuenf-haeufige-cyber-attacken-iot/.

[6] GlobalSign "Wie aus Botnetzen Thingbots werden" https://www.globalsign.com/de-de/blog/thingbots/.

[7] GlobalSign "Man-in-the-Middle-Angriffe im IoT" https://www.globalsign.com/de-de/blog/man-in-the-middle-attacke-im-iot/.

[8] GlobalSign "Vorsicht! Diebstahl von Daten & Identitäten im IoT" https://www.globalsign.com/de-de/blog/identity-theft-in-the-iot/.

[9] GlobalSign "Die Bedrohung der Sicherheit durch Social Engineering im IoT" https://www.globalsign.com/de-de/blog/social-engineering-iot/.

[10] GlobalSign "Heute geschlossen - Denial of Service Angriffe im IoT" https://www.globalsign.com/de-de/blog/denial-of-service-im-iot/.

[11] Nat Sakimura "Identity, Authentication + OAuth = OpenID Connect" https://www.youtube.com/watch?v=Kb56GzQ2pSk.

[12] Newman, Sam. Building Microservices. " O'Reilly Media, Inc.", 2015.

[13] InformationWeek "SAML: The Secret to Centralized Identity Management" http://www.informationweek.com/software/information-management/saml-the-secret-to-centralized-identity-management/d/d-id/1028656?.

[14] OneLogin "SAML Single Sign-On (SSO)" https://www.onelogin.com/saml.

[15] Google "Einmalanmeldung (Single Sign-On – SSO) für G Suite-Konten mithilfe von Identitätsanbietern von Dritten einrichten" https://support.google.com/a/answer/60224?hl=de.

[16] Google "SAML-basierte, föderierte Einmalanmeldung (SSO)" https://support.google.com/a/answer/6087519?hl=de.

[17] OpenID "OpenID Connect FAQ and Q & As" http://openid.net/connect/faq/.

[18] Lodderstedt, Thorsten et al. "Flexible und sichere Internetdienste mit OAuth 2.0". http://www.heise.de/developer/artikel/Flexible-und-sichere-Internetdienste-mit-OAuth-2-0-2068404.html.

[19] Bell, Gavin. Building social web applications. " O'Reilly Media, Inc.", 2009.

[20] AuthO "Introduction to JSON Web Tokens" https://jwt.io/introduction/.

[21] Ping Identity "SAML 101" https://www.pingidentity.com/content/dam/pic/downloads/resources/white-papers/en/saml-101-white-paper.pdf.

[22] Ping Identity "Ultimate Guide to SSO: Chapter 7 - Multi-Factor Authentication" https://www.pingidentity.com/en/ultimate-guides/ultimate-guide-to-sso/chapter-7.html.

[A] "Single Sign-On für Google Apps mit SAML" http://virtuallyhyper.com/2013/05/set-up-simplesamlphp-as-an-idp-to-be-used-in-an-sp-initiated-sso-with-google-apps/.

[B] "Authentifizierung mit Microsoft Azure Active Directory" https://azure.microsoft.com/de-de/documentation/articles/active-directory-authentication-scenarios/.
[C] "Rollen der am OAuth-2.0-Protokoll beteiligten Entitaeten" http://www.heise.de/developer/artikel/Flexible-und-sichere-Internetdienste-mit-OAuth-2-0-2068404.html.
[D] "Vergleich der Länge von verschlüsselten SAML und JWT Daten" https://jwt.io/introduction/.
[E] "Authentifizierung mit dem JSON Web Token" https://jwt.io/introduction/.

IDEs for Creating Mobile Experience Sampling Apps for Non-Programmers

Margarita Asenova*

Advisor: Anja Bachmann†

Karlsruhe Institute of Technology (KIT)
Pervasive Computing Systems – TECO
*uwdgn@student.kit.edu
†anja.bachmann@teco.edu

Abstract. Experience Sampling Methode ist eine Befragungstechnik zum Erfassen von Gefühlen, Stimmungen und Gedanken der Menschen mithilfe von täglichen Umfragen, die entweder per Hand, durch Audio-, Videoaufzeichnung oder einfach auf dem Smartphone ausgefüllt werden. Diese Seminararbeit beschäftigt sich mit dem Thema Experience Sampling Methode, deren Einsatzgebiete und Entwicklungsumgebung. Es werden bereits vorhandene IDEs und Tools vorgestellt und verglichen. Im Zuge diesen werden auch die meistbenutzten Sensoren und Event-Triggers vorgestellt. Zum Abschluss wird ein Überblick zum aktuellen Forschung ausgegeben.

Keywords: Experience Sampling Method, ESM Tools, Einsatzgebiete, Smartphones, Ambulatory Assessment

1 Einleitung

Experince Sampling Methode(ESM) ist inzwischen eine weitverbreitete Befragungstechnik und wird oft auch von Leuten ohne grundlegenden Programmierkenntnisse genutzt. Diese Methode zielt darauf ab bestimmte Information über Gefühle, Gedanken, Stimmungen usw. im Alltag der Probanden zu erfassen. Wegen der schnellen Verbreitung von Smartphones ist ESM sehr leicht einzusetzen, ohne dass dafür zusätzliche Kenntnisse erforderlich wären. Es sind auch keine Schulungen für die Benutzer notwendig, denn heutzutage kann jeder ein Smartphone benutzen. Im Vergleich zu der Vergangenheit, als diese Befragungen mit Kugelschreiber und Papier durchgeführt wurden, ist heutzutage nur ein Smartphone notwendig und diese können sogar wegen der aufgenommenen Daten von Sensoren während bestimmten Zeiten oder zwischen bestimmten Zeitintervallen durchgeführt werden. Genau dann, wenn solche Ereignisse eintreten, die für den Forscher interessant und relevant für die Studie sind. Wie genau eine Studie mit ESM durchgeführt wird, wird mithilfe von dem folgenden Beispiel veranschaulicht. Der Benutzer bekommt eine Benachrichtigung auf dem Handy, wenn ein für die Studie relevantes Ereignis eintritt. Ein solcher Ereignis können

die Änderungen der Wettertemperaturen sein. Immer wenn die Temperatur über 30, zwischen 20-30 und unter 20 Grad Celsius ist, bekommt der Befragte eine Benachrichtigung zum Ausfüllen einer Befragung. Wie er sich bei dieser Temperatur fühlt? Hat das irgendeinen Einfluss auf seine Arbeit? .... Die Antwort kann in Textform eingegeben oder eine aus vorgeschlagenen Möglichkeiten ausgewählt werden. So kann beispielsweise eine Studie durchgeführt werden zum Thema "Welche Temperatur der Benutzer als angenehmste bei seiner Arbeit im Büro empfindet". Nach dieser Studie kann eine Schlussfolgerung zum Thema "Sind Klimaanlagen bei der Arbeit im Büro notwendig¿' gezogen werden. Als erstes wird in dieser Seminararbeit die Experience Sampling Methode vorgestellt und erklärt was hinter dieser Methode steckt. In welchen Bereichen diese am meisten verbreitet ist und ein paar Beispiele dazu gegeben. Wie werden die unterschiedlichen Sensoren von Smartphones hier eingesetzt und was für Arten von Triggers existieren. Als letztes werden einige von den bekanntesten und meistbenutzten ESM-Tools vorgestellt und einen Vergleich zwischen ihnen gezogen.

2 Grundlagen

2.1 Definition von dem Experience Sampling Methode

Die Experience Sampling Methode wird von Schallberger als 'Befindenstagebuch' oder 'Erlebnisstichprobenmethode' definiert. Diese Methode ist eine Befragungstechnik, mit deren Hilfe die Stimmungen, Gedanken und Gefühle einer Person im Echtzeit gemessen werden. Dabei werden die Daten direkt in der Situation, in der sie erlebt werden gesammelt. Diese Situationen entsprechen bestimmten Perioden im Alltag der Befragten beispielsweise beim sonnigen Wetter, dem Aufenthalt an irgendeinem Ort, nach dem Essen usw. Nach Barett und Barett wurde dieses Verfahren früher auf Papier während festgelegten Uhrzeiten durchgeführt. Heutzutage erleichtert die Benutzung von Smartphones diese Methode erheblich, denn der Proband soll sich bis auf das eigentliche Ausfüllen der Befragung um fast nichts kümmern. Wenn es notwendig ist, eine Umfrage auszufüllen, erscheint eine Benachrichtigung auf dem Handy. So eine Situation ist zum Beispiel der Zeitpunkt nach dem täglichen Sport. Der Befragte bekommt immer nach einer Sporttätigkeiten eine Benachrichtigung. Die Antwort kann in Textform eingegeben oder eine aus vogeschlagenen Möglichkeiten ausgewählt werden.

2.2 Hintergrund

Die Experience Sampling Methode spielt heutzutage besonders große Rolle dank der sehr stark steigenden Entwicklung von Smartgeräte, mithilfe von denen kann man diese Methode sehr leicht benutzen, ohne dass irgendwelche Vorkenntnisse außer normalem Bedienen vom Smartphone verlangt zu werden. Dieses Verfahren verlässt sich nicht auf Erinnerungen und künstlicher Kontext, sondern auf den exakten Augenblick, wo die Beteiligten an dieser Befragung die benötigte Beschreibung von ihren vorübergehenden Gedanken, Gefühlen und ihrer Verhaltensweise in den zahlreichen alltäglichen Situationen liefern. Die Teilnahme von

Smartgeräten an ESM lässt den Forschern die Möglichkeit, diese Befragung in unterschiedlichen Umgebungen, wie z.B. zu Hause, im Urlaub, bei der Arbeit, durchzuführen und unter bestimmten Ereignissen wie Sport, Urlaub, schönes oder schlechtes Wetter usw. auszulösen.

Die Experience Sampling Methode kann entweder täglich auf bestimmte Intervalle auftreten(Interval contingent), als Reaktion zu bestimmten Ereignissen, für die sich der Forscher interessiert(Event Contingent), oder täglich zu einem zufällig ausgewählten Zeitpunkt(Signal Contingent). Mehr über diese Triggermethoden wird im Kapitel 4 erläutert. Die heutige Experience Sampling Methode, die mit der Teilnahme von Smartgeräte durchgeführt wird, reduziert deutlich die Verwendung von der alten Methode, die mithilfe von Papier und Stift durchgeführt wird. Experience Sampling Methode verlangt keinen Abruf und Wiederaufbau der Daten im Gedächtnis, sondern bringt den Zugang und präzisen Bericht von der momentanen Information im Gedächtnis.

Das Experience Sampling Verfahren nicht nur verringert die Nachteile von alter Befragungstechnik, sondern hat auch deutliche Vorteile. Diese Technik erlaubt dem Forscher, einen Probanden mithilfe von der gegründeten vom Webplattform Applikation zu beschreiben, wo der Probande die Befragung auf sein Handy ausgefüllt hat und weder der Forscher oder der Probande Programmierkenntnisse brauchen.

Als etwas Positives kann man auch bezeichnen, dass die Befragten sich in ihren natürlichen Umgebung befinden. Die Umgebung, in der man sich während der Befragung befindet, spielt sehr große Rolle, denn so sind die Antworte von den Befragten objektiver, da sie sich nicht auf vergangenen Erinnerungen, Situationen oder Umgebungen verlassen, sondern genau auf diesen Augenblick. Ein anderer Vorteil ist, dass die Befragungen jeder Zeit und mehrmals pro Tag durchgeführt werden können oder z. B. während einem bestimmten Ereignis, die mithilfe von Sensoren erkannt werden kann, aber die unterschiedlichen Sensoren werden später im Kapitel 4 aufgezählt.

3 Einsatzgebiete von ESM

Die Entwicklung von der Experience Sampling Methode fängt im Bereich der Psychologie an und heutzutage ist sie da am meisten verbreitet. Trotzdem gewinnt diese Technik immer mehr an Bedeutung und sie wird in letzten Jahren in zahlreichen Gebieten verwendet. Demzufolge werden hier einige von den Bekanntesten aufgezählt, und einige Beispiele dazu gegeben, wie diese Methode in unterschiedlichen Bereichen angewendet wird.

3.1 Psychologie

Die ESM findet weitgehend Verwendung in diesem Bereich. Die Existenz von zahlreichen ESM-Umfragen in diesem Feld ist einen Beweis wie oft diese Methode in der Psychologie benutzt wird. Psychlog ist ein Tool, das genutzt wird,

um physiologische, psychologische und allgemeine Information über die Aktivitäten des Benutzers zu sammeln zur Erforschung der psychischen Gesundheit. Es sammelt Puls- und Aktivitätsinformation von einem "Wireless Electrocardiogram mit dreiaxialem Beschleunigungssensor". Dieses Software ist kostenlos und kann auf Windowsgeräten ausgeführt werden. Mithilfe von diesem ESM Tool kann der Verlauf von des Pulses gemessen werden und demzufolge Krankheiten wie Depression, Panikstörungen, Sinus-Arrhythmie und andere erkannt werden. Die Forscher können sehr leicht die Umfragen anpassen, wenn sie die Text-Datei ändern. In diesen Dateien können sie die Fragen, die Art der Antworten (Text eingeben, Auswahl von ein paar Möglichkeiten oder mit ++, + , 0, - , – bewerten) und benutzte Trigger einstellen. Es werden auch ein paar Anpassungsmöglichkeiten von dem System an den Forschern vorgeschlagen, die mit Passwort geschützt sind, damit der Benutzer nichts an der Umfrage ändern kann. Obwohl Psychlog nicht so technisch ausgereift im Vergleich zu MyExperience oder andere Systeme ist, bietet es trotzdem einige Vorteile wie:

- Kann auch auf die günstigen Windowssmartphones ausgeführt werden
- Keine Programmierkenntnisse werden von den Forschern verlangt

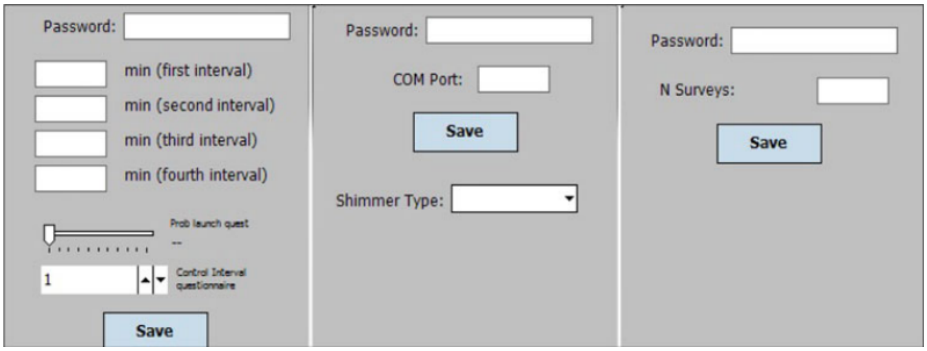


Fig. 1. Systemkonfigurationen von Psychlog; Trigger Konfiguration, ECG Sensorkonfiguration, Umfragekonfiguration

3.2 Ausbildung

Die Experience Sampling Untersuchungsmethode bietet den Erziehungswissenschaftlern ein innovatives Vorgehen zum Untersuchen des alltäglichen Lebens der Schüler, Eltern oder Lehrern. Dank dieser Methode kann sehr leicht die Veränderung im Verhalten der Studenten oder die Qualität in der Schule erfasast und verbessert werden.

3.3 Neugestaltung von Wohnungen

Die Fakultät für Architektur an MIT entwickelt die Idee für ein bild-basiertes Experience Sampling Verfahren um die Präferenzen und Bedürfnisse der Benutzer bei der Neugestaltung von Küchen aufzunehmen [8]. Die Forscher am MIT kombinieren die allgemeine ESM und Conjoint-Analyse um kontextspezifische Rückmeldung/Feedback zu erhalten. Die allgemeine Methode von Experience Sampling stört den Benutzern bei der Aktivität und es wird oft verlangt, dass der Benutzer seine aktuelle Tätigkeit unterbricht, um die Fragen zu beantworten. Genau deswegen haben die Forscher des MIT dieses bild-basierte Verfahren entwickelt, um diese Unterbrechungen zu vermeiden. Die Neuerung bei dieser bild-basierten Technik ist, anstatt dass der Benutzer durch den Fragebogen gestört wird, werden Fotos oder Videos während der aktuellen Tätigkeit aufgenommen. Der Benutzer wird erst später in einem passenden Moment befragt. Mithilfe dieser Fotos oder Videos erinnert er sich an diesem Moment. Damit dieses Experiment durchgeführt werden kann, sind zwei Kameras in gegenüberliegenden Ecken des Zimmers angebracht. Außer der Möglichkeit Videos und Fotos aufzunehmen, haben beide Kameras integrierte Bewegungsmelder, damit sie erkennen können, wenn der Benutzer das Zimmer betritt. Nach dieser Aktivität kann der Benutzer die Aufnahmen auf seinem Mobilgerät nachschauen und den Fragebogen ausfüllen.

4 Sensorik und Event-Triggers bei ESM

Wegen der ständig wachsenden Benutzung von sensorreichen Smartphones wird die Anwendung von ESM weiter erleichtert, denn diese Sensordaten können als Hilfsmittel zur Durchführung von ESM-Studien benutzet werden. Der Schwerpunkt einer sensorbasierten ESM Untersuchung liegt in der Entscheidung über das Design und wann oder unter welche Bedingungen eine Benachrichtigung verschickt werden soll, damit der Benutzer die Umfrage ausfüllt. Bei dem Einrichten des Triggers kann der Benutzer seine Präferenzen äußern, nach denen sich die Trigger richten sollen, zum Beispiel wie oft pro Tag eine Benachrichtigung ausgelöst werden darf, zwischen welche Uhrzeiten und wie lange diese Umfrage dauern soll.[1].

Bei dieser Befragungstechnik werden drei Grundmethoden benutzt, damit die Benachrichtigung ausgelöst wird:

- Zeit-basierter Trigger:
  - Die Benachrichtigungen werden täglich an den Beteiligten um eine festgestellte Uhrzeit geschickt,die jeden Tag unverändert bleibt.
  - Die Benachrichtigungen werden nur innerhalb von bestimmten zeitlichen Intervallen geschickt. Zum Beispiel: zwischen 08:00 - 20:00
- Signal-basierter Trigger - Die Benachrichtigungen werden immer um zufällig ausgewählte Uhrzeit geschickt
- Ereignis-basierter Trigger - Die Benachrichtigungen werden nur dann geschickt, während ein bestimmter Ereignis auftritt.

Außer der drei Grundtypen können die ESM Untersuchungen auch durch eine Kombination von den oben aufgezählten Typen ausgelöst werden. Ein Beispiel dafür ist die folgende ESM Studie. Die Auslösung von der Benachrichtigungen wird hier eine Kombination zwischen zeit-basierten und ereignis-basierten Triggers. Der Erste wird benutzt um ein Zeitintervall yu bestimmen, innerhalb dessen der Beschleunigungssensor abgefragt werden kann und der zweite Trigger wird dann benutzt, wenn die Sensordaten zeigen, dass es keine Bewegung gibt.Dann soll eine Umfrage durchgeführt werden.

Bei der Sensoren kann man auch zwischen zwei Oberarten einen Unterschied machen.

- Physikalischen Sensoren - Sensoren, die aktiv nach den Daten befragt werden sollen (Android Betriebssystem)
  - Beschleunigungssensor
  - Mikrofon
  - Näherungssensor
  - Bluetooth
  - GPS
  - Wi-Fi
- Software Sensoren - Sensoren, die die Daten mithilfe von dem Smartphone Hardware übertragen.
  - Telefonanrufe

- Bildschirm ein/aus
- SMS
- und andere

Die Speicherung und das Hochladen von Sensordaten wird von zwei unterschiedlichen Komponenten durchgeführt, wobei das Hochladen normalerweise nur unter existierende Wi-fi Verbindung ausgeführt wird. Mithilfe von diesem automatisierten Speichern und Hochladen von Daten wird die Beteiligung von den Benutzern an der Studie noch mehr erleichtert. Das System kann auch so eingerichtet werden, dass es sich automatisch aktualisiert ohne die Beteiligung vom Benutzer daran.

5 Programmierumgebungen für Experience Sampling

5.1 Bekannte Programmierumgebungen

Die Experience Sampling Methode ist sehr schnell populär geworden, besonders in den letzten Jahren mit der Entwicklung von Smartgeräten. Consolvo und Walker haben die ESM zur Auswertung von dem Inter Research System, die Personal Server genannt wird, benutzt. Bezogen auf diese zunehmende Beliebtheit ist die Entwicklung von Tools, um diese Methode zu unterstützen, notwendig. Intille et al. ....(Referenz hinzufügen) hat eine Software entwickelt, die den Forschern ermöglicht, eine Rückmeldung nur in bestimmten Situationen zu bekommen, die mithilfe von den Sensoren, die mit einem PDA verbunden sind, erkannt werden. Froehlich entwickelt MyExperience (Referenz hinzufügen), ein System zum Erfassen von objektiven und subjektiven in situ Daten von den Aktivitäten des Handys.

Tools.png

	MyExperience	Ohmage	movisensXS	MetricWire	PACO	ESMAC	Jeeves
OpenSource	Ja	Ja	Begrenzt	Begrenzt	Ja	Ja	Ja
Plattform	Windows	Android	Android	Alle	Android IOS	Android	Android
Endgerät	Computer PDA Smartphone	Smart-phone	Smartphone Tablets	Smart-phone Computer	Smart-phone Tablet	Smart-phone	Smart-phone
Funktion	Online Offline	Online Offline	Online Offline	Online Offline	Online	Online	Online Offline
Sensoren	kommunikation, Nutzung des Handys und der App, Benutzer-kontext, GPS, Blueto oth, wireless, GSM	Beschleun igungs-sensor, wifi, mobiles Radiozell, GPS	GPS, beschleunigu ngssensor, wearable sensors	Handy-nutzung und GPS	Handy-nutzung	Uhrzeit, call log, Bluetooth, Benach-richtigung en, Display-zustand, wetterapp beschleun igungs-sensor, GPS, Licht	Beschleunig ungs-sensor, Licht, SMS GPS, Gyroscope
Trigger	Ereignis-basierte Trigger	zeit-basierte, signal-basierte und location-basierte Trigger	signal-basierte und location-basierte Trigger	zeit-basierten und ereignis-basierten Triggers	zeit-basierten und ereignis-basierten Triggers	zeit-basierten und ereignis-basierten Triggers	zeit-basierten, ereignis-basierten und domain-spezifische(abhängig von der Antwort in der Umfrage)

5.2 Vergleich zwischen den unterschiedlichen Programmierumgebungen

Um den Forschern ihre Arbeit zu erleichtern, sind einige kommerzielle und einige open source EMS Tools entwickelt worden, so entfällt die Notwendigkeit des Lernens einer Programmiersprache. In der Tabelle ist eine kleine Übersicht über die meist benutzten Tools dargestellt. Im Normalfall setzen sich diese Tools aus einer Webschnittstelle, die die Spezifikationen von dem Studie aktiviert/freischaltet und aus zeitlichen Frameworks zur Aufstellung von diesen Umfragen zusammen. Nachdem wir schon einen Überblick über die aktuellen EMS-Tools bekommen haben, können wir die Schwachstellen von diesen Tools zeigen und ein paar Lösungen für diese Schwachstellen vorstellen.

Klare Unterschiedsgrenze bei den Software Tools[4]:

- Benutzeroberfläche
- Kommerzielles und kostenloses Software
- Mangel an Sensorverwendung
- Störung der Nutzer während der Umfrage

Eines des hervorragendsten ESM Tools heutzutage ist MyExperience. Dieses Tool ist nur für Windows Smartphones geeignet und verlangt die Installation von Microsoft SQL Datenbank, da die Information lokal auf dem Handy gespeichert wird. Dieses System wurde speziel für Handys entwickelt, um die Vorteile der zahlreichen Sensoren auszunutzen. Es unterstützt 50 integrierte Sensoren einschließlich GPS, GSM basiertes Bewegungssensor und Information der Gerätenutzung. Die Sensorereignisse können zum benutzerdefinierten Aktionen wie Synchronisation von der Datenbank, Senden eines SMS an Forschungsteam oder Darstellung von Selbstbeurteilungsumfragen in situ genutzt werden. Trotz aller Vorteilen und Neuigkeiten von MyExperience ist das System nicht komplett perfekt, denn es werden Grundprogrammierkenntnisse in XML verlangt um eine Umfrage zu erstellen[12].
Im Vergleich zu MyExperience werden bei MovisensXS keine Programmierkenntnisse verlangt, denn die Benutzeroberfläche von diesem Tool ist mithilfe von einer grafischen Programmiersprache erstellt. Momentan ist das eine der besten Benutzeroberflächen. MovisensXS ist inspiriert von MyExperience, aber es ist nicht für Windows Smartphones geeignet, sondern für Android und es sind signalbasierte und begrenzte sensor-basierte Trigger wegen des Sparens von Akkuenergie erlaubt. Hier können die Umfragen online erstellt und danach von dem dem Android Client heruntergeladen werden. Hauptnachteil dieses Systems ist , dass es nicht kostenlos und frei zugänglich ist.
PACO ist ein Tool, das sehr gut auf seiner Server und Client Seite dokumentiert ist, um die Integration mit anderen Anwendungen zu erleichtern. Beim Nutzen von PACO ist eine Anmeldung nicht erforderlich, damit der Befragte die App nutzen kann oder an der Umfrage teilnimmt. PACO verbindet sich automatisch mit dem Google Account des Benutzer. Allerdings kann dieses System nur dann benutzt werden, wenn Internetverbindung besteht und es verfügt nur über zeitbasierte Trigger[9].

Das Ohmage Tool benutzt zeit- und positionbasierte Trigger, aber die zweite Art von Trigger ist nur auf GPS-Daten begrenzt. Also, ohne Netzwerkposition. Die Umfragen sind als XML-Dateien(gezeigt in Fig.3) definiert und sie können nicht von den Benutzern verändert werden. Ein der Nachteile von Ohmage ist genau dieser Mangel an Automation. Anderer Nachteil von Ohmage ist, dass es keine Umfragedaten hochladen kann, bis eine Position mithilfe von den Sensoren gefunden wird[7][11].
Das MetricWire System kann sowohl offline als auch online benutzt werden, wenn die App schon heruntergeladen wurde. Die Daten, die erstellt werden, werden erst dann auf dem Server hochgeladen, wenn eine Verbindung zur Verfügung steht. Dieses Tool läuft auf allen Betriebssystemen und kann sowohl auf dem Computer als auch auf ein Handy benutzt werden, wenn es gekauft wird, denn dass System ist nur im begrenzten Fall kostenlos.
ESMAC kombiniert die Web-Schnittstelle für Nichtprogrammierer von MovisensXS zum Erstellen von individuellen ESM Apps und die XML-basierten Konfiguration von Event-triggers, die im System MyExperience benutzt wird[2].
Jeeves ist eine grafische Programmierumgebung, die die Erstellung von ESM Applikationen erleichtert. Die blockbasierte Notation Forschern grafisch eine ESM Umfrage zu erstellen, ohne dass irgendwelche Kenntnisse dafür vorausgesetzt sind. Die Autoren von "Jeeves – A Visual Programming Environment for Mobile Experience Sampling" haben nach dem Vorstellen von Jeeves eine Umfrage erstellt, wobei sie 20 Leute, Programmierer und Nichtprogrammierer, gefragt haben, wie sie mit Jeeves umgehen. Mithilfe von dieser Umfrage haben sie bewiesen, dass es keine Rolle spielt, ob man Programmierkenntnisse besitzt oder nicht, denn jeder kann genauso problemlos das System benutzen.

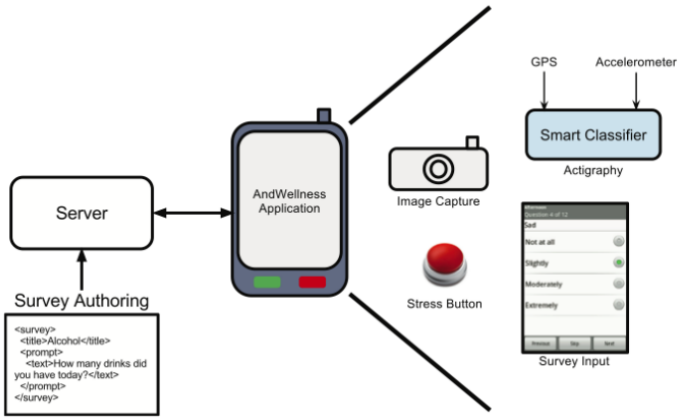


Fig. 2. Beispiel einer Benutzerumfrage mithilfe von Ohmage Tool

6 Fazit

Ein ideales Tool soll über eine zahlreiche Anzahl an Sensoren und Triggers verfügen[9], mithilfe von denen eine Umfrage zeit-, event-, signal- und position-basiert ausgelöst werden kann. Das System soll intuitiv und benutzerfreundlich sein, ohne irgendwelche Kenntnisse zu verlangen und möglichst energiesparend, denn die Verfügbarkeit aller dieser Sensoren und das aktive Programm auf dem System im Background wirkt sehr stark auf die Akkulaufzeit des Gerätes aus. Die Existenz einer Client/Server Architektur spielt auch sehr große Rolle, damit die App auch offline benutzt werden kann und wenn eine Wi-fi Verbindung besteht die Daten an dem Server geschickt werden können. Welches ESM-Tool kommt dem Ideal nahe?

References

1. Bachmann, A., Zetzsche, R., Riedel, T., Beigl, M., Reichert, M., Santangelo, P., Ebner-Priemer, U.: Identification of relevant sensor sources for context-aware esm apps in ambulatory assessment. In: Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers. pp. 265–268. ACM (2015)
2. Bachmann, A., Zetzsche, R., Schankin, A., Riedel, T., Beigl, M., Reichert, M., Santangelo, P., Ebner-Priemer, U.: Esmac: A web-based configurator for context-aware experience sampling apps in ambulatory assessment. In: Proceedings of the 5th EAI International Conference on Wireless Mobile Communication and Healthcare. pp. 15–18. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering) (2015)
3. Barrett, L.F., Barrett, D.J.: An introduction to computerized experience sampling in psychology. Social Science Computer Review 19(2), 175–185 (2001)
4. Fischer, J.E.: Experience-sampling tools: a critical review. Mobile living labs 9, 1–3 (2009)
5. Froehlich, J., Chen, M.Y., Consolvo, S., Harrison, B., Landay, J.A.: Myexperience: a system for in situ tracing and capturing of user feedback on mobile phones. In: Proceedings of the 5th international conference on Mobile systems, applications and services. pp. 57–70. ACM (2007)
6. Gaggioli, A., Pioggia, G., Tartarisco, G., Baldus, G., Corda, D., Cipresso, P., Riva, G.: A mobile data collection platform for mental health research. Personal and Ubiquitous Computing 17(2), 241–251 (2013)
7. Hicks, J., Ramanathan, N., Falaki, H., Longstaff, B., Parameswaran, K., Monibi, M., Kim, D.H., Selsky, J., Jenkins, J., Tangmunarunkit, H., et al.: ohmage: An open mobile system for activity and experience sampling. Los Angeles: UCLA Center for Embedded Network Sensing (2011)
8. Intille, S., Kukla, C., Ma, X.: Eliciting user preferences using image-based experience sampling and reflection. In: CHI'02 Extended Abstracts on Human Factors in Computing Systems. pp. 738–739. ACM (2002)
9. Kini, S.: Please take my survey: compliance with smartphone-based ema/esm studies (2013)

10. Lathia, N., Rachuri, K.K., Mascolo, C., Rentfrow, P.J.: Contextual dissonance: Design bias in sensor-based experience sampling methods. In: Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing. pp. 183–192. ACM (2013)
11. Ramanathan, N., Alquaddoomi, F., Falaki, H., George, D., Hsieh, C.K., Jenkins, J., Ketcham, C., Longstaff, B., Ooms, J., Selsky, J., et al.: Ohmage: an open mobile system for activity and experience sampling. In: 2012 6th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth) and Workshops. pp. 203–204. IEEE (2012)
12. Rough, D., Quigley, A.: Jeeves-a visual programming environment for mobile experience sampling. In: Visual Languages and Human-Centric Computing (VL/HCC), 2015 IEEE Symposium on. pp. 121–129. IEEE (2015)
13. Schallberger, U., Pfister, R., Venetz, M.: Qualität des erlebens in arbeit und freizeit. theoretische rahmenüberlegungen zum erlebens-stichproben-fragebogen (esf) und zu den operationalisierungen. Arbeitsberichte Aus Dem Projekt QualitäT Des Erlebens in Arbeit Und Freizeit (1999)

Privacy Issue Of User Modeling In Pervasive Systems

Nick Newill Tchouante Kembe*

Advisor: Antonios Karatzoglou†

Karlsruhe Institute of Technology (KIT)
Pervasive Computing Systems – TECO
*nick.kembe@student.kit.edu
†antonios@teco.edu

Abstract. The most widely used notion of privacy is that of control over personal information. The importance of social network sites and services have been increasing since some years. The introduction of ubiquitous systems, wearable computing and "Internet of Things" technologies in our digital society results in a large-scale data generation. In the same time there were an increasing number of socially intelligent agents who gather, manage and maintain the data. In this paper we will focus on two aspects: the location and user modeling. We learn about some techniques or methods used for providing privacy to the users. These methods are interesting, but not perfect. It's why we also take a look semantic-based methods, in order to have more reliable methods providing privacy.

Keywords: Privacy, Transparency, Anonimity, user modelling, location-based services

1 Introduction

It has been tacitly acknowledged for many years that personalized interaction and user modeling have significant privacy implications, due to the fact that personal information about users needs to be collected for performing personalization. The personalization is based on user models. Some user model dimensions can typically be observed by the system directly, some user model dimensions may require additional acquisition and inference steps, again others are entered into the system by self-reports by the users. The categorization of user model in figure 1 classifies several groups of user dimensions that can be identified. [3] contains more information about this ontology. The motion, location and orientation are important for the user model, so it's difficult to think about user model without location, since both are intrinsically intertwined. Computing and online services are increasingly being consumed through mobile devices, including smartphones and tablets. Location-based services (LBS) have become an integral part of users' experiences and an increasingly important market. They deliver to users targeted, relevant and highly convenient information, such as

Fig. 1: Groups of basic user dimensions[3].

up to-the-minute traffic reports; the location of the nearest petrol stations, hospitals, or banks; as well as targeted advertisements and coupons for services located in a consumer's immediate range.

As increasing amounts of personal data are being held, the risk of breaching data protection legislation and regulation has grown ever greater. At the same time, data protection laws are tightening across the world in response to consumers' and citizens' concerns. As part of a broader information governance strategy, some organizations are making greater use of more automated controls to manage data protection [1]. The situation of the privacy decreased in the past years for two reasons [2]: personalized systems moved to the web and to the mobile devices – gathering a lot of personal data about location –; and restriction are imposed by the privacy legislation.

Under the scholars, all the techniques used for realizing the data protection are commonly referred to as **Privacy Enhancing Technologies (PET)**. But there is not a common definition of PET. It typically refers to the use of technology to help achieve compliance with data protection legislation [1]. However, in the business application there are not limited to making personal information confidential, for example many of the technologies referred to as PETs can protect corporate confidential information and protect revenues by securing the in-

tegrity of data. There are 7 main PET technologies [1]: **Encryption; Metadata and Digital Rights Management; Application Programming; System Development Governance; User Interface; Identity Management** and **Architecture**

The rest of the paper is organized as follows. Section 2 presents a categorization of location privacy techniques. In Section 3 and 4 we take look presents respectively one privacy technique for user modeling and location-aware service. Section 5 and 6 present respectively semantic-based privacy technique user modeling and location-based services. Section 7 concludes the paper.

2 Location privacy techniques

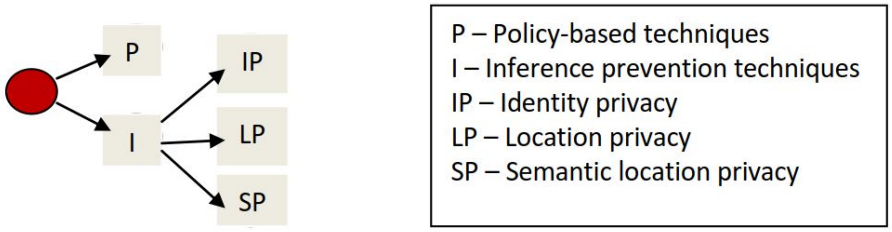


Fig. 2: A taxonomy of location PETs[4].

Since user modeling and location are strongly intertwined we will take a look on the state-of-the-art of location privacy techniques. LBS allow to gather trajectory information and to build mobility patterns. Mobility patterns reveal what people do, i.e. behavioral information. For example, people spend different amount of time in a location depending on what they do there, e.g. a user staying in a night-club at nightly hours is likely a customer of the nightspot[4]. In figure 2 we can see a taxonomy of the location PETs. The first class of privacy solutions are the **policy-based techniques**. For realizing the LBS there is LBS provider facilitating the communication between the application and the user. A policy is of a set of user-defined privacy preferences or rules typically enforced by the trustworthy LBS provider upon the request of service. Policy-based techniques are probably the most popular solutions for privacy in LBS, conceptually simple, in line with common practices in law, and endorsed by standardization bodies[4]. It's essential to understand that policy-based techniques do not prevent the extraction of mobility patterns because the LBS provider is generally aware of the positions of all clients and thus can record users' trajectories at the finer level of detail. Therefore, if the LBS provider is untrustworthy, the user's privacy is at stake. However policy specification languages have a peculiar feature, i.e. the capability of expressing conditions on contextual variables. The degree of

usability of these languages is generally assessed by involving users in the experimentation. The second class are the **inference-prevention techniques**. They basically aim at preventing the LBS provider from drawing sensitive information from exact positional data[4] and are further categorized in the following classes:

- **Identity privacy techniques** attempt to forestall the re-identification of users (deprived of their real identity) in LBSs providing anonymous services.
- **Location privacy techniques** apply to forestall the transmission of exact users' positions to the LBS provider. Knowing precisely the positions in which individuals are located (or not located) jeopardizes their privacy and physical safety.
- **Semantic location privacy techniques** aim at preventing the disclosure of the places in which users stay because those locations can reveal sensitive data and behavioral information.

Anonymization enables organizations to protect their data and systems from a diverse set of attacks and preserve privacy. Assertion about the quality of anonymization can only be made with some anonymization metrics. The techniques enumerated in this section are all based on some anonymity metrics; [6] gives an overview of the state-of-the-art anonymization metrics.

In the next section we'll take a look on a framework for privacy-aware user modeling in ubiquitous social networks.

3 User modeling privacy

The authors of [10] propose a framework for analyzing privacy requirements and for analyzing privacy-related data. They outline a combination of requirements analysis, conflict-resolution techniques, and a P3P[1]. The Platform for Privacy Preferences Project (P3P) enables Websites to express their privacy practices in a standard format that can be retrieved automatically and interpreted easily by user agents. P3P user agents allow users to be informed of site practices (in both machine- and human-readable formats) and to automate decision-making based on these practices when appropriate. Thus users need not read the privacy policies at every site they visit.

3.1 Privacy Challenges

There is a huge amount of knowledge that can be discovered by investigating people's textual/multimedia contributions to SN[2] and the links they set to their "friends"– in this sense, social network analysis is an important topic for Knowledge Discovery for Ubiquitous User Modeling. The most common use of this user data is targeted marketing. The very essence of social media is that user-profile information is public. The data can be classified in confidentiality levels,

[1] P3P: `https://www.w3.org/P3P/`
[2] Social Network

where the concrete details and the application of the confidentiality levels to data depends on the SNS[3] implementation[10]:

- **Private data:** is disclosed to the SNS operator for its internal purposes only.
- **Group data:** is disclosed to the SNS operator and can be accessed by other users of the same SNS that are also in the same group as the user.
- **Community data:** has been disclosed to the SNS operator and is available to all registered and logged-in users of the SNS.
- **Public data**: has been disclosed to the SNS operator and is made accessible for all SNS visitors, including anonymous visitors.

There is a problem with the privacy dependency. As an example, the friend of a friend can see my data. In addition, problems arise when systems support the interaction with the world outside the system. For example, Google Mail (Gmail) users consent to their emails' data being analyzed by Google; however, all incoming mails of a Google Mail account (whether sent by another Gmail user or by somebody else) are also analyzed. Thus, A's treatment of his privacy also has direct external effects on the privacy of C, who is a non-user of the system[10]. The distinction between "in the system"and "outside the system"vanishes in case of loosely coupled networks where members may engage in relationships spontaneously and without a central authority. So the potential privacy conflicts that arise by social-network interaction must be identified; privacy preferences and requirements must be formalized sufficiently such that software can automatically detect problems, alert the user, and assist her.

3.2 Privacy conflict detection

This princip is based on MSRA[4] method. The first step is to discover all the stakeholders and subsume all their privacy interests. We note that stakeholder are more than just the users but all the person who interact with the system.
Once the requirement – privacy interests – are specified there is requirement interaction management in order to identify the privacy conflicts. That's the second step. A requirement R is satisfied by a component C if the component exhibits all the properties specified in the requirement. There may be degrees of satisfaction of a requirements and this can be mapped to a range:

Definition 1. $Sat_R : C \longrightarrow [0, 1]$

Requirements interaction can then be defined as follows:

- **Perceived interaction:** Two requirements, labeled R_1 and R_2 interact if and only if the satisfaction of one requirement affects the satisfaction of the other.

[3] Social Network Service
[4] Multilateral Security Requirements Analysis: `http://vasarely.wiwi.hu-berlin.de/UKDU06/Proceedings/UKDU06-proceedings.pdf`

- **Operational interaction:** f component C_1 satisfies R_1 and component C_2 satisfies R_2, and the run-time behavior of C_1 affects the run-time behavior of C_2, then C_1 interacts with C_2, and indirectly R_1 interacts with R_2.

	R1 edit others entries	R2 only authors editors	R3 authors accountable	R4 anonymous authors
R1 edit others entries	0	-	?	-
R2 only authors editors	-	0	+	-
R3 authors accountable	?	+	0	-
R4 anonymous authors	-	-	-	0

+ positive correlation
- negative correlation
? unspecified correlation
0 no correlation

Fig. 3: Requirements interaction for a social network[10].

Interactions between requirements may be positively correlated (they strengthen each other), negatively correlated (they are in conflict), the correlation may be unspecified (the effect is unclear but exists) or non-existent (no effect). Figure 3 gives an overview of what the interactions between initial requirements for a various sets of stakeholders could look like. For example, the anonymity requirement R_4 is obviously in conflict with all the other requirements. If a user uses the services of the SNS anonymously, it is not possible to prove that information in an entry is about oneself (requirement R1), it is not possible to authenticate the users who edited entries through their identities (requirement R2), and accountability for requirements is not possible through user identities (requirement R3). Hence, some negotiation is necessary to resolve the negative and unspecified correlations between the different requirements. Resolutions of conflicts, which is the third step may also introduce new conflicts. Thus, an iterative requirements interaction management approach is needed. The following methods can be used to resolute the conflicts between the requirements:

- **Relaxation:** the conflicting requirements are relaxed or generalized to avoid conflict.
- **Refinement:** the conflicting requirements are partially satisfied.
- **Compromise:** a compromise is found between the requirements.
- **Restructuring:** a set of methods are used to modify the conflict context, which includes assumptions and related requirements.

Recognizing interactions in privacy and security requirements written in natural language is not a trivial activity. So we need an adequate modeling language that makes the identification of interactions easier[5].

3.3 Enhencing privacy with P3P

```
 <POLICY xmlns:PRINT="http://preibusch.de/namespaces/PRINT/PRINT.xsd">
<EXTENSION optional="no">
  <PRINT:NEGOTIATION-GROUP-DEF  id="friendship"
   standard="public_friend"  fallback="public_friend"  selected="public_friend"
   description="Choosing public (open) or private (hidden) friendship" />
</EXTENSION>
<STATEMENT>  <EXTENSION optional="no">
    <PRINT:NEGOTIATION-GROUP id="public_friend" groupid="friendship"
     serviceuri="/make-friend/public"
     description="Make this user a public friend of yours" />
  </EXTENSION>
  <CONSEQUENCE>Other visitors will see that you are friends</CONSEQUENCE>
  <RECIPIENT>  <ours/>
               <public/>  </RECIPIENT>
  <PURPOSE>    <contact/>
               <other-purpose> friendship </other-purpose>  </PURPOSE>
  <RETENTION>  <indefinitely/>  </RETENTION>
  <DATA-GROUP> <DATA ref="#user.login.id"/>  </DATA-GROUP>
</STATEMENT>
<STATEMENT>  <EXTENSION optional="no">
    <PRINT:NEGOTIATION-GROUP id="hidden_friend" groupid="friendship"
     serviceuri="/make-friend/hidden"
     description="Make this user a hidden friend of yours" />
  </EXTENSION>
  <CONSEQUENCE>Other visitors will not see that you are friends</CONSEQUENCE>
  <RECIPIENT>  <ours/>  </RECIPIENT>
  <PURPOSE>    <contact/>
               <other-purpose> friendship </other-purpose>  </PURPOSE>
  <RETENTION>  <indefinitely/>  </RETENTION>
  <DATA-GROUP> <DATA ref="#user.login.id"/>  </DATA-GROUP>
</STATEMENT>
</POLICY>
```

Fig. 4: Different friendship alternatives (public/hidden) are coded in a single P3P Privacy Policy[10].

To satisfy privacy requirements in an operational SNS appropriate measures must be taken: the conception and adaptation of technologies and processes, essentially privacy languages and tools to interpret and enforce these languages. There two essential goals to attain: mechanisms to ensure that data/information of one privacy level must not be made accessible via data/information of a lower privacy level; and mechanisms that prevent users from disclosing personal information about other users inside an SNS. P3P provides a way for a Web site to encode its data-collection and data-use practices in a machine-readable XML

[5] Adeniyi Onabajo and Jens H. Jahnke. Modelling and Reasoning for Confidentiality Requirements in Software Development. In ECBS, 2006.

format known as a P3P policy[6]. APPLE[7] makes it possible for users to express their privacy preferences. The user agents can use these preferences to take some decisions about the acceptance of privacy policies of SNSs.
Figure 4 shows an example of a P3P privacy policy where the choice between an open (public) or hidden (private) friendship is offered. The user agent parses alternative scenarios of friendship making and select the most appropriate option for the user. Since the friendship making process is realized through the SNS, the SNS operator can record the chosen option and integrate enforcement mechanisms into the site[8], so that when the user's friend list is displayed, only public friends are being displayed and the user are being disturbed at all by the whole process. Anyway the field of resolution generation is should further explored. If the policies gives semantics to confidentiality the privacy, the privacy can be guaranteed against the other users but not against the SNS.

4 Location privacy in Pervasive Computing

This section concentrates on location privacy, a particular type of information privacy that we define as the claim of individuals to determine for themselves when, how, and to what extent location information about them is communicated to others[5]. When location systems track users automatically on an ongoing basis, they generate an enormous amount of potentially sensitive information. This someone could inspect the history of all your past movements, recorded every second with submeter accuracy, you might start to see things differently. Some LBSs[9] work with the user's identity and others can operate anonymously. The main goal is to develop techniques that let users benefit from location-based applications while at the same time retaining their location privacy. Privacy of location information is about controlling access to this information.The need is not necessarily to stop all access – because some applications can use this information to provide useful and personalized services – but to be in control. In the following lines we will take a look on two techniques proposed by [7].

The goal of the approach is to develop a privacy-protecting framework based on frequently changing pseudonyms[10] so users avoid being identified by the locations they visit. For this purpose they introduce the concept of **mix zones** and show how to map the problem of location privacy onto that of anonymous communication. Since it's important to have metrics to quantify the quality of the privacy we introduce two metrics: **anonymity set** and **entropy**.
The goal is about developing techniques that let users benefit from location-based applications while at the same time retaining their location privacy. So our true identity should be hidden from the application receiving our location:

[6] W3C Working Group Note 13 November 2006: `http://www.w3.org/TR/P3P11/`.
[7] APPLE: A P3P Preference Exchange Language
[8] `http://research.sun.com/projects/xacml/XACML_P3P_Relationship.html`.
[9] LBS: location-based service
[10] The pseudonyms are used for the anonymization of location information

this can be considered as being the statement of the security policy. Our target is the class of location-aware applications that accept pseudonyms, and the aim will be the anonymization of location information. In this model there is a middleware – which is trusted and might help users hide their identity – between the user and LBS. So the users register interest in particular applications with the middleware; applications receive event callbacks from the middleware when the user enters, moves within, or exits certain application-defined areas. The middleware evaluates the user's location at regular periodic intervals, called *update periods*, to determine whether any events have occurred, and issues callbacks to applications when appropriate. An *anonymizing proxy* is required for all communication between users and applications. Using a long-term pseudonym for each user does not provide much privacy, because applications could identify users by following the "footsteps"of a pseudonym to or from such a "home"area[7]. The countermeasure is to have users change pseudonyms frequently, even while they are being tracked. In order to resolve the problem of application which could link – if the system's spatial and temporal resolution are not sufficiently high – the old and new pseudonyms we introduce the mix zone.

4.1 Mix zones

There are two essential constructions for anonymous communications (the mix network and the dining cryptographers algorithm) and the notion of anonymity sets. A mix zone is analogous to a mix node in communication systems. Using the mix zone to model our spatiotemporal problem lets us adopt useful techniques from the anonymous communications field[7].

A mix network can be considered as a store-and-forward network that offers

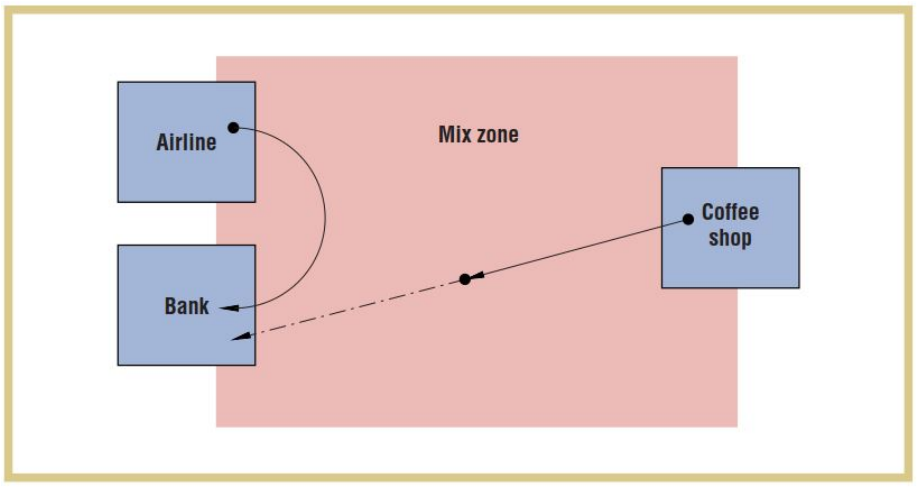


Fig. 5: A sample mix zone arrangement with three application zones[7].

anonymous communication facilities. This network contains normal message-routing nodes alongside special mix nodes so that hostile observers monitoring all the network links is not able to trace a message from the destination to the source without colliding mix nodes. A mix node simply collects n equal-length packets as input and reorders them by some metric (for example, lexicographically or randomly) before forwarding them, thus providing unlinkability between incoming and outgoing messages. A mix zone is a connected spatial region of maximum size in which none of these users has registered any application callback[7]. But in contrast, an *application zone* as an area where a user has registered for a callback. The identities are thus "mixed"because applications don't receive any location information when they are in a mix zone. Figure 5 provides a plan view of a single mix zone with three application zones around the edge: an airline agency (A), a bank (B), and a coffee shop (C). Zone A is much closer to B than C, so if two users leave A and C at the same time and a user reaches B at the next update period, an observer will know the user emerging from the mix zone at B is not the one who entered the mix zone at C. Furthermore, if nobody else was in the mix zone at the time, the user can only be the one from A.
There are another aspects to take into account in order to avoid some attacks. If a mix zone has a diameter much larger than the distance the user can cover during one location update period, it might not mix users adequately. If the maximum size of the mix zone exceeds the distance a user covers in one period, mixing will be incomplete. In order to quantify the anonymity provided by a mix node we need measure: the anonymity set.

4.2 Anonimity set

In a mix zone that user u visits during time period t, we define the anonymity set as the group of people visiting the mix zone during the same time period. The anonymity set's size is a first measure of the level of location privacy available in the mix zone at that time. A user might deliberately refuse to provide location updates to an application until the mix zone offers a minimum level of anonymity. The middleware can also calculate the average anonymity set size and present this information to the user who should decide if accepting the services of a new location-aware application.
After an experiment got some results. The anonymity set is a good quantitative measurement for assessing the anonymity. The protection technique does not work that in this special environment but could work in another one. It means the context of the mix zone can affect the quality of the anonymity. The update periods are crucial for the anonymity. The update periods in an area should be corresponding to the maximum walking speed of users in this area. As an example, if the maximum walking speed in the mix zone is 1.2 meters per second and the user needs 50 seconds from one extreme of the mix zone to another, the users cannot be considered to be "mixed"if the update periods shorter than 50 seconds.
There is a strong correlation between entry position and the exit position, which is dependent of the mix zone's geography but the anonymity sets does not model

user entry and exit motion.That's why a second measured is required to account the user movement.

4.3 Entropy

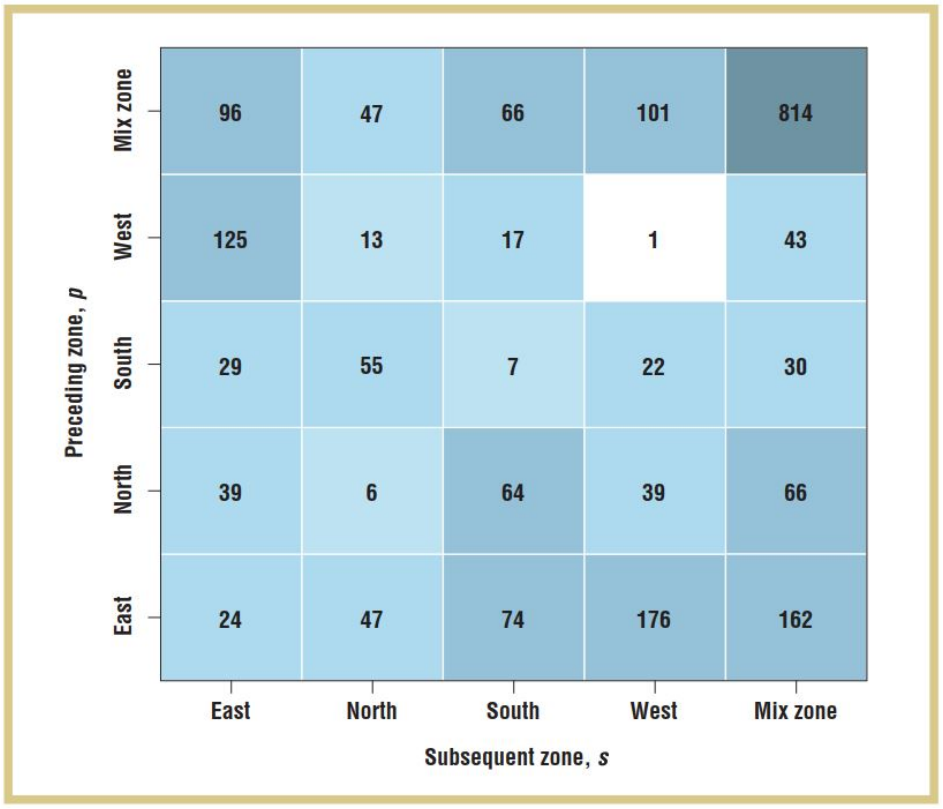


Fig. 6: The movement matrix M records the frequency of movements through the hallway mix zone[7].

Anonymity set's size is only a good measure when all the members of the set can equally potentially be the one of interest to the observer; this is the case of maximal entropy[7]. For a set of size n, if all elements are equiprobable, we need $\log_2 n$ bits of information to identify one of them. If they are not equiprobable, the entropy will be lower, and fewer bits will be enough. Here is how we can precisely compute how many bits suffice.
If we consider user movements through a mix zone z. For users traveling through z at time t, we can record the preceding zone, p, visited at time $t-1$, and the subsequent zone, s, visited at time $t+1$. We use historical data to calculate,

for all users visiting zone z, the relative frequency of each pair of preceding and subsequent points (p, s) and record it in a movement matrix M. M records, for all possible (p, s) pairs, the number of times a person who was in z at t was in p at $t-1$ and in s at $t+1$. The entries of M are proportional to normalized[11] joint probability:

$$P(prec = p, subs = s) = \frac{M(p,s)}{\sum_{i,j} M(j,i)}$$

The conditional probability of coming out through zone s, having gone in through zone p, follows from the product rule:

$$P(subs = s, prec = p) = \frac{M(p,s)}{\sum\limits_{j} M(p,j)}$$

. The Shannon's classic measure of entropy[12] can now be applied to this problem:

$$= -\sum_{i} p_i \cdot \log p_i$$

We get a set of possible outcome with probability p_i and the higher it is, the more uncertain a hostile observer will be about the true answer, and therefore the higher the anonymity will be. Figure 6 shows M, the frequency of all possible outcomes for users entering a mix zone, for a location update period of 5 seconds. It shows it is unlikely that the preceding zone, p, is the same as the subsequent zone, s, for a location period of 5 seconds, except in one case.

The entropy measurement shows a more pessimistic picture – in which the user has less privacy – because it models a more powerful adversary who might use historical data to de-anonymize pseudonyms more accurately.

This section and the previous one show example with simple techniques. In the following sections we will take a look semantic-based techniques.

5 Semantic-based user modeling: multimodal mobile device footprints

The actual mobile devices – smartphones and tablets – collect a wide variety of information about their environments and record "digital footprints"[13] about the location and activities of their owners. These devices are equipped with embedded sensors such as GPS, Bluetooth, WiFi, accelerometer, touch, light, and

[11] Normalization: `https://en.wikipedia.org/wiki/Normalization\_(statistics)`

[12] C.E. Shannon, "A Mathematical Theory of Communication,"Bell System Tech. J., vol. 27, July 1948, pp. 379–423, and Oct. 1948, pp. 623–656.

[13] Digital footprint: `https://www.smore.com/6pbw-what-s-your-digital-footprint`

many others. The new gathered information have stimulated people-centered mobile applications. The traditional usage of the mobile footprints is to infer daily activities like driving/running/walking, etc. and social contexts such as personality traits and emotional states. For these the applications need the data itself. **mFingerprint** is a framework that does not use the data itself but statistics about the data to model the user. Multimodal mobile usage data is analyzed to extract simple effective statistics that can uniquely represent mobile users. mFingerprint computes high-level statistical features that do not disclose sensitive information from the phone, such as raw location, browser history, and application names. Therefore, applications can share, transmit, and use these descriptive privacy-preserving feature vectors to enable personalized services on mobile devices with fewer privacy concerns. There are some problems to solve in order to achieve the goal of mFingerprint such as how to choose sources – which sensor data are suitable – to construct effective mobile fingerprints and avoid disclosing sensitive mobile data[15], ...

5.1 mFingerprint system overview

As showed in figure 7 the mFingerprint framework has four layers. The bottom layer collects various sensor readings using hardware sensors that trace location (e.g., GPS, WiFi, and cell tower), proximity (Bluetooth and microphone). Furthermore, soft sensor data such as application usage and web browser history are also recorded. The second layer of mFingerprint computes a set of privacy-preserving statistical features from these digital footprints. Here we particularly focus on designing frequency and entropy based statistical features to capture mobile device usage patterns[15]. Such features flow up to the third layer for user learning, which includes building user models via the feature vectors, identifying users via classification methods, and grouping users into meaningful clusters via unsupervised learning. The forth layer is the application layer where various applications can be established, such as inferring user profile, logging mobile behaviors, and creating personalized services and user interfaces.

5.2 Feature computation and user identification

Frequency based Footprint Features. The number of devices and cell towers that are observed by a phone throughout the day provides information about the owner's environment. For example, a phone in a busy public place will likely observe many wireless devices while a phone in a moving car observes different cell towers over time. Meanwhile, a user's app usage patterns throughout the day tell us something about his or her daily routine. We thus propose simple frequency-based features that measure how much activity of four different types (Wifi, Cell towers, Bluetooth, and App usage) is observed at different time intervals throughout the day.
If we divide the time into T-minute time periods and observe the phone's state every M minutes, with $M < T$ so that there are multiple observations per time period. In the i-th observation of time period t, we record: the number of Wifi

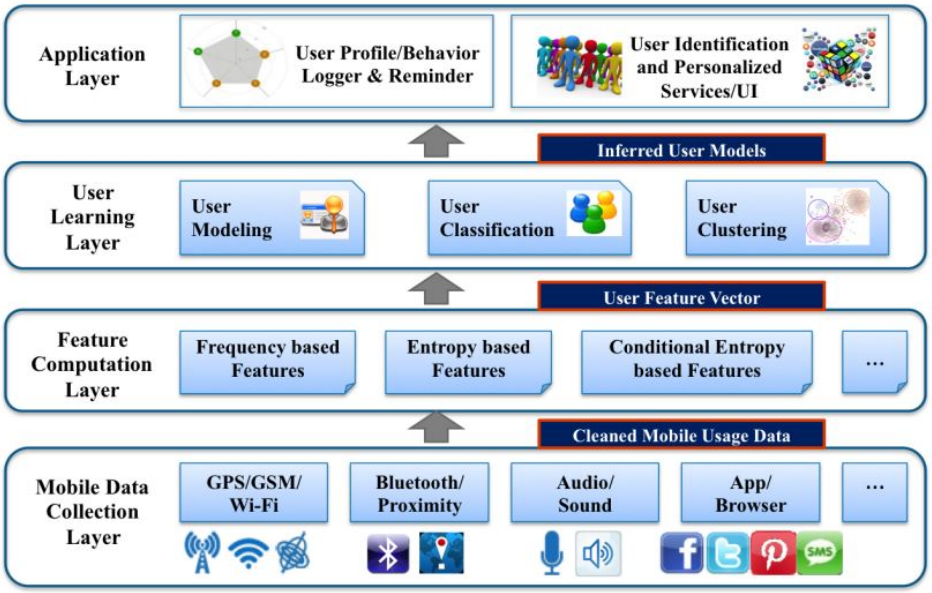


Fig. 7: The mFingerprint framework.[15].

devices that are observed ($W^{t,i}$); the number of cell phone towers that the phone is connected to ($C^{t,i}$); the number of bluetooth devices that are seen ($B^{t,i}$); and the number of unique apps that have been used over the last m minutes ($A^{t,i}$). We then aggregate each of these observation types to produce four features in each time period:

$$F_W^t = \sum_i W^{t,i}, \quad F_C^t = \sum_i C^{t,i}, \quad F_B^t = \sum_i B^{t,i} \quad and \quad F_A^t = \sum_i A^{t,i}$$

These features are incorporated in a single one, the vector $F^t = [F_W^t, F_C^t, F_B^t, F_A^t]$.

Entropy based footprint features. While the simple frequency features above give some insight into the environment of the phone, they ignore important evidence like the distribution of this activity. For example, in some environments a phone may see the same Wifi hotspot repeatedly through the day, while other environments may have an ever-changing set of nearby Wifi networks. To illustrate this, figure 8 compares observed frequency versus anonymized device IDs for two users across each of the four observation types, for a period of 10 days. We can observe that User 2 is less active in WiFi and cell mobility compared to User 1, but has more Bluetooth encounters and uses more diverse apps.
We thus propose using entropy of these distributions as an additional feature of our user fingerprints. The entropy feature summarizes the distribution over device IDs, but in a coarse way such that privacy concerns are minimized. For

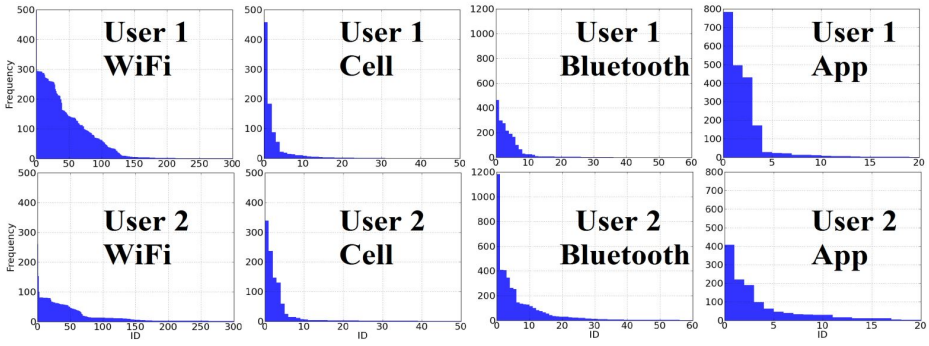


Fig. 8: Comparison of activity histograms for 2 users over 10 days.[15].

Wifi, let W_j^t denote the number of times we observe wifi hotspot j during time period t. Then we define the Wifi entropy during time period t as,

$$E_W^t = -\sum_j \frac{W_j^t}{F_W^t} \log \frac{W_j^t}{F_W^t}.$$

Entropy features for Cell towers, Bluetooth, and Apps (E_C^t , E_B^t , and E_A^t , respectively) are computed in the same way, and we define a multimodal entropy feature vector $E^t = [E_W^t \quad E_C^t \quad E_B^t \quad E_A^t]$, which incorporates all four perspectives.

Conditional entropy and frequency based features. The above entropy and frequency features are calculated conditioned on time and location. Intuitively at different times and at different locations, users have different patterns of application usage and surrounding devices (Bluetooth, WiFi, cell, etc.). Conditioning on time and space is thus useful to better differentiate users[15]:

- **Conditional features on time.** We distinguish between three fixed daily time intervals, mornings (0:00 - 8:59), working hours (9:00 - 17:59) and evenings (18:00 - 23:59), and two types of days, weekdays (Monday through Friday) and weekends (Saturday and Sunday).
- **Conditional features on location.** For each user, we filter and cluster their geo-locations in order to identify the top-k significant locations. From data collected at these k locations, we compute the conditional entropies and frequencies. We first try to divide the area in stationary and non-stationary periods according to Wifi readings. And then we cluster the stationary locations in order to identify the important locations.

A test of this framework in an evaluation shows that both frequency and entropy features outperform the random baseline significantly. Entropies have better performance for large numbers of users compared to frequencies. When features

including basic multi-modal entropies, location entropies and time entropies are all combined, the performance is the best in term of user identification. We also discover that more features improve accuracy, and more computation is required as well, especially for the clustering required by location conditioning[15].
This semantic-based user modeling privacy technique is very efficient according to the results. In the next section we will make an attempt to know whether semantic-based location privacy is also efficient in term of privacy.

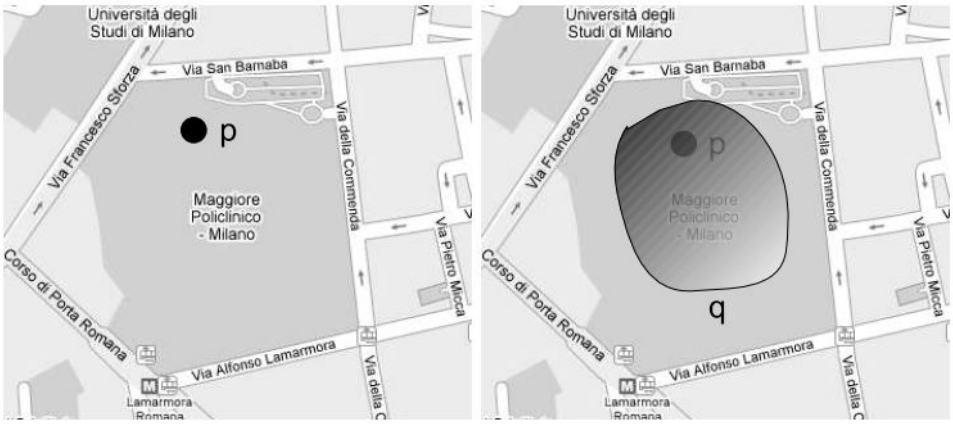
6 Semantic-based location obfuscation techniques

Personal location data refers to the association (u, p) between user identifier u and position information p. Protecting location privacy means thus preventing u and p from being both disclosed without the consent of the user[14]. The usual strategy for protecting location privacy is to obfuscate the actual position of the user with a coarse location and then forward the obfuscated location to the LBS provider[15]. Existing techniques for location obfuscation are only based on geometric methods[13], such as k-anonymity[6]. The authors claim that geometry-based obfuscation techniques do not protect against the following simple privacy attack.

6.1 Problem: location privacy attack

As an example of *spatial knowledge attack*, assume that John issues a LBS request from position p inside *hospital Maggiore* in figure 9(a). John however does not want to disclose the fact of *being inside* the hospital because that might reveal he has health problems. Now assume that location p is obfuscated by region q using some geometry-based technique (figure 9(b)). If an observer knows that John is in the obfuscated location q and q is entirely contained in the spatial extent of the hospital (the location of the hospital is publicly known), then such adversary can immediately infer that John is in the hospital. As a result, sensitive information is revealed without the user consent. Note however that if John would be a doctor, such a privacy concern would not arise because the location would be related to the user's professional activity.
The spatial knowledge attack arises because geometry-based obfuscation techniques do not consider the actual semantics of space, namely the spatial entities populating the reference space and their spatial relationships, in other terms the *spatial knowledge*. In order to avoid this kind of attack we want semantic-aware obfuscation techniques.

[14] A. R. Beresford and F. Stajano. Location privacy in pervasive computing. IEEE Pervasive Computing, 2(1):46–55, 2003.
[15] `http://searchnetworking.techtarget.com/definition/location-based-service-LBS`



(a) Position p (b) Obfuscated location q

Fig. 9: *Location data obfuscation*[13]

6.2 Approach

The basic idea of the obfuscation system is to collect users' preferences about sensitive places and the desired degree of location privacy in *privacy profiles* and then carry out the process of location obfuscation in two steps. Consider a privacy profile v.

1. A privacy model supporting the obfuscation of sensitive locations based on user preferences set by the user. Here we obfuscate the sensitive places specified in v based on the user's desired degree of privacy. This operation, that we call *obfuscated space generation*, results in the generation of a set of coarse locations hiding the actual extent of sensitive places in compliance with user preferences. We can abstractly think of obfuscated space generation as the function Obf:
$$Obf(v) = s.$$
This function maps profile v onto a set of regions enclosing sensitive places.
2. An algorithm, called *SensFlow* (i.e. Sensitivity Flow), implements the obfuscation strategy. It is carried out upon a user's LBS request. Consider a user with privacy profile v in position p. The operation that we call obfuscation enforcement can be abstractly represented by the function Oe:
$$Oe(p, v) = q.$$
The function maps the position p and profile v onto a location q where $q \in Obf(v)$ if p is contained in q and $q = p$ otherwise. As a result, when the location is obfuscated, an adversary cannot infer with certainty that the user is inside a sensitive (for the user) place. At most one can infer that the position may be in a sensitive place.

Each region is subdivided in cells where each cell has a sensitivity which depends on the user preferences in the privacy profile. Each cell is thus obfuscated separately through an obfuscation algorithm. To represent user preferences, we define a privacy model, called *obfuscation model*. In the next we will take a look on the privacy model.

6.3 The obfuscation model

A position is a point in a two-dimensional space S; region is a polygon; location broadly denotes a portion of space containing the user's position. Places are represented as simple features and a feature has just a name. FT and F respectively the set of features types and the set of corresponding feature. The privacy model is based on three concepts[13]: properties of places, their level of sensitivity and the obfuscated space. The pair (FT, F) is the geographical database of an application.

Properties of places. Places as classified into types. Users specify in their privacy profiles which types of places are *sensitive*, *non-sensitive* or *unreachable*. A place is sensitive when the user does not want to reveal to be in it; a place is unreachable when the user cannot be located in it; a place is non-sensitive otherwise.

Level of sensitivity. This the degree of sensitivity of a region for a user. For example a region entirely occupied by a hospital has a high level of sensitivity, if hospital is sensitive for the user. We emphasize that the level of sensitivity depends on the extent and nature of the objects located in the region as well as the privacy concerns of the user.
Formally, the score of feature type ft is defined by the function $Score(ft)$ ranging between 0 and 1: value 0 means that the feature type is not sensitive or unreachable while a value 1 means that the feature type has the highest sensitivity. The concept of score captures the subjective perception of the degree of sensitivity. The sensitivity level (SL) of a region r, written as $\mathrm{S}LReg(r)$, quantifies how much sensitive r is for the user. The weighted sensitive area is the surface in r occupied by sensitive features weighted with respect to the sensitivity score of each feature type. The relevant area of r is the portion of region not occupied by unreachable features.

Obfuscated space. This is a set of obfuscated locations associated with a privacy profile. Specifically, the locations of an obfuscated space have a level of sensitivity less or equal than a sensitivity *threshold value*. The sensitivity *threshold value* (θ_{sens}) is the maximum acceptable sensitivity of a location for the user. Its value ranges in the interval $(0, 1]$. A value equal to 1 means that the user does not care of location privacy in any point of space Since the threshold value is user-dependent, its value is specified in the privacy profile. The threshold value is another parameter specified in the privacy profile.

6.4 Computation of obfucasted space

The obfuscated spaces are calculated in two steps:

1. **Specification of the initial partition.** The reference space is subdivided in a set of small regions, referred to as cells, which constitute the initial partition denoted as C_{in}. The granularity of the initial partition, that is, how small the cells are, is application-dependent.
2. **Iteration method.** The current partition is checked to verify whether the set of cells is an obfuscated space. If not, it means that at least one cell is not privacy preserving. A cell c is thus selected among those cells which are not privacy- preserving and merged with an adjacent cell to obtain a coarser cell. The result is a new partition. This step is iterated until the solution is found, and thus all privacy preferences are satisfied or the partition degenerates into the whole space.

6.5 Experiment

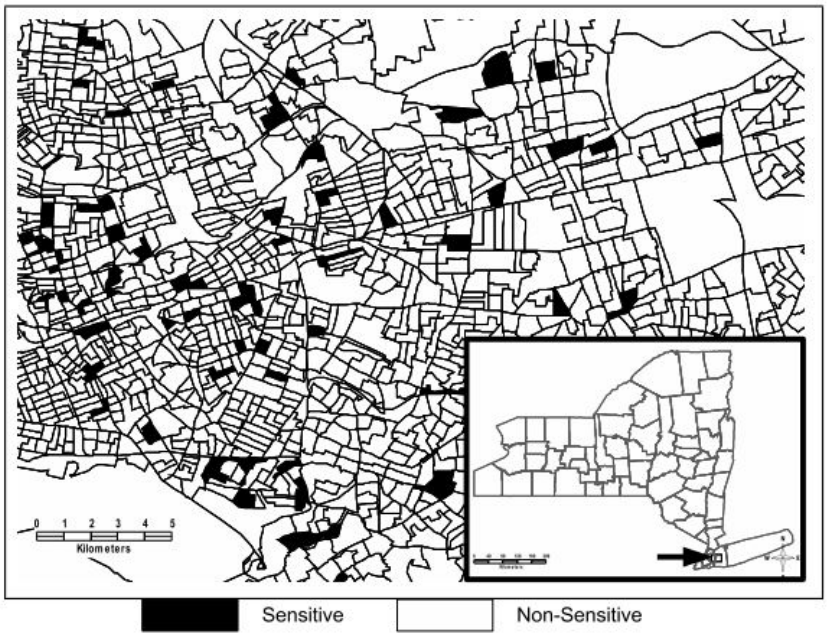
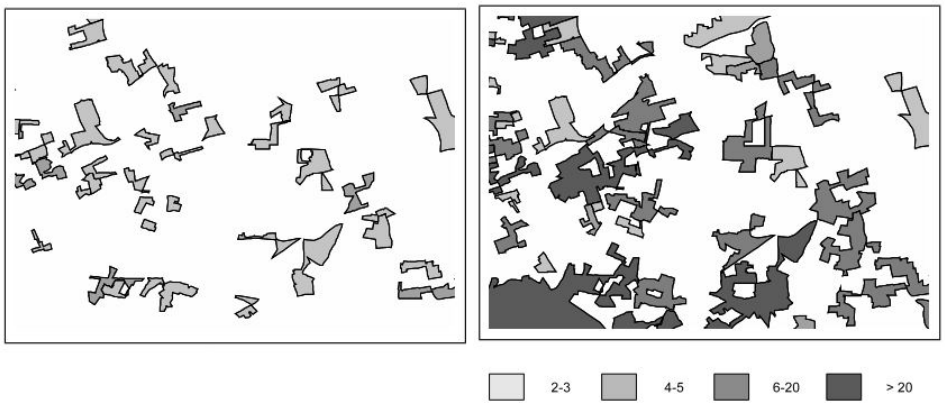


Fig. 10: Sensitive cells in the initial partition[13].

Figure 10 shows a data set consisting of 15000 polygons representing an aggregation of blocks. Each polygon is a cell of the partition and the percentage of



(a) $\theta_{sens} = 0.3$ (b) $\theta_{sens} = 0.1$

Fig. 11: *Visual representation of two obfuscated spaces relative to area in Figure 10* ($s = 0.05$)[13]

area which is sensitive in a cell is assigned randomly. The *SensFlow* algorithm has been run using different values of the sensitivity threshold. The experimental results are shown in the maps in Figure 11. The generalized regions are represented by polygons of different color, based on the number of aggregations: the color is darker for the more aggregated regions; white space denotes the original space. It can be observed that the granularity of the obfuscated space is coarser for lower values of the sensitivity threshold. The main limitation of this approach is that the publicly available data set is not sufficiently precise. Cells are generally too broad, especially in rural areas and that compromises the quality of service.

The grid-based approach to space subdivision is also deployed. Space is subdivided into a grid of regular cells. Features do not have any physical correspondence with cells. Features are thus contained in a cell or overlap multiple cells. The sensitive area in the cell results from the spatial intersection of the feature extent with the cell. We have run the algorithm over a grid of 100 squared cells, assuming again a unique feature type with maximum score.

Figure 12 shows the obfuscated spaces generated for different values of the sensitivity threshold. The result is visualized as follows: adjacent cells which have not been merged are assigned different gray tones; merged cells have an identical gray tone and are labeled by the same number. We can observe how the granularity of obfuscated locations (i.e. a set of cells with identical label) changes for different values of θ_{sens}. From the experiments it turns out that the grid-based approach is more flexible because the granularity of partition can be defined based on application On the other hand, the whole process of discretization of features in cells is much more complex.

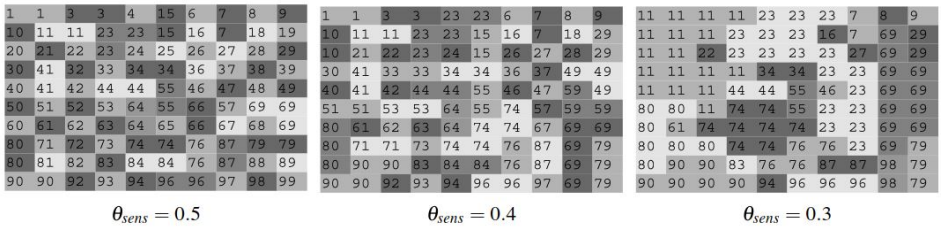


Fig. 12: Visual representation of the cell aggregation for different values of the sensitivity threshold θ_{sens} . Merged cells are indicated using both the same number and the same color.[13].

7 Conclusion

In this paper we have investigated the problem of privacy in pervasive computing. Here, the focus was on user modeling and location. Some techniques were without semantics and have disadvantages. The other ones were with semantics and could fix some drawbacks of the variants without any semantics. In the future it would interesting to search more in the direction of techniques for semantic-based privacy to both user modeling and location. Another critical point is the privacy executor. The question of who should be the third party actor between the user and the application is also an interesting search field.

References

1. Steve Kenny. *An Introduction to Privacy Enhancing Technologies* https://iapp.org/news/a/2008-05-introduction-to-privacy-enhancing-technologies/
2. Alfred Kobska. *Privacy-Enhanced Personalization* http://www.ics.uci.edu/ kobsa/papers/2006-CHI-kobsa.pdf
3. Dominikus Heckmann. *Ubiquitous User Modeling* http://d-nb.info/978586085/34
4. Maria Luisa Damiani. *Privacy enhancing techniques for the protection of mobility patterns in LBS: research issues and trends* https://air.unimi.it/retrieve/handle/2434/207836/242074/damiani-cpdp-revision.pdf
5. Maria Luisa Damiani. *Location privacy models in mobile applications: conceptual view and research directions* https://www.researchgate.net/publication/271095287
6. Douglas J., Kelly Rusty O., BaldwinRichard A., Michael R., GrimailaBarry E. Mullins . *A Survey of State-of-the-Art in Anonymity Metrics* http://dl.acm.org/citation.cfm?id=1456453
7. Alastair R. Beresford and Frank Stajano. *Location Privacy in Pervasive Computing*
8. Osman Abul, Francesco Bonchi, Mirco Nanni. *Never Walk Alone: Uncertainty for Anonymity in Moving Objects Databases*
9. Meiko Jensen. *Towards Privacy-Friendly Transparency Services in Inter-Organizational Business Processes*
10. Sören Preibusch, Bettina Hoser, Seda Gürses, Bettina Berendt . *Ubiquitous Social Networks: Opportunities and Challenges for Privacy-Aware User Modelling* https://www.diw.de/documents/publikationen/73/59994/dp698.pdf
11. Byoungyoung Lee, Jinoh Oh, Hwanjo Yu, and Jong Kim. *Protecting Location Privacy Using Location Semantics* http://www.cc.gatech.edu/ blee303/paper/locPriv.pdf
12. Anna Monreale, Roberto Trasarti, Chiara Renso. *Preserving Privacy in Semantic-Rich Trajectories of Human Mobility* http://dl.acm.org/citation.cfm?id=1868481
13. Maria Luisa Damiani, Elisa Bertino, Claudio Silvestri. *Protecting Location Privacy through Semantics-aware Obfuscation Techniques* http://download.springer.com/static/pdf/707/chp
14. Arif Tumer, Asuman Dogac, and Hakki Toroslu. *A Semantic-Based User Privacy Protection Framework for Web Services* http://download.springer.com/static/pdf/767/chp
15. Haipeng Zhang, Zhixian Yan, Jun Yang, Emmanuel Munguia Tapia, and David J. Crandall. *mFingerprint: Privacy-Preserving User Modeling with Multimodal Mobile Device Footprints* http://download.springer.com/static/pdf/454/chp

Keystrokes recognition using different sensing modalities

Marcus Gall*

Advisor: Long Wang†

Karlsruhe Institute of Technology (KIT)
Pervasive Computing Systems – TECO
*uycqo@student.kit.edu
†wanglong@teco.edu

Abstract. This document gives an overview over already researched approaches to sense finger-based keystrokes on physical or virtual keyboards. Since the typical way of finger-based input-methods measures the press of a key or use electric discharges on touch sensitive surfaces, there are also other information released during a keystroke, which can be used alternatively. On the following pages, some promising methods are introduced and discussed with respect to their advantages and disadvantages as a replacement for common text input methods. Hereby, the document focuses on keystroke sensing by device accelerations, acoustic emanations, video and wifi-based approaches.
As we will see, methods of acoustic sensing reaches already very accurate and robust solutions, but also other techniques have remarkable solutions and results.

Keywords: keystroke recognition, inference, text input method, sensing, acceleration, wifi, video, acoustic, motion

1 Introduction

Nowadays, the mainly used text-input methods for computers and smart-devices are the physical keyboard and touchscreens. Although the technology is mature, robust, and foolproof, it suffers from drawbacks in certain situations, which makes it hard to use then.

Physical keyboards are well-known for decades, but their appearance is bound to their bulky form factor, which makes it hard to use in mobile scenarios. Touchscreen-equipped devices like smartphones come with a virtual on-screen keyboard. This keyboard has only limited haptic feedback, which makes typing long texts tiring and reduces the effectively usable screen size for the main content a lot. On top, the need for smaller screens in the currently smaller and smaller demanded smart devices like smartwatches sets a new challenge, since small virtual keyboards are hardly manageable without a pen. As we can see, both of the main input methods have drawbacks in certain situations. This could

be resolved by exploiting domain knowledge for each device and switch to alternative sensing modalities.

Typically, finger-based keystrokes on a physical keyboard are sensed as an electric signal triggered from the pressed key while on touch sensitive screens, the electric discharge of the glass is measured. But other than we might assume, interactions with an input device causes much more diverse interactions with the surrounding environment. Keystroke actions on keyboards or touch screens release information such as sound, vibrations, accelerations and finger motions. These information leaked as side effects can be sensed and interpreted as keystrokes alternatively to the common sensing methods. Depending on the scenario, some sensing approaches work better than others. Although scientists are on that topic for over 20 years now, the results improved more and more in the recent years, since smart-devices are getting equipped with different sensor-types having increasingly higher resolutions.

In this work we want to focus on papers dealing with microphones, acceleration and gyroscope sensors, video material and WiFi channel strength variations. Derived data from these sensors were used to develop new input methods or to attack the used input method on a side channel and then evaluated in respect of their capabilities and dangers of using it in either case.

In the following sections, latest-state-of-the-art papers of alternative input methods are compared by their sensing type.

The beginning focuses on external sensor sensing: Section 2 carries out acoustic methods, section 3 is about video approaches, section 4 is about WiFi signal based sensing. The content in section 5 deal with possibilities by mostly sensing internal acceleration. Finally, section 6 gives a summary.

2 Acoustic emanations sensing methods

From an early point on, scientists were interested in the possibility to use sound emanations released from mechanic keyboards to reconstruct keystrokes. The basic assumption is, that pressing a mechanic key always creates the same emanation, since the conditions are the same. Due to that, the emanation's character can be used for latency triangulation or training classifiers, which match sensed features to keys.

In [1] from 2004, IBM researchers attacked mechanical keyboards using its acoustic emanations from pressing keys sensed by a microphone. A keystroke emanation consists of a touching, pressing, holding and releasing-phase of the key. As it turns out, the first emanation part, when the finger *touches* the key, contains the most separable information. This part can be windowed into a 2-3ms part and features from its frequency distribution can be calculated by FFT (Fast Fourier Transformation). For 20kHz buckets from 0-4kHz 200 features were calculated and fed into a multi-layer neural network (MLP) and trained by back-propagation. The MLP has 6-10 hidden nodes and the number of output nodes were set on the number of keys, which should be recognized (up to 30). The

output nodes are, as usual for neural networks, activated from 0 to 1 depending on its certainty.

In a two-keys-recognition scenario, the net did in average only 0,5 incorrect recognitions per 20 keystrokes, resulting in a 97,5% hit rate. Altering the sensing from a simple microphone at distance of 0,5m to a parabolic microphone at 15m including background noise didn't decrease the recognition quality.

In a 30-keys-recognition scenario, keys were recognized correctly with the largest output value in 79% of keystrokes, additional 7% in the second largest output and 2% in the third largest output. So in total, the pressed key was not found among the three largest candidates in 12% of the tests.

The latter testing was done on the same keyboard, where also the training samples were recorded. Switching the test keyboard to a different one of same type, recognition rate drops to 52% among the first four candidates (28%, 12%, 7%, 5%). This shows, that text recognition quality is turning bad, but still gives a valuable information gain during password snooping.

Other tests focused on the net's recognition rate when varying the test data. Changes in typing style like forceful typing and typing with many fingers instead of one had negative influence on the recognition rate. Same effect can be proved when different people take part in the test set. But adding these variation profiles to the training set improved the recognition rate, which ended up at 1 incorrect classifications out of 20 (5%) in a two-keys-recognition test.

As additional discovery, the biggest variance in the input features can be seen in the 300-3400Hz band, which also is sensable by a telephones standard microphone. Although proposed, an attack scenario on eavesdropping keystrokes by telephone was not carried out.

While the latter paper by IBM assumed labeled training data, which can be seen as a case of known-plaintext attack, researchers from the University of California Berkeley (UCB) extended the scenario in [2] by not using labeled data, but unlabeled one from only listening typing sounds for 10min. The assumption is, that the typed text is always based on the same natural language (English), which can be exploited by using the language's language model to create a strong recognizer. The continuous audio stream is segmented by its signal energy and features are calculated from 2,5ms shifted 10ms windows from the *touch*-phase part using Mel-Frequency-Cepstral Coefficients (MFCCs). Those features are k-means clustered for $K = 50$, although only 30 keys should be recognized. Having more clusters than keys let's keep the clusters have more precision, but also doesn't allow a one-to-one mapping. Therefore, each cluster just stands for a certain key probability distribution, instead of a key, what makes it possible to use them in a Hidden Markov Model (HMM). While the transition matrix depends on the English language, the output matrix, which links observations with k-means clusters needs to be calculated by the randomly-working Expectation Maximization (EM) algorithm. After EM found the set of weights which maximizes the likelihood and Viterbi optimized the best sequence of keys, the HMM is trained. Now a spelling and grammar checking corrects remaining spelling mistakes from the incorrect HMM classification. The completely correct recognized

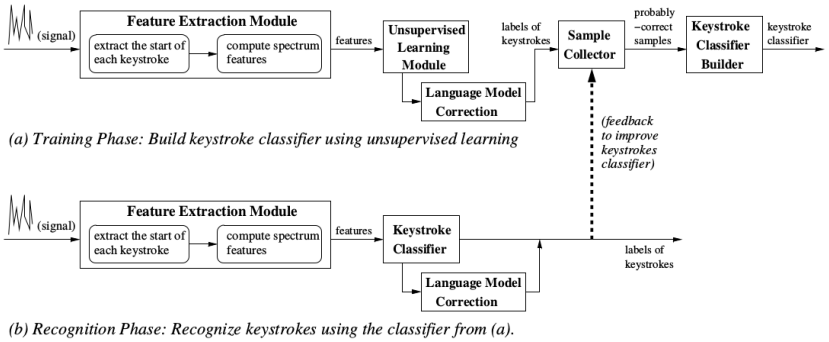


Fig. 1. Graphics of UCBs attack method

output from the HMM is used as feedback to train a linear classifier. Acting as a labeled data training, it improves the recognition rate for use cases like non-english input or passwords, where spelling or grammar correction has no use. The linear classifier is improved on the same training set 3 times until it's complete. In testing mode, the linear keystroke classifier output is again passed through a 3-gram language model correction to improve results. This way, it is possible to achieve recognition rates of 87,6% for words and 95,7% for characters. The language model correction greatly improves the correct recovery rate for words.

Shortly after these results were published, another approach was introduced in the bachelor thesis of [3] using key localization by acoustic triangulation of keystroke sounds. Instead of using sophisticated algorithms and feature extraction as done in the paper before, they propose to use the time delay between two microphones of the key-press sound to identify keys. Therefore, the two microphones are placed left and right hand side of the keyboard leveled with the tabulator-key's row. To identify when a keystroke is starting, they propose the maximum peak or cross correlation approach. Collecting training data was done by pressing keys of large distance on the keyboard to each other. 30 times for training, 10 times for testing. In this work, only two and five keys are used to show the separability. On this taken training data, the mean difference time for each key is calculated. It could be proved, that two far distant keys can be distinguished with a great certainty up to 95%. But since the mean difference time has a high variance mainly due to noise, the more keys are used, the ambiguous the guess gets. Testing five keys located in a horizontal axis on a English keyboard layout [Z, B, /, up, 3] with maximum peak segmentation shows a key recognition rate of [80%, 55%, 50%, 65%, 100%]. Cross correlation segmentation of the same test reveals worse [32%, 100%, 80%, 45%, 75%]. Testing the vertical precision between F5 and B works slightly with maximum peak segmentation, but not with cross correlation. Although the paper claims to have promising results, triangulation with two microphones only gives blurry results, especially when the distance between keys is less than 7cm. Also, it should be difficult to achieve a

localization in x and y axis, since the only two microphones are in use and the test scenario reduces the keys to only one dimension. As future improvements they suggest more and better microphones and better sound card as well as a dictionary for improved localization.

Some years later, another research team improved in [4] the latter approach of geometric triangulation by exploiting the time difference of arrival (TDoA). Again, it is tried to make a context-free key recognition to not be dependent on previous knowledge, high-costly learning and language-model limitations. Since smartphone microphones got better in quality and resolution, as well as the number of built-in microphones increases to at least two, using several smartphones positioned around a keyboard to detect key positions is promising. In the well-documented work, they firstly talk about the theoretical feasibility and inevitable errors of this sensing approach in general. So the minimal differential distance between two sound signal points is $c_{sound}/f_{sampling} \approx 0,77\,cm$ whereas the center of two keys is 1,9-2,2 cm apart. Thus, it is technically feasible to measure the distances, but still the low sampling rate $f_{sampling}$ of 44,1 kHz causes a deviation of distance estimation which expands the candidate region of typed keys. Besides, the other challenge is the unknown keyboard layout and unknown positioning, which both is contained in the context-free scenario approach. For the first challenge - finding the right key candidate - they use as already said before, several phones with multiple microphones, where the acoustic signal arrives asynchronously since the distances differ. Because in a device are at least two build-in microphones, this knowledge together with $f_{sampling}$can be used to model a hyperbola band per device, which stands for the zone where the key is assumed to be. Subsequently, overlying several hyperbola bands from smartphones create a better and smaller estimation area of the keystroke's origin.

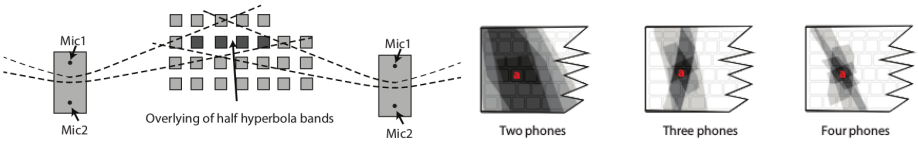


Fig. 2. Intersecting hyperbola planes estimating the keystroke position

For calculating a hyperbola band to a keystroke, several steps need to be done: Firstly, identifying and isolate the keystroke from all input streams. For identification, they take the greatest value over a 10-samples accumulated energy function as t_{hit} as a pin-point. From there, 10ms before and 90ms after are isolated to form a piece of acoustic covering the keystroke. Secondly, the TDoA must be estimated. Simply using pairwise $\triangle t_{hit}$ caused a too high deviation due to multipath effects and aliasing, so in the end generalized cross-correlation (GCC) in frequency domain with a heuristic-based Phase Transform (PHAT)

weighting function was applied pairwise on each piece of acoustic to get a more precise pin-point t_{gcc} for calculating the differences. Having the timing differences and the deviation error, the hyperbola bands can be set. This procedure is repeated with all keystrokes. In the end, there will be a heatmap of sets of hyperbola bands, which forms a keyboard-layout-like pattern. This clustering is used by an optimization algorithm to derive the keyboard's position and thus the mapping from clusters to characters. This approach was tested by two volunteers typing about 3300 characters of a text and over 72,2% of keystrokes could be recovered successfully. The robustness test in some noisy surroundings reveals an detection accuracy of at least 64% in a noise conference room and better. Other challenges they had to face were intra-, inter- and external clock-drift of sensors and optimization of smartphone positioning / rotation. A limitation of their approach was the exclusion of certain keys (e.g. CapsLock). Concluding, the combination of this context-free method with a context-based approach to increase the recognition rate is suggested.

The next work [5] comes up with the idea to build a virtual keyboard for mobile devices, which is called UbiK. They propose an input method like typing with fingers (including nail margins) on any solid surface with a printed keyboard layout attached onto it. This input method offers 10-finger writing on large workspaces, saves precious touchscreen area and is insensitive to gentle taps and touches. In the initial setup, the mobile device need to be placed very closely next to the typing area, whereby the user types all the printed keys (56 keys in this paper) at least once to generate training sounds. While touching sounds in previous works are used to extract energy- or FFT-features, this paper uses amplitude spectrum density (ASD), which allows more fine-grained centimeter-scale granularity and is insensitive to waveform ambiguities and unpredictable stretches. The thought behind UbiK is, that the audio waves from a sound source undergo a complex multipath reflection pattern on the surface and also the body of the smartphone as they propagate towards the microphone. These pattern of sound reflexion and cancellation can be characterized by the ASD. The ASD of different keystroke locations are highly distinguishable profiles and can be used as location signatures. Within a short distance of several millimeters, these signatures exhibit a certain level of correlation, which fades monotonically as distance increases. The presence of two microphones per device in modern smartphones makes it possible to access another feature set of the same event, what gains additional information for the decision process. Unfortunately, by default, Android allows only access to one of two microphones. To bypass this restriction, the app needs to load the tinyalsa driver, which gives access to both microphones at $f_{sampling}$ of 48 kHz. The teams developed virtual keyboard consists of three components: *detection*, *localization* and *adaption*. (See Fig. 3 for detailed architecture.) In the *detection* component, for detecting touching sounds, an adaptive threshold on the signal energy is used to separate noise floor from keystrokes by applying the Constant False Alarm Rate algorithm (CFAR). Thus, only signals having energy over a certain moving-average-window-plus-margined-value can pass. Additionally, loud environment noises are filtered out, when they not also

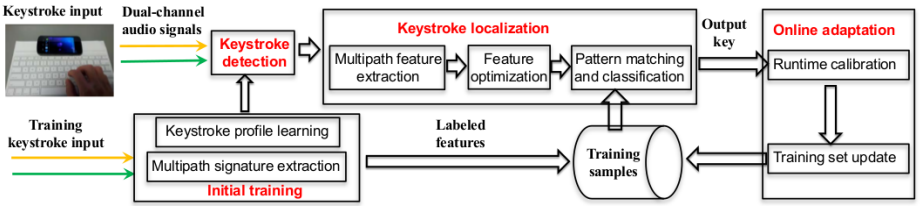


Fig. 3. UbiK system architecture

stimulate the smartphone's gyroscope within a certain empirically set threshold. So the detection algorithm only declares a keystroke if both, the audio and gyroscope confirm its presence. In such way, false alarms caused by human voice are eliminated since it does not cause surface vibrations. During the *localization* phase of keystrokes, the amount of samples, which are used to compute the ASD, is chosen dynamically depending on the noise floor and the energy peak level. From those samples, only a small frequency bandwidth, where the main ASD features are concentrated, is used for ASD. Capping the frequency range of the profile prevents noisy features from polluting localization accuracy and decreases calculation speed. Again, the concrete bandwidth is computed by an optimization algorithm. However, the band is somewhere between 100 Hz and 5000 Hz. The keystroke profile localization itself is a nearest-neighbor search with an Euclidian distance metric on both microphones features. To increase significance of the online localization results, UbiK calibrates itself by combining its localization suggestions with typed user-input and on-screen user corrections. This keyboard setup has an accuracy of 90% after only 3 training instances per key. Together with user's online feedback, accuracy reaches up to 95% at an average processing latency of 51,4 ms and standard deviation of 2,7 ms, which gives users no lag experience. Localization accuracy drops to around 80% if the phone is moved by one key's edge size. With less displacement (1/3 or 1/2 key), the decrease is much smaller. But because of the online adaption scheme, accuracy can be quickly restored to above 95% after a few tens of inputs. Although UbiK's results are already very good, although it's independent from language model and online usable, it still needs the user's cooperation in providing training data $(56 \times n)$ and online corrections.

This problem the following paper [6] wants to avoid by improving the triangulation approach on unlabeled data of [4] one year ago. They want to achieve a training-free and context-free eavesdrop-attack as their forerunners [4] did, but only using one single smartphone, instead of two. But for a proper triangulation in two dimensional space, there must be at least three distributed microphones. Although there are already smartphones existing which have three built-in mics, neither Android nor iOS currently offer interfaces to developers to access all three. So the scientists' work and contribution is to use only the two accessible but quite narrow-located microphones in one smartphone for localizing keystrokes in a single dimension by TDoA measurements. This leads again like

in [4] to an hyperbola band of indistinguishable keys. Having a-priori knowledge about the keyboard layout and the phone's position relative to the keyboard, the second dimension problem is solved by binning the indistinguishable keystrokes together and apply a k-means clustering with cityblock metric on their MFCC features. These from 400 Hz to 14kHz band-limited features carry a slightly different acoustic signature for each key due to physical imperfections across keys. Since the acoustic signatures are only used for separating clusters, there is no need of further labeling. The clustering of similar-sounding keys within the bins allows to calculate a mean TDoA-value, which is very close to the expected theoretical TDoA. So, labeling of the clusters is done by assigning the closest distances between these values.

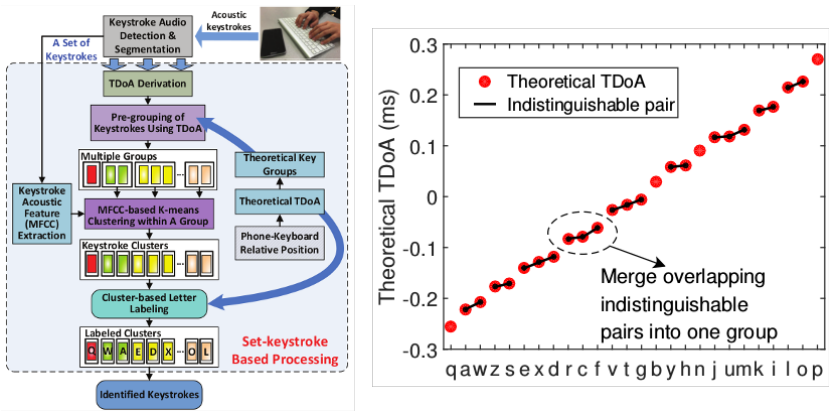


Fig. 4. System architecture of [6] and indistinguishable key groups

For testing, the researchers focused on recognizing the 26 letters of the alphabet, typed by three participants in laboratory environment under different test settings (sampling rate, keyboard) typing in total around 3600 keystrokes. Tested were three different keyboards models: An Apple wireless keyboard MC184LL/A, a Microsoft surface keyboard and a mechanical keyboard Razer Black Widow Ultimate. But it turns out that the keyboard model and key design doesn't have a significant effect on the results. As attacker smartphone, a Galaxy Note 3 was used. Although its audio chips are capable of 192 kHz playback and recording, the installed Android version 4.4.2 reduces the sampling rate to 48 kHz. It is said, that Android 5.0 claims to support 96 kHz and the research-team assumes, that one day newer Android versions even will offer 192 kHz support. So for testing their keystroke recognizer with that high sampling rates, two omni-directional microphones are placed in the exactly same distance of Note 3's microphones at 96 kHz and 192 kHz. The test result at 48 kHz got an accuracy in top-1 of 85,5%, top-2 of 94,9%, top-3 of 97,6%, precision of 87% and recall of 85%. Increasing the sampling rate to 192 kHz, the test results go up to a top-1 accuracy of 94,2% as

well as precision and recall over 90%. This is an amazing result for a context-free approach. But it should be noted, that the latter values are technically not yet possible with off-the-shelf smartphones, but an important proof of concept for the future based on a robust context-free TDoA approach.

2.1 Summary of acoustic methods

Giving a summery about the papers dealing with acoustic emanation sensing (Tab. 1), we have to state that there exist already very powerful methods of sensing. It doesn't matter, which aspect of the challenge we face, whether as text-input method, as a context-free attack or an attack with language model: In every section a solution is found with a high accuracy. Attacking with a big amount of unlabeled eavesdropped data, which is fed into an hidden markov model and combined with a language model to filter misclassification, we get the highest result of 95,7% accuracy ([2]). The big disadvantage of the machine learning approach is, that this method requires much training data, much know-how of data processing and a language model, which isn't able to recognize passwords. Accepting a little fewer accuracy of 94,2% but with less effort in data processing, the TDoA & MFCC approach of [6] can be used. The time difference of arrival is used to distinguish keys in a one dimensional plane and the MFCC to distinguish keys within a plane area in a second dimension. It is context-free, robust and doesn't need any training data and works well on passwords as well. Since both attacks processes the test data in a batch block (offline), they can't be used as a live text-input method (online). The approach to do acoustic keystroke sensing online is UbiK ([5]) achieving 95% accuracy with online user-feedback to improve the classification. The classification itself exploits the knowledge that finger typing on same surfaces causes same emanations. But for every single usage on a new surface, training data need to be manually collected at least once per key before typing can start, what makes the approach difficult in real world situations.

Paper	Year	Features	Training	Data processing	Acc
[1]	2004	Touch peak emanations, FFT	supervised	neural network	79%
[2]	2005	Touch peak emanations, MFCC	unsupervised	HMM, language model	95,7%
[3]	2006	Time Difference of Arrival	labeled data	mean value comparison	-
[4]	2014	Time Difference of Arrival	unlabeled data	certainty distribution	72,2%
[5]	2014	Amplitude Spectrum Density	labeled data	nearest-neighbor	95%
[6]	2015	TDoA & MFCC	unlabeled	nearest-neighbor	94,2%

Table 1. Comparison of acoustic related papers

3 Video-based sensing methods

Another way to eavesdrop keystrokes is to observe people during their text-input. An often classically in movies used approach is to track a person at an ATM or in the office by ceiling-mounted surveillance cameras. There are also other possibilities using a video camera for input sensing. As well, video sensors can be used for tracking keyboard motions or finger motions, just to name basic ideas. The recorded film material can then be processed to extract the keystrokes from the picture stream. Although it sounds an easy task to humans to extract desired information, in reality this approach faces several ambiguity problems during automated computation, as the following papers will show.

In "A Virtual Keyboard Based on True-3D Optical Ranging" ([7]) from 2005 researchers present a complete text-input system for contactless input for disabled users or sterile environments. The system consists of a pattern projector and a 3D range camera to detect typing events on arbitrary surface (Fig. 5) The camera is a Swissranger SR-2, using grey-scale and time-of-flight depth information for scene reconstruction. Thus, the finger tracking is not based on skin color detection but uses hand's contour and fits the depth curve with different feature models. This also makes it possible to skip the training for user adaption. The size of the working area is limited by the camera's spatial resolution of 160 x 124 px, which gives 15 cm x 25 cm detection space and maximum depth resolution of 0,6 cm.

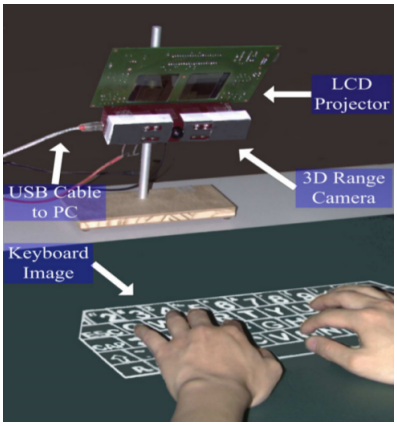


Fig. 5. Setup of the Virtual Keyboard

The hand is segmented by separating foreground from background using depth information from calibration, when the system is turned on. For estimating the fingertips, the hand shape contour is extracted from the foreground and curvature extrema are searched in it. Their approximation result in a local extrema finding around the tip centerpoint, which in the end can be expressed as a

sign change detection of the curve's second derivation. The keystroke detection is made by ploting the relation between fingertips position and its depth, which should fit a smooth second-degree parabola if a keystroke happened. In reality, this works pretty well at 33 fps, which is enough for normal typing, where fingers move at most 10cm/s. A Challenge is, as for all visual based hand tracking, the finger occlusion problem for two-handed typing. Here a potential solution would be applying more complicated 3D hand models to estimate the occluded fingers, what lowers the system frame rate a lot. The testing spent lots of effort in taking different parameters into account like users of different races, typing skills, genders, left- and right-handed subjects, in-/outdoor, different lighting conditions and a test pattern covering all key positions. What was found out was, that the most dominant detection error is the false detection, which is mainly caused by the low lateral resolution of the camera, which could not resolve very small floating distance between fingers and the table. But after some time, experienced user increased could increase accuracy and reach a normal typing speed of 30 words/min. But in average, false detection rate was at 7,4%, missed stroke rate at 3%, incorrect detection rate at 1%, typing accuracy at 88,6% and the average typing speed was 27,1 words.

A more mobile approach was presented 2006 from researchers from the university of Alberta [8]. Their paper introduces a wrist-worn wireless video camera, that tracks finger position from underside of each wrist. These features are transmitted wireless and processed live on a computer, where a Hidden Markov Model correlates features to small number of possible keystrokes. Then, a language model is fixing ambiguities and returns the typed keys. This process can be divided up in several stages. The *hand extraction and fingertip recognition*, the *keystroke detection* and the *HMM based character recognition*: The hand recognition is done by skin color discrimination on the color video material followed by a closing filter. The fingertips are detected as minima of the extracted hand contour and the phalanges ("the valleys in between the fingers") are detected as maxima. The scenery the camera sees and the extracted features are shown in Fig. 6

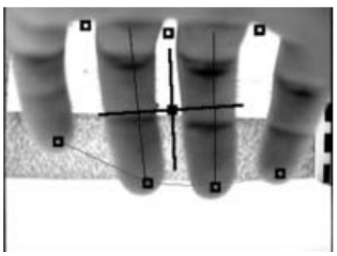


Fig. 6. Seven interest points and orientation of the hand

The keystroke detection uses the feature-set of four fingertips and three phalanges, which are moving relatively to each other during a keystroke. A keystroke is detected, if one or more fingers deviates from the rest position above a distance and speed threshold within 30 frames. For smoothing the noisy position readings, a first order low-pass filter is used to avoid spurious keystroke detections.

In the HMM-based character recognition stage, an Hidden Markov Model is used to correlate the feature-sets to possible keystrokes. So the HMM observations consist of a set of vectors that describe the position (in polar coordinates relative to the resting position) and the speed of the fingertip. The hidden states describe characters and the transition model is based on letter bigram probabilities and word unigram model. The observation model is learned during training by correlating each character in the training set with the output vector observed during its corresponding keypress. The implemented prototype system had a camera with a frame rate of 30 fps and a resolution of 320 x 240 px. During the training and test, which was run in a well-lit indoor environment, a person was left-handed trained on a 300 word test set over a period of one hour. Each character was typed in the training about approximately 25 times, and the mean and variance for each finger movement vector was calculated for the Gaussian probability distribution for the observation. Afterwards, the test having 200 words was used to measure recognition accuracy and speed, where false positives were classified as errors and undetected keystrokes were retyped. In the end, without any language model, the recognition accuracy reached 67%, with letter bigram support 84% and with both, letter and word model 90%. Since in the latter result also the word model is used, sometimes the word kept being misspelled until the last letter and then switch over to the correct one. Because this test only included the left hand, no space bar was defined, so the bar was pushed with the free right hand to indicate a words end. The typing speed was 14 words/min due to smoothing filter which adds 0,1s lag and the retyping of characters, when the detection was missed. To increase their results, the team suggests for the future an adaptive skin-color segmentation, deformable templates or dynamic contours for tracking the edges of fingers more accurately, Conditional Density Propagation algorithm to track contours in cluttered environments or Hough Transformation for fingertip tracking. Also fitting the observed hand contour to a 3D physical model of the hand could increase robustness.

The last of the most promising papers regarding video sensing, is a proposed analysis tool, which should supports analysts in transcribing text from a camera source, which observes somebody typing on a keyboard. “ClearShot” ([9]) from 2008 operates on an image stream produced by an off-the-shelf web cam which records the typing activity of a user. The tools consists of two main phases (see Fig.7) : The first phase is the *computer vision analysis*, where the recorded video is analyzed frame by frame and computes a set of keys that were likely pressed, certainly not pressed and the position of space characters. Because the results of this phase are noisy, another phase, called *text analysis* is required. The goal is to remove errors using language and context-sensitive techniques. The result of this phase is the reconstructed text, where each word is represented by a

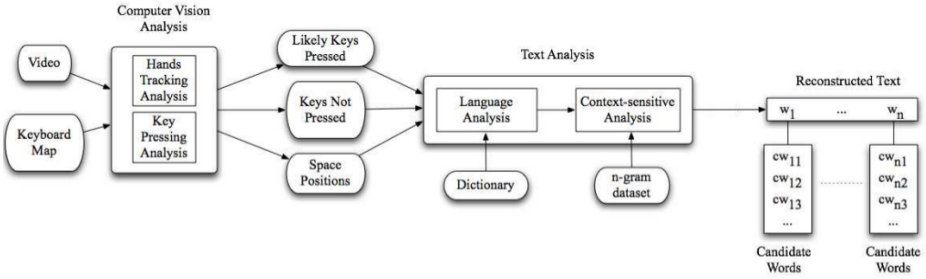


Fig. 7. Overview of the analysis steps in ClearyShot

list of possible candidates, ranked by likelihood. The computer vision analysis is divided up into sub-tasks *hands tracking* and *key pressing analysis*. The first task was tried to solve by optical flow analysis with Horn-Schunck algorithm, but they failed since the algorithm is sensitive to noise, what introduces errors in the estimation of movement intensity and direction. Additionally, typing pattern is are more complex than they originally assumed. In the end, contour analysis is used by frame-wise differentiating each frame from the previous one. This way, standing still objects like the keyboard is deleted from the result, while moving parts like the finger is kept. For the key pressing analysis, two methods are used: The first method is a light-based analysis, which shows sensitivity to a change in light diffusion in the area of a pressed key. The second method is occlusion-based, which separates keys which aren't covered by the hand from those, who are covered based on data provided from hands tracking analysis. The output of the key pressing analysis now relies on combination of the contour detection (areas, where finger has moved), with the lighting results (keys, which lighting has changed). From this list of keys, which are likely pressed, the information from occlusion-based technique (which holds information about keys), is subtracted. Besides of this, there's also a lighting-based tracking of the space bar to know the ending of a word. The major challenges in keystroke recognition analysis are for example, that the hand recognition analysis doesn't provide clear contours, and that occlusion sometimes hinders the detection of key pressings. So the light-based technique doesn't perform well, when the user's hand projects a shadow that makes the lighting uniform in a region of the keyboard. On top, touching a key, when a finger moves over it, doesn't necessarily imply, that the key has been pressed. For these reasons, the results produced by this phase are noisy in terms of false pressings or missed pressings. This motivates the last phase, the text analysis. There, likely-to-be-pressed-keys for every frame are grouped over time to key groups, when they occur sequentially for some time and don't overlap with other keys. Otherwise, a key group with several of those characters are created. Are there no detections of keys for a certain threshold-time, blank key groups are inserted to keep the likelihood of undetected keys at that time. A weighting is attached to each key group, which is proportional to the number of consecutive frames, which contains them. Then the key groups are fed into a

word model, which calculates a list of words, which are most likely typed based on the key group sequence without taking context into account. This isolated list of word candidates and their scores are then put 3-lists-wise in a 3-gram language model, which returns a sorted list of the most likely combinations of this small word sequence. In the end, the attacker has to manually review the sequences and choose the one which fits best. For testing the system, two persons with different typing styles typed 118 words each, observed by a web cam. This material was reviewed manually by two analysts what took them 59m at 89% accuracy respectively 1h55m at 96%. Clearshot's computer vision analyze time took two or three times the recording time ($< 15min$) while the language based analysis time wasn't told, but depends on the number of false pressings. For person A, the system has the correct word combination in top1: 46%, top5: 64%, top10: 72%. And for person B: top1: 36%, top5: 58%, top10: 68%, top25: 73%. This shows, that the accuracy is strongly dependent on the typist and anyhow just can support the attacker, which can guess the topic of the input very quickly and also gets supported a lot by the interpretation.

3.1 Summery of video based methods

There could just a few papers about video-based sensing be found (see Tab. 2), but they show, that it is technically possible to sense text-input from video material. At that time, good results only could be achieved by cooperation of the user in the sensing scenario as an alternative text-input method. There either the user has to type on a projected pattern which is observed by a time-of-flight camera, which achieves an accuracy of 88,6% or a wrist-located camera tracking the finger movements, achieving 90%. But because the first paper uses too many extra hardware, it can't be used as a mobile sensing device. Also as attacking method, the usage would be too intrusive, by the reason that the sensor's sensitivity range is very limited. The wrist-worn camera approach sounds promising, since smartwatches are more and more in use. The algorithms are rather simple and once they are calibrated to the user's physiology, they might converge very soon. However, the problem about what to do with the other hand wrist still exists. The only attacking approach with video sensing by observation has a very bad accuracy, but it never was meant to be that accurate, since it is a semi-automatic system, which needs human verification support.

Paper	Year	Features	Training	Data Processing	Acc
[7]	2005	greyscale, depth	none	fingertip tracking	88,6%
[8]	2006	fingertip, phalanges	labeled	HMM, language model	90%
[9]	2008	moved keys	none	key groups, language model	$\sim 41\%$

Table 2. Comparison of video based sensing methods

4 WiFi signal based sensing methods

The maybe most surprising sensing approach of text-input is the use of WiFi signals. Some years ago, there were experiments about exploiting the variation of WiFi signal strength to sense movements in an apartment, which are magnitudes bigger than caused from text-input: one approach was to detect drops of old persons on the floor and the other approach was hand gesture recognition [10][11]. The idea behind WiFi signal sensing is to use modern WiFi's channel state information (CSI), which are used in IEEE 802.11n/ac standard to control the multiple receive antennas for MIMO. The WiFi devices need to continuously monitor the state of the wireless channel to effectively perform transmit power allocations and rate adaptions for each individual MIMO stream such that the available capacity of the wireless channel is maximally utilized. These devices quantify the state of the channel in terms of CSI values. The CSI values characterize the Channel Frequency Response (CFR) for each subcarrier between each transmit-receive antenna pair. So when the number of subcarriers $S_c = 30$, the number of transmit antennas $M_T = 3$ and number of receive antennas $M_R = 2$, the total amount of data in a single CSI is S_c x M_T x $M_R = 180$ values.

The key idea, what the paper "WiKey - Keystroke recognition using WiFi Signals" [12] from 2015 proposes to use is the interpretation of CSI time-series. CSI values quantify the aggregate effect of wireless phenomena such as fading, multi-paths, and Doppler shift on the wireless signals in a given environment. Since CSI vectors are changing when the environment is changing, the same can be observed during typing: a certain key press forms a unique hand formation and thus a unique pattern in CSI time-series. This so called CSI-waveform can be sensed by normal off-the-shelf WiFi devices and distinguished in a trained classifier. All those devices need to do causing permanently traffic to produce enough CSI values within a keystroke to construct a high resolution waveform. The greatest possible sampling rate on this hardware was about 2500 samples/s. The challenges to overcome in this paper were *keystroke segmentation*, *feature extraction* of 37 keys, *feature comparison* and *evaluation*. For the *keystroke segmentation*, firstly the big amount of raw data (time-series of all subcarriers) need to pass a Butterworth low-pass filter. Due to observations, the important frequencies are between 3 Hz and 80 Hz, which were additionally set as cut-off frequencies in the filter. Because hand and finger movements result in correlated changes in the CSI-waveform, after normalization, principal component analysis (PCA) is applied to remove uncorrelated noise and returns hand and finger related movement data. From observation it was stated, that the first principal component captures the majority of correlated noise and thus can be dropped while the next three components contains useful features for further segmentation. Keystrokes can be observed in the CSI-waveform as increasing and decreasing trends in rate of change, what can be found by using a moving window approach which calculates the mean absolute deviation (MAD) for each of the three time series. After evaluating variances and applying energy-thresholds on the sliding window results, start- and endpoints of a keystroke can be identified. For *feature extraction*, these points are taken to cut relevant

data from the original CSI-waveform, which is again processed with PCA for the same reasons like before during keystroke segmentation. To reduce the amount of data points, but keeping the waveform shapes in time and frequency domain, Discrete Wavelet Transformation (DWT) is applied three times, what generates approximation coefficients which can be used as keystroke features. The amount of keystroke features in total is 3 x M_T x M_R. To *compare keystroke shape features* in a classifier, the researchers choose dynamic time warping (DTW) with Euclidean distance as a comparison metric to determine minimal distance between two waveforms. This metric is used in a k-nearest neighbor classifier for all transmit-receive antenna pairs, what makes in total 3 x M_T x M_R classifiers. The final result is calculated through majority voting of all kNN classifiers. To *evaluate* the accuracy, training and testing data from 10 users were collected. Nine of them provided 30 sample keystrokes for each of the 37 keys and typed an example sentence five times. The 10th user provided more training data: 80 sample keystrokes each, and five times five sentences. The training data were typed slowly with 1 second delay. WiKey's keystroke detection rate on that data was 97,5%, but they promised that the rate could be further improved by parameter tuning. The missed keystrokes are usually those, where fingers move very little when typing. For building classifiers, two sets of experiments were tested: The first set of classifiers consisted of classifiers for each of the 10 users using 30 samples for 10-fold cross validation accuracy. The second set was a classifier for the 10th user using 80 samples to see what impact more training data has on the accuracy. For the kNN classifier k was set to 15. The overall accuracy for 30 samples recognizing 37 keys was 82,8%, while reducing keys to 26 results in 83,5%. Testing the classifiers on real sentences gives 77,43%. Increasing the training samples by having the 80 samples set results in 85,95% accuracy for 37 keys and 89,7% for 26 keys. Testing the classifier on the real sentences of the 10th user ends with 93,47%. This is a very good result, but also the limitations of WiFi sensing must be seen. Stable results only can be achieved under controlled environments. During the test, only two persons were present and nothing was moved in the room during data collection. So the CSI-data is mainly catching variations caused by keystrokes, what improves finding clear features. For using WiKey in different settings, training must always be repeated because parameters are highly scenario dependent. Researchers found out, that linear distance changes towards the access point lets the waveform's shape diverge as well as changing orientation or even measuring on a different day. Those easily fading features makes it hard to keep quality in open environments. In the end, it can be stated, that WiKey is a user-dependent supervised classifier method, which works under very limited conditions.

4.1 Summary of WiFi based methods

The only WiFi sensing paper, which is really able to sense keystrokes, strikes with a very high accuracy. It could be used as an attack, as well as an text-input method. What clearly has to be said is, that the features are prone to any change in the environment after training, what changes CSI-based features and hence

decreasing accuracy. Although a cooperative user would be best to generate a good accuracy, also an attack can be successful with an attack vector like: a chat conversation with the victim, while the victim is observed in his room via a hacked router to train the classifier. To conclude, this is a remarkable approach which should be taken into account as a possible attack method, but lacks in robustness what makes it hard to use in reality.

Paper	Year	Features	Training	Data Processing	Acc
[12]	2015	Channel State Information	supervised	DWT, DTW, kNN	93,47%

Table 3. Overview of WiFi based sensing method

5 Acceleration and motion sensing methods

Coming from the external sensing methods like acoustics, video or WiFi signal interferences, also already existing gyroscope and acceleration sensors in mobile devices itself can considered as text-input sensing or at least information gain. Smartphones as well as smartwatches are usually equipped with both sensor types: gyroscopes for sensing spin acceleration and accelerators for sensing linear acceleration. Because every touch-device experiences vibrations and acceleration from the finger pressing a button, researchers want to find out what can be reconstructed through sensed acceleration data.

The first papers, interested in this topic is "TouchLogger" [13] which was published in 2011. This paper wants to make aware of the underestimated problem, that acceleration data is not protected well by rights management in modern smartphones, which can be exploited by malicious apps. Motion of a smartphone during typing depends on several factors: 1) striking force of the typing finger, 2) resistance force of the supporting hand, 3) the landing location of the typing finger and 4) location of supporting hand on the smartphone. The first two factors mainly affect the shift of the phone, while the latter two mainly affect rotation. The scientists observed, that just the latter two factors are user-independent and thus can be used as feature source. Also they observed gyroscope data contains senses the desired rotation data better although with a lower sampling rate. Androids motion data can be requested by a listener, which provides motion data in rotation angles for each axis as α, β, γ. Touchlogger discards the azimath angle α and only calculates features from the other two axis. Keystroke segmentation is done by calculating Peak-to-Average ratios of β and γ, which are much larger during typing. Their feature extraction approach is, motions caused by keystrokes do not reach their peaks regarding pitch and roll angle simultaneously. This vector path further abstracted into lobes, which can be seen as unique pattern (see 8). From this, two features can be extracted, expressing the angle of the direction of the upper lobe and the x-axis (AUB), as well as from the lower lobe (ALB). These features are used to model probability density functions in

a Gaussian distributions for during training for classification. Another feature, what are used for identifying keystrokes is the width of the lobes and also the angle of dominating edges (AU/AL). Trained was on 449 strokes of digit keys in landscape mode, and the classifier was able to correctly infer 71,5% of them. More details about how was trained or tested was not further explained in the paper.

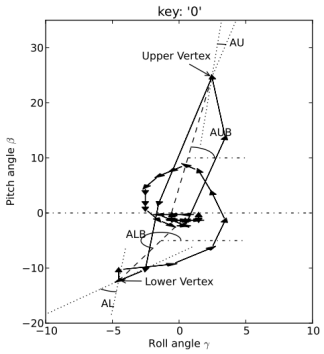


Fig. 8. Pitch and roll angles during keystroke and extracted lobes AUB and ALB

The next paper "ACCessory" [14] from 2012 goes further. They want to show that accelerometer readings are sufficient to extract sequences of entered text on smartphones. As threat model, a malicious app runs on Android only having the permissions to send information over the network as privilege. Running as service in background, the app tries to infer password entries by on-screen location estimation of the keypress. The used principle is a machine learning approach. For segmenting keys, spike detection on the acceleration stream by using root-meansquare was used giving 94-96% segmentation accuracy. From each keys motion data, 46 statistical features were extracted like root-mean-square value, min/max value, average sample by sample change, number of local peaks etc. On these data, a Wrapper algorithm searches exhaustively for features maximizing a pre-specified evaluation-measure. After this, researchers made two studies: The *first task* was formulated as hierarchical classification problem. They recursively partition each area of the screen into two parts, and then classify individual keys within each new subarea. Trying out several classification like Multilayer Perceptron, Support Vector Machine and a C4.5 decision tree, in the end it turns out, using Random Forest with n C4.5 decision trees works best. There, c randomly selected features of the feature space are looked at for splitting decision. They suspect that Random Forests performed well because of the propensity for significant variability of feature values between instances of the same label. Random Forests create multiple trees based on a varying subset of features, making it more robust than most other classifiers to intraclass variability. For this study, data were collected from four people doing 1300 keypresses on 60 screen

areas with having approximately 20 samples per area. Testing was done using k-fold cross validation and test results predictions have an accuracy of 80% in a concentration within an error of 0 or 1 key distance. Furthermore, 91%, 96%, and 99% of their predictions are within 2, 3, 4 keys neighborhood. The *second task* was about infering simulated password entries from motion data. Classification in this task is done by a conditional probability mass distribution on the statistical features mentioned before followed by a maximum likelihood search for this feature. For training the conditional probability distribution, 2700 keypresses from sentences and 99 6-character ([0-z]) passwords are collected. From this data, their model was able to correctly deduce 6 of the 99 passwords in a median 4,5 trials. A brute force attack on a 6 character length password takes approximately 2^{28} trials on average. The model presented here cracked 59 of 99 passwords in approximately 2^{15}median trials. From these results it can be concluded that accelerometers can be used to significantly reduce the search space for text entered on smartphones. To reduce the risk of snooping acceleration data, the researcher propose decreasing sampling rate for untrusted applications, what can substantially mitigate predictive accuracy to keystroke inference. The results of this paper is significantly better than from the previous, since it can distinguish 29-60 areas in a sequence instead of only 10 single digits.

Some years later, in 2015, another attack on number pads was researched based on accelerometer and magnetometer in smartphones. The magnetometer is used here, because in Android 4.4.x, accelerometer and magnetometer data are used to calculate the value of "orientation sensor". Again the idea is a malicious app as attack vector and was tested in three different real-life scenarios as the reader will see in the test results. The raw 3-axis accelerometer values are Kalman filtered to remove the mixed in gravitational component. Then the magnitude of external force on the touchscreen $F^2 = {A_x}^2 + {A_y}^2 + {A_z}^2$is measured as squared value of the three axis. It can be seen, that the magnitude curve of a keystroke correspond to a three-stage process: 1) the smartphone moves downward with increasing acceleration when a finger touches the screen; 2) the smartphone moves upward; 3) the hand holding the smartphone returns to initial position. From this process model, measures can be statistically extracted like maximum value of the first/second/third fluctuation, time differences between them, etc. for every tap action. Given the statistical analysis by using kernel density estimation method, the probability density function (PDF) for those values shows roughly normal distributed curves. The tap segmentation challenge is now solved by applying thresholds from the previous defined measures on the F^2data. The features, needed for key recognition can be extracted from three axes of acceleration and two axes of orientation. Again the statistical appraoch is choosen: A single tap event can be represented by the mean, median, mode, skewness, kurtosis and standard deviation extracted from the 5D sensor data. These features were used to train four classification algorithms for comparison reasons. The researchers used four types of two-class classifiers. Random Forest, Support Vector Machine (SVM), Neural Network and Nearest neighbor to construct inference models. The input inference task was regarded as a multi-

class classification problem that applies the one-against-many approach. Given k classes, one for each number on the number pad, k components are build for each of four multi-class classifiers, one for each class. The resulting label is the class with the highest score. The scenarios, for which the classifiers are trained and tested is "hand-hold-tap" - where tap actions are done while sitting or standing still, "table-hold-tap" - where the smartphones is placed on the table and input is done with a single hand, and "hand-hold-walk" - where input is done while walking. For each scenario, about 2400 number pad taps were collected with highest possible sampling rate on Android. The tap detection accuracy are 100% and 95% in the hand-hold-tap and table-hold-tap scenarios. While walking, accuracy becomes worse (74%), what might be caused by fluctuation induced from finger pressure and walking mixed together. User input inference was solved by SVM with a better performance than the others, while neural network classifier had worst results. Mostly, accuracy for SVM was over 70% what was claimed to be promising and competitive and better than guess. Unfortunately there's no better results in values available in that paper, but Fig. 9 shows the accuracy for SVM for all three scenarios, giving an idea how the distribution is.

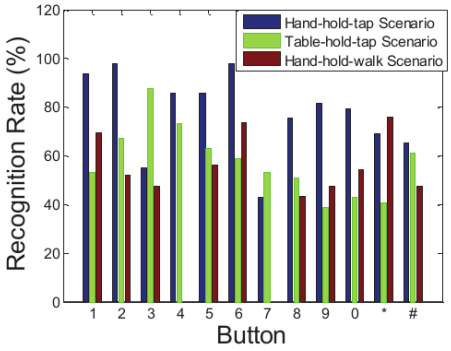


Fig. 9. Results for SVM classification in [15]

Another completely different way to exploit motion sensors from smart devices, is the use of wrist-worn smartwatches to recover keystrokes made on a keyboard. The unique constraint in this scenario is the other hands motion is missing in the data as well as the motions sensed with the watch can relate to 3 or 4 different keys. The following two papers deal with this topic, using different processing techniques.

"MoLe - Motion Leaks through Smartwatch Sensors" [16] from 2015 tried first to use the accelerometer and gyroscope data from a Galaxy Gear smartwatch to infer the words of a typing user. The key idea is to interpret motion data in a 2D keyboard plane instead of a regular 3D space. As well, many users using the reference position ("F" and "J" keys) what helps developing Bayesian decisions. Seen from this reference position, hand motion experience a large

positive displacement when typing "12345" and a negative displacement when typing "zxcvb". Near the rest position on the third row ("asdf") nearly no displacement is detected. But decoding characters get more complicated, when the user types a word rather than just a single character, since the motion is relative to the previous position of the key. Therefore, researchers modeled the system shown in Fig 10 to tackle the challenge: First, the attacker needs to build up a database for ground truth, where the attacker/s type each character on the keyboard multiple times. This is called a *"character point cloud"* (CPC) and is processed offline and stored for use later. The raw sensor data from attacker as well as the victim later, which is sampled at 200 Hz, is passed through *keystroke detection*. There, the keystrokes are segmented by discernible dips in the negative z-axis by bagged decision tree and a 2D-displacements map is computed by gravity removal, mean removal, double integration and Kalman filtering. The displacement map is called *"unlabeled point cloud"* (UPC) and contains tuples $<location_i, time_i>$ of the estimated location of the watch at the time, when a key was pressed. The UPC is forwarded to the *"Cloud Fitting"* module, whose task is to assign approximate labels to the points in UPC. For this, the cloud fitting module obtains the CPC that was computed earlier, and scales and rotates the convex hull of the CPC to best fit the convex hull of the UPC. The output is a rotated and scaled CPC which serves as the reference template for decoding unlabeled points in UPC. The *"Bayesian Inference"*-module (BIM) now takes three inputs: the template output from Cloud Fitting, the unlabeled points from UPC and a English dictionary W. Having this, BIM computes the a-posteriori probability that the unlabeled points form w_i for each word and returns a ranked list. In Bayesian computation, several aspects are considered such as: 1) Consecutive characters, which are adjacent on the keyboard, are pressed so fast, that they might not be separable. 2) Modeling the knowledge about key-layout and displacements as Gaussian distributions additionally onto the fitted CPC. 3) Current displacements are influenced by the location of their previous character. 4) Insertion of probable right hand characters with increasing time interval between two detected keystrokes. For offline-training data, two persons typed the 500 longest words from a dictionary. For testing, 8 persons typed 300 English words randomly selected from 5000 most frequent used words having an equally distributed word-length ranged from 1 to 14. The test result shows, that the median rank of a word is 24, while for 30 percentile, the rank is 5. In other words, there's a 30% change that MoLe narrows down the typed word to only 5 possibilities and 50% chance to only 24 possibilities. This is a appreciable reduction of search space, when we start with having 5000 words. It can be seen, that the ranking improves with word length greater 6, since more keystrokes are detectable and the amount of possible words is decreasing. The limitations of this work is 1) The inability to infer non valid English words or passwords; 2) The scalability across different watch models, since only Galaxy Gear was trained and tested; 3) Inability to parse sentences due to difficulties in detection the space bar, which is the reason why the volunteer victims only typed one word at a time, instead of free-flowing sentences. 4) It is assumed,

that the victim uses appropriate fingers for typing and is applying basic typing rules.

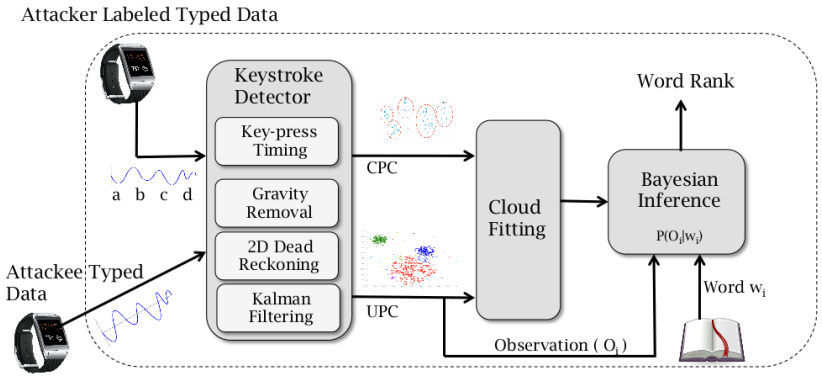


Fig. 10. System overview of "MoLe"

With "When Good Becomes Evil" [17] from 2015, there's another contribution to the young research topic about smartwatch motion sensor snooping. This paper presents a new and practical side-attack to infer user inputs on keyboards or numpads by exploiting acceleration and gyroscope sensors in smartwatch. Using motion data always comes with the problem of dealing with big variance what makes it unstable: Collected data is noisy, has small and irregular movements included, or shows different moving speed between two keys. Also sometimes the sampling rate is limited, what causes information loss. While previous papers tried to reconstruct the whole precise track of hand movements. This paper goes a new way, by only capturing hand movements between successive keystrokes and model them using displacements or motion directions. A transition diagram is built to assign probability to different combinations of keys. The paper applies this approach on two major categories of keyboards: *numeric keypad,* as used in POS-systems and *QWERTY-keyboards.* Although the same innovative approach is used on those keyboards, they need to be optimized separately on their specific constraints. Attacking numeric keypad has two phases: *Learning phase* and *attacking phase.* During learning phase, accelerations are converted into displacements along x- and y-axis by integration. Having a numpad, the amount of relative moves between keys are limited to 31. Additionally, when a submit button is modeled, another 10 vectors are necessary, what makes in total 42 displacement vectors. To reduce the amount of training and since the moves are symmetric, vectors are clustered with their reverse one ($0 \rightarrow 1$,$1 \rightarrow 0$), what lets 26 vectors remain. For *collecting training data*, 8 persons repeat typing the moves from each of the 42 cluster groups 15 times. In total 4920 training movements were done. This data gets re-sampling with cubic spline interpolation and filtered using Fast Fourier Transform (FFT) filter to get rid of linear noise and

high frequent noise. Then training data gets segmented by using the magnitude of external force F^2 (on page 19). From this points, where the threshold is surpassed, displacement vectors are calculated, optimized and stored in a state transition. In *testing phase*, the smartwatch continuously records the accelerations, covering the movement of typing 6-digit PINs and the "Enter" key. Three persons created in this scenario in total 300 6-digit-long test data. This data is segmented by a heuristic, that searches for a 6 times repeating pattern. After resampling and FFT-filtering, displacement vectors are extracted the same way as in the training. The Classification on displacement vectors was tried with several techniques including Random Forest, kNN, Support Vector Machine and Neural Network. It turns out, that k-Nearest Neighbor was the most accurate having up to 65% top3 accuracy, respectively 80% top10 accuracy. For performing an English text inference attack against *QWERTY keyboards*, for the first time acoustic signals and acceleration data from embedded smartwatch sensors are used. Using both input sources limits the negative impact of various noises, thus making attack more robust. The system was designed in three phases: *keystroke segmentation*, *keystroke modeling* and *word matching*. *Keystroke segmentation* is solved by using thresholds on the energy level of the DFT-filtered acoustic signal. For *keystroke modeling*, keypress events in this paper are clustered into "L" and "R" letters, dividing them up into left-hand and right-hand pressed keys. Hitting both letter types will produce noticeable sound, but only "L" letters introduces huge z-axis acceleration. "L" letters can be further divided into subclasses, depending whether the y-axis acceleration is positive, negative, or close to zero. This label [+1,0,-1] relates to whether the left hand moves towards a key on the keyboard, which is vertically a row higher, in the same row or a row lower. As an example, the word "quick" can be expressed as "L,R,R,L,R" or "+1,R,R,-1,R". This way, the whole English dictionary can be translated into that spelling scheme. In reality, user's left hand might also move to type a "R" letter. To make the scheme more robust, additional word profiles are added, having some "L-R" flips, when the word contains letters, which also could be typed with the other hand. To check, whether those transcription pattern are unique enough to infer words with high accuracy, a "corncob" wordlist was transformed to its tagged form: From 58110 words, 34121 tags were generated, while 26554 tags are associated with only one word. This shows, that the proposed tagging creates enough entropy to classify words well. Finally the system could be tested with 5 persons typing 27 test words, having a length from 7 to 13 characters out of "corncob" dictionary. The result was a top10 accuracy of in average 63%. To test the system in a context-aware approach, 5 persons typed some random sentences from four BBC-news articles and the related word dictionary was constructed using the context of the related articles. The dictionary contains 672 tags from 765 words and 615 tags were only associated with one word. Inferring accuracy was tested on 20 random sentences with 463 words in total. 57% of the words are correctly ranked top1 in the candidate list. If taking the top three candidates into consideration, the accuracy was raised to 88%. But from this 20 random sentences, only the words having not less than 4 characters were taken for this statistics,

since they contain more valuable information for guessing the meaning of the text. The advantages of this attack is the non-intrusiveness of to-attacking devices. Once the smartwatch is hacked, all input can be inferred. Additionally, by using acoustic signals and acceleration, high entropy is produced which improves accuracy compared to previous works. Limitations of this approach is the likely bad performance on words having less than 4 characters, which might be related to ambiguous tags, as well as inability of password or foreign language word recognition. Further improvements, the team plans to investigate is combining sensor data from smartwatch and smartphone, or using another noise filter to improve energy-level based keystroke detection.

With "(Smart)Watch your Taps" from 2015, there's another contribution on smartwatch motion data, but attacking a smartphone: This paper formulates an attack approach, which employs supervised learning techniques on linear acceleration data from smartwatches to infer typing on a smartphone's numeric touchscreen keypad. Additionally, the paper also uses smartphone's acceleration data as well, to infer typing on itself as other papers did already, but main focus is on smartwatch attacks smartphone. Two scenarios should be investigated: 1) *Non-Holding Hand Typing* (NHHT) - Wearing smartwatch and holding smartphone in one hand, typing with the other hand's index finger; 2) *Holding Hand Typing* (HHT) - Wearing smartwatch and smartphone in one hand, typing with its thumb. For training data collection, 12 persons typed 400 keystrokes of uniformly distributed random numbers in NHHT as well as HHT. During typing, linear acceleration is recorded on smartwatch and smartphone, additionally, keystroke ground-truth for labeling purposes is saved. Training data reveals, that keystroke events are clearly separated from one another with small but clear inactive time regions in both typing scenarios. A keystroke movement finishes after approximately 350 ms, what makes 18 samples at a sampling rate of 50 Hz. Samples from both the smartphone and the smartwatch are dissected into individual keystrokes, based on peaks in their linear acceleration on each axis individually. Then the sample with the highest magnitude is mapped as the fourth time sample in each keystroke segment. After a script removes erroneous training samples, 30 samples per key remained, which were divided up into 20 samples for training, 10 for testing. As features for classification, the researchers choose to build a 54-dimensional feature vector, consisting of the 18 samples' accelerometer magnitudes over 3 axes. So to say, they used all data they could get. The keystroke inference problem is modeled as a multi-class classification problem and three different classification algorithms are used: Simple linear regression (SLR), random forest (RF) and k-nearest neighbor (kNN). SLR training is optimized using LogitBoost, in kNN was $k = 1$ and linear search algorithm with distance and weight inversely proportional to each other. In RF, the depth of the tree is left unrestricted, number of random trees is set to 100, and number of features in each tree is chosen to be 6. Not only multiple classification algorithm were tested, but also three types of training datasets: "One vs. One", where the classifier is trained on the subjects training data (100 vs. 200), "One vs. Rest", where the classifier is trained on the other subjects training data (100

vs. 2220) and "All vs. All", where all subjects training data is used to train the classifier (1200 vs. 2400). In evaluation of the attack using only smartwatch data, SLR has high "One vs. One" classification accuracy of more that 90%, but less accurate in "One vs. Rest" with 70%. RF in "One vs. One" and "One vs. Rest" was around 70%, but higher in "All vs. All" having more than 80%. "One vs. One" and "All vs. All" accuracy of k-NN is high (close to 90%) while "One vs. Rest" classification accuracy is moderate (close to 80%). Because "One vs. Rest" refers to an attack scenario, when no training data from the attackee is available, the best overall accuracy on smartwatch data is kNN close to 80%. Keep in mind, that also smartphone acceleration data was recorded before: Working on these data the same way, the best accuracy was achieved on kNN as well, having better than 70%. It can be observed, that NHHT results have much better classification accuracy, while HHT classification results are mixed.

The last one of the promising paper about keystroke sensing is "spiPhone" from 2011. In this paper, scientists of MIT face an attack vector, which nobody used before and after: Their approach uses iPhones placed nearby a computer keyboard to sense keystrokes with the iPhone's accelerometer. Normally, accelerometers in smartphones have a quite low sampling rate, near at the Nyquist rate, what makes many traditional classifying techniques of unworkable. Their approach instead decodes keystrokes by measuring the relative physical position and distance between each vibration, processed by neural network. Then they match the abstracted words against candidate dictionaries. A system overview as introduction for the follwing explanations can be seen in Fig. 11. For their work, researchers used an iPhone 4, first time equipped with an 100 Hz acceleration sensor, what introduces the possibility of this attack for the first time. Lower sampling rates deliver results which aren't processable meaningfully, and anyway, 100 Hz is far lower than other papers (like Asonov [1] in the beginning) starting with 44 kHz. The interesting way how the scientists tackle the low sampling rate is defining a keypress event model stating the relation between two keystrokes. This model contains a keys horizontal location relative to a "central-line" dividing the keyboard into left (L, along t,g,b) and right (R) partitions. Also a certain distance α between two consecutive keypresses is modeled, what can be either near (N) or far (F). So two consecutive keystrokes is modeled as $\mathrm{loc}(\mathrm{P}_i)\|\mathrm{loc}(\mathrm{P}_j)\|\mathrm{dist}(\mathrm{P}_i,\mathrm{P}_j)$. Therefore, the word "canoe" is expressed as LLN.LRF.RRF.RLF. These abstract words can be processed for all words in a English dictionary. For training phase, all letters in the English alphabet are typed 150 times each. From this data, keystrokes are segmented in 100ms long parts and features are extracted from time-domain, frequency domain and cepstral features. The final feature-vector corresponding to x, y, z accelerations of a keypress is denoted as <mean, kurtosis, variance, min, max, energy, rms, mfccs, ffts>, so in the end each letter has a set of 150 feature-vectors. Each word in the training dictionary is broken down into its characters and character-pairs. For each character, randomly 100 feature vectors from the letters set is taken and labeled as left ("L") or right ("R") according to its position. The same is done for character-pairs , where 100 random vectors from the first, and 100 random

vectors from the second character are concatenated and labeled as "near" (N) or "far" (F). Then two neural networks are trained, one left-right neural network and near-far neural network, using 500 training cycles and learning rate of 0.3. The attack is done pretty straight forward. The features are extracted as before, and fed into the neural network to retrieve L/R resp. N/F label. Having the proposed label per key and key-pair, a word matcher assigns a score against each word with the same length N in the dictionary. What can be seen in the data is that differences between keypresses on the left and right end of the keyboard are significant (big difference in amplitude), while the difference between two adjacent keypresses is minimal (same amplitude), the same can be seen on two distant keypresses having big distance (big difference in amplitude). Since close proximity is hard to distinguish, the approach to identify a region rather than a specific key provides a more feasible means of differentiating between keypresses. Testing their neural networks, their L/R accuracy was 84% and N/F accuracy 65%. To calibrate the (spi)Phone system against previous work in this space, they referred to a test setting, which also was referred in [17], which used the "corn-cob" dictionary with approx 58000 words and recovered 27 test words ranging from 7-13 characters. In the researchers test, they slightly used different parameters (57500 words dictionary, 30 words, 4-9 characters). The result was 43% top10 accuracy and 56% top50 accuracy. For the teams defense, as mentioned already, is the sampling rate of acceleration generally two magnitudes smaller than in acoustic emanations, and their test used words smaller in length, where the dictionary still has many duplicate entries. Other challenges in this approach are ambient vibrations potentially garble information, surface characteristics influences capability to detect vibrations.

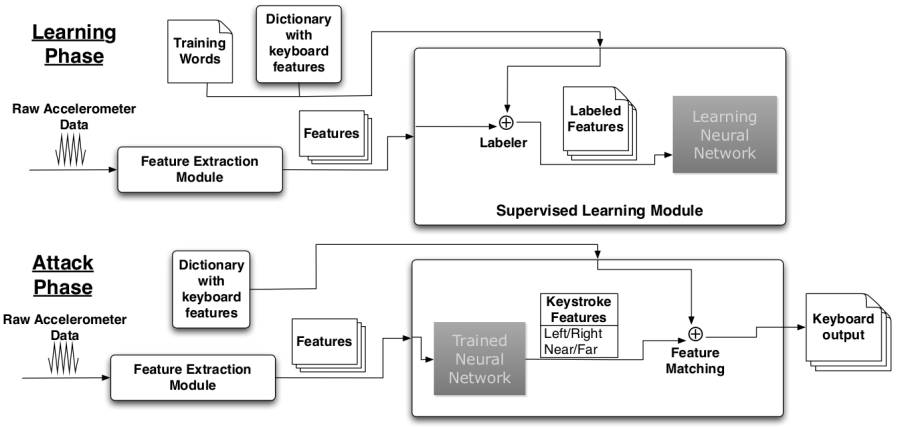


Fig. 11. System architecture of (spi)Phone

5.1 Summary of motion based methods

Summarizing the previous section, we saw several ways, how to exploit the gained information from motion sensors. Many papers emphasized the problem, that not privileged access of motion data on smartphones could be used to infer text-input from the user on the smartphone itself and developed an attack to proof it. But also, motion data from smartwatches can be used to infer keyboard, smartphone or numpad input, which was a second major topic of interest. Last but not least, one paper tried to use smartphone acceleration data to sense keystrokes made on a nearby keyboard. No paper could be found dealing with motion data as a alternative text-input method, which is related to the fact that todays motion sensor's sampling rate in smart devices is not sufficient for sensing input accurately in high resolution: Devices' motion sensor sampling rates vary from 30 - 100 Hz, but still 100 Hz is too less to catch high frequencies (e.g. emanated from keyboards) which carries important information. Another problem what make good results as alternative text-input difficult is the high dependency on how a user types. Only with lots of training data from the specific person on a specific device allows satisfying results for that specific person, since data from many users induce high variance, what generalizes bad. Formal typing rules, like which key has to be typed with which finger, could narrow down search space thus make classification results more reliable, but it is hard to formulate this as a precondition, because a user not always follows the rules. This way, strong assumptions, which would make sensing more reliable are not makable and thus, *attacks on smartphone input* stay on level around 70% accuracy for inferring a single entered digit on a touchscreen. On the other hand, for attacks, the sampling rate is sufficient to extract enough information, which reduces search space for brute-force attacks tremendously. For example [14] could find 6 character long [0-z] passwords in 2^{15} trials instead of 2^{28}. Mostly the extracted features on motion data were statistical data about the keystroke pattern to calculate probabilities in Gaussians or kNN classifiers. *Attacking text-input by leaked smartwatch motions* tries to interpret acceleration in the hand wrist to input on a keypad. Because the hand has more degrees of freedom, wrist's motion data carries only limited information about finger-tip movement, which can not be used with a strong confidence of which key was pressed. Also information of the right hand's typing can't be sensed directly. To reduce this downsides, most papers use a word model to increase accuracy by dictionary look-up. Unfortunately every paper dealing with word model couldn't take a language model as well, since they can not reliably track the space bar. By this limiting factor, an attack by smart-watch sensing can't process free-flowing text-input and can deliver 63% accuracy in the top10 guesses *per word*! Some papers use an intermediate dictionary to translate relative movements on the keyboard into keys. This intermediate dictionary works great on longer workds, but has lots of similar taggings on short words below 4 characters. These words have been left out in tests to raise accuracy, since "they don't have much information", what is bothersome in some way.

Summing up, motion data can be used as attack vector, but if the attack would work on free-flowing text some day, an attacker would not receive an instant transcript of the typed text. Instead he can quickly overlook the word listings and guess the topic and can put words manually together to a reconstructed text. Therefore, smart devices are easily exploitable, always close to the input device and no additional infrastructure is needed.

Paper	Year	Features	Training	Data Processing	Acc	Chars
SMARTPHONE-BASED ATTACKS						
[13]	2011	Gyroscope	labeled	Gaussian distribution	71,5%	[0-9]
[14]	2012	Accelerometer	labeled	Statistical, conditional prob.	?	$[0\text{-z}]^6$
[15]	2015	Accel. & Magneto	labeled	Statistical, SVM	>70%	[0-9]
[18]	2015	Acceleration	supervised	kNN/(random forest/lin. regression)	>70%	[0-9]
SMARTWATCH-BASED ATTACKS						
[16]	2015	Accel. & Gyroscope	labeled	Conditional prob, word model	$50\%_{top24}$	$[0\text{-z}]^w$
[17]	2015	Accel. & Acoustic	unlabeled	Relative paths, word model	$63\%_{top10}$	$[a\text{-z}]^w$
[18]	2015	Acceleration	supervised	kNN/(random forest/lin. regression)	80%	[0-9]
SMARTPHONE-BASED ON KEYBOARD						
[19]	2011	Acceleration	supervised	neural network, word model	$43\%_{top10}$	$[a\text{-z}]^w$

Table 4. Comparison of acceleration based sensing methods

6 Conclusion

Recapitulating what we have seen on the last pages were opportunities to alternatively sense keystrokes. Different from mechanical keyboard, different from touchscreens. Four categories were introduced. 1) acoustic sensing; 2) video sensing; 3) wifi sensing; 4) acceleration sensing. Keystroke sensing can be used to find new text-input methods, as well as used as an attack against the user. Most of the papers introduced new *side-attacks against keyboards, smartphone and numberpads*, using smart devices. Only three paper [5,7,8] proposed a *virtual keyboard approach*, the first uses acoustic pattern released by fingertips and margins on hard surfaces, the remaining two uses video cameras reaching promising 95%, 88,6% and 90% accuracy. Besides the second paper, which uses a projected keyboard as reference, all virtual keyboards need to be initialized with user or even situation dependent training data. The third paper uses a wrist-worn camera as extra hardware, which tracks the fingers. But when it comes to "involuntary" *data sensing (attack)*, methods needs to be more sophisticated. Having the opportunity to gain access to *labeled data* for training, WiFi sensing offers a method resulting in 93,47% accuracy by exploiting modern WiFi's Channel State Information. Although fascinating, the approach lacks on robustness, since many data need to be produced and updated when the environment changes. The actual goal of an key-logging attack is, of course, to design a system, which does not need

any labeled data (*unlabeled attack*) from the victim to minimize the adaption efforts and generalize the attack as much as possible. There, acoustic attacks with offline-procession exist, offering an accuracy of 95,7% using HMM and language model [2], as also 95% of accuracy using Time Difference of Arrival (TDoA) and Mel Frequency Cepstrum Coefficients (MFCC) [6] only needing a smartphone nearby and no need of a dictionary. No need of dictionaries is always "better", since dictionary-based approaches only can recognize words, which are included in the language model. Especially foreign language words and passwords can't be inferred successfully. Other paper in the section of unlabeled data attacks far below the previously mentioned accuracies. The approach [17] using smartwatch acceleration in combination with acoustic signal has the best results under all motion based approaches having 63% accuracy, finding a typed word in top10 suggestions. But this requires as well a word dictionary. The best dictionary-less motion approach using relative motions between keystrokes in a kNN-classifier recognizes only numbers with an accuracy of 80%. As the reader might have already recognized throughout the report, a big problem is the comparability of results between paper of different scientists, also because every paper looks at the topic with a different angle or different preconditions, making it hard to compare. Only three times, a paper tried to adapt their testing in regard to their predecessor. Generally, *acoustic approaches* need either massive training data to extract the key's specific sound, or three microphones try to triangulate the signal. A combination of both techniques can work with fewer microphones, but need to be placed on known positions. *Video data* has the challenge of hand segmentation, where often complex hand models are necessary to solve the problem of hidden fingers. Also, keypresses are difficult to identify, when no sound is taken into account. The video results are very type-style dependent. *WiFi-based sensing* senses very sensitively, but senses also lots of noise, which has different variability. Once an item in proximity of the typist is changed, all features of the tiny finger movements are expired. *Acceleration-based* techniques sense close to the typist, but have the lowest sampling rate of all, what is problematic having features close to the Nyquist-frequency.

Finally, it can be said, referring to the introduction question of finding alternative sensing methods: There are very good sensing approaches already existing for users as virtual keyboard with and without online-feedback, as well as for attackers. The attacks with the highest results based on offline-calculation, what is the reason why no labeled data is needed. Because of the same reason, those high accurate approaches aren't applicable as virtual keyboard. Further accuracy improvements are very hard reachable, because many combinations are already researched (motion and acoustic) and only minor changes in keystroke segmentation or classification seems to improve results.

References

1. Dmitri Asonov and Rakesh Agrawal. Keyboard acoustic emanations. *Proc. - IEEE Symp. Secur. Priv.*, 2004:3–11, 2004.

2. Li Zhuang, Feng Zhou, and J. D. Tygar. Keyboard acoustic emanations revisited. *Proc. 12th ACM Conf. Comput. Commun. Secur. - CCS '05*, (November):373, 2005.
3. A.H.Y. Fiona. Keyboard Acoustic Triangulation Attack. pages 1–40, 2006.
4. Tong Zhu, Qiang Ma, Shanfeng Zhang, and Yunhao Liu. Context-free Attacks Using Keyboard Acoustic Emanations. *Proc. 2014 ACM SIGSAC Conf. Comput. Commun. Secur. - CCS '14*, pages 453–464, 2014.
5. Junjue Wang, Kaichen Zhao, Xinyu Zhang, and Chunyi Peng. Ubiquitous keyboard for small mobile devices. *Proc. 12th Annu. Int. Conf. Mob. Syst. Appl. Serv. - MobiSys '14*, pages 14–27, 2014.
6. Jian Liu, Yan Wang, Gorkem Kar, Yingying Chen, Jie Yang, and Marco Gruteser. Snooping Keystrokes with mm-level Audio Ranging on a Single Phone. *Proc. 21st Annu. Int. Conf. Mob. Comput. Netw.*, pages 142–154, 2015.
7. Huan Du, Thierry Oggier, Felix Lustenberger, and Edoardo Charbon. A Virtual Keyboard Based on True-3D Optical Ranging. *Proc. Br. Mach. Vis. Conf.*, 1:220 – 229, 2005.
8. Farooq Ahmad and Petr Musilek. A keystroke and pointer control input interface for wearable computers. *Proc. - Fourth Annu. IEEE Int. Conf. Pervasive Comput. Commun. PerCom 2006*, 2006:2–11, 2006.
9. Davide Balzarotti, Marco Cova, and Giovanni Vigna. ClearShot: Eavesdropping on keyboard input from video. *Proc. - IEEE Symp. Secur. Priv.*, pages 170–183, 2008.
10. Qifan Pu, Sidhant Gupta, Shyamnath Gollakota, and Shwetak Patel. Whole-home gesture recognition using wireless signals. *Acm Mobicom*, pages 485–486, 2013.
11. Rajalakshmi Nandakumar, Bryce Kellogg, and Shyamnath Gollakota. Wi-Fi Gesture Recognition on Existing Devices.
12. Kamran Ali, Alex X Liu, Wei Wang, and Muhammad Shahzad. Keystroke Recognition Using WiFi Signals. *Proc. ACM MobiCom*, pages 90–102, 2015.
13. Liang Cai and Hao Chen. TouchLogger: inferring keystrokes on touch screen from smartphone motion. *Proc. 6th USENIX Conf. Hot Top. Secur.*, pages 1–6, 2011.
14. Emmanuel Owusu, Jun Han, Sauvik Das, Adrian Perrig, and Joy Zhang. ACCessory: Password Inference Using Accelerometers on Smartphones. *Proc. Twelfth Work. Mob. Comput. Syst. Appl.*, pages 9:1–9:6, 2012.
15. Chao Shen, Shichao Pei, Tianwen Yu, and Xiaohong Guan. On motion sensors as source for user input inference in smartphones. *2015 IEEE Int. Conf. Identity, Secur. Behav. Anal. ISBA 2015*, 2015.
16. He Wang, Ted Tsung-Te Lai, and Romit Roy Choudhury. MoLe: Motion Leaks through Smartwatch Sensors. *Proc. 21st Annu. Int. Conf. Mob. Comput. Netw. - MobiCom '15*, pages 155–166, 2015.
17. Xiangyu Liu, Zhe Zhou, Wenrui Diao, Zhou Li, and Kehuan Zhang. When Good Becomes Evil: Keystroke Inference with Smartwatch. *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur. - CCS '15*, pages 1273–1285, 2015.
18. Anindya Maiti, Murtuza Jadliwala, Jibo He, and Igor Bilogrevic. (Smart) Watch Your Taps : Side-Channel Keystroke Inference Attacks using Smartwatches. *ISWC '15 - Proc. 2015 ACM Int. Symp. Wearable Comput.*, pages 27–30, 2015.
19. Philip Marquardt, Arunabh Verma, Henry Carter, and Patrick Traynor. (sp) iPhone: decoding vibrations from nearby keyboards using mobile phone accelerometers. *Proc. 18th ...*, pages 551–562, 2011.

Business Models for the Internet of Things and the Cloud in an Industrial Environment

Aleksandar Kostov*

Advisor: Andrei Miclaus†

Karlsruhe Institute of Technology (KIT)
Pervasive Computing Systems – TECO
*ueduw@student.kit.edu
†miclaus@teco.edu

Abstract. Technological paradigms have always been shifting. Nowadays ever so much, and even more, faster. Increasing computational power, reduced size and enhanced connectivity capabilities of the new devices has ushered in a new technological era. Through the ever increasing integration of the cloud into the industrial landscape and the vast amounts of data generated through connected devices and their respective sensors, we see the rise of a new currency - information, that is subjectively of value no less than that of money. In this paper we explore how the incorporation of the cloud and the internet of things into the industrial sector has affected business models. We take a look into how existing business models have changed, what new models have arisen and how the access to vast amounts of monitoring data is disrupting value chains and reshaping the industrial landscapes.

Keywords: business models; the cloud; internet of things; industrial environment

1 Introduction

The industry is ever changing, ever evolving. New inventions and breakthroughs are revolutionizing different industrial sector on a regular basis. The recent industrial growth and progress is an unprecedented one, though. It is driven by the rapidly evolving technologies. Decrease in size, increase in sensory and computational power, lowered costs and increasing availability are all key factors to the rapid technological and industrial progress. We're almost caught in a negative feedback loop – the better our technologies become, the faster they help us progress, and the faster we upgrade our technologies. Fig.1 depicts the rapid shortening of the periods between each new industrial revolution and the resulting jumps in technological advancement. The result is a rapid shift in technological and subsequently industrial and business paradigms. New opportunities arise all the time, giving birth, or at least the possibility for it, to new business ventures, or discovering corners of the industrial process that have gone unnoticed or ignored due to lack of sufficiently technological advancements.

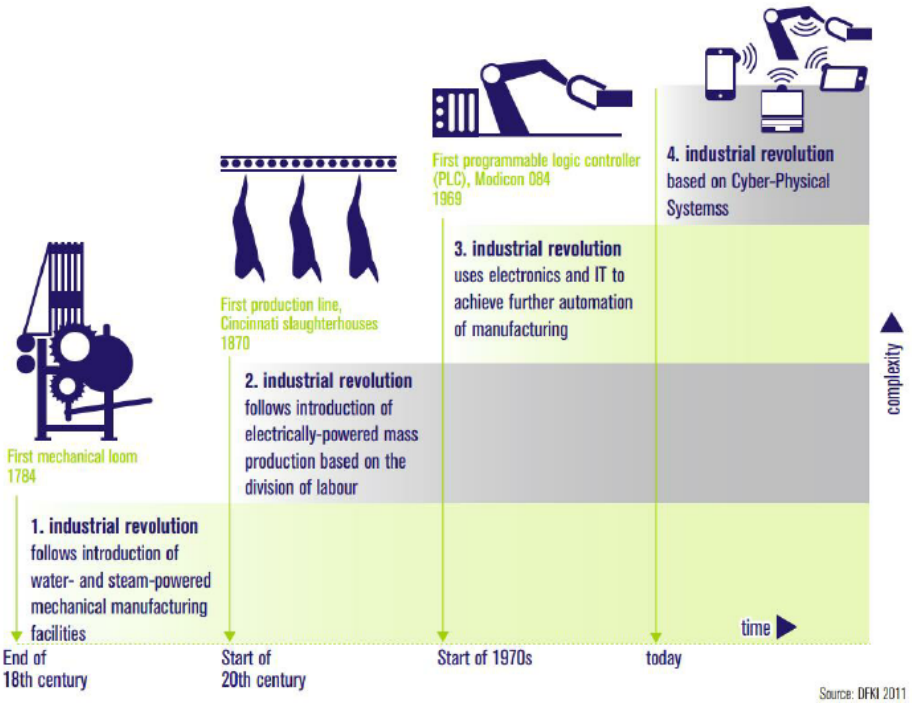


Fig. 1. The four big industrial revolutions.

The fourth industrial revolution is going to affect every sector of the industry – from health care and medicine, to the transport industry, to personal electronics, to energy production and distribution, to the service sector. And the thing that drives this industrial revolution is data. In this paper, we will go over how the ability to gather, analyse and utilize mass quantities of monitoring and feedback data has influenced the industry, and subsequently the businesses associated with them. We will explore the changes to business models ushered by the new insights provided by the analysis of the collected data, and how it steers the direction in which the industry is headed.

2 The Cloud and Internet of Things in the Industry

Today, the internet is being used as an information medium not only between people, but also between machines (also known as M2M connectivity). This has allowed machines to generate usage and monitoring data without the need for human intervention or participation, increasing the information generation and flow exponentially. As J. E. Porter has discovered in his article [13], the newly found intelligence and connectivity of the devices, brought forth by embedding

functionality through sensors, integrated microprocessors and operating systems, give them capabilities that can be grouped in four areas, as depicted in Fig. 2.

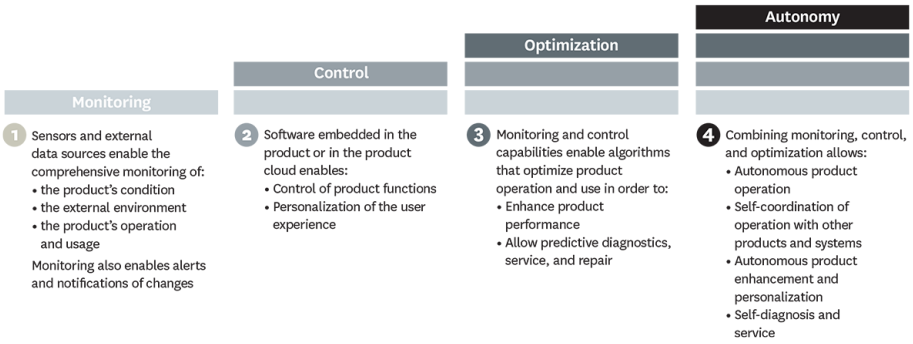


Fig. 2. Capabilities of smart, connected products, grouped into four areas. [13]

We can clearly see that the capabilities in all four areas work towards a common goal – generating, collecting, analyzing and utilizing data. For data is the resource that makes possible the optimization, evolution and fine-tuning of every segment of the industrial value chain – production, logistics, marketing, research and development, administration, servicing.

Although the internet of things plays a key role in the new industrial revolution, it's not the sole initiator. Without advancements in cloud computing, the mass collection and processing of data would be impossible. The cloud provides not only a way of storing the gathered data, but also a bridge between systems, and a medium for deployment of tools and utilities that take advantage of the stored data. And while smaller and more immediate tasks (like for example deploying safety measures in case of a fire in a factory) are performed on site with the help of the on-board logic of smart devices, the majority of the processing of data is performed off site, in the cloud, essentially turning it into the brains of the industry.

Physical products and smart devices are not the only things that generate useful data. Softwares and services, independent or supporting, also generate vast quantities of vital data for companies. That is why, as Oxford Economics have discovered in their study of cloud computing adoption [cite], the majority of the business and IT executives of the companies that participated in the survey have embraced the idea of cloud computing in their respective branches of the industry, and are expecting it to affect every sector positively, some more than others. Fig. 3 gives a quick overview of their expectations, or at least part of them. We can indeed see that the majority of the expectations are overwhelmingly positive.

And even so, even with the advantages the IoT and the cloud bring to the industry, some experts are concerned that they might remain underutilized by

the big companies. In his article about IoT business model innovation [19], Alex Scroxton describes how, even though the IoT offers clear advantages, it gets largely underused, especially for the development of new business models. As restructuring and reorganization are expensive and time consuming processes, the internet of things is largely used for optimization of performance, cost and expenses reduction and predictive maintenance. *This is not to criticise, but it raises a challenge in how the IoT is sold. As a result, movement to innovative business models is slow.*

However, there are more factors concerning the incorporation and utilization of the cloud and IoT than just costs and time. A general concern is security and privacy on multiple levels. Privacy and security breaches affect not only the companies, but the citizens as well. With an ever increasing quantity of data, both monitoring, operative and personal circulating the internet, it is becoming imperative that data, sensitive information and intellectual property get the proper security so they do not prove the downfall of the new technological age. Interestingly enough, as the majority of cloud service providers are big companies such as Google, Microsoft, Amazon and SAP, companies aren't generally concerned with the safety of the data. Priorities have shifted towards aspects such as API and interface security, IP security and cibercrimes (see [12]).

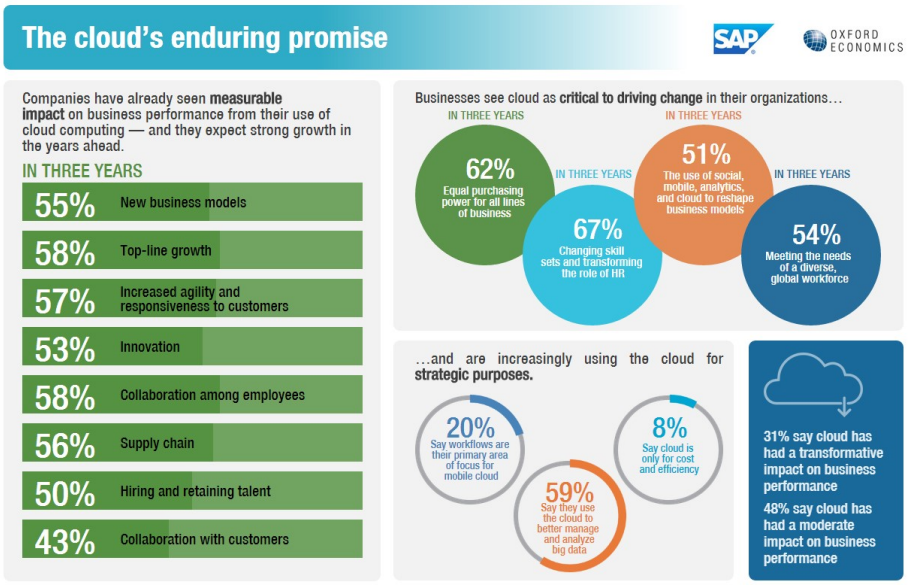


Fig. 3. The cloud's enduring promise. Source: http://www.oxfordeconomics.com/SAPcloudperformance

3 Business Models

The term business model is broad in the sense we want to explore it. It generally describes the organisation and inner workings of a company. A more apropriate term for our purposes would be revenue model, but let's stick to the original term.

To be able to better understand the business models in the industry, T. Kaufmann [8] describes their four dimentions, and namely:

- Who are our clients?
- What do we offer our clients?
- How do we produce and deliver?
- How do we profit?

These four key points are the core elements of every business model pattern. The clients are the ones who we will be trading with, and hopefully profiting from, so they make the basis for every business model, and as such need to be clearly identified and defined. The products and services that we will offer comprise our value proposition. How we produce and deliver is essentially the linchpin of the company, since the market is saturated with products and services of every kind. As such, it is essential for a company to stand out, if not with a unique product, then with an interesting and appealing ecosystems around existing products, making them more desirable. And finally, the way we actually generate profit from the goods and services we offer. While this is mostly fine tweaking of prices, picking the best and most profitable partnerships and collaborations, we can essentially point out which aspect of our product or service is most benifitial, and how it will bring increases in profits.

And because, as we've said before, the new wave of business model innovation is spurred on by the availability of data, and because their success lies more or less on the proper utilization of the data, we can safely define the new business models as datacentric.

Since a business model is a pretty broad definition, everything can warrant a new business model. Even something as simple as turning a normal product into a smart product can give old business models new flavor by improving performance, reducing cost and allowing predictive maintenance based on past and present data. A basic example would be outfitting a washing machine with sensors that allow for optimization of water and electricity usage.

But smart devices by themselves do not yet make the category of new or innovative business models, though. A step up from it would be the actual networking of smart devices, say through a centralized local hub. A good example would be more advanced home automation. Having a device by itself be smart is just the first step. Creating a system of smart devices offers a wider range of possibilities, such as more advanced monitoring, analytics and metrics, which pave the way for software solutions and services.

As the data is allowing companies to more effectively utilize their available resources, especially machinery, it is getting increasingly popular to buy instead of rent equipment. It may not seam like much, but this strategy has profound implications. Firstly, the responsibility for the maintenance of the equipment

shifts from the customer to the company itself. That way, customers are more inclined to use a certain product, as the costs of maintenance and the possibility of failure of equipment and the resulting repair costs can be a deterrent. With increased capabilities for analytics, metrics and data collection, companies can now more reliably predict the life cycles, eventual failure possibilities and therefore reduce the risk they take by leasing rather than selling equipment. This also drives costs down, and increases customer satisfaction.

Leasing instead of buying can have some effects of questionable benifit as well. Leasing or renting means that the usage cycles for each piece of equipment becomes shorter, and customers do not get locked in so easily (most of the times). This leads from one side to customers being more inclined to try out new products, as they do not have to make such a long term investment as with buying the equipment itself. Companies have to be careful, though, as shorter usage cycles means more opportunities for customers to switch their preferred supplier. Simply put, with every set of advantages come also a number of disadvantages that have to be considered before deciding on a business model.

As data can be gathered from every connected device and every customer, and the data can then in turn be used to optimize or innovate any part of the value chain of a given company, it is safe to assume that the possibilities for business models are enormous. From models like Add-On (the automobile industry, cheap flight companies), Freemium (Skype, Spotify, Dropbox), Lock-In, Razor and Blade (Gillette) to Hidden-Revenue (Google AdSense), Leveraging Customer Data (Amazon, Google, Facebook, Twitter) and Layer Player (PayPal), every company finds it's place in the industrial scene and selects a proper business model. That is why we will focus ourselves on only a couple of cases, and how they have been affected by the technological revolution.

3.1 Platform-as-a-Service (PaaS)

The PaaS business model has been adopted by a number of large companies, such as Google, Amazon and SAP. PaaS companies provide their clients with the means to develop, run and manage their software without having to build and maintain a complex infrastructure.

This has some distinct advantages, such as allowing the developers to focus on the applications, solutions and the data, while the PaaS provider takes care of the infrastructure (runtime, middleware, operating systems, virtualization, servers, storage and networking). PaaS offerings may also include facilities for application design, application development, testing and deployment, as well as services such as team collaboration, web service integration, and marshalling, database integration, security, scalability, storage, persistence, state management, application versioning, application instrumentation, and developer community facilitation. Besides the service engineering aspects, PaaS offerings include mechanisms for service management, such as monitoring, workflow management, discovery and reservation.

In cases like SAP, they have taken it one step further, and offer not only the platform itself as a service, but also the means to distribute your applications

via the company App Stores. We can distinguish two types of App Stores in companies - internal and external.

A comparison of the two App Stores can be seen on the Table 1 [23]

Table 1. Comparison of key capabilities of the public versus internal EAS model with focus on business involvement and IT control along the IT governance process

IT Gov. Process	SAP Store (public EAS)	SAP Enterprise Store (internal EAS)
IT Sourcing	▪ Business can identify, gather information about, and try new business applications ▪ IT defines buyers and proactively invites business reps to participate during external sourcing ▪ IT can enable selected business reps to make purchases	▪ Early involvement of business users in in-house development projects (internal sourcing)
IT Delivery	▪ Instant delivery of software can accelerate delivery process	▪ Business users select and consume apps in a self-service mode using a consumer-friendly app catalog ▪ Provide apps to ecosystem ▪ Secure and instant delivery to user devices
IT Support		▪ Internal EAS can be used to distribute updates of applications ▪ Distribution of e-learnings
Monitoring	▪ IT can monitor all purchases on the EAS via a central order view	▪ Monitoring of app usage, downloads, reviews, ratings ▪ License monitoring
IT Control	▪ Define buyer roles ▪ Prevent business users from buying non-authorized app-lications	▪ Define target groups for applications (who can access which apps) ▪ Fully define catalog content and visual styling of EAS

3.2 Pay-Per-Use

A widely used business model is the so-called Pay-Per-Use business model. The Pay-Per-Use business model is suitable for companies that expect non-linear business growth or unpredictable spikes/dips in product usage. And because the usage tends to average out over time (spikes and dips cancel each other out), with proper data aggregation and analysis, the user base can be segmented, and users with simmilar use interests can be banded together and receive pricing offers accordingly. [21]

The Pay-Per-Use business model can also be appealing to consumers, not only to companies, as, depending on the service provided and the user interests, Pay-Per-Use can provide the flexibility other business models might lack. The lack of contractual obligation is an example of such flexibility, because, as the name states, you only pay for what you use.

An example of a good Pay-Per-Use implementation would be Amazon's EC2 service. As the majority of applications and services require servers, Amazon has found a way to offer an attractive perspective for developers. Like we mentioned above, equipment is being rented or leased instead of bought more and more widely, and that is what Amazon are doing with their EC2 service. Instead of buying, maintaining and managing servers themselves, clients can rent servers from Amazon. The renting itself is flexible, as developers can rent as much server units as they need, with options for increasing/decreasing of server numbers depending on current demand for computational power. This means, Amazon offers the opportunity to use their already deployed equipment and infrastructure, as much as you need. Consumers pay according to the hours of usage and number of server instances, without any contractual obligations.

But while Amazon is a huge company with the means of deploying and maintaining infrastructure with attractive offers, smaller companies might not have the same success initially. Smaller or starting service providers will have to distribute their costs over a smaller base, which means higher costs and lower initial discounts.

3.3 Subscription

The Subscription based business model is viable for businesses with linear and relatively predictable growth, and a predictable consumer base behavior. Implementing a subscription based business model requires a fair amount of fine tuning, as an unbalanced subscription model can quickly lead a company to failure. Things such as proper pricing and a hardware buffer to handle an unexpected overflow of demand have to be taken into consideration.

According to KineticD[21] there are three different type of subscription based revenue models: per-user; per-device and per-enterprise. These three basically regulate what access a certain user or user group has to the software or solution based on the license they have received. A per-enterprice license can allow for example anywhere from 10 to 100 or even more users access to the software or utility if they are company members (Antivirus software companies), while a per-user license would allow only a single user or a single account with a smaller number of concurrent users (a good example would be Netflix and their subscription model).

4 Conclusion

In this paper, we have given an overview of how the interconnectivity of the new smart devices and the opportunities the cloud gives for interoperability, cooperation and collaboration, as well as mass monitoring and operational data gathering, analytics and processing is changing not only the industrial, but also the business landscape. As the emerging technologies and the advancements in existing ones allow us to better utilize the vast amounts of data devices produce every day, it is up to the companies to find business models that suit them best.

For many of them, this would not be just one business model. It might be a combination of existing models, or it might be an entirely new model they have developed through innovative ideas. Whatever the case, technology is moving forward in an incredible pase, and business models have to keep up in order to make best use of the emerging opportunities.

References

1. Ballon, P., Walravens, N., Spedalieri, A., Venezia, C.: An Advertisement-based Platform Business Model for Mobile Operators. Constellations (NOVEMBER) (2008)
2. Chen, N., Lin, J., Hoi, S., Xiao, X., Zhang, B.: AR-miner: mining informative reviews for developers from mobile app marketplace. Icse (2014), 767–778 (2014), http://www.cais.ntu.edu.sg/{~}nchen1/AR-Miner/icse14-preprint.pdf
3. Cramer, H., Rost, M., Belloni, N., Bentley, F., Chincholle, D.: Research in the large. using app stores, markets, and other wide distribution channels in Ubicomp research. Proceedings of the 12th ACM international conference adjunct papers on Ubiquitous computing - Ubicomp '10 p. 511 (2010), http://www.scopus.com/inward/record.url?eid=2-s2.0-78650022798{\&}partnerID=tZOtx3y1
4. Emmrich, V., Döbele, M., Bauernhansl, T., Paulus-Rohmer, D., Schatz, A., Weskamp, M.: Geschäftsmodell-Innovation durch Industrie 4.0. Dr.Wieselhuber & Partner GmbH und Fraunhofer IPA p. 56 (2015)
5. Fleisch, E., Weinberger, M., Wortmann, F.: Business Models and the Internet of Things. Bosch IoT Lab White Paper (SEPTEMBER), 19 (2014), http://cocoa.ethz.ch/downloads/2014/10/2090{_}EN{_}Bosch$\backslash$nLab$\backslash$nWhite$\backslash$nPaper$\backslash$nGM$\backslash$nim$\backslash$nIOT$\backslash$n1{_}2.pdf
6. Helmrich, K., Siemens: On the Way to Industrie 4.0 – The Digital Enterprise. Industry4.0 (2014)
7. Hillis, D.: The new world order for open-source and commercial software (2016), https://techcrunch.com/2016/06/13/the-new-world-order-for-open-source-and-commercial-software/?ncid=rss&utm_source=feedburner&utm_medium=feed&utm_campaign=Feed%3A+Techcrunch+%28TechCrunch%29&sr_share=twitter
8. Kaufmann, T.: Geschäftsmodelle in Industrie 4.0 und dem Internet der Dinge, vol. 53 (2013)
9. Kim, J., Park, Y., Kim, C., Lee, H.: Mobile application service networks: Apple's App Store. Service Business 8(1), 1–27 (2013), http://link.springer.com/10.1007/s11628-013-0184-z
10. Kimbler, K.: App store strategies for service providers. 2010 14th Int. Conference on Intelligence in Next Generation Networks: "Weaving Applications Into the Network Fabric", ICIN 2010 - 2nd Int. Workshop on Business Models for Mobile Platforms, BMMP 10 pp. 1–5 (2010)
11. Lindgardt, Z., Reeves, M., Stalk, G., Deimler, M.S.: Business Model Innovation: When the Game Gets Tough, Change the Game. Boston Consulting Group (December), 9 (2009)
12. Louis Columbus, Oxford Economics, S.: 55 percent of Enterprises Predict Cloud Computing Will Enable New Business Models In Three Years (2015), http://www.forbes.com/sites/louiscolumbus/2015/06/08/

55-of-enterprises-predict-cloud-computing-will-enable-new-business-models-in-three-years/#ceaba01c0b27
13. Michael E. Porter, J.E.H.: How Smart, Connected Products Are Transforming Competition (2014), https://hbr.org/2014/11/how-smart-connected-products-are-transforming-competition#
14. M&M, S.: Industrial internet of things und cloud – strategien, lösungen und praxis 07. (April) (2016)
15. Mobile, V.: The European App Economy. Vision Mobile (2014), www.visionmobile.com/product/the-european-app-economy/
16. Müller, R.M., Kijl, B., Martens, J.K.J.: A comparison of inter-organizational business models of mobile app stores: There is more than open vs. closed. Journal of Theoretical and Applied Electronic Commerce Research 6(2), 63–76 (2011)
17. Oxford Economics: The new engine of business 1(1), 16 (2015)
18. Pagano, D., Maalej, W.: User feedback in the appstore: An empirical study. 2013 21st IEEE International Requirements Engineering Conference, RE 2013 - Proceedings pp. 125–134 (2013)
19. Scroxton, A.: Internet of things needs more innovative business models (2016), http://www.computerweekly.com/news/4500278603/Internet-of-things-needs-more-innovative-business-models
20. Smith, M.: Milions of sensitive services exposed on the internet reveal most hackable countries (2016), http://www.networkworld.com/article/3080003/security/millions-of-sensitive-services-exposed-on-internet-reveal-most-hackable-countries.html
21. Summary, E.: Sustainability is the Key : Cloud Computing Business Models
22. Wee, D., Kelly, R., Cattel, J., Breunig, M.: Industry 4.0 - how to navigate digitization of the manufacturing sector. McKinsey & Company pp. 1–62 (2015)
23. Wenzel, S.: App store models for enterprise software: A comparative case study of public versus internal enterprise app stores. Lecture Notes in Business Information Processing 182 LNBIP, 227–242 (2014)
24. Xia, R., Rost, M., Holmquist, L.E.: Business models in the mobile ecosystem. ICMB and GMR 2010 - 2010 9th International Conference on Mobile Business/2010 9th Global Mobility Roundtable pp. 1–8 (2010)

Methoden des Experience Sampling mit Mobilgeräten

Denis Gaus*

Betreuer: Anja Bachmann†

Karlsruher Institut für Technologie (KIT)
Pervasive Computing Systems – TECO
*uobdo@student.kit.edu
†bachmann@teco.edu

Zusammenfassung. Die rasante Verbreitung und Entwicklung der Mobilgeräte in den letzten Jahren hat den Sozialwissenschaften neue Möglichkeiten zur Datenerhebung an die Hand gegeben. Die in diesem Paper vorgestellte Mobile Experience Sampling Methode erfasst in der natürlichen Umgebung des Probanden vielerlei Daten ohne jegliches Zutun dessen, unterstützt von gelegentlichen Befragungen des Benutzers, womit eine möglichst detailgetreue und unverfälschte Beobachtung von Verhalten, Gedanken und Gefühlen des Teilnehmers erreicht werden soll, ohne auf dessen Erinnerungen angewiesen zu sein. Sie stellt eine Weiterentwicklung der klassischen Experience Sampling Methode dar, die bereits seit Jahrzehnten in den Sozialwissenschaften zum Einsatz kam, und wurde erst durch die neuen technischen Möglichkeiten von vor allen Dingen Smartphones möglich gemacht.

Schlüsselwörter: mobile sensing, ESM, mESM, BCI

1 Einleitung

Sozialwissenschaftler waren seit je her interessiert daran, das menschliche Verhalten zu beobachten und aufzuzeichnen, ohne es zu verfälschen, also Einfluss darauf zu nehmen. Die Verfügbarkeit und Allgegenwärtigkeit von Smartphones bietet Forschern dabei die Möglichkeit, dies in einer Art und Weise zu tun, wie es vorher nicht möglich war. Fortschritte in Wissenschaft und Technologie gehen sehr häufig Hand in Hand. Dabei können Smartphones mit all den in ihnen verbauten Sensoren nicht nur zur Datenaufzeichnung dienen, sondern dank ihrer erheblichen Rechenkraft auch einen großen Teil der Vorverarbeitung übernehmen.

2 Technischer Fortschritt

Es gab in den letzten Jahren eine Vielzahl an technologischen Entwicklungen, vor allem bei den Smart-Wearables, die völlig neue Möglichkeiten im Bereich

des Social Sensing geschaffen haben. Das Mobiltelefon hat unsere Kommunikationsnetzwerke dezentralisiert. Kommunikation geschieht also nicht mehr nur von einem festen Punkt zu einem anderen festen Punkt, die Punkte sind nun beweglich und jederzeit und überall erreichbar.

Smart-Devices sind mittlerweile mehr oder weniger allgegenwärtig, im Juli 2015 besaßen 46 Millionen der Bundesbürger ein Smartphone und somit die Mehrzahl der deutschen Bevölkerung [1]. Weltweit ist die Anzahl an Smartphonebesitzern ebenfalls stetig am wachsen, hier vor allem in den Entwicklungsländern, da in den industrialisierten Ländern bereits eine gewisse Sättigung erreicht wurde (Abb. 1) . Niemals zuvor hat ein technisches Gerät eine solch wichtige Rolle in unserem Leben eingenommen wie das Smartphone. Das Mobiltelefon hat dabei die Rolle des persönlichsten Objektes vieler Menschen eingenommen. Laut citesrivastava2005mobile haben viele Menschen Angst, ihr Zuhause ohne Mobiltelefon zu verlassen und fühlen sich unwohl, wenn andere Menschen ihre Nachrichten lesen oder das Telefon in sonstiger Weise verwenden.

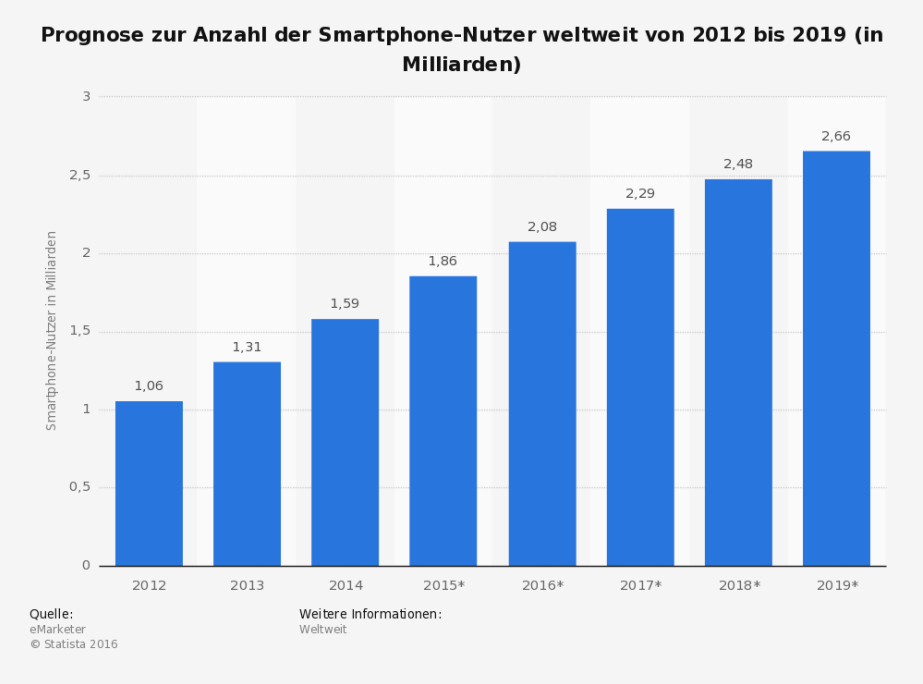


Abb. 1. Anzahl der Smartphone-Nutzer auf der Welt [6]

Smartphones eignen sich aufgrund der Vielzahl der in ihnen verbauten Sensoren, wie GPS, Beschleunigungssensor, Gyroskop, Kompass, Annäherungssensor, Lichtsensor, oder auch Bluetooth sehr gut zur Aufzeichnung von sozialem Kontext im üblichen sozialen Umfeld des Benutzers, ohne diesen von Außen zu

beeinflussen. Mit ihrer Hilfe kann festgestellt werden wo sich jemand aufhält und was derjenige gerade macht, oder auch wer sich sonst in der Umgebung befindet.

Ebenfalls gab es wesentliche Fortschritte im Bereich der Funktechnologien, so sind bereits heute im deutschen Mobilfunknetz mit LTE Advanced Cat-6 bis zu 300Mbit/s im Download und 50Mbit/s im Upload möglich. Im Gegensatz dazu waren Anfang dieses Jahrhunderts mithilfe des damals verbreiteten GPRS nur Übertragungsgeschwindigkeiten von um die 56Kbit/s möglich, bei gleichzeitig höheren Kosten für die Mobilfunktarife, insbesondere was die Datenübertragung angeht. Ein weiterer Ausbau der Geschwindigkeiten ist seitens der Provider angekündigt und in Erprobung, während auch die LTE-Abdeckung durch den fortwährend stattfindenden Ausbau der Telekommunikationsanbieter sich weiter verbessert. Somit ist, wenn nötig, auch eine Übertragung der Daten in Echtzeit oftmals machbar.
Rechenleistung und Speichermengen von Smartphones sind weiter im Begriff zu wachsen, so das die für die Aufzeichnung und Auswertung der Daten notwendige Rechenleistung üblicherweise gegeben ist und auch das Speichern dieser Daten auf den im Smartphone verbauten Speicher eine kleinere Hürde darstellt.

3 Verhaltensbeobachtung

Das Menschliche Verhalten ist üblicherweise vom Kontext abhängig in dem man sich befindet. Der Kontext wird beschrieben durch die physische Umgebung, also dem Ort an dem man sich befindet, wie etwa der Arbeitsplatz, die Schule oder das Zuhause, unserem physischen Zustand also ob wir beispielsweise laufen oder schlafen und unserem inneren Zustand. Das menschliche Verhalten ist sehr komplex und drückt sich auf vielfältige Art und Weise aus. Für Psychologen ist es wünschenswert, dieses Verhalten fortwährend aufzuzeichnen, um es verstehen und ergründen zu können. Frühe Studie bestanden daraus, Teilnehmer entweder künstlichen Situation im Labor auszusetzen, um sie dabei zu beobachten oder auch aus nachträglichen Interviews, in denen Teilnehmer über vergangene Erfahrungen befragt werden. Da das menschliche Verhalten aber stark vom Kontext abhängig ist, welches niemals komplett in einem Labor nachgestellt werden kann, sind diese Arten von Studien wenig geeignet um natürliches Verhalten nachzustellen.

Die Experience Sampling Method (ESM) ist eine sozialwissenschaftliche Methode, die aus gelegentlicher Befragung des Benutzers besteht. Das Ziel ist es, eine unmittelbare Antwort auf gestellte Fragen zu erhalten, das menschliche Verhalten also aufzuzeichnen, während es passiert. Diese Vorgehensweise vermeidet sowohl die Notwendigkeit von direktem Kontakt mit dem Untersuchenden, als auch künstlich erschaffene Laborumgebungen.

Die Daten werden im gewöhnlichen Kontext des Probanden erhoben, ohne Beeinflussung von Außen. Außerdem werden die Daten unmittelbar nach dem Geschehen abgefragt, so dass bis dahin noch wenig für die retrospektive Datenerhebung übliche Wahrnehmensverschiebung stattgefunden hat, die für die nachträgliche Datenerhebung symptomatisch ist. ESM erlaubt darüber hinaus

langfristige Untersuchungen und Analysen der Studienteilnehmer und könnte auch Aufzeichnungen von selten vorkommenden Ereignissen hervorbringen.

Anfangs wurden für das ESM programmierbare Pager in Kombination mit Papiertagebüchern verwendet. Der Pager benachrichtigt dabei den Probanden durch ein Klingeln, wenn es an der Zeit ist, einen Eintrag im Tagebuch vorzunehmen. Dabei kamen in älteren Studien unterschiedliche Formen der Datensammlung und Befragung zum Einsatz, wie beispielsweise ein Anrufen der Teilnehmer auf ihrem Mobiltelefon oder Benutzung eines PDA zur Datenaufzeichnung. All diese Studien haben die Benutzung von nicht alltäglichen und vielen auch nicht vertrauten Geräten wie beispielsweise Pagern, PDAs und Tagebüchern gemein und erfordern ein trainieren des Benutzers zur korrekten Benutzung dieser. Außerdem erlaubten diese Geräte nicht die Erkennung und Aufzeichnung von tatsächlich relevanten Momenten, da sie allesamt kontextunanhängig arbeiten. Darüber hinaus verlassen sich diese klassischen ESM-Ansätze auf die Ehrlichkeit der Benutzer, was einen großen Nachteil darstellt.

Im Laufe des letzten Jahrzehnts wurden Mobilgeräte wie Smartphones und auch immer mehr Verbreitung findende Wearables wie Smartwatches ein Teil unseres Alltags. Die feste Verankerung dieser Geräte in den Alltag vieler Menschen, ihre mittlerweile beträchtliche Rechenkraft und die Vielzahl der in ihnen verbauten Sensoren eröffnen den sozialwissenschaftlichen Studien völlig neue Möglichkeiten um persönliche Daten von einer Vielzahl an Individuen zu erhalten. Die Verfügbarkeit dieser Daten hat zur Erschaffung eines neues Forschungsgebietes beigetragen, der rechnerbasierten Sozialwissenschaften (engl. computational social science).

3.1 Smartphonebasiertes ESM

Mobilgeräte sind in der Lage, das Experience Sampling in der Verhaltenspsychologie grundlegend zu verändern. Einfachere Teilnehmerrekrutierung, Datensammlung und die wesentlich größere Anzahl an tatsächlich gewonnenen Daten, als bei traditionellen ESM Ansätzen.

Grundsätzliche Vorgehensweise: Eine smartphonebasierte mobile ESM-Applikation (mESM), die beispielsweise aus einem der Appstores bezogen werden kann, verwaltet die von den Sensoren aufgezeichneten Daten und verarbeitet diese, um interessante Momente zu entdecken, in denen Benutzerdaten aufgezeichnet werden sollten. Wird ein solcher Moment erkannt, wird der Benutzer dazu aufgefordert, einen Fragebogen auf dem Smartphone auszufüllen. Diese vom Benutzer eingegebenen Daten werden gemeinsam mit den sonstigen Sensordaten des Mobilgerätes an einen Datenserver zur weiteren Analyse übergeben.

mESM-Studien sind in einigen entscheidenden Punkten besser, als traditionelle ESM-Studien mit Pager und Tagebuch. Der wohl wichtigste Vorteil ist, das Smartphones bereits fest zum Alltag der meisten Menschen gehören, im Gegensatz zu den bei der klassischen Methode eingesetzten Hilfsmitteln. Es wird somit vermieden, das der Benutzer mit einem ihm ungewohnten Gerät umgehen muss. Somit muss auch kein zusätzliches Gerät mit sich herumgetragen werden als üblich. Gleichzeitig führt das auch zu einer Kostenreduzierung der Studie, da man

ein für gewöhnlich beim Benutzer bereits vorhandenes Smartphone verwenden kann. Laut [24] schämen sich Teilnehmer auch weniger auf bestimmte Fragen zu antworten, wenn sie ein ihnen gewohntes Gerät zur Beantwortung benutzen können.

Ein wichtiger technologischer Unterschied von Smartphones zu anderen Mobilgeräten ist die so gut wie immer vorhandene Datenverbindung, um fortwährend mit dem Untersuchenden kommunizieren zu können.

Jede ESM-Studie kann des Weiteren als unabhängige Applikation auf einer Vielzahl von Geräten über die verschiedenen Appstores der Smartphonebetriebssysteme ausgerollt werden. Darüber hinaus, sind moderne Mobiltelefone mit einer Vielzahl an Sensoren ausgestattet. Somit lassen sich Rückschlüsse auf die Umgebung der Testperson ziehen, was bei klassischen Ansätzen nicht möglich war. Ziel von ESM-Studien ist es oftmals, Benutzererfahrungen zu bestimmten Situationen zu sammeln, zB. wenn der Teilnehmer einen Ort betritt oder verlässt.

Pager benutzen dagegen vordefinierte Zeiten und machen somit die Erfassung von Ereignissen, die zeitlich nicht exakt vorhergesagt werden können, unmöglich. Im Gegensatz dazu, sind Smartphones in der Lage, den Kontext in dem sich der Benutzer befindet mithilfe der Sensoren zu erkennen und ihn aufzufordern, den Fragebogen auszufüllen, noch während das gewünschte Ereignis stattfindet. Ortsbasierte Erinnerungen sind dabei bereits heute in vielen Applikationen in Verwendung, unter anderem bietet Google Now diese Funktionalität im am weitesten verbreiteten mobilem Betriebssystem Android bereits von Hause aus an. Der größte Nachteil der Pager-Methode ist, das man vollständig abhängig ist von der Ehrlichkeit und Zuverlässigkeit der Benutzer bei ihren Tagebucheinträgen. Beim mESM dagegen ist vieles mithilfe der Sensoren bekannt, ohne das der Benutzer es eingeben muss, wie beispielsweise der Aufenthaltsort des Smartphones, was der Benutzer gerade macht und wer sich sonst noch in der Umgebung befindet. Somit können einige der vom Benutzer verursachten Fehler und Falschangaben vermieden werden.

Ältere Studien versuchten oftmals, Tagebucheinträge mit Körperdaten wie Blutdruck und Herzschlagfrequenz zu verknüpfen [7]. Die beim mESM aus den Sensoren gewonnenen Daten können für eine Bewertung sowohl des körperlichen als auch des psychischen Zustandes des Benutzers verwendet werden. Geräte wie Smartwatches sind in der Lage, unter anderem Herzschlag und elektrodermale Aktivität des Benutzers messen. Zur Messung der elektrodermalen Aktivität wird dabei der elektrische Leitungswiderstand der Haut ermittelt. Emotionen oder auch Stress verändern die Hautleitfähigkeit [5]. Diese Messungen ermöglichen ein besseres Einschätzen des Situation, in der sich der Benutzer befindet.

Moderne Smartphones besitzen eine Vielzahl an Sensoren, was sie zusammen mit ihrer mittlerweile beträchtlichen Rechenkraft attraktiv macht für eine Kontexterkennung und Auswertung in Echtzeit. Mobiltelefone werden von einem Großteil der Benutzer die meiste Zeit bei sich getragen. Die von ihnen ermittelten Sensordaten geben somit die tatsächlichen Aktivitäten und das Verhalten ihres Besitzers oder auch Veränderungen in seiner Umgebung wieder.

GPS-Daten werden unter anderem von Google Now bereits seit längerem dazu genutzt, um beispielsweise pünktlich zur üblichen Feierabendzeit die Dauer des Nachhauseweges zu berechnen und auf eventuelle Beeinträchtigungen auf der Strecke hinzuweisen.

Der Beschleunigungsmesser kann dafür verwendet werden um körperliche Aktivitäten des Benutzers zu erkennen. Vom eingebauten Mikrofon aufgezeichnete Töne können dazu verwendet werden, um zu ermitteln, ob eine Unterhaltung in der Umgebung stattfindet oder aber sogar wie gestresst der Benutzer gerade ist oder auch seine sonstige allgemeine Gefühlslage [18,22]. Mithilfe von Bluetooth ist es möglich, weitere Smartphonebesitzer in der Umgebung durch co-location pattern zu identifizieren [21]. Darüber hinaus kann auch einiges an Daten aus der Kombination der unterschiedlichen Sensordaten gewonnen werden, unter anderem auch Kontextinformation aus Sozialen Netzwerken [16,19].

ESM-Forscher sind dank Smartphones nun erstmals in der Lage, kontextsensitive Studien entwerfen zu können. Auch der Empfang von Studiendaten in Echtzeit ist dank dauerhafter Internetverbindung und der immer besser werdenden Mobilfunknetze möglich und somit auch ein Anpassen der Datenerhebung "on the fly".

3.2 Moderne mESM Software Frameworks

Das Design, die Implementierung und der Einsatz von Experience Sampling mithilfe von Mobilgeräten benötigt weitergehende technische Kenntnisse als die klassische Sozialwissenschaftslehre.

Eine auf Mobiltelefone aufbauende mESM-Studie macht einiges an Programmieraufwand notwendig um die Applikation zu erstellen, die Sensordaten zu erfassen und auszuwerten. Ebenso ist die Erstellung anspruchsvoller Maschinenlernmodelle für die Kontextermittlung Voraussetzung. Darüber hinaus muss sichergestellt werden, dass ein zuverlässiger Datenfluss vom Smartphone zum zentralen Server gewährleistet ist. Diese Aufgaben gehen nicht nur üblicherweise über das Fachwissen von Psychologen hinaus, sondern führen zu unnötigem, sich wiederholendem Aufwand bei jeder neuen Studie. Die ersten ESM-Frameworks gab es bereits vor dem Aufkommen der Smartphones und wurden unter anderem für PDAs entwickelt. Diese verfügten aber über eine sehr eingeschränkte Kontexterkennung, da diesen Geräten die notwendige Sensorik fehlte. Großer Vorteil gegenüber der klassischen Pager und Tagebuchmethode war hierbei, das sowohl die Benachrichtigung, dass eine Benutzerinteraktion notwendig ist, als auch die Datensammlung, nun auf dem selben Gerät stattfinden konnten. Eines der Hauptproblem war hierbei aber, das PDAs niemals große Verbreitung fanden und somit groß angelegte ESM-Studien nicht möglich waren, ohne enorme Kosten für die den Studienersteller zu verursachen. Das Erkennen von Kontext war das am meisten vermisste Feature der traditionellen ESM-Untersuchungen. Das Aufzeichnen von Daten während bestimmter Ereignisse ist dabei von ganz besonderem Interesse für Psychologen [23]. Insbesondere wenn das gewünschte Geschehnis eher selten vorkommt, nur von kurzer Dauer ist, oder zeitlich nicht

vorhersehbar, ist es ohne Kontexterkennung recht wahrscheinlich, gar nichts, oder nur sehr wenig davon zu erfassen.

Spätere Frameworks lieferten wesentlich bessere Ergebnisse, da mithilfe der in den Mobilgeräten verbauten Sensoren eine Kontexterkennung möglich wurde. Dabei gibt es entweder die Möglichkeit die Sensordaten nur aufzuzeichnen, um sie anschließend zur Analyse auf einen zentralen Server hochzuladen, oder aber sie werden unmittelbar auf dem Gerät verarbeitet. Die Weiterverarbeitung ermöglicht dabei, die Situation zu erkennen, in der sich der Benutzer gerade befindet. Anschließend kann dem Benutzer der passenden Fragebogen ausgeliefert werden, falls die Situation als relevant erkannt wird.

Mit aufkommen der Smartphones und den neuen Möglichkeiten die ihre Vielzahl an Sensoren bieten, kam auch immer mehr freie und quelloffene mESM Software auf den Markt, aber auch eine Vielzahl an kommerziellen Produkten. Die reinen Sensordaten die mit dem Verhalten des Benutzers in Zusammenhang stehen, sind oftmals schwierig unmittelbar zu interpretieren. Man nehme die Fortbewegung als Beispiel. Man sieht den Sensordaten für sich einzeln genommen nicht unbedingt direkt an, um welche Art der Fortbewegung es sich dabei handelt. Deswegen ist es notwendig, das mESM-Frameworks in der Lage sind, die in diesem Fall dafür notwendigen Sensordaten zu kombinieren und daraus abzuleiten, ob jemand beispielsweise geht, rennt, oder Auto fährt.

Einige Frameworks wie Psychlog [10] und Open Data Kit [12] bieten auch Unterstützung für externe Sensoren, die die Benutzbarkeit von ESM für Ambulantes Assessment verbessern.

Ein gutes mESM-Framework sollte dabei sowohl einfach vom jeweiligen Studienentwickler zu bedienen sein, als auch trotzdem noch genügend Möglichkeiten bieten, so dass eine Vielzahl an unterschiedlichen Studien unterstützt werden kann.

Dabei gibt es einige unterschiedliche Ansätze. MyExperience beispielsweise, ein für Pocket PC noch vor dem aufkommen der Smartphones geschriebenes Framework, nutzt eine auf XML basierende Oberfläche. In dieser kann der Studiensteller unter anderem festlegen welche Sensordaten benutzt werden sollen, den Inhalt der jeweiligen Fragebögen und wann der Benutzer aufgefordert werden soll, diese Fragebögen auszufüllen [9].

Das später für Android-Smartphones entwickelte Framework OpenDataKit, hatte zum Ziel, eine möglichst einfache Erstellung von Studien zu ermöglichen. Damit kann sich ein Studiensteller per drag-and-drop eine Applikation zusammenstellen. Darüber hinaus unterstützt es auch noch den automatischen Datenupload und die Speicherung der Daten [12].

Auch Soziale Netzwerke können als Sensor genutzt werden, um das menschliche Verhalten besser zu verstehen. Ein immer größerer Teil an sozialer Interaktion findet über diese statt, wie über 1,5 Milliarden monatlich aktive Benutzer auf Facebook zeigen [4]. Die Beobachtung und Auswertung von Daten aus sozialen Netzwerken gewinnt dadurch immer mehr an Bedeutung in sozialwissenschaftlichen Studien. Die Informationen der sozialen Netzwerke können mit den Sensoren des Smartphones verknüpft werden, um so mehr über den Beziehungsstatus

der Personen in der Umgebung, oder auch über Einträge des Telefonbuches des jeweiligen Mobiltelefons zu erfahren. Von dieser Art der Informationsverknüpfung in Echtzeit wird in SenSocial Gebrauch gemacht. Dabei handelt es sich um eine verteile Middleware, die sowohl auf den Smartphones der Benutzer, als auch auf einem zentralen Server verweilt und dafür sorgt, das aus sozialen Netzwerken gewonnene Daten mit den Daten der Smartphonesensoren verknüpft werden [19].

4 Herausforderungen bei der Datenaufzeichnung mit Mobilgeräten

Mobilgeräte haben das traditionelle ESM grundlegend verändert und revolutioniert. Sie ermöglichten eine kontextabhängige Datenerhebung, schnelle Entwicklung von Prototypen, eine bessere Skalierbarkeit von Studien und auch eine zum Großteil automatische Ergebnisanalyse. Jedoch gibt es immer noch einige Einschränkungen, die bereits in der traditionellen Methode vorhanden waren. Es stellt sich zum einen die Frage, wie man bestimmte Erlebnisse aufzeichnen kann, ohne auf den Alltag des Teilnehmers Einfluss zu nehmen. Außerdem ist es schwierig, Daten eines wichtigen Geschehens aufzuzeichnen, wenn sich der Lebensstil des Probanden als wechselhaft und schwierig vorherzusagen erweist.

Benutzer für die mESM-Applikation zu rekrutieren gestaltet sich oftmals schwierig, schließlich kann man die Applikation potenziell an Millionen von Benutzern testen. Mehr Teilnehmer bedeutet auch bessere Studienergebnisse, diese aber oft schwierig zu finden. Außerdem kamen auch neue technische Herausforderungen dazu, die Einfluss darauf nehmen, wie Studien erstellt werden sollten.

4.1 Rekrutierung und langfristige Motivation von Teilnehmern

Das Internet gab Sozialwissenschaftlern einen relativ einfachen Zugriff auf eine große und vielfältige Menge an potenziellen Teilnehmern. Es schwächte somit das Problem ab, das man sozialwissenschaftliche Studien größtenteils mit einer kleinen Gruppe an Studenten als Probanden durchführen musste [11].

Die ersten mESM-Studien hatten trotz Nutzung des Internets Probleme mit der Rekrutierung, da eine Teilnahme an diesen weiterhin durch eine geringe Verbreitung der benötigten Hardware beschränkt war, in diesem Fall die Pocket PCs, die bei den ersten mESM-Studien zum Einsatz kamen. Mit Aufkommen und immer weiterer Verbreitung von Smartphones mit geschätzten 1.86 Milliarden Smartphone-Nutzern weltweit im Jahre 2015 und immer noch steigender Tendenz [6], hat man nun so zuvor nicht vorstellbare Möglichkeiten zur Ausrollung von Studien. Diese kann relativ einfach über die auf allen größeren Smartphonebetriebssystemen vorhandenen Appstores geschehen und erhöht die Studienreichweite wesentlich. Die Teilnehmerrekrutierung über das Internet ist darüber hinaus auch noch repräsentativer für die tatsächliche Bevölkerung, als es mit den traditionellen Methoden der Fall war [11].

Darüber hinaus bietet ein smartphonesbasiertes mESM zusätzliche Anonymität, da der Benutzer weder verpflichtet ist seinen richtigen Namen zu nennen, wenn er es nicht wünscht, noch mit irgendjemandem von denjenigen, die die Studie betreiben, direkt in Kontakt treten muss.

Einer der Nachteile davon, das es keine direkte Kontrolle über Studienteilnehmer gibt, besteht darin, das man nur schwierig kontrollieren kann, das tatsächlich nur berechtigte, bzw. gewünschte Probanden an einer Studie teilnehmen. Beispielsweise könnte es durch Falschangaben zu einer unerwünschte Teilnahme von Minderjährigen an Studien die ausschließlich für für Erwachsene gedacht sind kommen. Das Problem von bewussten Falschangaben wird ein Stück weit dadurch abgeschwächt, dass die Applikation einiges an Informationen über den Teilnehmer liefert. Unter anderem seine Aktivitäten, Aufenthaltsorte, seine Beweguns- und Kommunikationsmuster und weitere Eigenschaften. Mithilfe dieser aus der Studie gewonnen Daten können Rückschlüsse gezogen werden in Bezug auf Alter, Geschlecht und sogar zu einem gewissen Teil den sozialen Status des Probanden [8]. Dieses Wissen kann außerdem auch dazu verwendet werden, um gezielt eine demographische Gruppe auszuwählen, oder Studien an unterschiedliche Gruppen jeweils anzupassen. Beispielsweise kann die Zeit der Stichprobennahme an die üblichen lokalen Gewohnheiten angepasst werden oder auch unterschiedliche Fragen zu stellen, in Abhängigkeit der sozialen Gruppe des Teilnehmers.

Die Teilnahme an einer Studie kann vom Benutzer jederzeit ohne weiteres beendet werden, indem er entweder die Applikation entfernt oder auch einfach nur ignoriert. Deswegen ist es oftmals relativ schwierig, die Benutzer zur aktiven Teilnahme, vor allem auch über längere Zeiträume zu bewegen. Um das aktive Teilnehmen attraktiver zu gestalten, geben manche Applikationen bestimmte Information an die User zurück. Die mESM Applikation Emotion Sense zeichnet Gefühlszustände der Benutzer und kontextabhängige Sensordaten auf und versucht dabei, ihre Benutzer durch Gamification und Anzeige von einigen aus der Teilnahme gewonnenen Erkenntnissen über den Benutzer zu motivieren [17]. Teilnehmer können neue Teile der Applikation freischalten, indem sie aktiv mitmachen und neue Daten liefern.

4.2 Zeitpunkt der Befragungen

Smartphones gehören fest zum Alltag der Teilnehmer. Deswegen ist es wichtig, dass Benutzerinteraktionen zu einem passenden Zeitpunkt angefordert werden. Das Ziel sollte sein, sowohl nicht all zu störend auf den Benutzer zu wirken, als auch seine eigentliche Aufgabe der Datensammlung zu erfüllen und das in diesem Moment Erlebte aufzuzeichnen. Ein Benutzer der momentan nicht gestört werden will oder auch kann, wird nämlich mit hoher Wahrscheinlichkeit nicht unmittelbar auf die ihm gestellten Fragen antworten, was wiederum zu einer Verfälschung des Erlebten bei einer späteren Beantwortung oder auch zu einem Ausbleiben dieser führen kann [3].

Darüber hinaus werden Benutzer üblicherweise zum Ausfüllen eines Fragebogens zum Erlebten aufgefordert, indem sie eine Benachrichtigung auf das Smart-

phone erhalten, so wie sie es auch sonst im Umgang mit dem Smartphone gewohnt sind, wenn das Smartphone eine neue Interaktion bereit hält. Ein Benutzer wird eher dann schnell auf die Benachrichtigung reagieren und sich in dem, was er gerade tut unterbrechen lassen um den Fragebogen zeitig auszufüllen, wenn die Benachrichtigung in einem günstigen Moment ausgeliefert wird.

Einige Untersuchungen befassten sich damit, wie man einen guten Zeitpunkt für eine Unterbrechung erkennt und stellten fest, das es größtenteils von der Situation abhängt in der sich der Benutzer gerade befindet [14,15]. Beispielsweise kann mithilfe des Beschleunigungssensors der Wechsel von einer Tätigkeit in die nächste festgestellt werden. Während dieses Übergangs reagieren Benutzer für gewöhnlich wohlwollend, wenn sie vom Smartphone gestört werden [14]. Laut der Studie InterruptMe können diese guten Momente für eine Unterbrechung des Benutzers aber nicht isoliert für sich alleine betrachtet werden, sie sind wesentlich von der ansonsten in letzter Zeit aufgetretenen Menge an Unterbrechungen abhängig [20]. Das ganze spielt eine immer wichtigere Rolle, je mehr Applikationen beim Benutzer in Benutzung sind und somit auch mehr Benachrichtigungen um Aufmerksamkeit wetteifern.

4.3 Technische Herausforderungen beim mESM

Bei Smartphones handelt es sich nicht mehr wie bei den klassischen Mobiltelefonen um ein reines Kommunikationsgerät. Es sind heutzutage Allzweckgeräte mit denen sowohl in jeglicher Form kommuniziert werden kann, aber auch Sachen wie Surfen im Internet, Terminverwaltung, der morgendliche Wecker, Gaming, Navigation und viele weitere Aufgaben erledigt werden.

Diese ganzen Aufgaben limitieren die Ressourcen, die für die Datenerhebung zur Verfügung stehen. Darüber hinaus muss eine mESM-Applikation auch immer aktiv auf dem Gerät laufen. Sie kann nicht, wie bei den meisten Apps üblich, in der Zeit in der sie nicht genutzt wird in den Schlafmodus versetzt werden, da sie dann die Situation, in der sich der Benutzer befindet, nicht erkennen kann, ihm somit auch nicht die richtigen Fragen zu seinen Erlebnissen zum richtigen Zeitpunkt vorlegen kann und somit ihre eigentliche Aufgabe nicht wie gewünscht ausführen könnte.

Eine der problematischsten Ressourcen für viele mobile sensing Applikationen ist die beschränkte Batteriekapazität von Mobilgeräten. Viele der in Smartphones verbauten Sensoren sind nicht für das fortlaufende Messen gedacht und verbrauchen bei häufiger Aktivität eine Menge an Energie. Viele Smartphonebesitzer sehen es als Voraussetzung an, das ihr Gerät den ganzen Tag über mit einer Akkuladung durchhält, bevor es dann Nachts zum Laden angesteckt wird, um für den nächsten Tag bereit zu sein. Oftmals ist tagsüber auch gar keine Lademöglichkeit gegeben, oder aber wird als unkomfortabel angesehen. Die aktiven Sensoren, die im Vordergrund laufende mESM-Applikation und von ihr getätigte Datenverarbeitung können den Energieverbrauch entscheidend erhöhen und zu einer Ablehnung bei potenziellen Benutzern führen.

Dem wird häufig versucht durch adaptives Messen entgegenzuwirken, also entweder seltenere Messvorgänge oder aber Messen mit einer geringeren Genau-

igkeit. Oftmals kann es ausreichend sein, anstatt der genauen GPS-Koordinaten des Benutzers nur die Cell-ID der verbundenen Basisstation zu ermitteln und damit seinen ungefähren Standort, um dadurch Energie zu sparen.

Das AndWellness Framework ermöglicht es Studienerstellern beispielsweise das Verhältnis zwischen Messgenauigkeit und Energieverbrauch zu verändern und implementiert die hierarchische Sensoraktivierung um den Energieverbauch noch weiter reduzieren zu können [13]. Dabei werden sparsame, aber dafür nicht so exakte Sensoren verwendet um herauszufinden, ob die genaueren, aber dafür auch energiehungrigen Sensoren aktiviert werden sollen, weil genauere Informationen benötigt werden.

Wenn das Mobilgerät sich beispielsweise mit einem neuen Access Point verbindet oder den Kontakt verliert, wird angenommen, dass der Benutzer sich fortbewegt und das GPS schaltet sich hinzu um eine genauere Aufzeichnung der Fortbewegung zu ermöglichen. Ziel ist aber weiterhin, trotz Anpassen der Abtastrate mit adaptivem Messen oder Benutzung der hierarchischen Aktivierung von Sensoren sicherzustellen, keine wichtigen Ereignisse zu verpassen und derzeit Gegenstand aktueller Forschungsarbeiten.

Für die Datenspeicherung und die Datenanalyse, die Visualisierung von Daten und auch zum Verändern von Einstellungen der mESM-Applikation auf den aufzeichnenden Mobilgeräten kommen zentrale Server zum Einsatz. Dabei kann es eine entscheidende Rolle für die Leistungsfähigkeit einer mESM-Applikation spielen, wo welche Funktionalität im Einzelnen beheimatet ist, also welche Funktionalität auf dem Server liegt und welche auf dem Mobilgerät. Dies beeinflusst sowohl die Performanz als auch die Möglichkeiten die der Applikation zur Verfügung stehen.

Serverseitige Verarbeitung von Daten hat die Vorteile von hoher Rechenleistung im Vergleich zu den einzelnen Mobilgeräten. Der Server verfügt darüber hinaus über eine globale Sicht auf das gesamte System und ist somit in der Lage, bessere Rückschlüsse zu ermöglichen als es mit einer rein lokalen Sicht der Fall ist, da auch die Daten anderer Nutzer zur Auswertung verwendet werden können.

Außerdem kann es zu Energieeinsparungen auf Seiten der Mobilgeräte kommen, da diese die für die Datenverarbeitung nötige Energie einsparen. Da das Senden von Daten aber auch wiederum einiges an Energie verbraucht, kann es vorkommen, dass die durch die Datenverarbeitung stattgefundene Komprimierung der Daten wiederum Energie beim Senden einspart, dabei hängt es im einzelnen Fall davon ab, welche Art der Vorverarbeitung auf Seiten des Mobilgerätes stattfindet und kann sich von Fall zu Fall unterscheiden, auch abhängig von der Netzqualität des Mobilgerätes.

Andererseits kann aber der Datentransfer von großen Mengen an Sensordaten zum Server ohne Vorverarbeitung teuer, oder auch einfach nicht machbar sein, insbesondere wenn die Datenübertragung über das Mobilfunknetz geschieht.

Der Mobilfunkempfang ist immer noch nicht flächendeckend, selbst in industrialisierten Ländern und vor allem nicht innerhalb von Gebäuden, insbesondere was die schnellen 4G-Netze angeht. Die Netzabdeckung ist dabei auch relativ

stark vom jeweiligen Netzbetreiber abhängig, bei dem man unter Vertrag steht. Der Ausbau der Mobilfunknetze und die Kosten der Mobilfunktarife unterscheiden sich erheblich zwischen unterschiedlichen Ländern. Insofern also häufiges Versenden von größeren Datenmengen nötig ist, kann das zu einem großen Problem werden.

Aus diesem Grund bieten einige der Frameworks Appentwicklern die Möglichkeit an, Richtlinien für die Datenübertragung festzulegen. Beispielsweise das der Sendevorgang der Daten zum Server nur bei vorhandener WiFi-Verbindung geschehen soll, oder auch das bei niedrigem Batteriestand keine Daten hochgeladen werden sollen.

Ein weiterer Faktor der noch hinzukommt, warum ein ausgewogenes Verhältnis zwischen lokaler- und serverseitiger Datenverarbeitung vorhanden sein sollte ist die Privatsphäre. Viele potenzielle Studienteilnehmer sorgen sich bei einer solchen Rundumüberwachung natürlich um die Wahrung und Sicherstellung dieser. Es muss sichergestellt werden, das Daten nicht an unbefugte Dritte gelangen können. Verschlüsselung und Anonymisierung von personenbezogenen Daten sind in diesem Kontext unerlässlich.

Insbesondere durch die NSA-Affäre und zuletzt auch der gerichtlichen Auseinandersetzung um das Entsperren eines iPhones zwischen Apple und dem FBI, die sich über Wochen hinzog und medial stark begleitet wurde [2], ist das Thema Datenschutz und Privatsphäre stärker in das öffentliche Interesse gerückt worden.

Aufenthaltsorte und Stimm- oder Videoaufnahmen sind Privatsache und sollten entsprechend geschützt werden. Eine Möglichkeit dies zu erreichen besteht darin, für ein ausgewogenes Verhältnis zwischen serverseitiger- und lokaler Datenverarbeitung zu sorgen.

Bei Sprachaufnahmen muss beispielsweise nicht das gesamte Audiofile an den Server gesendet werden, sondern es genügt, Eigenschaften des Sounds zu extrahieren die für die Sprachklassifizierung relevant sind, der Mel-Frequenz-Cepstrum-Koeffizient wäre eine Möglichkeit. Diese extrahierten Eigenschaften können für die Analyse an den zentralen Server gesendet werden. Somit ist sichergestellt, dass die Audiodatei nicht wieder hergestellt werden kann, sollten diese Daten in die Hände Unbefugter gelangen.

Anstatt die vollständigen GPS-Koordinaten an den Server zu versenden, kann die Applikation auch einen internen Klassifizierer für den Aufenthaltsort des Benutzers ermitteln und weitergeben, beispielsweise wären Zuhause und Arbeit zwei mögliche solcher Ortsklassifizierer. Diese werden nach Ermittlung durch die Applikation anstatt der tatsächlichen Koordinaten an den Server gesendet, und minimieren die Menge an Daten die über den Nutzer ohne weiteres Wissen gewonnen werden können.

Smartphonenutzer tragen das Gerät nicht immer durchgehend bei sich oder schalten es auch mal komplett aus. Weitere Probleme mit der Verfügbarkeit und Genauigkeit einiger Sensoren unter bestimmten Bedingungen existieren. Üblicherweise ist kein GPS Empfang in Gebäuden gegeben und GPS-Sensoren neigen je nach Smartphonemodell dazu, nicht völlig exakt zu arbeiten und einen

auch mal eine Straße weiter zu positionieren, als es tatsächlich der Fall ist, trotz Aufenthalt im Freien. Bluetooth hat eine stark begrenzte Reichweite. GPS und Bluetooth werden auch gerne aus Gründen der Akkuschonung und zur Wahrung der Privatsphäre ausgeschaltet. Weitere Sensoren, wie die Pulsmessung von Smartwatches stellen sich immer wieder als nicht völlig zuverlässig heraus.

All diese Faktoren führen zu teilweise verfälschten Messungen und können zu falschen Schlussfolgerungen führen.

4.4 Einbau von Verhaltensintervention

Der nächste logische Schritt, nachdem man das menschliche Verhalten relativ genau beobachten und auswerten kann ist es, diese Informationen zu nutzen um zukünftige Handlungsweisen positiv zu beeinflussen. Behaviour change interventions (BCIs) bestehen aus dem Sammeln von Informationen über die Zielperson, setzen von Zielen, Beobachtung des Verhaltens vom Benutzer und das Geben von Feedback.

Später wurde das Konzept der digital BCIs (dBCIs) entwickelt, dabei handelt ist sich um die Verhaltensintervention über das Internet. Damit wurde es möglich Inhalte der Intervention wesentlich schneller zur Teilnehmer zu bringen und dabei die Kosten zu reduzieren, da man keinen direkten Kontakt zum Benutzer mehr braucht. Inhalte können damit auch automatisch auf die jeweilige Zielperson zugeschnitten werden. Es wurde in einer Studie zur Selbstbehandlung von kleineren Krankheitsberschwerden wie Husten, Fieber und Halsschmerzen gezeigt, das Patienten noch besser auf je nach Krankheitssymptome maßgeschneiderte Inhalte reagieren, als auf eine statische Webseite mit Ratschlägen zu Krankheiten, für welche zuvor gezeigt wurde, das sie Arztbesuche reduzieren und das Zutrauen des Patienten zur Eigenbehandlung steigern kann. Diejenigen Patienten die auf der nach krankheitssymptomen angepassten Seite mit ausschließlich für sie relevantem Inhalt nach Hilfe suchten, fühlten sich sicherer im selbstständigen Umgang mit ihrer Krankheit und wollten auch weniger häufig weitergehende gesundheitliche Hilfe, wie einen Doktor in Anspruch nehmen, als dies bei der statischen Webseite der Fall war [25].

Da Smartphones mittlerweile solch allgegenwärtige Geräte geworden sind und ihre Besitzer sie auch so gut wie nie aus den Augen lassen, eignen sie sich natürlich auch hervorragend für Versuche der Verhaltensveränderung des Besitzers. Es gibt mittlerweile einige Projekte, in denen versucht wurde dBCIs aus dem Internet auf die Smartphones zu verfrachten. Leanne G. Morrison et al. entwickelten eine hybride Web- und Appbasierte Intervention die Benutzern bei der Gewichtsabnahme helfen soll [39].

Teilnehmer bekamen dabei Zugriff auf die webbasierte Gewichtsmanagement-Intervention “POWeR” und in 2 alternierenden Wochen auf die Gewichtsmanagement-Applikation “POWeR Tracker”. Die Benutzer waren sich in den Wochen mit Nutzung der Applikation dessen, was sie aßen und auch ihrer Ziele für körperliche Aktivitäten merklich bewusster, als dies in den Wochen ohne Zugang zu der App der Fall war. Aufgezeichnete Daten über die Nutzungsdauer zeigten dabei, das die Nutzungsdauer der Webseite sich in den Wochen mit und ohne Zugriff

kaum voneinander unterschieden. Der POWeR Tracker wurde dabei üblicherweise für kurze Zeitspannen von um die 3 Minuten verwendet, üblicherweise wenn es gerade einen gut passenden Moment gab, wenn der Teilnehmer meinte das die von der App angebotenen Tools gerade hilfreich wären oder weil er gerade eine automatische Benachrichtigung der Applikation erhalten hatte. Wie die Studie zeigt, lassen sich Smartphones dazu nutzen den Benutzer sich seiner gesteckten Zielen mit Bezug auf die Gesundheit mehr verbunden zu fühlen und als eine Motivationshilfe.

5 Fazit

Die Allgegenwärtigkeit von Smartphones verändert momentan die Möglichkeiten der Sozialwissenschaften enorm. Der technische Fortschritt beim mobile computing führte zur Entwicklung von mobile ESM, was Sozialwissenschaftler in die Lage versetze, das Verhalten und Handeln einer vorher so nicht denkbar großen Menge an Menschen beobachten und Analysieren zu können, ohne teure technische oder sonstige Anschaffungen tätigen zu müssen. Auch die Qualität, und Granularität der Daten die man mithilfe der Smartphonesensoren und sonstiger Interaktion des Benutzers mit dem Mobilgerät erhält, versetzt Forscher in die Lage, sehr genau über das Verhalten des Benutzers Bescheid zu wissen.

Jedoch gibt es immer noch Probleme im Umgang mit den Benutzern. Auch wenn mESM durch die Integration ins Smartphone bereits deutlich weniger Aufwand für den Teilnehmer bedeutet, als es beim klassischen ESM der Fall war, liefern sie doch immer noch häufig falsche Informationen oder Beantworten die Fragen nicht rechtzeitig, wenn sie sich von ihnen in ihrem Alltag gestört fühlen.

Es muss mehr Wert gelegt werden auf Applikationen, die auf den Benutzer so wenig wie möglich störend wirken und der Teilnehmer muss einen tatsächlichen eindeutigen Nutzen für sich feststellen können, um sich die Mühe zu machen aktiv teilzunehmen. Denn die Langzeitmotivation ist sehr schwierig und es scheint noch niemand einen wirklich wirksamen Weg gefunden zu haben, Teilnehmer langfristig erfolgreich zu motivieren, um auch länger andauernde Beobachtungen von Teilnehmern zu ermöglichen.

Aus den traditionellen Sozialwissenschaften hat sich ein neues, interdisziplinäres Wissenschaftsfeld gebildet. Aus vom Benutzer generierten Inhalten wird dabei unter Einsatz von unter anderem Statistik und Maschinenlernmethoden versucht, sozialwissenschaftliche Konzepte zu erklären.

Insbesondere was die Beeinflussung von Verhalten des Teilnehmers betrifft, steckt die Forschung noch in den Kinderschuhen und es kann auf Grund von fehlenden größer angelegten Studien relativ wenig zur Effektivität von mobile dBCIs gesagt werden.

In Zukunft wird es immer mehr smarte Geräte geben, ob es nun der Fernseher, der Kühlschrank, die Waschmaschine oder die Kaffeemaschine ist. Es werden sich somit noch eine ganze Reihe an neuen Möglichkeiten zur Datengewinnung ergeben, als es jetzt schon der Fall ist und es bleibt eine Menge für die Wis-

senschaftler zu tun, um noch besser beobachten und das menschliche Verhalten verstehen und analysieren zu können.

Literatur

1. Anzahl der Smartphone-Nutzer in Deutschland in den Jahren 2009 bis 2015 (in Millionen). In Statista - Das Statistik-Portal. Zugriff am 16. Dezember 2015, von http://de.statista.com/statistik/daten/studie/198959/umfrage/anzahl-der-smartphonenutzer-in-deutschland-seit-2010/
2. Apple versus FBI - iPhone-Streit zwischen Apple und US-Behörden beendet aus Focus Online von http://www.focus.de/digital/apple-versus-fbi-iphone-streit-zwischen-apple-und-us-behoerden-beendet_id_5389574.html
3. Ask, but Don't Interrupt: The Case for Interruptibility-aware Mobile Experience Sampling. In: Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers. UbiComp/ISWC'15 Adjunct
4. Facebook. (n.d.). Anzahl der monatlich aktiven Facebook Nutzer weltweit vom 3. Quartal 2008 bis zum 4. Quartal 2015 (in Millionen) . In Statista - Das Statistik-Portal. Zugriff am 10. März 2016, von http://de.statista.com/statistik/daten/studie/37545/umfrage/anzahl-der-aktiven-nutzer-von-facebook/
5. Prof. Dr. Tanja Schultz: Cognitive Systems Lab, KIT (TH), Fakultät für Informatik, Vorlesung WS 2012/2013, Biosignale und Benutzerschnittstellen, Elektrodermale Aktivität (EDA), http://csl.anthropomatik.kit.edu/downloads/vorlesungsinhalte/BSBS1213_E_ElectroDermalActivity.pdf
6. Prognose zur Anzahl der Smartphone-Nutzer weltweit von 2012 bis 2019 (in Milliarden). In Statista - Das Statistik-Portal. Zugriff am 16. Dezember 2015, von http://de.statista.com/statistik/daten/studie/309656/umfrage/prognose-zur-anzahl-der-smartphone-nutzer-weltweit/
7. Fahrenberg, J., Myrtek, M.: Ambulatory assessment: Computer-assisted psychological and psychophysiological methods in monitoring and field studies (1996)
8. Frias-Martinez, V., Virseda, J.: On the Relationship Between Socio-economic Factors and Cell Phone Usage. In: Proceedings of the Fifth International Conference on Information and Communication Technologies and Development. pp. 76–84. ICTD '12, ACM, New York, NY, USA (2012), http://doi.acm.org/10.1145/2160673.2160684
9. Froehlich, J., Chen, M.Y., Consolvo, S., Harrison, B., Landay, J.A.: MyExperience: A System for in Situ Tracing and Capturing of User Feedback on Mobile Phones. In: Proceedings of the 5th International Conference on Mobile Systems, Applications and Services. pp. 57–70. MobiSys '07, ACM, New York, NY, USA (2007), http://doi.acm.org/10.1145/1247660.1247670
10. Gaggioli, A., Pioggia, G., Tartarisco, G., Baldus, G., Corda, D., Cipresso, P., Riva, G.: A mobile data collection platform for mental health research. Personal and Ubiquitous Computing 17(2), 241–251 (2011), http://dx.doi.org/10.1007/s00779-011-0465-2
11. Gosling, S.D., Vazire, S., Srivastava, S., John, O.P.: Should we trust web-based studies? A comparative analysis of six preconceptions about internet questionnaires. American Psychologist 59(2), 93 (2004)

12. Hartung, C., Lerer, A., Anokwa, Y., Tseng, C., Brunette, W., Borriello, G.: Open Data Kit: Tools to Build Information Services for Developing Regions. In: Proceedings of the 4th ACM/IEEE International Conference on Information and Communication Technologies and Development. pp. 18:1–18:12. ICTD '10, ACM, New York, NY, USA (2010), http://doi.acm.org/10.1145/2369220.2369236
13. Hicks, J., Ramanathan, N., Kim, D., Monibi, M., Selsky, J., Hansen, M., Estrin, D.: AndWellness: An Open Mobile System for Activity and Experience Sampling. In: Wireless Health 2010. pp. 34–43. WH '10, ACM, New York, NY, USA (2010), http://doi.acm.org/10.1145/1921081.1921087
14. Ho, J., Intille, S.S.: Using Context-aware Computing to Reduce the Perceived Burden of Interruptions from Mobile Devices. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 909–918. CHI '05, ACM, New York, NY, USA (2005), http://doi.acm.org/10.1145/1054972.1055100
15. ter Hofte, G.H.H.: Xensible Interruptions from Your Mobile Phone. In: Proceedings of the 9th International Conference on Human Computer Interaction with Mobile Devices and Services. pp. 178–181. MobileHCI '07, ACM, New York, NY, USA (2007), http://doi.acm.org/10.1145/1377999.1378003
16. Lane, N., Miluzzo, E., Lu, H., Peebles, D., Choudhury, T., Campbell, A.: A survey of mobile phone sensing. Communications Magazine, IEEE 48(9), 140–150 (Sept 2010)
17. Lathia, N., Rachuri, K.K., Mascolo, C., Rentfrow, P.J.: Contextual Dissonance: Design Bias in Sensor-based Experience Sampling Methods. In: Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing. pp. 183–192. UbiComp '13, ACM, New York, NY, USA (2013), http://doi.acm.org/10.1145/2493432.2493452
18. Lu, H., Frauendorfer, D., Rabbi, M., Mast, M.S., Chittaranjan, G.T., Campbell, A.T., Gatica-Perez, D., Choudhury, T.: StressSense: Detecting Stress in Unconstrained Acoustic Environments Using Smartphones. In: Proceedings of the 2012 ACM Conference on Ubiquitous Computing. pp. 351–360. UbiComp '12, ACM, New York, NY, USA (2012), http://doi.acm.org/10.1145/2370216.2370270
19. Mehrotra, A., Pejovic, V., Musolesi, M.: SenSocial: A Middleware for Integrating Online Social Networks and Mobile Sensing Data Streams. In: Proceedings of the 15th International Middleware Conference. pp. 205–216. Middleware '14, ACM, New York, NY, USA (2014), http://doi.acm.org/10.1145/2663165.2663331
20. Pejovic, V., Musolesi, M.: Interruptme: designing intelligent prompting mechanisms for pervasive applications. In: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing. pp. 897–908. ACM (2014)
21. Rachuri, K.K., Mascolo, C., Musolesi, M., Rentfrow, P.J.: SociableSense: Exploring the Trade-offs of Adaptive Sampling and Computation Offloading for Social Sensing. In: Proceedings of the 17th Annual International Conference on Mobile Computing and Networking. pp. 73–84. MobiCom '11, ACM, New York, NY, USA (2011), http://doi.acm.org/10.1145/2030613.2030623
22. Rachuri, K.K., Musolesi, M., Mascolo, C., Rentfrow, P.J., Longworth, C., Aucinas, A.: EmotionSense: A Mobile Phones Based Adaptive Platform for Experimental Social Psychology Research. In: Proceedings of the 12th ACM International Conference on Ubiquitous Computing. pp. 281–290. UbiComp '10, ACM, New York, NY, USA (2010), http://doi.acm.org/10.1145/1864349.1864393
23. Reis, H.T., Gable, S.L.: Event-sampling and other methods for studying everyday experience. Handbook of research methods in social and personality psychology pp. 190–222 (2000)

24. Trull, T.J., Ebner-Priemer, U.: Ambulatory assessment. Annual review of clinical psychology 9, 151 (2013)
25. Yardley Lucy, Joseph Judith, Michie Susan, Weal Mark, Wills Gary, Little Paul: Evaluation of a Web-based Intervention Providing Tailored Advice for Self-management of Minor Respiratory Symptoms: Exploratory Randomized Controlled Trial. Journal of Medical Internet Research 12(4), e66 (Nov 2010), http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3056528/

Proximity and Velocity Recognition of External Objects on a Mobile Observer

Florian Leicher*

Advisor: Prof. Dr. Michael Beigl†

Karlsruhe Institute of Technology (KIT)
Pervasive Computing Systems – TECO
*ugfan@student.kit.edu
†beigl@teco.edu

Abstract. The TECO institute is working on an enhancement for its project *ProximityHat*, an electronic travel aid (ETA). This paper is a state of the art analysis about new possible sensors for the ProximityHat. The following sensor types are being analyzed: laser range detection, ultrasonic range detecting, Doppler velocity measurement and image processing with a stereo camera. These sensors are used for gathering proximity and velocity data of external objects. Relevant works for each sensor type mounted on a mobile observer (human or robot) are discussed with a special focus on ETAs.

Keywords: Mobile Sensor, Human Mounted Sensor, ProximityHat, Electronic Travel Aid, ETA, Velocity Sensor

1 Introduction

There is a huge number of electronic devices tested in science (and some also on the market) that are designed to help handicapped persons to cope with certain tasks in their lives [1]. When talking about aiding the blind, one important field is vision substitution – representing visual clues a healthy person could perceive from the outside world in another way. The majority of such vision substitution devices use auditory or tactual (mostly vibration) stimuli as a replacement [2].

Farcy et al. name two categories of such substitution devices, *Electronic Travel Aids* (ETA) and *Electronic Orientation Aids* (EOA) [3]. ETAs "transform information about the environment that would normally be relayed through vision into a form that can be conveyed through another sensory modality" [2] and as such help a user e.g. avoiding obstacles. EOAs help the user to find a route in an unknown environment. While for EOAs the environment is usually prepared in some way (e.g. there are beacons placed on a predefined route to send location clues to the sensors [4], there is a map provided for GPS location [3], or the experiment assumes that navigational clues are provided [5]), ETAs usually work only with the sensory data gathered on the blind person. As the ProximityHat and our proposed extensions are in the category of ETAs, this work will mostly focus on such aids.

1.1 The ProximityHat

The ProximityHat is the product of a Bachelor's Thesis by Florian Braun at the TECO Institute in 2015 [6]. The idea of this ETA is to enable the wearer to *feel the space* around them. The set-up consists of a head-worn array of six ultrasonic distance sensors facing in different directions equally spaced at 60° and six servomotors located underneath the sensors that can exert pressure to the underlying skin. These sensors detect the proximity of walls or other objects on head level in those six directions. Depending on the measured value the corresponding motor pushes lighter or harder against the head. The novelty of this approach is that unlike most other projects in this field it does not use vibration but pressure as a tactile way to communicate or "display" information. One advantage of this is that the spatial resolution of vibration is less fine than the one of pressure. And furthermore, constant pressure – unlike continuous vibration – can easily be ignored by the brain if an object stays close.

Fig. 1: The ProximityHat

The ProximityHat tests well in that the users feel comfortable wearing it and are able to fulfill tasks like navigating along hallways and through open doors. Yet the biggest drawback is the high reaction time of about half a second until the hat presents the user a change in proximity of an external object so that the wearer has already walked past an object when it is finally shown. This latency is partly due to the fact that the sensor only works with the speed of sound and partly that requesting the distances is not parallelized and executed on an Arduino Mega[1] with 16 MHz clock speed. Another negative feature on a psychological level is that the ProximityHat has a high *social weight* [7]. Even though the set-up is comfortable a user might feel uneasy wearing such a sensor array on their head as the ProximityHat does not integrate with regular clothing. As it is the goal to enhance the ProximityHat, the following sections will discuss how other functionalities can be added.

[1] Arduinos are open-source microcontroller boards `http://www.arduino.cc`

1.2 Augmenting Human Senses

Taking the ProximityHat as a starting point, our goal is to improve its set-up and to find new applications. This means we want to detect events other than the current one (detection of static close objects) and display them on the head. Such events can be detected by using a suited sensor. The challenge is to identify meaningful yet not too complex events from this sensor and communicate these to the wearer of the ProximityHat in an intuitive way. If this succeeds, we are able to extend the human senses.

One can think of a variety of possible events like: fire detection, poisonous gas in the air, warning of sick people, objects on collision course with the wearer, crime warnings and more. Our goal is to build a mobile ETA, so the event has to be detectable by a mobile sensor. A combination of several sensor inputs could generate such an event, e.g. a rise in CO_2 concentration through a particle detector and an increase in temperature could suggest a fire. The goal is to additionally display this event on the ProximityHat. The most promising idea seems to be displaying (fast) *movement* of external objects on the head as the ProximityHat is already detecting some but not all movements (figure 2).

In the current ProximityHat the location of a signal on the head is directly used to represent the direction of the event (e.g. where are close walls around the wearer). Such a set-up makes the most sense if the to-be-displayed event can be mapped to a location relative to the wearer [8], which is the case for movement.

A pressure display as used by the ProximityHat is great for presenting non-complex information as proximity. If more complex outside events are to be displayed, one has to combine different representation methods (like tactile and auditory) or can use vibration patterns of any sort.

2 Exploring the Representation of Moving Objects on the Head

Fast moving objects are potentially an indicator of danger. Especially in traffic situations on the street information about fast moving objects could be beneficial and help orientating. But also in daily life information about any rapid movement – also at the backside of the wearer – is likely useful. One could detect balls, rocks etc. being thrown at oneself or is able to sense if someone else is entering the room. For such scenarios the sensors, unlike the sensors of the current ProximityHat, will have to cover heights other than (just) the head level and must have a shorter latency when representing information to the wearer. Such a system will primarily help blind wearers but one should always examine if other users can profit as well [8].

Research Question Taking the considerations from the previous sections into account, the research question on an enhanced ProximityHat is the following:

> *We will explore the human sensitivity for temporal and spatial vibration patters on the head for a sensory augmentation for velocity and movement of external objects.*

This research question can be split up into a sensor and an actuator part. Section 2.1 describes the confining implications for the sensor part, and section 2.2 for the actuator one. Section 3 provides a state of the art analysis of the sensors that can be used for augmenting the ProximityHat; a state of the art analysis of the actuators and tactile representation is done by Vincent Diener.

2.1 Scenario Implications for the Sensors

When looking at the movement of external objects, one can identify four different cases that could potentially help the user (observer) of our wearable to identify an external danger: A foreign object is moving (a) tangential to the observer, (b) towards the observer, (c) away from the observer, or (d) the observer is moving towards a stationary subject. The ProximityHat is currently only covering the last case. For accurate movement detection, a sensor device needs to operate fast

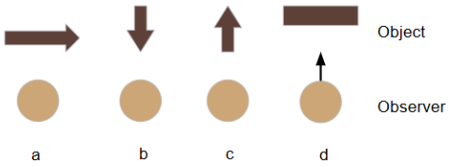


Fig. 2: Different Danger Scenarios: an external object moving (a) tangential, (b) towards, or (c) away from the observer, or (d) the observer moving towards a stationary obstacle.

enough. This has to be considered, looking at the high latency of the Proximity-Hat. Another constraint that applies is that the planned experiments are to be done with a limited budget so only low-cost sensors are taken into consideration.

Taking detecting cars as a main scenario for the ProximityHat, it is necessary to gather data at the waist/chest height of a human. If the sensor array is mounted on top of the head as it is the case with the current ProximityHat, either the detection angle has to be sufficiently wide or the sensor has to be tilted at the correct angle to identify the object. This causes additional difficulties when considering that the moving external objects can be positioned at varying distances from the head of the observer. The head as a location for a sensor has the advantage that even when walking, it is fairly stable. Other body region that are comparably stable are the chest and the waist. Building a set-up for the chest is likely to have a high social weight as a noticeable device has to be worn in front of the t-shirt. If the sensor is located on the belt, a person might have their arms in front of a sensor and thus falsifying the measurements. Also it might not be possible to warn the user about low hanging objects like trees.

2.2 Scenario Implications for Tactile Representation (Actuators)

Our set-up idea for the improved ProximityHat focuses on vibration patterns. This allows representing several events which is not possible with pressure. The tactile display around the head consists of an array of vibration motors arranged in three rows around the head. The middle row is displaced by half the length of a vibrator. This helps to keep the mounting process simple and to maximize the distance between any two actuators. Further tests will show what the ideal distance for sensing vibrators will be. There are some studies that have evaluated sensitivity on the head [9], e.g. Dobrzynski et al. identified that the back is the most sensitive, the sides the least [10]. One primary goal will be to identify the best vibration patterns for the identified scenarios. A variety of possible patterns is depicted in figure 3.

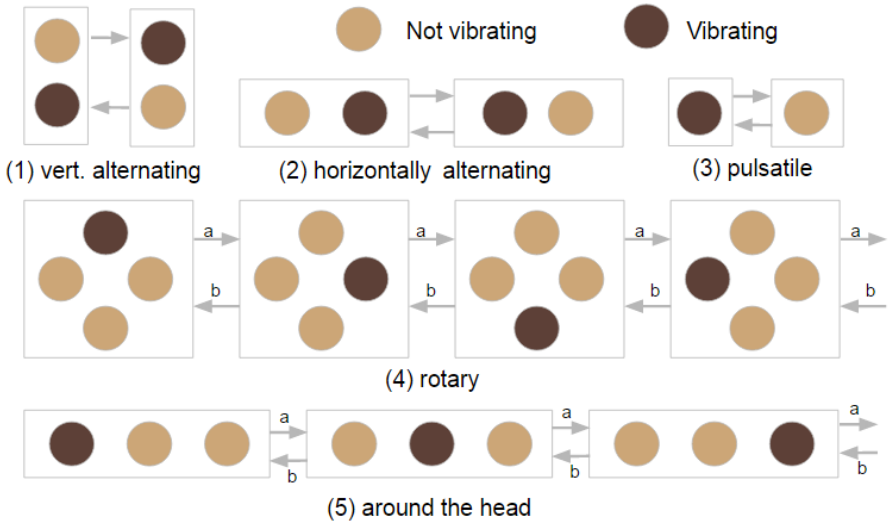


Fig. 3: Possible Vibration Patterns on an actuator set-up with three rows: an alternation of two actuators vibrating in (1) vertical or (2) horizontal direction or (3) just one actuator repeatedly turning on and off; (4) a rotation movement of four vibrators (a) clockwise or (b) counter-clockwise, or (5) a rotating pulse around the head going (a) right or (b) left.

Such patterns have to be evaluated in terms of how easily they are recognized, how intuitively they represent a given danger, and how well any two or more can be combined with each other. A possible mapping could be to use the rotation patterns to describe objects passing by the observer. The proximity of the moving object could be represented through the strength of the vibrators, the object's direction through the orientation of the rotation (4a and 4b) and the speed of the object through the speed that the activated vibrator is moving within a circle.

A first prototype with four actuators mounted on a flexible cotton head band was built(figure 4). This showed that displaying such patterns on the head is feasible.

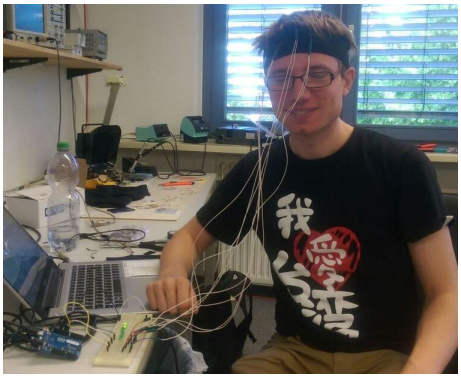


Fig. 4: The first test prototype containing four vibrating motors

2.3 A Possible Set-Up

Taking these considerations into account the following set-up is proposed. The user will wear a headband with around 30 actuators placed in three rows around their head. The actuators are fixed to the headband by weaving small flexible pockets onto the outside of the headband. The flexible cotton of the pockets presses the actuators towards the head, since the pocket is stretched more at the outside (see figure 5). For testing purposes it is also very easy to insert and remove the actuators from these pockets. In general the flexible cotton used in the first test proved to be very well suited for the headband as it fits on different head shapes and is air permeable so a wearer is less likely to sweat.

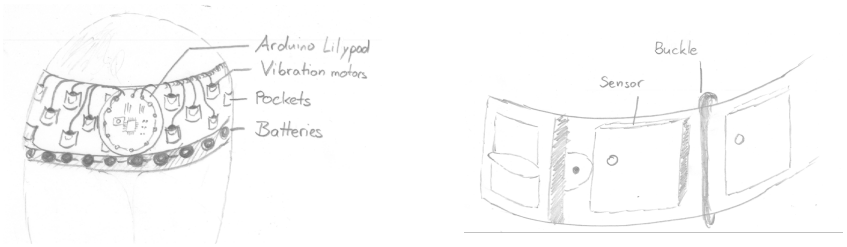
The actuators are connected to three thin Lilypad Arduinos[2] in groups of ten vibrators each (input restriction of this Arduino). Small coin cells weaved uniformly (for a good weight balancing) around at the bottom of the headband will power these Arduinos. In a later stage the headband with its attached actuators and thin Arduinos on top can be easily covered by a second layer of said cotton material. Such a prototype has a low social weight as it is hardly distinguishable from a regular headband e.g. worn by sportsmen as the cables etc. are covered.

The question what sensors will be used for the improved ProximityHat is subject of section 3. In this sample set-up we assume that they send their signals to an Arduino and can be placed at waist height. In this case all three LilyPads can be connected via cable along the wearers back to a main Arduino in a trouser pocket. (A wireless communication would also be possible.) The communication

[2] `https://www.arduino.cc/en/Main/ArduinoBoardLilyPad`

can easily be done with i2c[3]. The main Arduino will be powered by a 5V battery also located in the trouser pocket.

The sensory set-up would be directly connected to this main Arduino. In the case of e.g. Doppler sensors these would be fixed in equal distances around an "outer" belt. This outer belt has some 3D-printed buckles attached which will enable the user to attach it to a regular belt worn around the waist. This enables the wearer to quickly put it on and off. This set-up can be seen in Fig. 5



(a) Actuator Headband (b) Sensor Belt

Fig. 5: Sketches for a first prototype

3 Measuring Proximity and Velocity on Mobile Observers

The scenario for the ProximityHat requires measuring the proximity, velocity and direction of external objects locally. Sensors on mobile observers are mostly discussed in the field of robotics. When considering such works, one has to keep in mind that robots usually move smoother and thus the application of results from this field on a shaky walking person is limited.

There are two fundamental mechanisms for a device to gather data from the outside world without contact. A sensor can emit some sort of signal and collect data on its surroundings by analyzing how the signal is reflected back to the sensor (*active sensor*) or just measure incoming signals (*passive sensor*). These sensors are discussed in sections 3.1 and 3.2 respectively. Section 4 will give a sample implementation for a stereo camera (a passive sensor) and section 5 will finally conclude the findings.

3.1 Active Sensors

An active sensor emits a wireless signal which is then reflected by the object that the sensors gathers information on. Usually this signal is a wave with a particular wavelength. There are two types of waves, sonic waves that travel at the speed

[3] A sample i2c set-up: `http://www.instructables.com/id/I2C-between-Arduinos/`

of sound (approximately $0.343km/s$) and electromagnetic waves that travel at the speed of light (approximately $299,700km/s$). The wave types have different propagation behaviors depending on the surrounding (e.g. sonic waves travel well in water and electromagnetic waves in the infrared spectrum are blocked out by sun light). In order for an object to reflect a signal its wavelength has to be in the range of the object's size. Sonic sensors thus usually use *ultrasound* and electromagnetic ones *infrared* or *microwaves*. This section discusses two physical properties – time of flight (TOF) (section 3.1.1) and the Doppler effect (section 3.1.2) that theoretically can be observed in the reflection of any of these waves.

	Ultrasound	**Infrared Laser**	**Microwaves**
TOF for 6m	$35ms$	$0.00004ms$	$0.00004ms$
Range Coverage	about $30m$	$2m$, high end lasers up to $100m$	Long distance (50-1000m)
Beam characteristics	Wide angle Signal must be reflected from the object in a normal angle	Singular point (TOF cameras or LIDAR can be used) Doesn't work in light environments	Passes through thin objects (e.g. walls) Used for Doppler radars and motion tracking.

Table 1: Comparison of different wave types in sensors

3.1.1 Time of Flight Sensors

General Concept A popular method to detect proximity data about the surrounding of a mobile observer are *time of flight* (TOF) Sensors. By knowing the speed of the wave (in air) and measuring the time it takes for an emitted signal to come back to the sensor, one can calculate the distance between the sensor and the external object. By measuring the change of proximity one can calculate the velocity of that object in the line of the sensor. Time of flight sensors that operate with electromagnetic waves have to have high processing rates as the signal will travel very fast. For example travelling a path of 6 meters and back will take $0.00004ms$. On the other hand sonic sensors have fairly high latencies as in this case the signal will travel for $35ms$ [11].

Ultrasound sensors have a relatively large detection range of up to 30 meters. Yet their measurements are not very precise as the ultrasonic signal is usually emitted in a cone with a wide angle. As a consequence the external object's surface must be standing approx. perpendicular to the emitted beam as it is otherwise not reflected back to the sensor [12]. A problem that occurs due to the mostly piston-like radiation pattern of a sonic sensor is that unwanted reflections from e.g. a corner coming back to the sensor are falsifying the time of flight results [13]. High end devices can solve this problem by modulating the emitted frequency and thus giving each pulse an internal time stamp.

Let us assume that a car is moving at $50km/h$ tangentially to the observer on a street at a distance of about 5 meters (a likely distance to the curb) and that the observer has detection sensors positioned at an angle θ equally spaced

around the body (*angular resolution*). If a sensor can cover a range of r meters it means that for each measurement one has to wait twice this distance times the speed of the signal until a new measurement can be started in order to be sure that no old signal is coming back (*reaction time*) [11]. We will say the first sensor to detect the object is located at the angle θ_1, the second one at θ_2. In order for the system to register a movement both detectors have to have first detected that the object is there and then that it is gone. If we assume that they can emit signals simultaneously, it will still take at least twice the reaction time starting from the time the second sensor emits its signal until the movement can be quantified. During this time the car will have moved forward with $50km/h$. If we calculate this example for sonic sensors with a big cone, θ cannot be smaller than 30° [11], thus $\theta_1 = 60°$; $\theta_2 = 30°$. The reaction time for such a sensor with a range of 20 meters is physically at least $2 \cdot 20m/c_{sound} \approx 120ms$. In twice this reaction time the car will have moved about 3.3 meters and thus already past the perpendicular (figure 6 (b)). By including processing times etc. one receives a high latency.

Infrared waves are easily reflected by small dust particles etc. in the air, therefore they are usually used in the form of a concentrated laser with a very small beam. Usually their distance measurement reaches only about two meters. Some high-end lasers like the DMT10-2-1111[4] claim to be able to reach over 100 meters, but these use a special technique to modulate the amplitude of the laser[5]. As the laser has a small beam, *light detection and ranging* (LIDAR) sensors use a moving beam to scan the environment. So called time-of-flight cameras work with the same principle except that several beams are emitted at the same time, scanning a 2D image at once. These signals are distinguished by modulation, thus giving each a unambiguous signature [14]. A problem that all infrared based sensors face is that they cannot be reliably used under sun light.

Microwave sensors can be used with sunlight and have a large distance coverage of about 50-1000 meters [15]. Microwaves are only reflected by thicker objects and pass through walls. Low cost TOF sensors do not use them, however there are microwave Doppler sensors (section 3.1.2).

Time of Flight Applications There are several examples of **sonar** detectors being used in ETAs. One very influential example is the *NavBelt* developed by Borenstein in 1990. Eight ultrasonic sensors are placed around a belt and continuously detect the proximity of nearby objects. They then run an obstacle avoidance algorithm to calculate where the wearer should go next. This is then communicated to the wearer through acoustic commands [16].

Other systems like the *Wearable Obstacle Detection System* that places two sonar sensors on top of the wearer's shoulders and represents distance through a vibration motor on the shoulder [2] or the ProximityHat make likewise use of sonic distance sensors. The principal is also used commercially in the Ultracane[6],

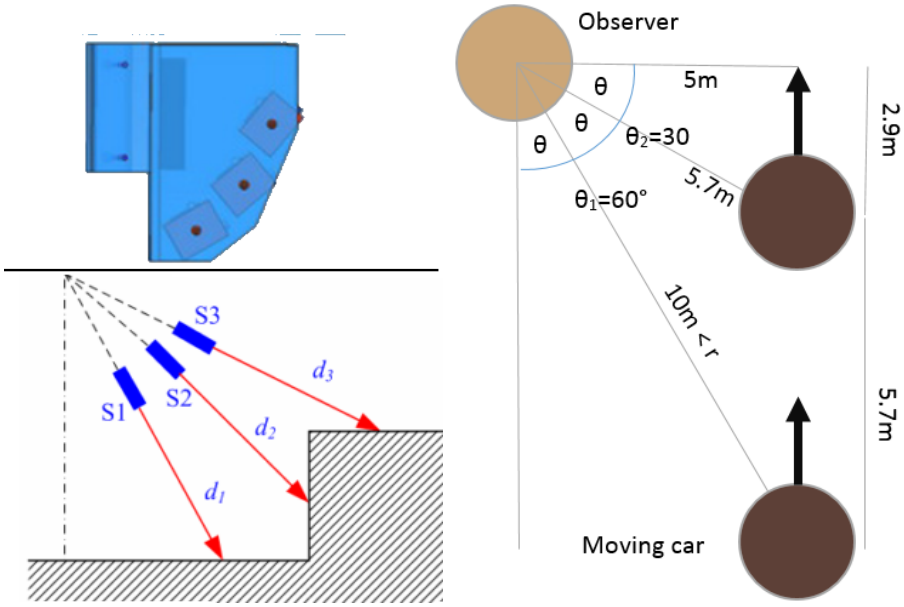
[4] `https://www.sick.com/us/en/product-portfolio/distance-sensors/long-range-distance-sensors/dmt/c/g168313`

[5] `https://www.bosch-professional.com/de/de/glm-100-c-34166-ocs-p/`

[6] `https://www.ultracane.com/`

a white cane that lets the user feel a vibration at the handle if the sonic sensors along the cane detect a close object.

There are not so many applications of **infrared** laser sensors in ETAs as one usually needs to detect objects at larger distances. It is very commonly used in mobile robotics though. An example of a pure infrared laser ETA is a system for floor-state estimation by Lee and Lee [17]. They mounted three laser sensors facing the floor at different angles to the wearer's belt. Thus they were able to reliably predict staircases in front of the user.



(a) Floor state estimation (b) A car passing an observer ($\theta = 30°$)

Fig. 6: Time of flight Principle

In contrast to ultrasound sensors, infrared laser sensors can provide proximity data fast and precise enough for creating a detailed depth map. For such a set-up the infrared laser has to be rotated (either through a motor or a moving mirror) to collect data at several continuous points. Such systems are called LIDAR (or sometimes LADAR). Sensors, like the one-dimensional RP Lidar[7] that scans the surrounding at a constant height level are commercially available at a low price and could be used with an Arduino. These circular LIDAR sensors have an angular resolution of $\theta = 1°$. But as only a one-dimensional line is scanned, an algorithm cannot reliably rule out that a movement was caused by a shaking observer. Wang et al. have solved the object tracking problem

[7] `http://www.slamtec.com/en/lidar`

(DATMO, introduced in section 3.2) for a high-end 2D LIDAR scanner (a TOF camera) mounted on a moving vehicle [18] and thus are able to measure the velocity of objects. This works on a shaking moving observer (human) as objects are segmented but it takes a lot of processing power.

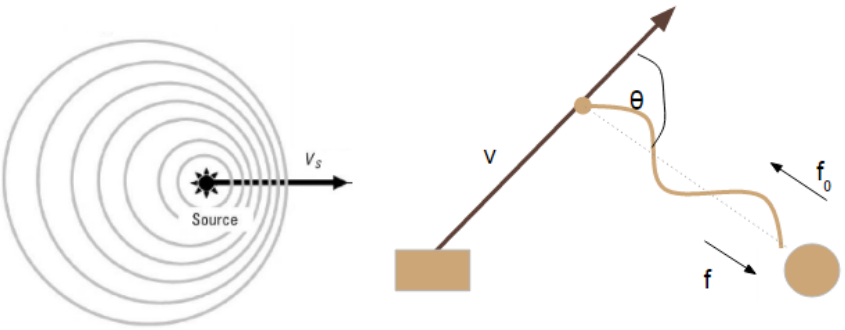
Summary From the calculations we can see that sonic proximity sensors are not suitable for velocity measuring on the ProximityHat due to their latency and not precise measurements. Laser infrared sensor could assist in distance measuring but only with a more expensive one- or two-dimensional long distance LIDAR one can gather reliable data on object movement and filter out shaking by the observer. But with sunlight exposure their results are not reliable.

3.1.2 Doppler Effect Sensors

General Concept The Doppler effect describes a wavelength shift that takes place when the emitting source of a signal is moving relative to the observer. If the source is moving away from (towards) the observer, the wave crests are pulled apart (pushed together) and thus the frequency received by the observer is shifted lower (higher) compared to the original signal. This frequency shift f_D taking place due to the Doppler effect can be described with the formula:

$$f_D = f_0 - f = \frac{f_0}{c} \cdot 2v \cdot \cos\theta$$

where f_0 is the original frequency emitted, f the received frequency at the observer, c the speed of wave (usually speed of light), v the velocity that the sender is moving relative to the observer and θ the angle between the wave and the path of the source relative to the observer [19]. (See figure 7)



(a) Shifted wave crests

(b) Relationship between the shifted Doppler frequency f_D and the velocity of the object

Fig. 7: The Doppler effect

A (stationary) Doppler sensor can measure the velocity of a moving external object by comparing the original frequency of the emitted wave and the received shifted signal. One can see from the formula that the angle at which the object is moving relative to the sensor plays a role. Only if the object is moving directly towards or away from the sensor the (absolute) value of the cosine term is 1 and thus cancels out. Otherwise the Doppler shift can only be used to determine the part of the object's velocity vector that is moving directly in the line of sight of the sensor. This is called *cosine effect* [20]. If one doesn't know the direction of a moving object, a Doppler velocity sensor cannot yield the speed of a moving object accurately. In part this problem can be solved if one also has the proximity data of the moving object to derive an estimate of the object's direction.

Regular low cost Doppler sensors use an operation mode called "continuous wave". In this mode the same wavelength is continuously emitted and relative differences of the frequency are reported. This doesn't allow proximity measurements by the previously described time of flight method [15]. There are other operation modes that work with pulsed signals (with different modulated frequencies) and allow the sensor to distinguish these pulses and thus measure the distance through time of flight. Yet all low-cost sensors use continuous wave [13].

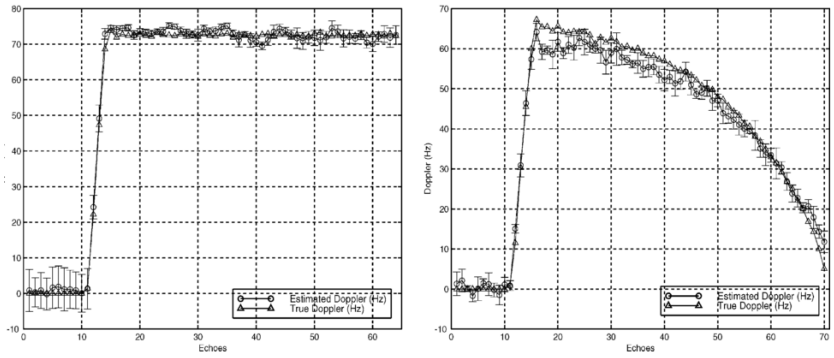


Fig. 8: The recorded Doppler shifts f_D over time (number of echoes sent) while a RoBat moves at a constant velocity orthogonal to a wall (left) or past a post positioned to one side of its path, showing the cosine effect (right). After echo 15 the RoBat is close enough to retrieve a signal.

Doppler Effect Applications There are few applications that make use of the Doppler effect in robotics [21] and thus far none in the field of ETAs. If the Doppler effect is used, it is just an additional data collected by a multi-purpose sensor. The *RoBat* is a mobile robot using a sonar sensor that was inspired by CF-FM bats. (These bats apply echolocation through a mix of constant-frequency (CF) and frequency-modulated (FM) calls to detect position and speed of surrounding objects (especially prey). The RoBat employs two sonar sensors to –

additionally to range information – detect velocity through the Doppler shift [13]. As the signal will reach one sensor slightly before the other one, the direction the signal is coming from can also be determined (*binaurality*). With this additional information the researchers developed an obstacle avoidance algorithm that incorporates predicted future positions of other robot. Due to the cosine effect this only works well, if the other robot is coming at a suitable angle close to 0°, that is in the line of the sensor. This can be seen in the left experiment of figure 8.

Summary From this we see that a Doppler sensor works well for detecting velocities coming towards the observer (Situations (b) and (c) in figure 2). But one cannot use this raw data, as it is not clear if the object is actually moving in the line of the sensor or at some angle to it. It might be possible to use other sensors to identify the object's direction.

3.2 Passive Sensors – Image Processing

A passive sensor is only receiving (but not emitting) signals of a certain type and processing them. For example a video camera detects all visible light coming to its lens, a thermal camera detects incoming infrared waves and a microphone does so with sound waves. In the low cost segment one generally uses webcams that have a high enough resolution to analyze the video stream with a suited algorithm. These are discussed in the remainder of this section but the algorithms would likewise work with expensive thermal cameras.

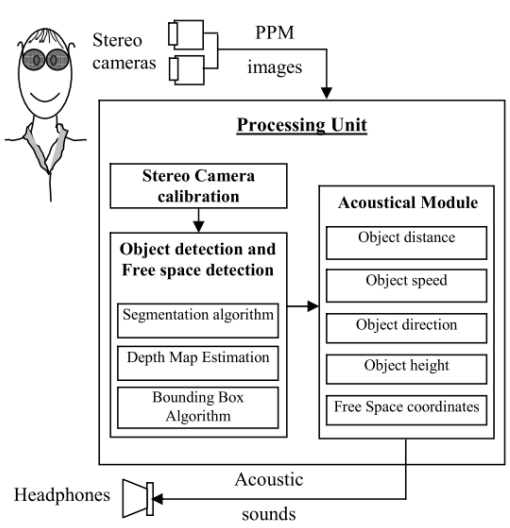
General Concept There are several applications that utilize cameras to detect motion of objects. A very basic method to detect if there is motion is to compare any two frames of a video stream and see if the pixel by pixel difference is above a threshold. But this only works if the camera is stable (a Gauß mixture distribution can help to neglect small movements). *Detection and Tracking of Moving Objects* (DATMO) algorithms usually want to find a desired object (e.g. a human, usually done through machine learning [22]) in an image and then track its motion. Since the important objects in the video stream are segmented, DATMO can give relative movements and thus also works if the camera is shaking.

With the technique *Detection by Tracking* one can then analyze where certain features of an object appear in the next camera frames and thus interpolate the line of movement of this object. With model based tracking it is possible to predict where an object is likely to appear next. The calculated hypothesis is then checked with reality. This has the advantage that objects can also be tracked if the object recognition doesn't work precisely enough or if an object is temporarily hidden (e.g. behind a light pole). For such operations one can use particle filters (most popular the Kayman filter). Such models are often used for predictable movement like that of planes or cars.

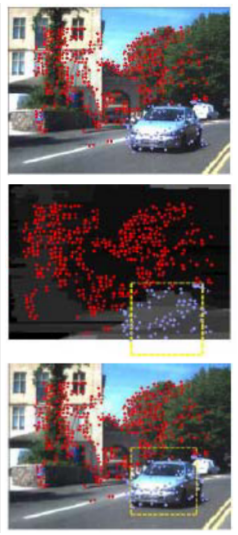
In order to get positions and velocities a stereo camera can be used. If an object is detected in both cameras one can – knowing the distance and orientation of the two cameras – triangulate its absolute position and velocity [23].

Image Processing Applications For many ETAs retrieving distance information with active sensors is enough as this is less complex than with passive ones. One the other hand with video stream processing one can also detect a variety of surrounding features (e.g. zebra crosswalks [24]).

Dunai et al. developed the *Real-time Assistance Prototype* to display the surrounding environment to a blind user through spatial hearing [25]. The prototype has a pair of stereo cameras mounted to a helmet that records the front of a wearer with a cone of 64°. An algorithm then segments the objects and creates a depth map. A Kanade-Lucas-Tomasi algorithm is used for tracking to find out how the bounding boxes of the identified objects are moving. The relevant information about one object is then transmitted to the wearer by an acoustic signal. For the wearer it seems as if the signal is coming from a certain direction as the acoustic signal is played slightly earlier on one speaker than the other. The sound signals are supposed to cover object distance, direction, height and speed as well as free space detection. With this device is possible to complete simple tasks like following a moving object. It is reported that the algorithm has problems if the objects are very close to the observer as no good depth map can be calculated. The authors also note that the system works far better if the observer is static and the objects are moving than vice versa. The concept of displaying objects through sound is already sold commercially with the device vOICe[8].



(a) Concept: Display object through sound



(b) Object Segmentation

Fig. 9: The Real-Time Assistance Prototype by Dunai et al.

[8] https://www.seeingwithsound.com/

Another ETA that uses stereo camera vision is a navigation aid built by Johnson and Higgens [26]. It consists of a stereo camera located at belt height and 14 vibration motors along the front part of the belt. For image processing the user is also wearing a backpack with a laptop. A 2D depth map is calculated based on the camera data and then divided into 14 vertical sections. Each vibrator is associated with one of these sections. The closest object in each section determines with what frequency the corresponding motor will vibrate (figure 10). Test runs with this system had the problem that the depth recognition algorithm didn't always work properly and that irrelevant close objects on the side were overriding more important information in the middle. Yet the system can be used well for orientation.

Fig. 10: Johnson and Higgins' navigation aid: the real situation (left), calculated depth map (middle), and the corresponding signals sent to the 14 vibration motors (right)

There are also some systems that combine camera image processing and time of flight sensors discussed in section 3.1.1 [27]. This has the advantage that it is less complex to measure the distance to an object with an active sensor while the camera can still be used to recognize other features.

Summary Generally the scenario for the ProximityHat can be solved with inexpensive webcams. It remains to be seen if the problems with determining the velocity of close objects is a problem. To solve this problem, it might be possible to additionally use TOF sensors. The only real drawback of image processing is that an Arduino is not capable of doing the calculations and more processing power is needed. Section 4 shows an implementation approach for this.

4 Image Processing with OpenCV

OpenCV is an open source library of computer vision functions running on different platforms such as Windows, Linux or Raspbian. In the scenario for the ProximityHat, a camera system should recognize the speed of objects in the near surrounding. As the displaying capabilities of vibration patterns are not very precise, it could be enough to determine speed and direction of the *fastest* object in a defined area of the camera. (One could also display the movement of the *closest* object or the *biggest* one.) In order to determine the velocity vector of a single object, the camera has to measure the velocity portion within the plane

```
coordinatesOfGoodFeatures = cv2.goodFeatuesToTrack(
        frame, mask, corners, maxCorners,
        qualityLevel, minDistance, blockSize
)
```

Code 1: goodFeaturesToTrack in OpenCV

of the camera chip and the one normal to that plane (thus of objects coming towards or going away form the camera). This is discussed in the remainder of this section.

4.1 Velocities within the Plane of the Camera

It is easy to measure velocities within the camera plane as one can do all calculation in a two dimensional model with just one camera. For the ProximityHat it could be enough to know that something is moving fast within an area without identifying the object. If so, one can ignore the process of object recognition in a frame. It is enough to determine how much a feature (belonging to an object) moved from one frame to the next. The fastest of these features will belong to the fastest object. This calculation is called *Dense Optical Flow*. An optical flow algorithm determines for a particular pixel of one frame which pixel position in the next frame (within a set circumference of the old position), if any, is the most likely place that the pixel moved to. This is based on a match of the neighboring pixels from the old frame in the new frame.

A standard example project from the OpenCV tutorial uses the Gunnar Farneback algorithm that tracks every nth pixel in a grid[9]. On an Intel i5-4200U CPU at 1.60 GHz it takes about 100ms for one frame of 640x480 pixels to process. If one assumes that the algorithm is to be executed on a much weaker machine such as a Raspberry Pi, this might take a lot of evaluation time.

But one doesn't need to calculate the optical flow for every pixel of the frame. OpenCV provides the method goodFeaturesToTrack[10] that filters suitable corners within a frame that can be very easily tracked by an optical flow algorithm [28]. Here one can also adjust, how easy to track and how far apart these corner points are supposed to be. For this sparse feature set one commonly uses the Lucas-Kanade *Sparse Optical Flow* algorithm[11] that determines the new position of a corner. The calculation times of this algorithm largely depend on how many good features are found in the image but with a number approximately equal to the points of the Farneback algorithm it is still about four times faster.

By calculating the optical flow in both directions and looking at the differences to the original points, one can filter out flows that could not be determined reliably (code 2). By looking at the differences in the provided and the newly

[9] `github.com/npinto/opencv/blob/master/samples/python2/opt_flow.py`
[10] `docs.opencv.org/2.4/modules/imgproc/doc/feature_detection.html`
[11] `docs.opencv.org/master/d7/d8b/tutorial_py_lucas_kanade.html`

```
calculatedCoordinates, st, err = cv2.calcOpticalFlowPyrLK(
        oldframe, newframe,
        coordinatesOfGoodFeatures, **lk_parameters
)                       #going backwards
coordinatesBackwards, st, err = cv2.calcOpticalFlowPyrLK(
        newFrame, oldframe,
        calculatedCoordinates, **lk_parameters
)                       #filtering out bad tracks
goodPoints = abs(coordinatesOfGoodFeatures -
        coordinatesBackwards) < threshhold
```

Code 2: Farneback Optical Flow in OpenCV

calculated coordinates one can easily calculate the speed in that 2D image. This does not represent the velocity in reality yet, as a single camera cannot perceive how close to the lens the movement is happening.

4.2 Velocities normal to the Plane of the Camera

To calculate the absolute velocity one needs the distance an object is away from the camera. This is also necessary to calculate velocities within the axis of the camera. This is usually done with a stereo vision camera. *Stereovision*[12] is a GNU licensed library based on OpenCV that provides several functionalities for 3D construction from stereo cameras. After one goes through the process of calibrating one's stereo camera, one can use images_to_pointcloud.py and create a live 3D point cloud from the images of the two cameras. This presumably takes a lot of calculating power but enables one to determine the distance from the camera and thus calculate the necessary absolutes speeds. The algorithm could be sped up if one only focuses on points that are interesting (possibly those that have a high relative velocity within the plane of the camera).

Generally OpenCV can be run on a Raspberry Pi; it will depend on the invested processing power if the algorithm can be executed fast enough. There is also a project[13] to implement stereo vision directly on a Raspberry Pi that could be used for the ProximityHat.

5 Conclusion

This work shows the difficulties that arise when measuring velocities on a mobile observer. Of the options shown stereo image processing seems to be the best solution as absolute velocities in all directions can be perceived. A work by Dunai et al. shows that our ProximityHat scenario is feasible with stereo cameras and

[12] `github.com/erget/StereoVision`

[13] `www.argondesign.com/case-studies/2014/oct/21/stereo-depth-perception-raspberry-pi/`

that there are only minor problems measuring objects in the range of up to five meters. The biggest drawback is that a high processing power is necessary for processing the algorithm, especially to determine the most dangerous velocities that are in the line of the camera (coming towards or going away from it).

One-dimensional LIDAR TOF sensors allow velocity detection in the line of the sensor and to some extent tangential to it. But because only a small range is covered, no object segmentation is possible which makes the measurements on a moving human unreliable. This could only be solved by using more expensive TOF cameras that will also need more processing power. Sonar TOF sensors as used in the current ProximityHat cannot be used for the proposed scenario. Doppler sensors don't need a lot of calculating power and can be connected to an Arduino but one faces the problem that the data effectively can only be used in combination with another sensor.

For some scenarios a LIDAR and a Doppler sensor could work but it is likely that an image processing solution as shown in section 4 will provide the best results for displaying velocities through a tactile display based on vibration. For the final decision which set-up to use and where to position the sensors it has to be decided how to chose which object's movement (the fastest, the closest, the one on collision course, etc.) shall be displayed. For this it is necessary to do further tests what is the most intuitive and the most helpful for the users.

References

1. D. Dakopoulos and N. G. Bourbakis, "Wearable Obstacle Avoidance Electronic Travel Aids for Blind: A Survey," *APPLICATIONS AND REVIEWS*, vol. 40, no. 1, 2010.
2. S. Cardin, D. Thalmann, and F. Vexo, "Wearable Obstacle Detection System for visually impaired People," *VR Workshop on Haptic and Tactile Perception of Deformable Objects*, pp. 50–55, 2005.
3. R. Farcy, R. Leroux, A. Jucha, R. Damaschini, C. Grégoire, and A. Zogaghi, "Electronic Travel Aids and Eletronic Orientation Aids," in *Conference & Workshop on Assistive Technologies for People with Vision & Hearing Impairments*, 2006.
4. D. Bin, Y. Haitao, L. Jiang, and Z. Xiaoning, "The research on blind navigation system based on RFID," *2007 International Conference on Wireless Communications, Networking and Mobile Computing*, no. 070416277, pp. 2058–2061, 2007.
5. K. Tsukada and M. Yasumura, "ActiveBelt : Belt-Type Wearable Tactile Display," pp. 384–399, 2004.
6. M. Berning, F. Braun, T. Riedel, and M. Beigl, "ProximityHat," *Proceedings of the 2015 ACM International Symposium on Wearable Computers - ISWC '15*, no. April 2016, pp. 31–38, 2015.
7. A. Toney, B. Mulley, B. H. Thomas, and W. Piekarski, "Social weight: Designing to minimise the social consequences arising from technology use by the mobile professional," *Personal and Ubiquitous Computing*, vol. 7, no. 5, pp. 309–320, 2003.
8. H. Z. Tan and A. Pentland, "Tactual Displays for Sensory Substitution and Wearable Computers," *Fundamentals of Wearable Computers and Augmented Reality*, pp. 579–598, 2001.
9. K. Myles, "Vibrotactile Sensitivity of the Head," *Engineering*, no. January, 2009.

10. M. K. Dobrzynski, S. Mejri, and S. Wischmann, "Quantifying information transfer through a head-attached vibrotactile display: Principles for design and control," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 7, pp. 2011–2018, 2012.
11. G. Benet, F. Blanes, J. Simó, and P. Pérez, "Using infrared sensors for distance measurement in mobile robots," *Robotics and Autonomous Systems*, vol. 40, no. 4, pp. 255–266, 2002.
12. T. Mohammad, "Using Ultrasonic and Infrared Sensors for Distance Measurement," *Academy of Science, engineering and Technology*, vol. 3, no. 3, pp. 273–278, 2009.
13. J. M. Carmena and J. C. T. Hallam, "The use of Doppler in Sonar-based mobile robot navigation: Inspirations from biology," *Information Sciences*, vol. 161, no. 1-2, pp. 71–94, 2004.
14. F. Chiabrando, R. Chiabrando, D. Piatti, and F. Rinaudo, "Sensors for 3D Imaging: Metric Evaluation and Calibration of a CCD/CMOS Time-of-Flight Camera.," *Sensors (Basel, Switzerland)*, vol. 9, pp. 10080–96, jan 2009.
15. J. Fang, H. Meng, H. Zhang, and X. Wang, "A low-cost vehicle detection and classification system based on unmodulated continuous-wave radar," *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pp. 715–720, 2007.
16. S. Shoval, J. Borenstein, and Y. Koren, "Mobile robot obstacle avoidance in a computerized travel aid for\nthe blind," *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pp. 2023–2028, 1994.
17. M. Lee and S. Lee, "Design and analysis of an infrared range sensor system for floor-state estimation," *Journal of Mechanical Science and Technology*, vol. 25, no. 4, pp. 1043–1050, 2011.
18. C. C. Wang, C. Thorpe, and A. Suppe, "LADAR-based detection and tracking of moving objects from a ground vehicle at high speeds," *IEEE Intelligent Vehicles Symposium, Proceedings*, pp. 416–421, 2003.
19. V. Chen, *The Micro-Doppler Effect in Radar*. Artech House, 2011.
20. Charvat, *Small and Short Range Radar Systems*. CRC Press, 2014.
21. I. Amundson, X. Koutsoukos, and J. Sallai, "Mobile sensor localization and navigation using RF doppler shifts," *Proc. of the first ACM intl. workshop on Mobile entity localization and tracking in GPS-less environments*, pp. 97–102, 2008.
22. T. Lindeberg, "Edge detection and ridge detection with automatic scale selection," *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR '96, 1996 IEEE Computer Society Conference on*, vol. 30, no. 2, pp. 465–470, 1996.
23. N. Slesareva, A. Bruhn, and J. Weickert, "Optic Flow Goes Stereo: a Variational Method for Estimating Discontinuity-Preserving Dense Disparity Maps," *Dagm*, vol. 42, pp. 267–282, 2009.
24. M. Uddin and T. Shioyama, "Bipolarity and Projective Invariant-Based Zebra-Crossing Detection for the Visually Impaired," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 22–22, 2005.
25. L. Dunai, G. P. Fajarnes, V. S. Praderas, B. D. Garcia, and I. L. Lengua, "Real-time assistance prototype- A new navigation aid for blind people," *IECON Proceedings (Industrial Electronics Conference)*, pp. 1173–1178, 2010.
26. L. A. Johnson and C. M. Higgins, "A navigation aid for the blind using tactile-visual sensory substitution," *Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings*, pp. 6289–6292, 2006.
27. A. Kumar, R. Patra, M. Manjunatha, J. Mukhopadhyay, and A. K. Majumdar, "An electronic travel aid for navigation of visually impaired persons," *2011 3rd Intl Conference on Communication Systems and Networks, COMSNETS*, 2011.
28. Jianbo Shi and C. Tomasi, "Good features to track," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition CVPR-94*, pp. 593–600, 1994.