

Karlsruhe Reports in Informatics 2014,7

Edited by Karlsruhe Institute of Technology,
Faculty of Informatics
ISSN 2190-4782

**Ubiquitäre Systeme (Seminar)
und
Mobile Computing (Proseminar)
WS 2013/14**

Mobile und Verteilte Systeme
Ubiquitous Computing

Teil X

Herausgeber:
Martin Alexander Neumann, Anja Bachmann,
Matthias Berning, Markus Scholz, Yong Ding, Matthias Budde

2014



Fakultät für **Informatik**

Please note:

This Report has been published on the Internet under the following
Creative Commons License:

<http://creativecommons.org/licenses/by-nc-nd/3.0/de>.

**Ubiquitäre Systeme (Seminar)
und
Mobile Computing (Proseminar)
WS 2013/14**

**Mobile und Verteilte Systeme
Ubiquitous Computing
Teil X**

Herausgeber

Martin Alexander Neumann

Anja Bachmann

Matthias Berning

Markus Scholz

Yong Ding

Matthias Budde

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

**Interner Bericht 04/2014
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 Wintersemesters 2013/2014 zusammen. Die Themenvielfalt der hier zusammengetragenen Aufsätze umfasst Mobile Augmented Reality, sowie Anwendungen von Data Mining in Ubiquitären Systemen, und Analyse von Methoden zum 3D-Körperscan. Wir danken den Studierenden für ihren besonderen Einsatz, sowohl während des Seminars als auch bei der Fertigstellung dieses Bandes.

Karlsruhe, den 2. April 2014

Martin Alexander Neumann
Anja Bachmann
Matthias Berning
Markus Scholz
Yong Ding
Matthias Budde

Inhaltsverzeichnis

<i>Alina Saalfeld</i> Service Dependency Discovery in the Web	1
<i>Adrian Hoppe</i> Übersicht zu expliziter Interaktion in Intelligenten Umgebungen durch Gesten ..	16
<i>David Timmermann</i> Knotenmobilität in industriellen drahtlosen Sensornetzwerken	33
<i>Christoph Klebsattel</i> Privatsphäre und Datenschutz beim Participatory Sensing	50
<i>Johannes Bleier</i> Power System Analysis	67
<i>Philipp Schmurr</i> Sensor Emulation with Machine Learning Methods	81
<i>David Vogelbacher</i> Methoden der Anomalieerkennung mit besonderem Hinblick auf Sensornetzwerke	95
<i>Christopher Bazant</i> Literaturüberblick über Methoden zur expliziten Interaktion in intelligenten Umgebungen	111
<i>Robert Zetsche</i> Zugriff auf lokale und verteilte Daten im Kontext der Android Plattform	157

<i>Alexandru Albu</i>	
Overview over current Radio-based, Device-Free Context Recognition Research	186
<i>Nikolai Hartmann</i>	
Einfluss von Fenstergröße, Überlappung der Fenster und Abtastrate bei funkbasierter gerätefreier Aktivitätserkennung	202
<i>Frederik Schöll</i>	
Gamification for Environmental Sensing	221

Service Dependency Discovery in the Web

Alina Saalfeld*

Betreuer: Martin Alexander Neumann[†]

Karlsruher Institut für Technologie (KIT)
Pervasive Computing Systems – TECO

*uadnv@student.kit.edu

[†]mneumann@teco.edu

Zusammenfassung. Dependency Discovery ist ein Forschungsgebiet, mit dem sich in der Vergangenheit bereits viele Arbeiten auseinandergesetzt haben. All diese Arbeiten beschäftigen sich jedoch nur damit, Abhängigkeiten in lokalen Netzen wie Universitäts- oder Firmennetzen zu erkennen. Ob die erdachten Algorithmen auch im Internet anwendbar wären, wird von diesen Arbeiten nicht thematisiert.

In dieser Arbeit werden vier dieser Algorithmen zur Erkennung von Abhängigkeiten zwischen Diensten in einem Netzwerk betrachtet und darauf untersucht, ob sie sich sinnvoll im Internet einsetzen ließen. Diese Algorithmen sind Sherlock, Orion, NSDMiner und Rippler. All diese Ansätze sind nicht an ein bestimmtes Betriebssystem gebunden oder benötigen Anpassung für die untersuchten Dienste oder an diesen, welches eine Grundvoraussetzung für ihre Anwendung in einem so heterogenen Netzwerk wie dem Internet ist.

Diese Arbeit zeigt auf, welche Schwächen die Algorithmen besitzen und wo Stärken liegen. Sie zeigt dabei, dass keiner der bisher bekannten Algorithmen es wirklich schafft, eine gute Abhängigkeitserkennung im Internet zu gewährleisten. Zwar scheint Orion die beste Wahl dafür zu sein, aber auch dieser Ansatz hat im Internet Schwächen.

Abschließend wird darauf eingegangen, welche Forschungsbereiche zum Thema Dependency Discovery im Internet interessant sind und neue Erkenntnisse einbringen könnten.

1 Einleitung

In digitalen Netzwerken jeder Art werden heutzutage verschiedenste Dienste und Anwendungen angeboten und genutzt. Sind diese Netzwerke von privater Natur, so sind sie meist recht gut überschaubar und Abhängigkeiten zwischen diesen Diensten meist leicht ersichtlich. Werden in dem Netzwerk jedoch viele verschiedene Dienste angeboten, wie in vielen staatlichen oder Unternehmensnetzwerken, so ist es keine triviale Aufgabe mehr, diese Abhängigkeiten zu erkennen. Das Forschungsgebiet der *(Service) Dependency Discovery* (auch *Dependency Mining* oder zu deutsch *Abhängigkeitserkennung*) beschäftigt sich mit diesem Problem.

In dieser Arbeit wird zunächst auf den Zweck von Dependency Discovery eingegangen. Es wird die Frage beantwortet, warum die Kenntnis über Abhängigkeiten von Diensten in einem Netzwerk nützlich ist und welchen praktischen Nutzen dieses Wissen hat.

Es werden anschließend verschiedene Ansätze, die bereits entwickelt wurden, vorgestellt. Ein Hauptaugenmerk liegt hierbei auf den Arbeiten von Sherlock [3], Orion [7], NSDMiner [9] und Rippler [10], aber auch andere Ansätze werden kurz behandelt.

Danach werden diese vier großen Lösungen miteinander verglichen. Bei diesem Vergleich fließt noch nicht ihre Anwendbarkeit im Internet mit ein.

Die vier genannten, bereits entwickelten Ansätze behandeln in ihren Arbeiten alle lediglich die Verwendung der Tools in Unternehmens- oder ähnlich großen Netzwerken. Diese Arbeit untersucht, ob sich die Ansätze auf das Internet als Netzwerk übertragen ließen und welche Maßnahmen dafür zu treffen wären. Dies passiert im nächsten Abschnitt der Arbeit. Es wird dabei analysiert, wie welche Ansätze unter den Eigenheiten des Internets leiden.

Abschließend folgt ein Fazit, welches die Ergebnisse dieser Arbeit zusammenfasst. Hier werden außerdem einige Ansätze für weiterführende Forschungsarbeiten genannt.

2 Verwendungszwecke von Dependency Discovery

Es gibt verschiedene Gründe, warum die Kenntnis von Abhängigkeiten zwischen Diensten nützlich ist. Im Groben lassen sich diese Gründe in Fehlererkennung und Fehlervorbeugung unterteilen. „Fehler“ ist hierbei allerdings sehr weit zu fassen, da auch Performanceeinbußen bereits als Fehler zu verstehen sind.

Bei der Fehlererkennung geht es darum, herauszufinden, warum es in einem Netzwerk Probleme gibt. Wenn beispielsweise ein Webserver nicht mehr korrekt arbeitet, könnte dies statt am Webserver selbst vielleicht auch am DNS-Server oder an der Datenbank, auf die der Webserver zurückgreift, liegen.

Oftmals ist Fehlererkennung aber nicht ganz so einfach. In einem Netzwerk mit mehreren Dutzend oder noch mehr Systemen und ebenso vielen angebotenen Diensten fällt es ohne Kenntnis über die Abhängigkeiten zwischen den Diensten schwer, die Fehlerquelle ausfindig zu machen. Kennt man jedoch die Abhängigkeiten, kann man die Fehlerquelle eingrenzen und leichter finden. Es kann sogar möglich sein, weitere Dienste, die von dem Fehler auch betroffen sind, vorherzusagen.

Letzteres ist auch in der Fehlervorbeugung relevant. Sind die Abhängigkeiten innerhalb eines Netzwerks bekannt, kann vorhergesagt werden, welche Dienste betroffen sind, wenn ein anderer Dienst ausfällt. Für diesen Fall können Backup-Strategien geplant werden, um so dem Fehlerfall vorzubeugen.

Diese Gründe scheinen zunächst einmal vor allem in kleineren Netzwerken als dem Internet relevant zu sein, beispielsweise in Netzwerken einer Firma oder einer öffentlichen Einrichtung. Relevant werden sie jedoch durch das Internet of Things auch im Internet selbst. Im bisherigen Internet waren Abhängigkeiten

oft offensichtlich: Wird eine Website aufgerufen, wird hierfür Zugriff auf einen DNS-Server benötigt. Hat der Server eine Datenbankanbindung, so fällt die Abhängigkeit zur Datenbank wieder in das Netzwerk des Serverbesitzers, spielt sich also wieder nur in einem kleineren Netzwerk ab, welches vom Internet direkt aus oft kaum erreichbar ist. Durch das Internet of Things geht diese Übersicht aber verloren. Wie unter anderem [2] aufzeigt, stehen die Dinge, die nun über das Internet kommunizieren, komplex miteinander in Verbindung. Es ist offensichtlich, dass in einem derart komplexen und stark verknüpften Netzwerk die Erkennung von Abhängigkeiten nicht mehr durch bloßes Hinsehen möglich ist.

Doch auch im klassischen Internet ist Dependency Discovery interessant. Erstreckt sich das Intranet einer Firma über mehrere Standorte, die räumlich weit voneinander getrennt sind, so werden diese durch das Internet miteinander verbunden. Zwar handelt es sich hierbei nur um eine Untermenge des Internets, die in sich stark verknüpft ist, jedoch unterliegt die Verbindung zwischen den Standorten den Komplikationen, die das Internet mit sich bringt: Langsamere Verbindungen und schwankende Reaktionszeiten spielen hierbei eine wichtige Rolle. Natürlich sind Fehlererkennung und -vorbeugung auch in solchen verteilten Netzen interessant.

3 Bisherige Ansätze

In der Literatur sind im Wesentlichen vier Ansätze relevant. Es handelt sich dabei um Sherlock [3], Orion [7], NSDMiner [9] und der im letzten Jahr erschienene Algorithmus Rippler [10]. Diese Ansätze werden im Folgenden noch detaillierter umrissen. Da dies natürlich nicht die einzigen Ansätze sind, sollen in diesem Abschnitt zunächst weitere bekannte Ansätze kurz beschrieben und begründet werden, warum sie in dieser Arbeit nicht näher betrachtet werden.

Pinpoint, X-Trace und ADD sind Ansätze, die auf ein Netzwerk zugeschnitten werden müssen. Pinpoint modifiziert dabei einen bestimmten Webserver, sodass allen HTTP-Zugriffen auf diesen eine ID zugewiesen wird, die dieser an andere Dienste, die er aufruft, weiterleitet [6]. X-Trace modifiziert die in einem Netzwerk verwendeten Protokolle, um mit diesen Metadaten zu verschicken [8]. ADD erzeugt absichtlich Last auf einzelnen Diensten des Netzwerks um zu untersuchen, wie dadurch andere Dienste beeinflusst werden. Dafür benötigt es selbstverständlich Kenntnisse darüber, wie auf dem betrachteten Dienst diese Last erzeugt werden kann [5].

Da alle diese Ansätze speziell für das überwachte Netzwerk zugeschnitten werden müssen, sind sie im Web nicht sinnvoll verwendbar und deshalb für nähere Betrachtungen irrelevant.

Magpie untersucht auf Kernebene Events, um zu erkennen, auf welche anderen Dienste ein Dienst zugreift [4]. Dieser Algorithmus bietet zwar Grundlagen zur Erkennung von möglichen Abhängigkeiten, bewerten jedoch nicht, ob es tatsächliche Abhängigkeiten sind.

Project5 besitzt einen ähnlichen Ansatz wie NSDMiner, bei dem nach Verbindungen gesucht wird, die in andere Verbindungen eingebettet sind [1]. Aufgrund

der hohen Ähnlichkeit der Algorithmen wird Project5 hier nicht gesondert betrachtet.

3.1 Verwendete Begriffe

Eine Reihe von Begriffen spielt bei der Dependency Discovery eine wichtige Rolle. Diese Begriffe sollen in diesem Abschnitt zunächst geklärt werden, bevor die von dieser Arbeit betrachteten Ansätze erklärt werden.

Ein Dienst ist definiert als eine Schnittstelle, über die ein Client über ein Netzwerk Funktionen aufrufen kann. Dieser Client kann dabei selbst ein Dienst sein. Dienste werden in dieser Arbeit oft mit einem Buchstaben, meist A oder B, angegeben, um einen beliebigen, aber eindeutig identifizierten Dienst zu repräsentieren. Beispiele für Dienste sind ein Webserver, ein DNS-Dienst oder eine Datenbank.

Viele Ansätze ([7], [9], [10]) unterscheiden zwischen Local-Remote-Abhängigkeit (LR-Abhängigkeit) und Remote-Remote-Abhängigkeit (RR-Abhängigkeit). Wenn ein Dienst A um seine Arbeit erfolgreich auszuführen auf einen Dienst B zugreifen muss, sagen wir *Dienst A ist von Dienst B abhängig* und es handelt sich um eine LR-Abhängigkeit. Ein Beispiel für diese Art der Abhängigkeit ist ein Webserver (Dienst A), der zum Ausliefern einer Website von einer Datenbank (Dienst B) abhängt. Muss ein Client um einen Dienst A zu verwenden zunächst auf einen Dienst B zugreifen, sagen wir ebenfalls *Dienst A ist von Dienst B abhängig*, hierbei handelt es sich jedoch um eine RR-Abhängigkeit. Auch hierfür kann ein Beispiel gefunden werden, bei dem ein Webserver involviert ist: Versucht ein Client auf einen Webserver (Dienst A) zuzugreifen, so muss er zunächst einen DNS-Dienst (Dienst B) nach dessen IP-Adresse fragen. Der Webserver ist also vom DNS-Dienst abhängig.

Zu beachten ist, dass diese Abhängigkeiten nicht notwendigerweise bei jedem Aufruf eines Dienstes bestehen. Beispielsweise könnte der Webserver Caching betreiben, um nicht bei jedem Zugriff auf die Datenbank zuzugreifen. Dies ändert jedoch nichts an der Tatsache, dass die Abhängigkeit existiert. Anders ausgedrückt: Dienst A ist nur dann von Dienst B unabhängig, wenn Dienst A nie auf Dienst B angewiesen ist, um korrekt zu funktionieren. Dies gilt sowohl für LR- als auch für RR-Abhängigkeiten.

Dienstpaare sind Paare von zwei Diensten ($A \rightarrow B$). Diese Paare sind gerichtet. Das bedeutet, dass das Paar ($A \rightarrow B$) ein anderes Paar als das Paar ($B \rightarrow A$) ist. Zumeist werden diese Dienstpaare genutzt, um festzustellen, ob eine Abhängigkeit besteht oder nicht. Besteht bei einem Dienstpaar ($A \rightarrow B$) eine Abhängigkeit, so ist A von B abhängig.

Die Ergebnisse der Dependency Discovery werden im Regelfall in zwei Kategorien eingeteilt: Positives sind Dienstpaare ($A \rightarrow B$), bei denen A von B abhängt, Negatives sind Dienstpaare ($A \rightarrow B$), bei denen A nicht von B abhängt. Diese Kategorien können weiterhin jeweils in True und False eingeteilt werden: Hierbei sind True Positives Dienstpaare ($A \rightarrow B$), bei denen korrekt erkannt wurde, dass A von B abhängt, während ein False Positive ein Dienstpaar ist, das fälschlicherweise als Positive erkannt wurde, bei dem also keine tatsächliche

Abhängig von A von B existiert. True Negatives sind folglich Dienstaare, die korrekt als nicht-abhängig erkannt wurden, während False Negatives eigentlich eine existierende Abhängigkeit darstellen, die aber nicht erkannt wurde.

3.2 Sherlock

Der einfachste Ansatz, der im Rahmen dieser Arbeit betrachtet wird, ist Sherlock [3]. Sherlock überprüft, ob der zeitliche Abstand zwischen den Aufrufen eines Dienstes A und eines Dienstes B meist kleiner ist als ein gewisser Schwellwert.

Um Sherlock nutzen zu können, ist es nötig, den Netzwerkverkehr von möglichst vielen Clients im Netzwerk aufzuzeichnen, da die Wahrscheinlichkeiten für eine Abhängigkeit aus Sicht eines jeden Clients bestimmt werden. Erst danach werden dann Clients, die als ähnlich zu betrachten sind, zusammengefasst.

Sherlock überprüft die relative Häufigkeit, mit der Dienst B höchstens 10 ms vor Dienst A aufgerufen wird, und vergleicht diese mit der sogenannten „chance of co-occurrence“. Für diese wird über alle Messungen die durchschnittliche Zeit I zwischen Zugriffen auf denselben Dienst berechnet und anschließend 10ms durch I geteilt. Ist die relative Häufigkeit nun „viel größer“ als die chance of co-occurrence (wobei in [3] nicht genauer spezifiziert wird, was genau dies bedeutet), dann wird Dienst A als abhängig von Dienst B angenommen. All diese Annahmen finden dabei aus Sicht eines festen Clients statt.

Die Ergebnisse dieser Berechnungen werden periodisch an eine Inference Engine gesendet, die die Ergebnisse auswertet. Hierbei werden aus den relativen Wahrscheinlichkeiten, die ein Dienstaar auf den verschiedenen Clients für eine Abhängigkeit besitzt, der Durchschnitt und die Standardabweichung berechnet. Weicht die Wahrscheinlichkeit, die ein Client berechnet hat, mehr als das fünf-fache der Standardabweichung vom Durchschnitt ab, so wird sein Ergebnis bei der Berechnung der Endwahrscheinlichkeiten nicht beachtet. In [3] wird nicht erläutert, wie diese Endwahrscheinlichkeiten berechnet werden. [3]

3.3 Orion

In [7] wird ein Algorithmus zur Dependency Discovery vorgestellt, der den Namen Orion trägt. Orion überprüft, ob der zeitliche Abstand zwischen dem Aufruf eines Dienstes A und dem Aufruf eines Dienstes B immer ähnlich groß ist, beziehungsweise, in einem Histogramm dargestellt, gut sichtbare Spitzen zeigt. Diesem Verfahren liegt die Vermutung zu Grunde, dass in einem Dienst wiederkehrende Ausführungspfade vorhanden sind, die immer in etwa die selbe Zeit zum Ausführen benötigen.

Orion wird entweder auf jedem Host und Client in einem Netzwerk installiert oder arbeitet mit Packet Sniffern, die auf Routern laufen, die möglichst nahe an den zu beobachtenden Hosts stehen. Aus der Kommunikation zu und von diesen Hosts kann der Algorithmus dann die Abhängigkeiten ermitteln.

Orion untersucht jeweils einen Zugriff auf einen Dienst A darauf, auf welche anderen Dienste innerhalb der nächsten 3 Sekunden zugegriffen wird. Jeder

Dienst B, der in diesem Intervall gefunden wird, wird als Hinweis auf entweder eine mögliche LR-Dependency ($A \rightarrow B$) oder eine mögliche RR-Dependency ($B \rightarrow A$) aufgefasst. Es wird dabei immer gespeichert, mit welcher Verzögerung Dienst B nach Dienst A aufgerufen wurde.

Die gewonnenen Daten eines Dienstpaars werden dann in einem Histogramm gespeichert. Die Klassenbreite entspricht dabei 10ms, die Höhe der Anzahl der Delay-Samples, die in die Klasse fallen. Um Rauschen im Histogramm zu unterdrücken, wird nun zunächst eine schnelle Fouriertransformation durchgeführt und dann ein Kaiser Window als low-pass Filter angewendet.

Anschließend werden im entstandenen Histogramm Spitzen gesucht. Eine Spitze wird in diesem Algorithmus dadurch definiert, dass sie über der Geraden vom Durchschnittswert der Klassenhöhen addiert mit dem k -fachen der Standardabweichung liegen. k ist hierbei so zu wählen, dass die Spitzen zu den 0,1% bis 1% der höchsten Klassenhöhen gehören. In [7] wird daher $k=3$ vorgeschlagen, da hiermit die Spitzen zu den 0,3% der höchsten Höhen zählen.

Werden eine oder mehrere Spitzen gefunden, wird angenommen, dass das untersuchte Dienstpaar eine Abhängigkeit darstellt. In [7] wird nicht darauf eingegangen, ob an dieser Stelle noch LR- und RR-Abhängigkeiten unterschieden werden (können). [7]

3.4 NSDMiner

Der Algorithmus NSDMiner wird in [9] vorgestellt. NSDMiner nutzt aus, dass ein Dienst A, der von einem Dienst B LR-abhängt, oft mit Dienst B kommuniziert, bevor er einem aufrufenden Client antwortet. Dieser Algorithmus kann jedoch nur LR-Abhängigkeiten erkennen.

NSDMiner braucht von allen Hosts im Netzwerk einen Mitschnitt der Netzwerkkommunikation, um diesen auszuwerten. Wie auch bei Orion kann daher entweder auf den Hosts selbst hierfür Software installiert werden, oder sie wird auf Routern installiert, die diesen Netzwerkverkehr verwalten.

Für NSDMiner sind Netzwerkkommunikationsflüsse von zentraler Bedeutung. Während eine TCP-Verbindung aufgrund ihres Designs einen festen Anfang und ein festes Ende besitzt, ist die Ermittlung von Anfang und Ende einer UDP-Verbindung schwieriger. NSDMiner geht daher folgendermaßen vor: Wird ein neues UDP-Paket gefunden, so wird geprüft, ob es ein anderes Paket mit dem selben Ziel und der selben Quelle gibt, welches vor dem neuen Paket geprüft wurde. Gibt es eines und ist dieses nicht zu alt, wird das neue Paket zur Verbindung des alten Pakets hinzugefügt. Wird kein passendes Paket gefunden, so begründet das neue Paket eine neue Verbindung.

NSDMiner untersucht nach dem Zusammenstellen dieser Verbindungen, welche von anderen komplett umschlossen werden. Dafür wird für jede Verbindung, die von einem Host A ausgeht, überprüft, ob es eine Verbindung gibt, die vorher begann, später endet und als Ziel Dienst A (auf Host A) besitzt. Ist dies der Fall, wird dies als Indiz dafür gewertet, dass Dienst A von einem Dienst B abhängt, der das Ziel der inneren Verbindung ist.

Dieses Indiz wird folgendermaßen gewichtet: Gibt es nur eine Verbindung, die die innere Verbindung umfasst, wird der sogenannte dweight-Wert vom Dienstpaar ($A \rightarrow B$) um eins erhöht. Gibt es mehrere Verbindungen, die in Frage kommen, gibt es zwei Möglichkeiten. Im sogenannten „shared“-Modus wird der Punkt auf alle denkbaren Dienstpaare gleichmäßig verteilt. Gibt es also sowohl eine Verbindung zu Dienst A als auch zu Dienst B, die eine Verbindung zu Dienst C umschließen, dann wird $dweight(A \rightarrow C)$ und $dweight(B \rightarrow C)$ um jeweils 0,5 erhöht. Im „exclusive“-Modus werden stattdessen keine dweight-Werte erhöht.

Ob ein Dienstpaar ($A \rightarrow B$) von NSDMiner als Abhängigkeit erkannt wird, hängt vom Verhältnis des zugehörigen dweight zu den Aufrufen von Dienst A ab. Übersteigt dieses Verhältnis eine Schwelle α , die von [9] als $< 0,5\%$ empfohlen wird, gilt ($A \rightarrow B$) als Abhängigkeit. [9]

3.5 Rippler

Rippler, welches in [10] vorgestellt wird, verfolgt einen völlig anderen Ansatz als andere Ansätze zur Dependency Discovery: Es werden Pakete künstlich verzögert und gemessen, ob dies auf die Aufrufe anderer Dienste Einfluss nimmt. Hierbei spricht man vom „Watermarking“.

Um diesen Ansatz umsetzen zu können, muss Rippler dazu in der Lage sein, alle Pakete an oder von untersuchten Diensten zu verzögern. Der empfohlene Ansatz dafür ist es, allen Netzwerkverkehr von den Routern des Netzwerks zunächst zu einer zentralen Maschine leiten zu lassen, die sie dann an die korrekten Ziele weiterleitet. Diese zentrale Maschine entscheidet, ob die jeweiligen Pakete verzögert werden. Alternativ kann auf jedem untersuchten Host ein solcher Verzögerer laufen, allerdings müssen hierbei die Uhren aller Hosts möglichst genau aufeinander abgestimmt sein, da dies kritisch für den Algorithmus ist. Außerdem muss, möglichst nahe an den zu untersuchenden Diensten, der Netzwerkverkehr mitgeschnitten werden. Es ist dabei kritisch, dass dies so nahe wie möglich an den Diensten (optimal: auf der selben Maschine) geschieht, damit die Ankunftszeiten der Netzwerkpakete so genau wie möglich gemessen werden. Außerdem müssen die Uhren des Verzögerers und der Mitschnitte so synchron wie möglich gehalten werden.

Rippler unterteilt den Untersuchungszeitraum in Zeitfenster gleicher Größe. Diese Zeitfenster sollten eine Größe s_w besitzen, die größer ist als die Zeit, die ein Paket trotz Verzögerung zum Erreichen seines endgültigen Ziels benötigt, jedoch möglichst klein, um viele Stichproben zu generieren. Diese Zeitfenster, w_1, w_2, \dots, w_{2n} , werden in Zeitfenster mit ungeradem Index ($w_1, w_3, \dots, w_{2n-1}$) und solche mit geradem Index unterteilt. Verbindungen zu dem aktuell untersuchten Dienst, die in einem ungeraden Zeitfenster beginnen, werden dabei um eine Zeit t_d verzögert.

Es werden dabei immer nur Verbindungen zu einem Dienst gleichzeitig verzögert. Sei dieser Dienst Dienst A. Es werden nun für jeden in Frage kommenden Dienst B überprüft, ob es eine LL-Abhängigkeit ($A \rightarrow B$) oder eine LR-Abhängigkeit ($B \rightarrow A$) besteht. In Frage kommt hierbei alle Dienste außer Dienst A selbst.

Es werden nun die Verbindungen zu Dienst B überprüft. Es wird durch das Watermarking beim Aufruf des Dienstes A ein Muster mit einer bestimmten stochastischen Verteilung erzeugt. Kann dieses Muster in den Verbindungen zu Dienst B wiedergefunden werden, besteht eine Abhängigkeit. Es ist Rippler anhand der Quellen der Verbindungen möglich zu erkennen, ob es sich dabei um eine LL-Abhängigkeit ($A \rightarrow B$) oder eine LR-Abhängigkeit ($B \rightarrow A$) handelt.

Mittels verschiedener stochastischer Verfahren, auf die im Rahmen dieser Arbeit nicht genau eingegangen wird, können nun die Wahrscheinlichkeiten für eine Abhängigkeit zwischen A und B bestimmt werden. Die Verfahren, die von [10] vorgeschlagen werden, sind ein Zweistichproben T-Test (mit unabhängigen oder abhängigen Stichproben) oder der Wilcoxon-Vorzeichen-Rang-Test. Diese Tests versuchen mittels der oben genannten Muster durch das Watermarking die Null-Hypothese, dass die Dienste voneinander unabhängig sind, abzulehnen. [10]

4 Vergleich der Ansätze

Der Algorithmus von Sherlock basiert auf der Annahme, dass ein Dienst B, von dem ein anderer Dienst A abhängt, kurze Zeit vor Dienst A aufgerufen wird. Damit erkennt Sherlock nur RR-Dependencies, und kann, da nur auf Clients gemessen wird, LR-Dependencies nicht erkennen. Weiterhin werden diese Abhängigkeiten nur dann erkannt, wenn derselbe Client sowohl Dienst A als auch Dienst B aufruft. Letztlich kann auch Caching (der Client cacht eine Antwort von Dienst B) dazu führen, dass Sherlock die Abhängigkeit zweier Dienste nicht erkennt. Dies führt zu einer hohen Rate an False Negatives. Weiterhin erzeugt Sherlock jedoch auch viele False Positives: Werden zwei Dienste häufiger hintereinander aufgerufen, obwohl sie nicht voneinander abhängig sind, erkennt Sherlock dies trotzdem als Abhängigkeit. [10]

Auch bei Orion kann das Problem einer hohen Anzahl an False Positives auftreten, aus dem gleichen Grund wie bei Sherlock. Da Orion jedoch voraussetzt, dass der zeitliche Abstand zwischen den Aufrufen zweier Dienste immer sehr ähnlich sein muss, ist dieses Risiko bei Orion deutlich geringer. Es kann jedoch beispielsweise dadurch auftreten, wenn ein Dienst A von zwei Diensten B und C LR-abhängt, und diese weitestgehend gleichzeitig aufruft. Orion könnte dies durchaus als RR-Abhängigkeit zwischen B und C erkennen. [7]

NSDMiner kann aufgrund seines Designs nur LR-Abhängigkeiten erkennen. [9]

Rippler ist der einzige der betrachteten Algorithmen, der Korrelation und Kausalität nicht gleichsetzt. Rippler verspricht aus diesem Grunde, weniger False Positives zu erzeugen als die anderen untersuchten Algorithmen. [10]

Wichtig zu beachten ist, dass False Negatives schwieriger zu finden und zu korrigieren sind als False Positives, da die Liste der gesamten Negatives schnell sehr groß wird, während die Liste aller Positives überschaubarer bleibt.

Tabelle 1. Vergleich der Algorithmen

	Erkennt LR- Abhängig- keiten	Erkennt RR- Abhängig- keiten	Hohe Rate an False Positives	Hohe Rate an False Negatives	Niedrige Rate an False Negatives
Sherlock	-	+	+	+	-
Orion	+	+	+	-	-
NSDMiner	+	-	-	-	-
Rippler	+	+	-	-	+

5 Anwendbarkeit im Web

Um die Anwendbarkeit der vier verschiedenen Ansätze im Web beurteilen zu können, werden verschiedene Use-Cases betrachtet. Es wird dabei untersucht, ob die Ansätze die verschiedenen Abhängigkeiten innerhalb der Use-Cases entdecken können, und wo dazu Software installiert oder Netzwerkverkehr analysiert werden muss. In diesem Kapitel sollen diese Use-Cases beschrieben werden. Die Use-Cases decken sowohl ein einfaches Beispiel ab, als auch eine Konfiguration, bei der die Abhängigkeiten mehrstufig sind.

Weiterhin gibt es im Internet als Netzwerk ein Problem, welches bei kleineren Netzwerken nicht vorliegt: Aufgrund der heutigen Routingverfahren und der Länge der Leitungen ist die Übertragungsgeschwindigkeit nicht nur wesentlich geringer als in lokaleren Netzen, sie schwankt auch deutlich: Es kann ein Paket B, welches nach einem Paket A abgeschickt wurde, vor Paket A beim Empfänger ankommen. Auch auf dieses Problem wird im folgenden Kapitel eingegangen.

Tabelle 2 bietet eine Zusammenfassung darüber, was bei den verschiedenen Algorithmen beim Einsatz im Web zu beachten ist.

Use-Case 1: Einfacher Webserver Ein einfacher Webserver hat eine LR-Abhängigkeit von einer Datenbank, die auf derselben Maschine läuft. Weiterhin besteht eine Abhängigkeit des Webserver von einem DNS-Server, der auf einer anderen Maschine läuft.

Use-Case 2: Ein Webserver mit entfernter Datenbank Wie in Use-Case 1 hat ein Webserver eine LR-Abhängigkeit von einer Datenbank, die jedoch auf einer entfernten Maschine läuft. Diese Datenbank besitzt eine RR-Abhängigkeit von einem DNS-Server 2. Ebenso ist auch der Webserver selbst RR-abhängig von einem (anderen) DNS-Server 1.

5.1 Sherlock

Der Algorithmus von Sherlock erkennt Abhängigkeiten dann, wenn der zeitliche Abstand der Aufrufe zweier Dienste, von denen der eine vom anderen abhängt, klein genug ist. Um Sherlock also sinnvoll im Internet verwenden zu können, muss die Schwelle für den maximalen Zeitunterschied zwischen den Aufrufen

Tabelle 2. Vergleich der Algorithmen im Web

	Sherlock	Orion	NSDMiner	Rippler
Delays im Internet	Problematisch; unter Umständen mit stark erhöhter Menge an False Positives verbunden und erhöhter Rechenaufwand	Unproblematisch; nur erhöhter Rechenaufwand	Problematisch; unter Umständen mit stark erhöhter Menge an False Positives verbunden und erhöhter Rechenaufwand	Sehr Problematisch; Uhren müssen synchron gehalten werden und stark erhöhte Menge an False Negatives oder großer Delay im Betrieb
Use-Case 1	Erkennt RR-Abhängigkeiten, wenn auf Client installiert	Erkennt Abhängigkeiten, wenn auf Client und Webserver installiert	Erkennt LR-Abhängigkeiten, wenn auf dem Webserver installiert	Überall Verzögerer nötig, Netzwerk von Client und Webserver müssen aufgezichnet werden
Use-Case 2	Erkennt RR-Abhängigkeiten, wenn auf Client und Webserver installiert	Erkennt Abhängigkeiten, wenn auf Client und Webserver installiert	Erkennt LR-Abhängigkeiten, wenn auf dem Webserver installiert	Überall Verzögerer nötig, Netzwerk von Client und Webserver müssen aufgezichnet werden

dieser Dienste groß genug gewählt sein. Eine große Schwelle bedeutet aber auch, dass es zu vielen False Positives kommen kann.

Sherlock ist in seiner Funktionsweise grundsätzlich darauf beschränkt, RR-Abhängigkeiten zu erkennen. Um Sherlock nutzen zu können, muss es auf „Clients“ den Netzwerkverkehr mitschneiden. Mit „Clients“ sind in diesem Zusammenhang auch Dienste gemeint, die wiederum andere Dienste aufrufen (also LR-abhängig von diesen sind). Betrachten wir vor diesem Hintergrund die Use-Cases.

Use-Case 1 Hier müsste Sherlock einzig den Netzwerkverkehr der Maschine mitschreiben, die auf den Webserver zuzugreifen versucht. Sherlock ist dann in der Lage zu erkennen, dass eine RR-Abhängigkeit des Webserver vom DNS-Server besteht.

Use-Case 2 In diesem Szenario reicht es nicht, Sherlock den Netzwerkverkehr des Clients mitschreiben zu lassen, der auf den Webserver zuzugreifen versucht. Durch diese Installation würde nur die Abhängigkeit des Webserver vom DNS-Server erkannt werden. Um auch die RR-Abhängigkeit der Datenbank vom DNS-Server zu erkennen, muss auch der Netzwerkverkehr des Webserver mitgeschnitten werden. Dazu muss entweder auf einem Router zwischen dem Webserver, dem DNS-Server und der Datenbank oder direkt auf dem Webserver der Netzwerkverkehr mitgeschnitten werden. In diesem Szenario, in dem nur ein Webserver betroffen ist, ist es wahrscheinlich am praktischsten, direkt auf dem Webserver den Netzwerkverkehr aufzuzeichnen.

5.2 Orion

Wie auch Sherlock ist Orion in seiner Funktionsweise davon abhängig, dass bei abhängigen Dienstpaaaren ($A \rightarrow B$) die Aufrufe von Dienst A und Dienst B zeitlich nah genug beieinander liegen, sodass der zeitliche Abstand unter einen Schwellwert fällt. Bei Orion allerdings sollten durch eine größere Schwelle nicht mehr False Positives erkannt werden, es sollte sich lediglich der Rechenaufwand beim Analysieren der Netzwerkaktivität erhöhen.

Orion muss auf jedem Host, auf dem zu beobachtende Dienste laufen, sowie auf einigen Clients, die diese Dienste aufrufen, den Netzwerkverkehr mitschneiden. Hierzu muss entweder direkt auf diesen Maschinen oder aber mittels Packet Sniffers von Routern zwischen den Maschinen der Netzwerkverkehr aufgezeichnet werden.

Use-Case 1 In diesem Szenario ist es wohl unumgänglich sowohl auf dem Client als auch dem Webserver den Netzwerkverkehr mitszuschneiden. Da hiermit auch die relevante Kommunikation zum DNS-Server bereits betrachtet wird, muss dieser nicht gesondert betrachtet werden, die RR-Abhängigkeit des Webserver vom DNS-Server wird auch so erkannt.

Use-Case 2 Auch hier reicht eine Analyse des Netzwerkverkehrs des Clients und des Webservers. Auf dem Webserver können die Daten gewonnen werden, die nötig sind, um die RR-Abhängigkeit der Datenbank vom DNS-Server 2 und die LR-Abhängigkeit des Webservers von der Datenbank zu erkennen. Für die RR-Abhängigkeit des Webservers vom DNS-Server 1 reichen dann die Daten, die aus dem Netzwerkverkehr vom Client gewonnen werden können.

5.3 NSDMiner

Auch NSDMiner ist von den langsameren Antworten, die ein großes Netzwerk wie das Internet mit sich bringt, betroffen, wenn auch nur indirekt. Zwar analysiert NSDMiner immer ganze Verbindungen ohne zeitliche Einschränkungen zu haben, wann eine Verbindung nicht mehr untersucht wird (beziehungsweise sind diese Einschränkungen von anderen Verbindungen abhängig). Da jedoch untersucht wird, welche Verbindungen im zeitlichen Intervall einer anderen komplett eingebettet sind, kann es häufiger als in lokalen Netzen vorkommen, dass eine Verbindung zu einem Dienst B zufällig in einer Verbindung zu Dienst A eingebettet ist, und dies als ein Hinweis auf eine LR-Abhängigkeit ($A \rightarrow B$) gewertet wird. Hierdurch kann es vermehrt zu False Positives kommen.

Aufgrund des Designs von NSDMiner kann der Algorithmus nur LR-Abhängigkeiten erkennen. Um diese zu erkennen, benötigt es einen Mitschnitt des Netzwerkverkehrs der Maschinen, auf denen Dienste laufen, die von anderen LR-abhängig sind. Bei einer Abhängigkeit ($A \rightarrow B$) sind also die Daten von A relevant.

Use-Case 1 und 2 Hier reicht es, wenn der Netzwerkverkehr vom Webserver aufgezeichnet wird. Mit diesem kann die LR-Abhängigkeit des Webservers von der Datenbank erkannt werden. Andere Abhängigkeiten kann NSDMiner in diesen Use-Cases nicht erkennen.

5.4 Rippler

Die langsamen Verbindungen im Internet sind für Rippler ein großes Problem. Dies beginnt bereits beim Synchronisieren der Uhren: Wie in den Use-Cases gezeigt wird, ist es im Internet nahezu unumgänglich an verschiedenen Stellen Verzögerer einzusetzen. Die Uhren der Verzögerer müssen so synchron wie möglich laufen. Es ist jedoch im Internet ein Problem, diese Synchronität zu gewährleisten.

Die stochastischen Tests, die in Rippler die Abhängigkeit zweier Dienste bewerten, sind darauf angewiesen, dass die Verbindung immer etwa gleich lange zum Erreichen ihres Ziel braucht. Brauchen Verbindungen immer unterschiedliche Zeit um ihr Ziel zu erreichen, so kann nicht richtig erkannt werden, ob eine Verbindung verzögert wurde oder nicht. Da die verschiedenen stochastischen Tests dazu tendieren, eher die Nullhypothese, dass zwei Dienste unabhängig

sind, anzunehmen, führt eine kleine Verzögerung daher zu mehr False Negatives. Wird die Verzögerung jedoch zu groß gewählt, sind spürbare Verzögerungen oder gar Time-Outs denkbar. Wie und ob hier ein Mittelweg gefunden werden kann, bei dem nicht zu viele False Negatives entdeckt werden, aber die Verzögerung gleichzeitig nicht zu groß ist, bedarf weiterer Untersuchungen.

Rippler muss alle Pakete betrachteter Dienste manipulieren können und von all diesen den Netzwerkverkehr mitschneiden. Wenn dies nicht zentral möglich ist, ist zu beachten, dass die Uhren der Verzögerer und der Mitschnitte so synchron wie möglich sein müssen.

Use-Case 1 Hier muss auf allen beteiligten Systemen ein Verzögerer installiert werden und der Netzwerkverkehr mindestens vom Client und Webserver aufgezeichnet werden, um alle Abhängigkeiten zu entdecken. Ein Verzögerer könnte auf der selben Maschine wie der Webserver installiert werden, da alle Kommunikation sowieso durch diese Maschine fließt.

Use-Case 2 Hier gilt Ähnliches wie im Use-Case 1. Es müssen auf allen Systemen Verzögerer installiert werden und es muss der Netzwerkverkehr vom Client und Webserver analysiert werden. Hier können sogar zwei Verzögerer eingespart werden, sollten der Webserver, die Datenbank und der DNS-Server 2 im selben Netzwerk liegen beziehungsweise ein zentraler Router ihren Netzwerkverkehr verwalten. Dann kann, statt auf dem Webserver, dem DNS-Server und dem Datenbankserver Verzögerer zu installieren, ein Verzögerer auf dem Router installiert werden.

6 Fazit

Alle vorgestellten Ansätze zur Dependency Discovery im Internet haben ein gemeinsames Problem: Es ist nötig, dass die Betreiber der Dienste oder der Infrastruktur zwischen den Diensten (und Clients) den Netzwerkverkehr aufzeichnen und Ausschnitte daraus an denjenigen weitergeben, der die Abhängigkeiten analysieren möchte. Im Falle von Intranet-Netzwerken, die sich über mehrere Standorte aufspannen, mag dies gut möglich sein, in anderen Fällen jedoch nicht.

Dieses Problem ist jedoch ein allgemeines Problem an Verfahren zur Dependency Discovery, welche den Netzwerkverkehr analysieren: Diese Verfahren können nur funktionieren, wenn aller Netzwerkverkehr aufgezeichnet wird, der die beobachteten Dienste betrifft. Da aber bisher keine anderen Verfahren existieren, die im Web anwendbar wären (siehe „Bisherige Ansätze“), ist dieses Übel zumindest aktuell ein unumgängliches.

Von diesem Problem abgesehen ist Orion der einzige Algorithmus, der durch die betrachteten Besonderheiten des Internets als Netzwerk nicht beeinflusst wird. Die Time-Outs des Algorithmus müssen zwar erhöht werden, womit ein erhöhter Rechenaufwand verbunden ist, jedoch ist diese Schwäche vernachlässigbar verglichen mit den Problemen, die die anderen Algorithmen im Internet haben.

Interessant ist, dass Rippler, welcher in lokalen Netzen der stärkste Algorithmus ist, im Internet kaum anwendbar ist. Die Annahmen, die in das Design des Algorithmus eingeflossen sind, sind im Internet ungültig.

Um das allgemeine Problem zu beheben, dass es für eine Dependency Discovery im Internet nötig ist, Software auf fremden Systemen zu installieren, wäre es denkbar, ein anderes Protokoll zu verwenden, bei dem Abhängigkeitserkennung bereits protokollseitig vorgesehen ist. Hierfür wäre jedoch die Einführung eines neuen Standards nötig, der dann auch umgesetzt werden müsste. Im Hinblick darauf, wie langsam sich schon IPv6 nur durchsetzt, ist es jedoch unwahrscheinlich, dass dies in naher Zukunft Wirklichkeit wird, auch wenn dies sicherlich ein interessantes Forschungsfeld darstellt.

Interessant wäre es auch zu erforschen, ob es möglich ist, Rippler im Internet zu verwenden, wenn man Bordergateways in Betracht zieht: Ist es also möglich, Rippler mit weniger Nachteilen zu verwenden, wenn der untersuchte Netzwerkverkehr immer durch ein oder wenige Gateways verläuft? In diesem Falle wäre es möglich, die Verzögerer an diesen Bordergateways anzubringen, um die Synchronisierung der Uhren so zu erleichtern. Da Rippler selbst der jüngste der vorgestellten Algorithmen ist, ist davon auszugehen, dass an diesem Algorithmus in Zukunft noch geforscht werden wird, womit unter Umständen die entdeckten Hindernisse in vielen Use-Cases umgangen werden können.

Auch wenn diese Arbeit vom Internet of Things motiviert ist, so gibt es in diesem weitere Eigenschaften, die in dieser Arbeit nicht untersucht wurden. Denkbar ist zum Beispiel, dass ein Ding nicht von einem bestimmten Dienst abhängt, sondern von einem bestimmten Diensttypen. Hierfür ist eine automatische Gruppierung von Diensten, die dieselbe Funktion bieten, erforderlich. Auch dies bietet sicher einen interessanten Ansatz für weitere Forschungen.

Es ist davon auszugehen, dass in Zukunft das Thema Dependency Discovery im Internet weiter beleuchtet wird. Dabei bleibt zu hoffen, dass dadurch ein neuer, innovativer Ansatz entwickelt wird, der die Abhängigkeitserkennung im Internet vereinfacht. Bis dahin jedoch wird hierfür wohl Orion die erste Wahl bleiben.

Literatur

1. Aguilera, M.K., Mogul, J.C., Wiener, J.L., Reynolds, P., Muthitacharoen, A.: Performance debugging for distributed systems of black boxes. In: Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles. pp. 74–89. SOSP '03, ACM, New York, NY, USA (2003), <http://doi.acm.org/10.1145/945445.945454>
2. Atzori, L., Iera, A., Morabito, G.: The internet of things: A survey. *Computer Networks* 54(15), 2787 – 2805 (2010), <http://www.sciencedirect.com/science/article/pii/S1389128610001568>
3. Bahl, P., Chandra, R., Greenberg, A., Kandula, S., Maltz, D.A., Zhang, M.: Towards highly reliable enterprise network services via inference of multi-level dependencies. In: Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications. pp. 13–24. SIGCOMM

- '07, ACM, New York, NY, USA (2007), <http://doi.acm.org/10.1145/1282380.1282383>
4. Barham, P., Donnelly, A., Isaacs, R., Mortier, R.: Using magpie for request extraction and workload modelling. In: Proceedings of the 6th Conference on Symposium on Operating Systems Design & Implementation - Volume 6. pp. 18–18. OSDI'04, USENIX Association, Berkeley, CA, USA (2004), <http://dl.acm.org/citation.cfm?id=1251254.1251272>
 5. Brown, A.B., Kar, G., Keller, A.: An active approach to characterizing dynamic dependencies for problem determination in a distributed environment. In: Pavlou, G., Anerousis, N., Liotta, A. (eds.) Integrated Network Management. pp. 377–390. IEEE (2001), <http://dblp.uni-trier.de/db/conf/im/im2001.html#BrownKK01>
 6. Chen, M.Y., Accardi, A., Kiciman, E., Lloyd, J., Patterson, D., Fox, A., Brewer, E.: Path-based failure and evolution management. In: Proceedings of the 1st Conference on Symposium on Networked Systems Design and Implementation - Volume 1. pp. 23–23. NSDI'04, USENIX Association, Berkeley, CA, USA (2004), <http://dl.acm.org/citation.cfm?id=1251175.1251198>
 7. Chen, X., Zhang, M., Mao, Z.M., Bahl, P.: Automating network application dependency discovery: Experiences, limitations, and new solutions. In: Draves, R., van Renesse, R. (eds.) OSDI. pp. 117–130. USENIX Association (2008), <http://dblp.uni-trier.de/db/conf/osdi/osdi2008.html#ChenZMB08>
 8. Fonseca, R., Porter, G., Katz, R.H., Shenker, S., Stoica, I.: X-trace: A pervasive network tracing framework. In: Proceedings of the 4th USENIX Conference on Networked Systems Design & Implementation. pp. 20–20. NSDI'07, USENIX Association, Berkeley, CA, USA (2007), <http://dl.acm.org/citation.cfm?id=1973430.1973450>
 9. Natarajan, A., Ning, P., Liu, Y., Jajodia, S., Hutchinson, S.: Nsdminer: Automated discovery of network service dependencies. In: INFOCOM, 2012 Proceedings IEEE. pp. 2507–2515 (2012)
 10. Zand, A., Vigna, G., Kemmerer, R., Kruegel, C.: Rippler: Delay injection for service dependency detection. Tech. rep., University of California, Santa Barbara (2013)

Übersicht zu expliziter Interaktion in Intelligenten Umgebungen durch Gesten

Adrian Hoppe*

Betreuer: Matthias Berning[†]

Karlsruher Institut für Technologie (KIT)
Pervasive Computing Systems – TECO

*adrian.hoppe@student.kit.edu

[†]berning@teco.edu

Zusammenfassung. Im Folgenden wird eine Übersicht zum Thema Gestensteuerung in intelligenten Umgebungen geliefert. Zum Einen wird die geeignete Auswahl einer Geste betrachtet und die damit verbundenen Kriterien erläutert. Zum Anderen widmet sich diese Arbeit dem Thema der Erkennung von Gesten durch Computertechnik. Dabei wird zwischen den Gebieten mit und ohne Hardware am Körper, insbesondere auch Wearable Computing unterschieden. Die jeweiligen Bereiche erlauben spezifische Mechanismen zur Positionsbestimmung. Des Weiteren wird die Interaktion durch Gesten mit anderen Modalitäten verglichen.

Schlüsselwörter: Explizite Interaktion, intelligente Umgebung, Geste, Gestensteuerung, Gestenauswahl, Gestenerkennung, Wearable Computing, Mechanismen zu Positionsbestimmung, Mensch Maschine Interaktion, HCI

1 Motivation

Die bisherige Entwicklung von Computer Eingabetechnik setzte voraus, dass der Benutzer direkt über Maus und Tastatur mit einem Computer interagiert. Aufgrund der rapiden Ausbreitung der letzten Jahre sind Computersysteme immer mehr in unseren Alltag eingedrungen. Das sogenannte Ubiquitous (allgegenwärtig) und Pervasive (alles durchdringend) Computing erfordert aber eine Modifikation der Interaktion zwischen Mensch und Maschine.

Ein Beispiel für Ubiquitous Computing ist eine intelligente Umgebungen, auch Ambient Intelligence genannt. Dies ist ein vernetzter, mit Computertechnik ausgestatteter Bereich mit folgenden Eigenschaften: Er ist kontext-sensitiv und reaktionsfähig, besitzt die Möglichkeit sich anzupassen, arbeitet diskret und verborgen und unterstützt den Mensch im täglichen Leben [5]. Intelligente Umgebungen können eine Vielzahl von Geräten, wie z.B. Elektrogeräte und Sensoren, vereinen, diese können sich frei im Raum verteilen. Zur Ansteuerung all dieser Objekte sind Maus und Tastatur ungeeignet, da sie dem Benutzer keine natürliche und effiziente Eingabemöglichkeit bieten. Forscher versuchen daher

Nachfolger für die klassische Eingabe zu finden. Ein Ansatz stellt die explizite Interaktion in intelligenten Umgebungen durch Gesten dar. Explizit bedeutet hier, dass der Benutzer mit einer aktiven Handlung eine Änderung seiner Umwelt erzielen möchte.

Eine Geste als Eingabe für Computer zu benutzen scheint auf den ersten Blick intuitiv, der Mensch agiert mit der Umwelt in dem er sich in ihr bewegt, sie anfasst, sie verändert. Die Umsetzung einer solchen Eingabe ist aber mit deutlich mehr Arbeit verbunden, als der erste Blick vermuten lässt. Komplexe Aspekte, wie Bewegungsanalyse, -modellierung, die Wiedererkennung von Mustern, sowie das eventuelle maschinelle Lernen beziehungsweise die Anpassung der Maschine an den Menschen spielen eine Rolle.

Es existieren bereits umfangreiche Forschungen zu diesem Thema. Anwendungsgebiete für Human Computer Interaction (HCI) durch Gesten sind unter anderem die Interpretation von Zeichensprache, die Kontrolle von Robotern, als auch die, häufig in Filmen eingesetzte, Methode des Motion Capturing.

Diese Arbeit bietet eine Zusammenfassung über die Themengebiete Gestenauswahl und Gestenerkennung. Es wird im Anschluss eine Definition einer Geste geliefert und daraufhin einige Methoden gezeigt, nach denen Gesten auszuwählen sind. Hierbei spielen vor allem der physische Aufwand, als auch die gute Erkennbarkeit einer Geste durch die Hardware eine Rolle. Anschließend werden verschiedene Mechanismen zur Gestenerkennung vorgestellt und anhand von Beispielen weiter illustriert. Die drei Themengebiete stellen eine Interaktion des Anwenders mit und ohne tragbarer Hardware, als auch eine Mischung aus Beiden, mit Hardware ausgestatteter Kleidung, dar. Abschließend wird die Interaktion durch Gesten in einen Vergleich mit anderen Eingabetechniken gebracht und evaluiert.

2 Definition Geste

Zuerst soll hier der Begriff Geste genauer definiert und erläutert werden. Die bisher vorhandenen Studien haben teilweise abweichende Vorstellungen einer Geste.

Nach Stokoe [15] kann eine Geste durch die eindeutige Kombination aus Konfiguration, Position und Bewegung der Hand ausgedrückt werden.

Als anschauliches Beispiel betrachten wir einmal die Begrüßung einer Person durch eine Winkbewegung. Die ausgestreckten Finger der Hand stellen hier die Konfiguration dar. Die Hand befindet sich vor dem Körper, Handfläche nach vorne zeigend, dies ist die Position. Die pendelnde Bewegung von links nach rechts ist die Aktion, die ausgeführt wird. Alles zusammen beschreibt die Wink Handlung.

Auch Bobick und Ivanov [1] legen verstärkt Wert auf die Dynamik, die eine Geste ausmacht. Bewegung, Aktion und Handlung sind hier die ausschlaggebenden Begriffe.

Aufgrund der bereits teilweise angesprochenen Anwendungsgebiete von Gesten lassen diese sich nach Wu und Huang [21] in 4 Kategorien wie folgt einteilen: Es gibt dialogorientierte Gesten, wie die beiläufige, unterstützende Gestik des allgemeinen Sprachgebrauches. Hinzu kommen kommunikative Gesten, diese sind aktiv, Zeichensprache, die von Taubstummen eingesetzt wird, ist ein gutes Beispiel. Kontrollgesten bieten die Möglichkeit virtuelle und reale Objekte, durch z.B. die Analyse von Zeigegesten, zu lokalisieren. Zusätzlich kann sich der Anwender, durch den gezielten Einsatz der Gesten, in virtuellen Umgebungen bewegen. Eng Verbunden dazu stehen die manipulativen Gesten, die die tatsächliche Steuerung von bzw. Interaktion mit virtuellen/realen Objekten vorsehen. Als Beispiel kann man sich hier eine Fernsteuerung von Robotern vorstellen.

3 Wahl der Geste

Aufgrund des dynamischen Aspekts von Gesten ist es offensichtlich, dass die passende Auswahl einer Geste kritisch für eine gute Performance des Computersystems ist. Für den Benutzer spielt die Umsetzbarkeit eine große Rolle. Eine vom Computer leicht zu erkennende Geste, die vom Anwender physisch nicht umsetzbar ist, ist unangebracht. Diverse Ansätze von Gestensteuerung sollen nun an drei Beispielen erläutert werden. Dabei werden verschiedene Konzepte zur Selektion, mit und ohne Rückmeldung an den Benutzer, beschrieben.

3.1 Selektion eines Objektes

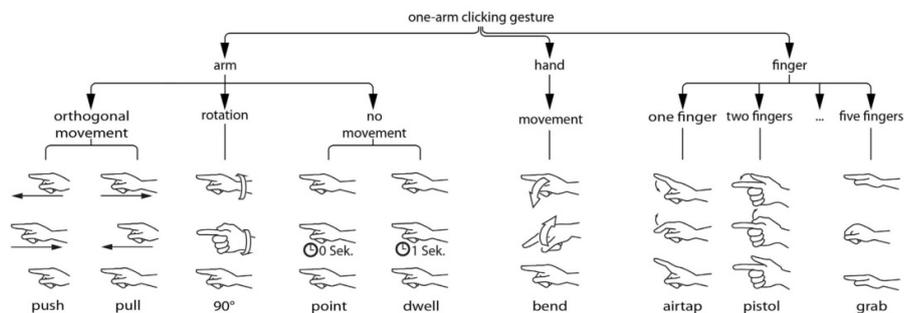


Abb. 1. Übersicht der analysierten Gesten [3]

Van de Camp et al. [3] beschäftigten sich ausgiebig mit dem Thema der Selektion eines Objektes und analysierten die in Abbildung 1 aufgelisteten Gesten. Untersucht wurden Gesten, die eine Auswahl Interaktion über eine gewisse Entfernung mit dem Einsatz nur eines Armes auslösen konnten. Die Studie wurde mit 18 Teilnehmern, im Alter von 20 bis 64 Jahren, verschiedenen Geschlechts,

links- als auch rechtshändig, durchgeführt. Diese sollten auf einer Videowand einen Knopf mit den jeweiligen Gesten drücken. Die zwei am besten bewerteten Gesten waren „airtap“ und „point“.

„airtap“ stellt eine Art Mausclick in der Luft dar und wurde von den Teilnehmern dadurch als sehr intuitiv und außerdem als wenig mühsam beschrieben. Diese Geste wird daher empfohlen. Bei „point“ handelt es sich um eine Auswahl durch Zeigen, sobald die Handbewegung an einem gewissen Punkt stoppt wird ein Klick ausgelöst. Die gute Bewertung dieser Geste wird auf die angenehme und schnelle Bedienung zurückgeführt, bei der kein Mehraufwand, mit Ausnahme der Armbewegung, entsteht. Jedoch wird hier angemerkt, dass die Teilnehmer ein wenig Kontrolle vermissten und die Geste nur in Anwendungen verwenden würden, in denen ein falscher Knopfdruck nicht schwerwiegend wäre. „pistol“, „bend“, „90°“, „grap“, „dwell“ und „push“ wurden zwar nicht so gut bewertet, die Teilnehmer waren ihnen aber nicht vollkommen abgeneigt und sie wurden immer noch als nützlich angesehen. Die Geste „pull“ war den Benutzern zu anstrengend oder nicht intuitiv genug in der Ausführung und wird daher nicht zur Verwendung empfohlen.

3.2 Selektion mit Rückmeldung (Feedback) an den Benutzer

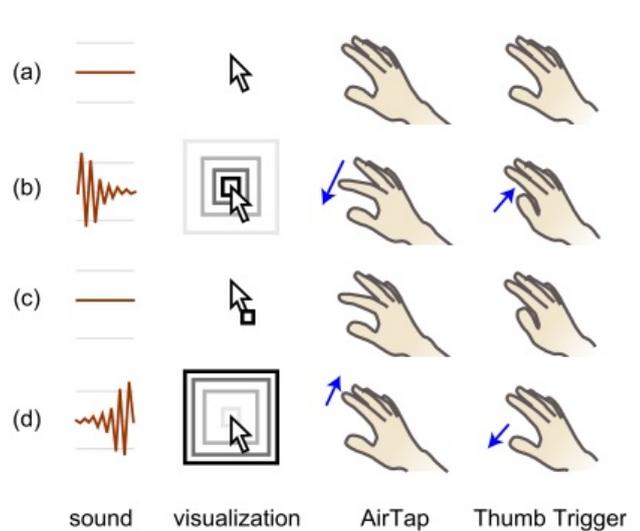


Abb. 2. Klick Gesten mit audio-visuellem Feedback [18]

Die Arbeit von Vogel und Balakrishnan [18] befasst sich auch mit der Auswahl einer geeigneten Selektionsgeste. Die Benutzung einer Verweilzeit zur Auswahl wird jedoch als unpassend eingestuft, da der Benutzer eine dauerhafte Verzögerung während der Benutzung ertragen muss. Auch die Genauigkeit stellt dabei ein Problem dar, da der physische Aufwand den Körper still zu Halten sehr hoch ist und daher nach kurzer Zeit Verwacklungen auftreten können. Es wird weiter beschrieben, dass der zusätzliche Einsatz von Sprache für einen einfachen Klick zu exzessiv ist. Außerdem schildern die Forscher das Problem des fehlenden Feedbacks einer Geste. Nach Wang et al. [19] wird die Bedienleistung ohne eine Rückmeldung immens beeinträchtigt.

Unter Einbeziehung dieser Probleme werden zwei Klick Gesten vorgestellt, die jeweils durch audio-visuelles Feedback unterstützt werden. „AirTap“ stellt hier, wie auch oben, einen Klick mit dem Zeigefinger in der Luft dar. Bei „Thumb Trigger“ wird der Daumen zum Zeigefinger bewegt und kann diesen auch berühren. Eine Berührung bei „Thumb Trigger“ bietet den Vorteil einer haptischen Wahrnehmung und erhöht dadurch die Rückmeldung an den Benutzer, obendrein gibt es einen absoluten Anschlag, der die Erkennung durch das System erleichtert.

Abbildung 2 beschreibt den Ablauf eines Klicks mit Einsatz von Feedback, der nun erläutert werden soll. Abb. 2 (a) bildet hier den Ruhezustand des Systems ab. In Abb. 2 (b) wird eine Klick Geste ausgeführt. Hierbei wird nun ein markanter Ton abgespielt, sowie eine Reihe von kleiner werdenden Quadraten um den Cursor animiert, um eine visuelle Unterstützung zu geben. Bei Abb. 2 (c) befindet sich das System in einem Verweil-Status, der aktuell gehaltene Klick des Benutzers wird nun durch eine Erweiterung des Cursors um ein Rechteck signalisiert. Abb. 2 (d) zeigt das Lösen des Klicks, wieder begleitet von einem charakteristischen Ton, als auch visuell durch nun größer werdende Quadrate. Die beschriebenen Klick Gesten wurden mit drei verschiedenen Zeige Gesten kombiniert:

„RayCasting“ nutzt eine einfache Verlängerung des Zeigefingers zur Bestimmung der Cursor Position, siehe Abb. 3 (a). In Abb. 3 (b) „Relative Pointing with Clutching“ wird eine offene Hand zur relativen Bestimmung der Position genutzt, die geschlossene Faust bietet dem Benutzer die Möglichkeit des Clutching, des Lösen von der Interaktion durch die Hand. Das aktive Clutching wird durch den Wechsel auf einen aufrecht stehenden Cursor angezeigt. „Hybrid RayToRelativePointing“ ist eine Mischung aus beiden Varianten, die eine relative, als auch absolute Kontrolle der Position des Zeigers erlaubt.

Die beiden Gesten „Relative Pointing with Clutching“ und „Hybrid RayToRelativePointing“ unterscheiden sich kaum, sowohl in der Fehleranfälligkeit, als auch in der Auswahlzeit. Beide scheinen für die Aufgabe der Auswahl von 16mm kleinen Zielen aus 4m Entfernung gut geeignet zu sein. „RayCasting“ weist eine hohe Fehlerrate (22,5%) während der Selektion auf und wird daher als nicht praktikabel eingestuft.

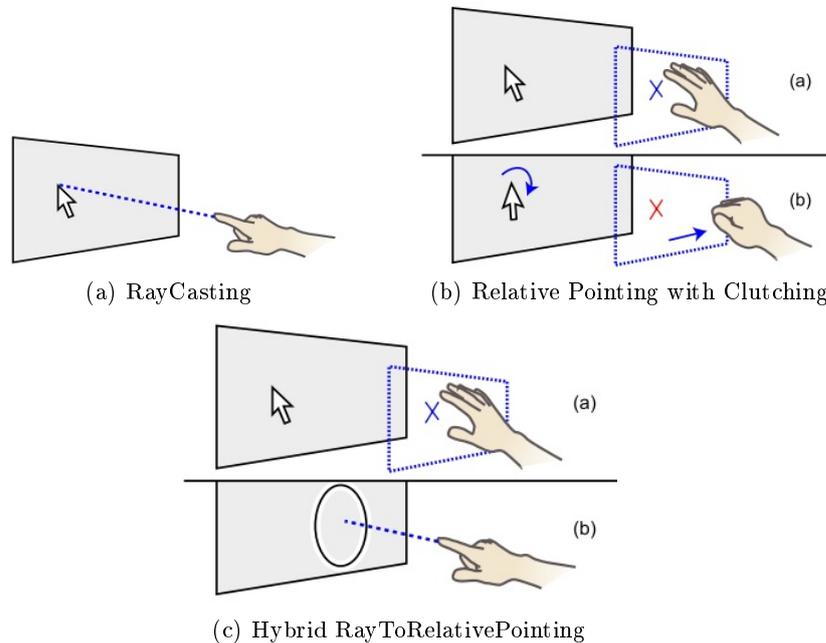


Abb. 3. Diverse Zeige-Gesten [18]

3.3 Selektion in zwei Schritten

Wie im vorherigen Abschnitt erläutert ist die genaue Auswahl von Objekten in größerer Entfernung ein Problem, das durch relatives Zeigen gelöst werden kann. Falls aber diese Objekte nicht nur klein sind, sondern auch noch dicht beieinander liegen, treten erneut Auswahl-Probleme auf, die nicht unbedingt optimal durch „Relative Pointing“ gelöst werden können.

Delmare et al. [7] versuchten das Problem der Ungenauigkeit von „RayCasting“ auf großer Distanz und das Problem einer Bereichsauswahl, die nicht genügend Kontrolle bietet und zur Auswahl falscher Objekte führen kann, zu entschärfen. Dazu teilten sie die Zeigeoperation in zwei Schritte auf und kombinierten die zwei Methoden.

Wie Abbildung 4 (a) zeigt wird eine grobe Auswahl durch die Ausrichtung des Armes bestimmt, die genaue Bestimmung des gewünschten Objekts wird durch eine Drehung des Handgelenks erreicht. In diesem Fall handelt es sich bei den Objekten um LED Leuchten, die bei der Auswahl aufleuchten. Es wird erwartet, dass diese Methode gut geeignet ist für Auswahloperationen in dicht bestückten Umgebungen, unabhängig von Größe und Entfernung der Objekte. Des Weiteren wird davon ausgegangen, dass Benutzer, die sich an das System gewöhnt haben, eine effiziente Interaktion durch Kombination der zwei Schritte erreichen können. Instabilitäten der Auswahl durch ungewollte Armbewegung während z.B. der Armrotation kann vermieden werden. In Abb. 4 (b) hat der

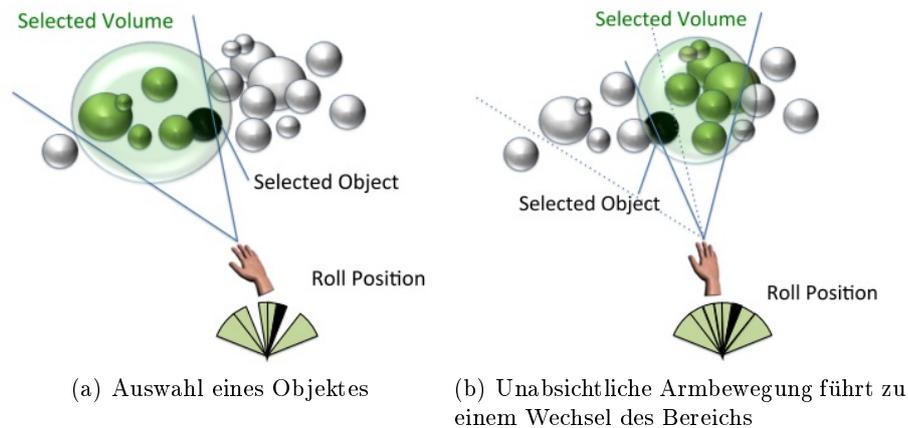


Abb. 4. Bereichsauswahl (grün) mit Zeigegeste und Präzision (schwarzes Objekt) per Handdrehung [7]

Benutzer durch eine ungewollte Armbewegung den ausgewählten Bereich gewechselt. Um ein Flackern der Auswahl zu verhindern wird die relative Rotation der Hand angepasst. Alternativ kann eine Bestätigung zwischen der Bereichs- und der Objektauswahl stattfinden, dies würde aber die Auswahlzeit insgesamt erhöhen.

4 Methoden der gerätefreien Gestensteuerung

Der folgende Abschnitt befasst sich mit dem Thema Gestensteuerung ohne Einsatz eines Zusatzgeräts am Körper des Benutzers, die Erkennung wird von Hardware im Raum übernommen. Es wird auf verschiedene Verfahren zur Umsetzung dieser Aufgabe eingegangen und deren Einsetzbarkeit evaluiert.

4.1 Optische Positionsbestimmung

Die Position einer Person im Raum kann mit einfachen oder stereoskopischen Kameras ermittelt werden. Dazu wird versucht aus dem Kamerabild die Silhouette des Menschen zu bestimmen. Unter Anderen entwickelte Krueger [9] diverse Mechaniken, die eine Interaktion möglichst natürlich wirken lassen.

Zusätzlich zur rein optischen Kamera kann durch eine Verwendung einer Tiefensensor Kamera, eine Kombination aus Infrarot Laser Projektor und einem Bildsensor, der reflektierende Strahlen aufnimmt, die Raumwahrnehmung des Systems verbessert werden.

Jedoch wird die rein optische Bestimmung von Personen im Raum durch technische Aspekte eingeschränkt. Zum Einen sorgt eine niedrige Auflösung der Kameras zu Ungenauigkeiten, zum Anderen ist die Geschwindigkeit, herkömmlich 60 FPS (Frames per Second), der Aufzeichnung nicht für schnelle Bewegungen

geeignet. Hinzu kommt die Tatsache, dass verschiedene Körperteile (z.B. Finger) sich gegenseitig verdecken können und sich somit eine Position nicht immer permanent herleiten lässt.

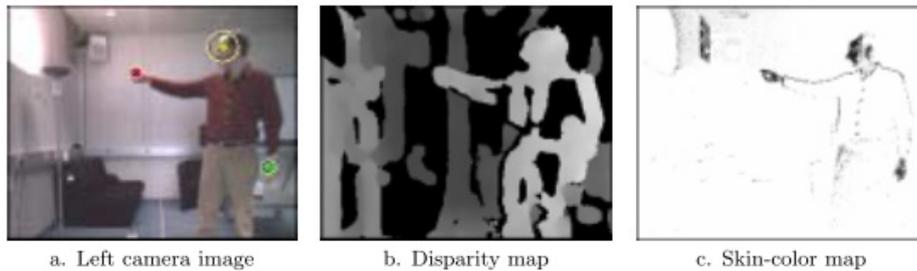


Abb. 5. Ausgangsbild (a) und zusätzlich berechnete Bilder (b,c) [10]

Beispiel: Stereoskopische Kamera, Aufzeichnung der Ausrichtung von Hand, Kopf und Gesicht Ein Gesten-Erkennungssystem wurde von Nickel und Stiefelhagen [10] mittels einer stereoskopischen Kamera umgesetzt. Dazu werden die Entfernungsmessungen, die die stereoskopische Kamera liefert (s. Abb. 5 (a)), mit Hilfe einer Bibliothek, in eine Disparity Map umgewandelt (s. Abb. 5 (b)). Diese enthält für alle Pixel die jeweiligen 3D Koordinaten. Das laufende System durchsucht nun das Bild nach einem Kopf einer Person. Falls ein Objekt mit der Form eines Kopfes gefunden wurde, werden die Pixel innerhalb der Region für die Erstellung einer Skin-Color Map (s. Abb. 5 (c)) verwendet. Basierend auf der gefundenen, gemittelten, Hautfarbe des Kopfes werden in dieser Darstellung farblich passenden Objekte (z.B. Hände) markiert. Die Kombination von Disparity und Skin-Color Map führt dann zu der Klassifikation von Händen und Kopf der Person.

Mit jedem neuen Bild, das die Kamera liefert, wird die Zuordnung erneut unter Zuhilfenahme folgender Punkte durchgeführt:

Die Beobachtung eines menschlichen Körperteils ist umso wahrscheinlicher, je größer die Anzahl an hautfarbenen Pixel im erfassten Objekt ist.

Die Körperhaltung der berechneten Person muss den Gesetzen der menschlichen Bewegung entsprechen.

Die Abfolge der aktuellen und der vorhergegangenen Berechnung sollte in Bezug zueinander stehen.

Durch Einhaltung dieser Schritte ist es möglich die Position der Hände einer Person sicher zu bestimmen, sowohl bei einer sich bewegenden Kamera, als auch einem komplexen Hintergrund. Durch die automatisch stattfindenden Berechnung muss das System nicht initialisiert werden. Gelegentliche Falschdeutungen

von Händen werden auf Verdeckung, schnelle Bewegungen und die sich bei Bewegung ändernde Handform zurückgeführt.

Bei der Nutzung einer einfachen Zeigeoperation mit dem ausgestreckten Arm wurden in einem Versuch 88% vermeintlicher Gesten erkannt, 75% davon waren korrekte Gesten. Dies führte zu einer 90%-igen Wahrscheinlichkeit der korrekten Auswahl eines Objektes. Anschließend wurde das Gerät speziell auf den jeweiligen Benutzer eingestellt, was in einer Erhöhung der Präzision auf 89% und insgesamt in einer 97%-igen Erkennungsrate der ausgewählten Ziele resultierte.

Beispiel: MS Kinect Oikonomidis et al. setzten die, für die Microsoft XBOX 360 entwickelte, Erweiterung Kinect ein um eine 3-dimensionale Bestimmung einer Handposition umzusetzen [11]. Mit Hilfe der Farb- und Tiefenbildkamera erreichten sie eine Positionsbestimmung in Echtzeit mit einer Rate von 15 Hz. Auf 2,5m Entfernung blieb der Fehler dabei unter 7,5mm. Ähnlich zum vorherigen Beispiel wurde auch die Hautfarbe des Benutzers genutzt um die Form der Hand genauer zu bestimmen. Ein großer Vorteil des Einsatzes von kommerziellen Sensoren ist offensichtlich der, durch die hohe Stückzahl, reduzierte Preis und die hohe Verbreitung, die den Einsatz im Wohnzimmer des normalen Bürgers begünstigt.

5 Methoden der Gestenerkennung mit Hilfe von technischer Kleidung

Nun sollen Verfahren erläutert werden, bei denen der Benutzer Hardware am Körper trägt, um mit der virtuellen Umgebung interagieren zu können. Die operierende Person muss dazu spezielle, mit Hardware ausgestattete, Kleidungsstücke tragen, auch Wearable Computing genannt. In diesem Bereich sind vor allem Handschuhe verbreitet, die den Benutzer kaum einschränken, da er z.B. immer noch andere Gegenstände halten kann.

5.1 Optische Positionsbestimmung

Nach Sturman und Zeltzer [16] gibt es die Möglichkeit auf dem Körper einer Person diverse Marker anzubringen. Diese können entweder Infrarot-LEDs oder aber auch Infrarot-Licht reflektierende Punkte sein. Eine Reihe von im Raum aufgestellter Kameras, mindestens zwei, nehmen die Markierungen wahr. Mit Hilfe von Software werden dann die Positionen der verschiedenen Blickwinkel miteinander korreliert um daraus eine 3D Koordinate für jeden Marker im Raum zu bestimmen. Selspot, Op-Eye und Optotrak bieten LED Systeme, die bereits von diversen Laboren eingesetzt werden. Im Bereich der reflektierenden Marker existieren Systeme von Elite und Vicon.

Ein Problem der beschriebenen Methode ist die benötigte Berechnungszeit der Koordinaten. Bei Kamerasystemen, die mit 60 FPS aufzeichnen fallen so viele Daten an, dass einige Systeme nicht in Echtzeit arbeiten können. In diesen Fällen wird das Bildmaterial aufgezeichnet und anschließend verarbeitet. Es ergibt

sich ein weiteres Erschwernis bei der Nutzung von LED Markern, diese werden in einer Sequenz nacheinander eingeschaltet und jeweils einzeln aufgezeichnet, was die Geschwindigkeit bei einer hohen Anzahl von Markern stark limitiert. Aber auch bei reflektierenden Markierungen gibt es ein Problem, die Zuordnung der Marker kostet erhöhten Rechenaufwand und ist nicht immer eindeutig, diese Angelegenheit kann aber durch den Einsatz von zusätzlichen Kameras eingeschränkt werden.

Die teils verminderte Echtzeitfähigkeit und die Probleme mit zu dicht aneinander liegenden Markierungen dämmen die Verwendung dieser Methode für virtuelle Umgebungen ein.

Eine Erweiterung der Positionsbestimmung im Raum ist die Lokalisierung von Fingerpositionen. DeFanti und Sandin entwickelten den sog. Sayre Glove [6], der es ermöglicht über Schläuche, die auf dem Handrücken entlang der Finger verlaufen, den Grad der Finger Kontraktion zu ermitteln. Hierzu werden die flexiblen Schläuche an einem Ende mit einer Lichtquelle, an dem anderen Ende mit einem Lichtsensor ausgestattet. Bei ausgestreckten Fingern empfängt der Sensor einen maximalen Licht Wert, der durch das Knicken der Finger, und dadurch auch des Röhrchens, stetig abnimmt. Durch Analyse der Werte kann dann die relative Position der Finger zur Hand bestimmt werden.

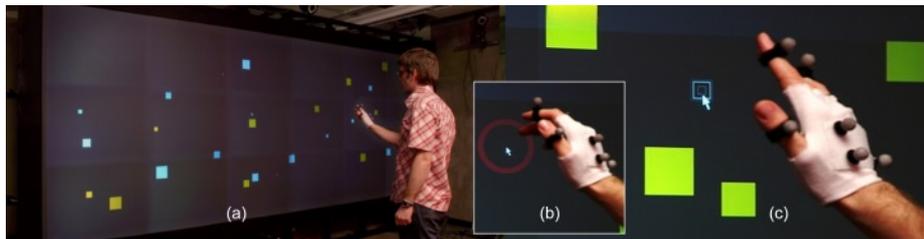


Abb. 6. (a) großes, hochauflösendes Display, (b,c) Nahaufnahmen der Hand während der Interaktion [18]

Beispiel: Optisches Tracking, reflektierende Markierungen Vogel und Balakrishnan [18] verwendeten ein Vicon Tracking System für die Ermittlung der Handposition und -orientierung. Wie in Abb. 6 (b,c) zu sehen wurden dafür reflektierende Markierungen an den Fingern sowie dem Handrücken befestigt. Die im Raum befindlichen Kameras zeichnen den Benutzer dabei von verschiedenen Seiten auf. Ihr System ist in der Lage die 3D Position jedes einzelnen Markers, mit 120 Hz, millimetergenau zu bestimmen und an andere Anwendungen weiter zu geben. Ein Zusammenschluss von 18 Beamern generiert eine Display-Fläche von 5 x 1,8 Metern Größe, mit einer Auflösung von 6144 x 2304 Pixeln, auf der der Benutzer Selektionen ausführen kann. 18 verbundene Computer berechnen

das Bild mit einer Geschwindigkeit von 45 FPS, was einer Verzögerung von 22 ms zur Umsetzung einer Bewegung entspricht. Die Forscher weisen darauf hin, dass ihr Prototyp nur eine Übergangslösung sei, bis die markerlose Erkennung von Gesten im Raum präzise möglich wäre. Es ist daher als Mittel zum Zweck zu sehen, um weitere Forschungen im Gebiet der Gesten zu erleichtern.

5.2 Magnetische Positionsbestimmung

Es ist möglich die Position eines Objektes mit Hilfe eines magnetischen Feldes zu ermitteln [16]. Hierzu wird eine Quelle benötigt, die dieses Magnetfeld ausstrahlt. Sensoren auf der Kleidung des Benutzers ermitteln ihre eigene Position und Ausrichtung relativ zur Quelle und übermitteln diese an einen zentralen Apparat. Existierende Systeme von Polhemus oder Ascension Technologies bieten die Möglichkeit Punkte im Raum auf bis zu 5 Meter Entfernung aufzunehmen und verzeichnen eine Genauigkeit der Position von bis zu 0,25 Zentimeter auf 0,1 Grad Rotation. Ein Vorteil dieser Systeme ist die Eigenschaft, dass sie keine direkte Sicht auf die Sensoren benötigen. Ein Nachteil stellt die Einwirkung metallischer Objekte im Raum dar, die das Magnetfeld zerstreuen und so zu fehlerhaften Positionsbestimmungen führen können.

Beispiel: Magnetsensor Tracking An dieser Stelle sei auf das Beispiel des Kapitels 6.1 verwiesen, das das Thema der magnetischen Positionsbestimmung behandelt. Ciger et al. [4] nutzten einen „Magic Wand“, einen Stab an dem Sensoren angebracht wurden, um eine Gestenerkennung durchzuführen. Die Erkennung mehrerer Gesten erlaubte die Steuerung von und Interaktion in einem eigens kreierte Spiel.

Der Unterschied von einem Handschuh zu einem Stab kann vom technischen Aspekt her vernachlässigt werden.

5.3 Akustische Positionsbestimmung

Akustische Bewegungserfassung benutzt einen Hoch-Frequenz-Ton, der vom Benutzer ausgeht. Mikrofone im Raum nehmen diese Signale auf und bestimmen daraus die Position der Quelle [16]. Eine gute Platzierung erlaubt eine Genauigkeit von wenigen Millimetern. Wie die optischen Systeme benötigt die akustische Positionsbestimmung eine direkte Sichtlinie zwischen der Quelle und den Mikrofonen. Des Weiteren können Störungen durch akustische Reflexionen auftreten. Beim Einsatz mehrerer Quellen muss sichergestellt werden, dass diese auf verschiedenen Frequenzen arbeiten, um Beeinflussungen untereinander zu vermeiden. Die Geschwindigkeit mit der sich Schallwellen ausbreiten ist relativ träge, daher arbeitet das System vergleichsweise langsam [12]. Hinzu kommt die Tatsache, dass sich die Geschwindigkeit des Schalls je nach Temperatur, Feuchtigkeit und Druck der Luft ändert, was für Ungenauigkeiten sorgen kann. Logitech und Mattel bieten entsprechende Systeme an.

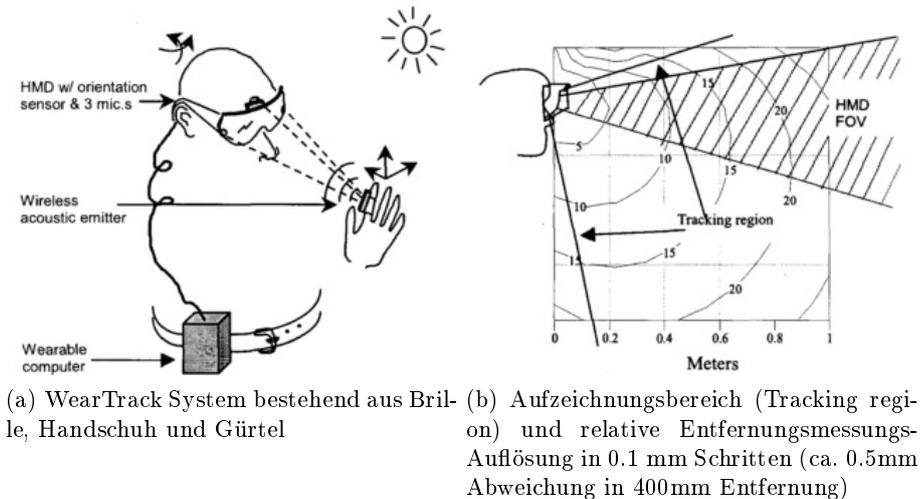


Abb. 7. WearTrack System Aufbau und Tracking [8]

Beispiel: Akustiksensor Tracking, WearTrack System Foxlin und Harrington präsentierten das sog. WearTrack System [8]. Die Idee war ein Interface zu entwerfen, welches, ohne Aufbau von anderer Hardware, direkt verwendet werden konnte. Dazu wird der Benutzer mit einem Sender an der Hand (z.B. ein Ring, hier Pegasus FreeD) ausgestattet, der ein akustisches Signal ausstrahlt, dieses Signal wird von einer Brille auf dem Kopf aufgezeichnet, welche zusätzlich noch einen Orientierungssensor enthält (siehe Abb. 7 (a)). Da die typische Entfernung zwischen Kopf und Hand eher gering ist, erzielten die Forscher eine gute Positionsaufösung mit hohen Messraten. Weiterverarbeitet werden die Daten von einem kleinen tragbaren Computer am Gürtel des Benutzers. Wie in Abbildung 7 (b) zu sehen ermöglicht WearTrack eine genaue Aufzeichnung der Handposition, auch wenn diese sich nicht direkt vor dem Gesicht des Benutzers befinden sollte. Die Gestensteuerung erfolgt durch eine Bewegung der Hand relativ zum Kopf, eine direkte Auswahl durch Zeigen auf Objekte ist ebenfalls möglich.

6 Methoden der Gestenerkennung durch Geräte in der Hand

Der folgende Abschnitt handelt von Geräten, die der Nutzer aktiv in der Hand halten muss, um mit ihnen seine Gesten aufzeichnen zu können. Die starke Verbreitung von aktuellen Spielkonsolen, die in ihren Controllern zum Teil eine Vielzahl von Sensoren vereinen, macht diesen Bereich besonders leicht zugänglich und dadurch attraktiv.

6.1 Positionsbestimmung

Ähnlich zu Kapitel 5 gibt es die Möglichkeiten der optischen, magnetischen und akustischen Positionsbestimmung. Der einzige Unterschied besteht nur in der Art der Anbringung an den Nutzer. Dadurch, dass der Anwender ein Gerät in den Händen halten muss, sind die Finger nur eingeschränkt für den Einsatz von Gesten nutzbar und daher muss dies durch entsprechende Hilfsmittel kompensiert werden. Häufig wird die Unterbringung von Knöpfen auf den Geräten beobachtet, die die fehlende Selektionsfähigkeit durch Gesten ausgleichen.

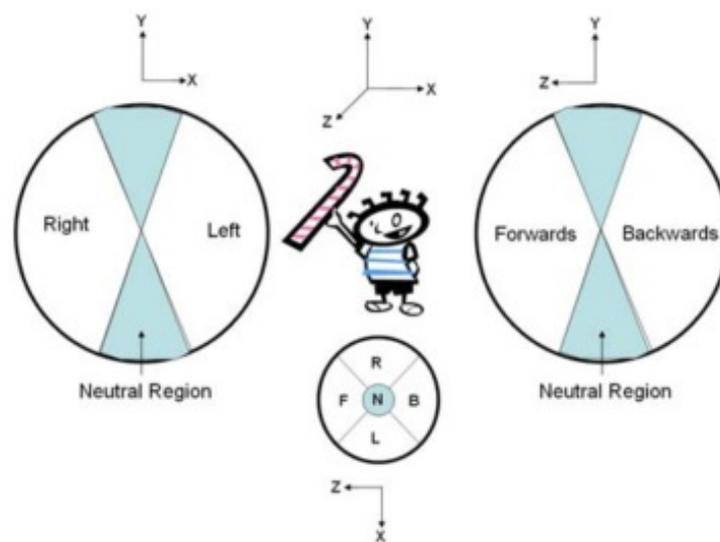


Abb. 8. Ausrichtungsregionen [4]

Beispiel: Magnetsensor Tracking, Magic Wand Ciger et al. [4] nutzen einen hölzernen Stab an dem sie einen magnetischen Sensor, den Ascension Flock of Birds, anbrachten. Der Sensor nutzt eine pulsierende Gleichstrom-Spannung und ermöglicht eine Erkennung in allen 6 Freiheitsgraden (Degrees-of-Freedom, DOF). Es können also Bewegungen in alle 3 Richtungen, als auch Drehungen um die 3 Achsen registriert werden. Untersuchungen zeigten, dass die Positionsbestimmung sehr ungenau ist, die Orientierung des Stabs kann aber sehr präzise bestimmt werden. Für das Experiment wurde nur die Stellung des Stabs verwendet, daher spielte die mangelnde Präzision der Position keine große Rolle. Wie in Abb. 8 zu sehen wurden eine Neutrale, sowie vier Regionen zu Steuerung deklariert. Ciger et al. entwarfen eine Art Spiel in dem sich der Benutzer mit Hilfe des Stabes bewegen konnte. Der Benutzer platzierte sich vor einer Leinwand

und zeigte auf selbige. Durch eine Erweiterung des Systems durch eine Spracherkennung zur Befehlseingabe konnte der Benutzer ein Kommando geben um sich im Spiel zu bewegen und zu stoppen. Das System bestimmte die Zeige Position des Stabes, solange er noch auf den Bildschirm gerichtet war. Eine Neigung des Stabes außerhalb des Bildschirms veranlasste eine Rotation der Kamera und erlaubte so eine Navigation in der virtuellen Welt.

Die Forscher betonen das ihre Umsetzung sehr praktikabel sei, da es eine wesentlich geringere Lernkurve als z.B. ein Joystick vorweise. Außerdem benötige das System keine Vorbereitung durch den Nutzer, dieser müsse lediglich den Stab in die Hand nehmen und könne direkt mit der Eingabe beginnen.

Beispiel: WiiMote Der von Nintendo entwickelte Controller WiiMote nutzt eine kabellose Verbindung zur Konsole via Bluetooth. Auf eine Entfernung von bis zu 9 Metern zeichnet er durch Sensorik im Gerät 3D-Beschleunigungsdaten auf [20]. Die Wii Konsole ist mit einer LED Leiste verbunden, die 10 Infrarotlampen enthält. Diese Sensorleiste wird entweder über oder unter den Bildschirm platziert. Aufgezeichnet wird das Licht von einer Infrarotkamera am Kopf des Controllers. Seine Zeige Position kann dann, relativ zu der LED Leiste, vom System bestimmt werden, dies ist bis zu einer Entfernung von 5 Metern akkurat möglich [17]. Erweitert wird die WiiMote durch den Nunchuk und Wii Motion Plus. Der Nunchuk ist eine an die WiiMote kabelgebundene Steuerungseinheit, die weitere Eingabetasten und Beschleunigungssensoren beinhaltet. Wii Motion Plus ist ein Steck-Modul für den Controller, welches einen gyroskopischen Sensor verwendet, der die Präzision erhöhen soll.

Wingrave et al. [20] beschreiben einige Limitierungen in der Genauigkeit der Positionsbestimmung der WiiMote. Sie untersuchten die Einsetzbarkeit im Bereich Gestensteuerung und fanden einige Optimierungen für den Controller. Hauptvorteil des Systems sei jedoch nicht der geringe Preis, sondern die Auswirkung auf das Ubiquitous Computing. Die Plattform biete einen Anstoß für ein ganze Reihe von neuen Anwendungen.

Schlömer et al. [14] etablierten 5 Gesten für die Erkennung durch die WiiMote und erreichten eine korrekte Erkennungsrate von 85 bis 95 Prozent. Sie entwickelten eine Bibliothek, die es sogar ermöglicht dem System eigene Gesten beizubringen. Diese Bibliothek ist öffentlich zugänglich.

7 Vergleich zu anderen Eingabetechniken

Die Interaktion mit virtuellen Umgebungen durch Gesten ist nur eine Möglichkeit. Blicksteuerung oder der Einsatz des gesamten Körpers, sprach-basierte Systeme oder die Verwendung einer klassischen Benutzerschnittstelle stehen auch zur Verfügung. Brumitt et al. [2] zeigten in einem Experiment, dass eine Steuerung von z.B. Lichtern in einem Raum präferiert durch den Einsatz von Sprache bewältigt wurde. Jedoch kristallisierte es sich heraus, dass nur mit diesem Mittel die Beschreibung des gewünschten Objektes nicht einfach fällt und daher eine Form von lokalisierender Eingabe benötigt wird. Eine Kombination von Sprache

und Geste erzielte hierbei Bestnoten im Bereich Benutzerfreundlichkeit. Die Verbindung mehrerer Modalitäten, wie eben Sprache, Gesten oder anderen Eingabemethoden, wird multimodale Interaktion genannt.

Rukzio et al. [13] verglichen drei verschiedene Methoden zur Auswahl. Zum einen der direkte Kontakt, durch den Benutzer (Touching). Zweitens das Zeigen auf das gewünschte Objekt (Pointing) und als dritte Variante die Erfassung der Objekte in der näheren Umgebung durch ein Gerät und die Auswahl mit dem Selbigen (Scanning). Abhängig von der Aktivität des Benutzers (stehen, sitzen, liegen) und vom Aufwand der jeweiligen Methode, trafen die Testpersonen im Versuch eine Entscheidung für eine der drei Varianten. Es fiel in den Versuchen auf, dass die Nutzer, bei bestehendem Blickkontakt, zu den ersten beiden Möglichkeiten griffen, um die Problemstellung der Auswahl zu lösen. Dem Menschen ist ein solcher Umgang sehr intuitiv und daher die erste Wahl. Touching wurde bei einer geringen Entfernung gewählt, Pointing um größere Entfernungen zu überbrücken bzw. körperliche Anstrengung zu vermeiden. Scanning ist ein komplexer, indirekter Vorgang und wurde daher möglichst umgangen. Anders aber, wenn das auszuwählende Objekt sich nicht in direkter Reichweite, beispielsweise in einem anderen Raum, befand. In diesem Falle überwog der Wunsch des Menschen den physischen Aufwand möglichst gering zu halten und es wurde Scanning verwendet.

8 Fazit

Abschließend lässt sich sagen, dass weitere Forschungen im Bereich der Gestensteuerung sinnvoll und auch notwendig sind. Der zukünftige Einsatz im Heimbereich ist durch bereits weitreichende Verbreitung von gestenfähiger Hardware mehr als wahrscheinlich. Hinzu kommt noch die teils mangelnde Genauigkeit der vorhandenen Systeme, die aber sehr wahrscheinlich durch eine weitere Steigerung der Hardwareleistung, als auch durch die Erforschung neuer und besserer Algorithmen weitreichend kuriert werden kann.

Die Gruppe von Gesten, die ein System zur Interaktion mit der intelligenten Umgebung bereitstellen will, muss sorgfältig ausgewählt werden um Performance und Benutzerfreundlichkeit zu steigern.

Meiner Meinung nach sollte in Zukunft ein Fokus auf eine gerätefreie HCI gelegt werden. So könnte jede Person im Raum mit dem System interagieren und der Bedarf der Verbindung durch Anlegen eines Gerätes entfiere völlig. Aber auch Wearable Computing könnte den zusätzlichen Aufwand durch einen erhöhten Nutzen kompensieren. Es ergibt sich eine immense Erhöhung der Präzision durch die Nutzung von Markierungen am Körper des Anwenders. Gleichzeitig schränken die Kleidungsstücke aber auch nur minimal ein, so könnte man die Kleidung z.B. morgens an- und erst abends wieder ablegen. Die Umsetzung einer Gestensteuerung durch den Einsatz von Wearable Computing hat den weiteren Vorteil, dass auch der Mechanismus der magnetischen oder akustischen Positionsbestimmung eingesetzt werden kann. Dies könnte dem System, je nach Um-

gebung, eine eventuelle Leistungssteigerung bieten, die durch eine rein optische Erkennung vielleicht nicht möglich ist. Das Halten eines gestenfähigen Eingabegerätes ist zum Einen eine physische Anstrengung und zum Anderen auch nicht zu jeder Zeit, während jeder Aktivität durchführbar. Daher sehe ich persönlich keine größere Bewandnis in dieser Form der Gestensteuerung für die Zukunft. Jedoch ist diese Aussage nicht generell gültig und es muss bei der Auswahl der technischen Grundlage des Systems auch der Anwendungszweck mit einbezogen werden. Anwendungen die eine sehr hohe Anforderung an Effizienz und Genauigkeit haben, wie z.B. Computerspiele, können dem Benutzer deutlich mehr Einsatz abverlangen. Dadurch, dass sich der Anwender aktiv entscheidet mit dem Computer über eine längere Zeit zu interagieren, ist die Nutzung eines physikalischen Gerätes erträglich. Leichtgewichtige Aufgaben, wie die Betätigung des Lichts oder das Einschalten einer Stereoanlage, sollten den Benutzer aber nicht stark belasten und erlauben keinen erheblichen Mehraufwand seinerseits. Gesten als unimodale Eingabe wirken unpraktisch und gezwungen, die Kombination von Gesten mit anderen Eingabefunktionen erscheint eine gut umsetzbare und auch leicht durchführbare Strategie zu sein. Bisherige Forschungen zeigen, dass der Mensch eine Mischung aus Sprach- und Gestensteuerung bevorzugt. Diese Form der Interaktion ist dem Menschen sehr natürlich und könnte die Einbindung von Computern in unsere Umgebung so unauffällig und gleichzeitig so nützlich wie möglich gestalten.

Literatur

1. Bobick, A.F., Ivanov, Y.A.: Action recognition using probabilistic parsing. In: Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on. pp. 196–202. IEEE (1998)
2. Brumitt, B., Krumm, J., Meyers, B., Shafer, S.: Let there be light: Comparing interfaces for homes of the future. *IEEE Personal Communications* 28, 35 (2000)
3. van de Camp, F., Schick, A., Stiefelhagen, R.: How to click in mid-air. In: Distributed, Ambient, and Pervasive Interactions, pp. 78–86. Springer (2013)
4. Ciger, J., Gutierrez, M., Vexo, F., Thalmann, D.: The magic wand. In: Proceedings of the 19th spring conference on Computer graphics. pp. 119–124. ACM (2003)
5. Cook, D.J., Augusto, J.C., Jakkula, V.R.: Ambient intelligence: Technologies, applications, and opportunities. *Pervasive and Mobile Computing* 5(4), 277–298 (2009)
6. Defanti, T., Sandin, D.: Final report to the national endowment of the arts. US NEA R60-34-163. University of Illinois at Chicago Circle (1977)
7. Delamare, W., Coutrix, C., Nigay, L., et al.: Pointing in the physical world for light source selection. In: Proceedings of Designing Interactive Lighting workshop at DIS 2012 (2012)
8. Foxlin, E., Harrington, M.: Weartrack: A self-referenced head and hand tracker for wearable computers and portable vr. In: Wearable Computers, The Fourth International Symposium on. pp. 155–162. IEEE (2000)
9. Krueger, M.W.: Artificial reality II, vol. 10. Addison-Wesley Reading (Ma) (1991)
10. Nickel, K., Stiefelhagen, R.: Pointing gesture recognition based on 3d-tracking of face, hands and head orientation. In: Proceedings of the 5th international conference on Multimodal interfaces. pp. 140–146. ACM (2003)

11. Oikonomidis, I., Kyriazis, N., Argyros, A.A.: Efficient model-based 3d tracking of hand articulations using kinect. In: BMVC. pp. 1–11 (2011)
12. Rabbi, I., Ullah, S.: A survey on augmented reality challenges and tracking. *Acta Graphica znanstveni časopis za tiskarstvo i grafičke komunikacije* 24(1-2), 29–46 (2013)
13. Rukzio, E., Leichtenstern, K., Callaghan, V., Holleis, P., Schmidt, A., Chin, J.: An experimental comparison of physical mobile interaction techniques: Touching, pointing and scanning. In: *UbiComp 2006: Ubiquitous Computing*, pp. 87–104. Springer (2006)
14. Schlömer, T., Poppinga, B., Henze, N., Boll, S.: Gesture recognition with a wii controller. In: *Proceedings of the 2nd international conference on Tangible and embedded interaction*. pp. 11–14. ACM (2008)
15. Stokoe, W.C.: Sign language structure: An outline of the visual communication systems of the american deaf. *Journal of deaf studies and deaf education* 10(1), 3–37 (2005)
16. Sturman, D.J., Zeltzer, D.: A survey of glove-based input. *Computer Graphics and Applications, IEEE* 14(1), 30–39 (1994)
17. Vaughan-Nichols, S.J.: Game-console makers battle over motion-sensitive controllers. *Computer* 42(8), 13–15 (2009)
18. Vogel, D., Balakrishnan, R.: Distant freehand pointing and clicking on very large, high resolution displays. In: *Proceedings of the 18th annual ACM symposium on User interface software and technology*. pp. 33–42. ACM (2005)
19. Wang, Y., MacKenzie, C.L.: The role of contextual haptic and visual constraints on object manipulation in virtual environments. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. pp. 532–539. ACM (2000)
20. Wingrave, C.A., Williamson, B., Varcholik, P.D., Rose, J., Miller, A., Charbonneau, E., Bott, J., LaViola Jr, J.J.: The wiimote and beyond: Spatially convenient devices for 3d user interfaces. *Computer Graphics and Applications, IEEE* 30(2), 71–85 (2010)
21. Wu, Y., Huang, T.S.: Human hand modeling, analysis and animation in the context of hci. In: *Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on*. vol. 3, pp. 6–10. IEEE (1999)

Knotenmobilität in industriellen drahtlosen Sensornetzwerken

David Timmermann*

Advisor: Matthias Berning[†]

Karlsruhe Institute of Technology (KIT)
Pervasive Computing Systems – TECO

*uagta@student.kit.edu

[†]berning@teco.edu

Abstract. In den letzten Jahren sind drahtlose Sensornetzwerke in der Industrie zu einer Selbstverständlichkeit geworden. Neue Standards für diese Netzwerke sorgen dafür, dass viele Anforderungen an Netzwerke, wie etwa Fehlertoleranz und Latenz, problemlos umgesetzt werden können. Auch der Faktor der Knotenmobilität, welcher sich mit der optimalen Einbindung eines Knotens in ein Netzwerk beschäftigt, wird von allen Standards behandelt. Dabei lassen sich jedoch oft Unterschiede in der Art der Behandlung finden. Es gibt Standards die durch eine zentrale Kontrolle gewährleisten wollen, dass jeder Sensor optimal in das Netzwerk eingebunden ist; andere setzen auf Router in dem Netzwerk, um so die Belastung auf dem zentralen Knoten auf das Netzwerk zu verteilen. Weitere Standards zielen darauf, gänzlich ohne einen zentralen Knoten zu arbeiten.

Keywords: Industrielle drahtlose Sensornetzwerke, Knotenmobilität, IDSN

1 Einleitung

Die Anzahl an Sensoren hat in den letzten Jahren rapide zugenommen; auch in der Industrie sind nahezu alle Maschinen mit Sensoren ausgestattet. Zu Beginn waren diese Sensoren nur direkt an der Maschine auslesbar. Im Laufe der Jahre entwickelte sich, begünstigt durch Verbesserungen im Bereich der Funk und Netzwerktechnologie, ein Trend zu drahtlosen Sensornetzwerken. In diesen Netzwerken werden alle Sensordaten an eine zentrale Station im Netzwerk gesendet. Diese zentrale Station, auch Senke genannt, erfüllt zwei Aufgaben: Zum einen steuert und kontrolliert sie das Netzwerk, zum anderen wertet sie alle erhaltenen Daten aus. Da die Sensordaten meist entscheidende Informationen darüber beinhalten, ob ein Bearbeitungsschritt erfolgreich war, zum Beispiel eine Bohrung exakt genug, ist es wichtig, dass diese Daten auch an der Senke ankommen. Problematisch kann diese Übermittlung der Sensordaten sein, weil die Maschinen durch ihre Emissionen, welche beim Betrieb entstehen, die Funkverbindung des Netzwerkes negativ beeinflussen. Auch ist nicht bei jedem Sensor gewährleistet, dass dieser immer an einer festen Position ist, zum Beispiel wenn sich dieser

Sensor an einem Fertigungsroboter befindet. Die ständige Veränderung des Netzwerkes und die daraus resultierende Notwendigkeit, die Knoten immer neu in das Netzwerk einzupflegen, um eine optimale Verbindung zu garantieren, wird als Knotenmobilität bezeichnet.

Um sich die Bedeutung von Knotenmobilität in diesen Netzwerken einfacher vorstellen zu können, sollten zwei mögliche einfache Szenarien betrachtet werden.

In Szenario 1 wird eine mobile Roboterplattform betrachtet, die in einer Fabrik Waren zu beliebigen Punkten liefert. Hier sollen Sensoren zum Beispiel den Zustand der Ladung überwachen. Durch die Bewegung verweilen die Sensorknoten nie lange an einem bestimmten Ort des Netzwerkes. Dieses hat zur Folge, dass die Sensoren immer wieder neu in das Netzwerk eingebunden werden müssen, um einen Verbindungsabbruch zu vermeiden.

Szenario 2 handelt von einem Schweißroboter an einer Bandstraße, welcher Sensordaten sendet, diese geben Rückschluss darauf, ob ein Schweißpunkt gut gesetzt wurde oder nicht. Der Sensor bewegt sich nur in einem sehr kleinen Umkreis im Netzwerk und eine Beeinflussung der Signalstärke tritt meist über Interferenzen durch den Schweißvorgang auf.

In beiden Szenarien ist die Knotenmobilität unterschiedlich stark ausgeprägt und daher bieten sich für jedes Szenario andere Standards aus dem Bereich der industriellen drahtlosen Sensornetze an.

In der folgenden Arbeit wird sich primär damit beschäftigt, wie verschiedene Standards aus dem Bereich der industriellen drahtlosen Sensornetze (im folgenden mit IDSN abgekürzt) diese Knotenmobilität behandeln. Dazu wird im ersten Abschnitt, die Knotenmobilität und die Möglichkeiten diese zu behandeln betrachtet. Im zweiten Abschnitt, wird der grundlegende Aufbau eines IDSNs genauer vorgestellt und auf grundlegende Anforderungen an dieses eingegangen. Im letzten Abschnitt folgt eine Vorstellung der Standards für IDSNs sowie deren Bewertung im Hinblick auf die Knotenmobilität.

2 Knotenmobilität

Der Begriff Knotenmobilität wird vor allem im Bereich Mobile Computing viel genutzt. Er umschreibt das Verhalten eines Knoten bei der Bewegung innerhalb eines drahtlosen Netzwerkes und was gegebenenfalls zu unternehmen ist, um den Knoten bestmöglich im Netzwerk einzubinden. Dieses kann zum Beispiel bedeuten, den Knoten bei einer neuen Basisstation anzumelden oder seine Routingtabelle zu aktualisieren. Allgemein lässt sich die Reaktion hierbei unterscheiden in ein statisches Verhalten, dem Handoff oder Routing, und einem eher dynamischen Verhalten dem on-demand Routing.

Alle Informationen aus dem Bereich Knotenmobilität sind in dem Buch [12] ausführlicher vorzufinden.

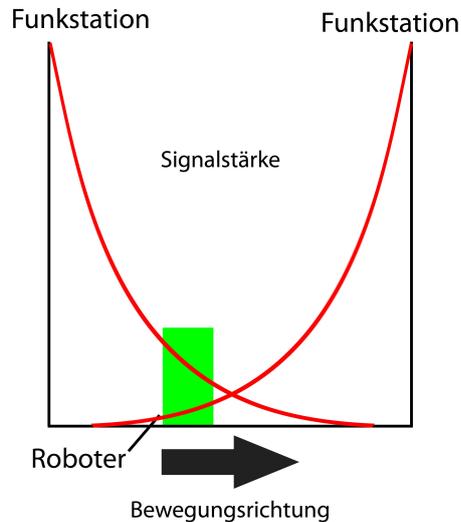


Fig. 1. Szenario Handoff

2.1 Handoff

Im Kontext der Knotenmobilität wird unter Handoff verstanden, dass ein Knoten dauerhaft kontrolliert wird, ob und wie er innerhalb des Netzes positioniert ist und ob es eine bessere Möglichkeit gibt ihn in das Netz einzubinden. In einem Handynetz wird zum Beispiel versucht immer das stärkste Signal zu nutzen. Ob und wann der Wechsel von einem schwächeren zu einem stärkeren Signal durchgeführt werden soll, wird durch den Handoff bestimmt.

In dem Beispielszenario in Fig. 1 ist zu sehen, wie ein Roboter sich durch einen Raum bewegt. Dabei ist er zu Beginn mit der Station auf der linken Seite verbunden. Ab einem gewissen Zeitpunkt ist das Signal jedoch so schlecht, dass ein Wechsel zu der zweiten Station sinnvoll ist und ein Handoff ausgeführt wird.

Um diesen Handoff zu initiieren, muss zunächst ein Algorithmus entscheiden, ob es ein sinnvolles Signal gibt, zu welchem der Handoff ausgeführt werden soll. Um dieses zu entscheiden, kann einer der folgenden Algorithmen genutzt werden.

Relative Signal Strength: Bei der „Relative Signal Strength“ Methode werden dauerhaft die eingehenden Signale gemessen. Falls ein Signal gefunden wird, welches stärker ist als das momentane, wird dieses genutzt. Das kann allerdings zu ständigem Signalwechsel führen, obwohl sich die Signalstärken nur marginal unterscheiden.

Relative Signal Strength with Threshold: Die „Relative Signal Strength with Threshold“ Methode wechselt erst zu einem besseren Signal, sobald das aktuelle Signal einen gewissen Grenzwert unterschreitet.

Relative Signal Strength with Hysteresis: „Relative Signal Strength with Hysteresis“ initiiert erst einen Wechsel zu einem stärkeren Signal, wenn dieses um ein gewisses Vielfaches stärker ist als das aktuelle Signal.

Relative Signal Strength with Hysteresis and Threshold: Bei der „Relative Signal Strength with Hysteresis and Threshold“ Methode wird die „Relative Signal Strength with Threshold“ Methode und die „Relative Signal Strength with Hysteresis“ Methode kombiniert. Hier erfolgt ein Wechsel zu einem stärkeren Signal also erst, wenn das aktuelle Signal schwächer als ein gewisser Grenzwert ist und das neue Signal um ein gewisses Vielfaches stärker ist als das Aktuelle.

Die bisherigen Algorithmen ähneln sich alle in der Hinsicht, dass bereits eine Bewegung des Knotens stattgefunden hat und auf diese Bewegung reagiert wird. Es ist jedoch auch möglich die bisher gesammelten Daten zu analysieren und auf Basis dieser Analyse zu versuchen, die Zukunft vorherzusagen. An diesem Punkt setzt das Verfahren „Prediction Techniques“ an.

Prediction Techniques: Bei dieser Methode wird versucht vorherzusagen, wo sich der Knoten in der Zukunft befindet. Auf Basis dieser Vorhersage ist es möglich zu bestimmen, welches Signal am besten geeignet ist.

Um den Handoff Vorgang durchzuführen existieren, mehrere Protokolle die vorgeben, welche Rolle die verschiedenen Komponenten (Netzwerk, Basisstation, mobile Station) bei einem Handoff übernehmen. Dabei gilt es zu entscheiden, wer die Messergebnisse bewertet und dann entscheidet den Handoff zu initiieren, als auch festzulegen, wer für die einzelnen Teile des Wechsels auf ein neues Signal zuständig ist.

Network-Controlled Handoff: Hier misst das Netzwerk die Signalstärken an bestimmten Basisstationen und entscheidet dann aufgrund eines Entscheidungsalgorithmus, ob ein Handoff ausgeführt wird. Durch dieses Verfahren ist die Information für die Signalqualität aller Knoten nur an einem Punkt im Netzwerk zu finden. Das gesamte Handoffverfahren nimmt mit diesem Protokoll ca. 100 - 200ms in Anspruch.

Mobile-Assisted Handoff: Bei diesem Verfahren misst die mobile Station die Stärke des Signals und leitet die Messergebnisse an das Netzwerk weiter. Dieses entscheidet dann auf Grundlage des benutzten Algorithmus ob es zu einem Handoff kommen soll. Der komplette Handoff bei diesem Verfahren benötigt bis zu

einer Sekunde, was vor allem an der Einbindung des Controllers der Basisstation liegt, welcher für die Allokierung und Freigabe von Funkkanälen zuständig ist und auch den eigentlichen Handoff ausführt.

Mobile-Controlled Handoff: Das Protokoll „Mobile-Controlled Handoff“ zeichnet sich dadurch aus, dass der Knoten vollkommen selbstständig agiert. Er misst die verschiedenen Signalstärken der Basisstationen und entscheidet dann, ob gegebenenfalls ein Handoff ausgeführt werden soll. Durch die vollkommene Selbstständigkeit ist eine Gesamtzeit vom ca. 100ms erreichbar.

Da ein Handoff stets Zeit benötigt und es gegebenenfalls zu einer schnellen Abfolge von Handoffs kommen kann, existieren die sogenannten Handoffschemata. Diese unterstützen dabei, realistisch zu bewerten, wie häufig ein Handoff stattfinden wird und wie lange ein Knoten mit einem Signal verbunden ist. Auf Basis der Schemata ist es möglich, besser zu entscheiden, ob ein Handoff notwendig ist oder nicht. Bei dem Schemata kann einerseits unterschieden werden zwischen den eher einfach strukturierten Schemata, wie etwa dem "Hong and Rappaport's Traffic Model", "El-Dolil et al.'s Traffic Model", "Steele and Nofal's Traffic Model" und "Xie and Kuek's Traffic Model". Andererseits gibt es Schemata, die besondere Anforderung haben, zum Beispiel Prioritäten "Priority Scheme", oder mit einer Schlange arbeiten "Handoff Call Queuing Scheme". Welches Schemata zum Einsatz kommt hängt zum einem von der Umgebung ab in welcher sich bewegt wird, zum anderen von dem eingesetzten System.

2.2 Routing

Routing in einem mobilen Netzwerk, insbesondere in einem mobilen ad-hoc Netzwerk, lässt sich generell unterscheiden in das proaktive Routing und in das reaktive oder on-demand Routing. Im proaktiven Routing existiert zu jedem Zeitpunkt zu jedem Knoten im Netzwerk eine Route; Veränderungen im Routing werden sofort an alle Teilnehmer des Netzwerkes weitergeleitet. Beim reaktiven Routing wird von den Protokollen erst eine Route zwischen zwei Knoten erzeugt, wenn diese benötigt wird. Insbesondere das on-demand Routing gehört zu einem dynamischen Verfahren im Bereich der Knotenmobilität. Als ein Mittelweg zwischen den Routing Verfahren existiert das Hybrid Routing.

Proaktives Routing: Beim Proaktiven Routing wird oftmals das "Destination-sequenced distance vector routing" (DSDV) Protokoll eingesetzt. Dieses basiert auf einer Erweiterung des Bellman-Ford Algorithmus. Aufgrund der Tatsache, dass für alle Knoten im Netzwerk eine Route erzeugt wird, unabhängig davon, ob diese eine Route benötigt oder nicht, entsteht eine unnötige Belastung des Netzwerkes. In Fig. 2 ist ein Beispielszenario dargestellt in welchem eine Route von Knoten A nach E(roter Pfad) zu sehen ist.

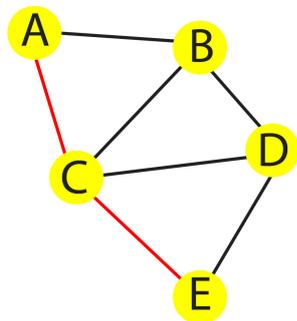


Fig. 2. Route DSDV, DSR, AODV

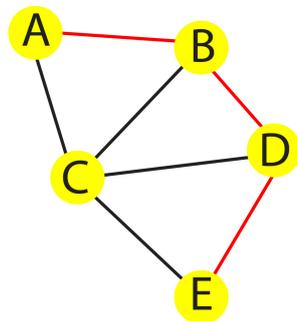


Fig. 3. Route SSA

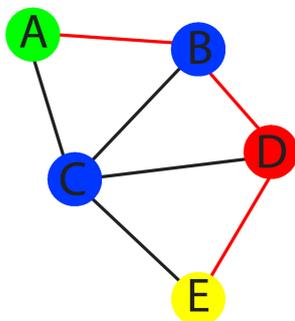


Fig. 4. Route TORA

On-Demand Routing: Um das Problem der nicht benötigten Route zu minimieren, wird beim on-demand Routing nur eine Route erzeugt, wenn diese auch benötigt wird. Ein Protokoll, welches in diesem Bereich arbeitet, muss drei grundlegende Anforderungen erfüllen. Erstens muss es die genaue Route bestimmen können, zweitens sind Regeln erforderlich, wie die Daten über diese Route geleitet werden und drittens muss die Route überwacht und gewartet werden. Die bekanntesten Protokolle in diesem Bereich heißen "dynamic source routing" (DSR), "signal stability-based adaptive routing" (SSA), "ad hoc on-demand distance vector routing" (AODV) und "temporally ordered routing algorithm" (TORA).

DSR: DSR ist ein Verfahren, welches beim Bestimmen einer Route vor allem auf einen Broadcast vom Ausgangspunkt aus setzt, welcher über die Nachbarknoten weiterläuft und die endgültige Route per DSDV auswählt. Durch die Anwendung des DSDV Verfahrens ist die Route zwischen Knoten A und E im

Beispielszenario Fig. 2, die gleiche wie beim proaktiven Routing. Um Daten über die gefundene Route zu versenden, wird in jedem Header eines Datenpaketes die komplette Route gespeichert. Diese Art der Routenübermittlung wird auch Source-Routing genannt. Sollte bei der Weiterleitung eines Paketes festgestellt werden, dass ein Knoten auf der Route nicht mehr erreichbar ist, wird eine Error-Meldung an Ausgangsknoten geschickt.

AODV: AODV ähnelt stark dem DSR Verfahren, nutzt jedoch zum einem das next-hop Verfahren zur Weiterleitung von Datenpaketen, dabei wird im Header des Datenpaket nur die Route zum nächsten Knoten gespeichert. Die weitere Route wird aus den Routingtabellen der Knoten ausgelesen. Als weitere Besonderheit minimiert diese Variante die Kosten für die Wartung der Route.

SSA: Beim SSA Routing wird versucht stets eine Route zu finden, welche eine lange Lebenszeit hat. Als Anhaltspunkt für eine langlebige Verbindung wird die Signalstärke genommen. In Fig. 3 ist eine mögliche langlebige Route zwischen Knoten A und E in rot eingezeichnet, die schwarzen Verbindungen können hier als nicht so stabile Verbindungen angesehen werden. Bei der Weiterleitung von Daten wird hier ebenfalls das next-hop Verfahren genutzt.

TORA: TORA nutzt beim Finden der Route die Möglichkeit, den Knoten bestimmte Höhen zuzuweisen. Dabei hat der Ausgangsknoten die niedrigste Höhe und der Zielknoten die höchste. Eine Route hat bei TORA die Einschränkung, dass sie nie von einer höheren Höhe auf eine niedrige Höhe verlaufen kann. In Fig. 4 ist diese Höheneinteilung und die daraus resultierende Route für eine Route von Knoten A nach E eingezeichnet. In diesem Beispiel ist die Höhenreihenfolge wie folgt: Grün, Blau, Rot, Gelb (von niedrig nach hoch). Zur Weiterleitung wird bei TORA ebenfalls das next-hop Verfahren genutzt.

2.3 Hybrid Routing

Bei Hybrid Routing Verfahren wird versucht, die proaktive Routing Verfahren mit on-demand Routing Verfahren zu verbinden. Ein Beispiel hierfür ist das "zone routing protocol" (ZRP). Dieses unterteilt zunächst das Netzwerk in verschiedene Zonen. Zwischen den Zonen wird on-demand Routing genutzt, innerhalb Zonen wird proaktives Routing angewandt. So wird versucht, aus beiden Routingarten die positiven Aspekte zu verbinden.

3 Industrielle drahtlose Sensornetzwerke

3.1 Aufbau eines industriellen Sensornetzwerkes

In Fig. 5 ist der schematische Aufbau eines industriellen drahtlosen Sensornetzwerkes skizziert. Die *Sensorknoten* sind Sensoren, welche in der industriellen

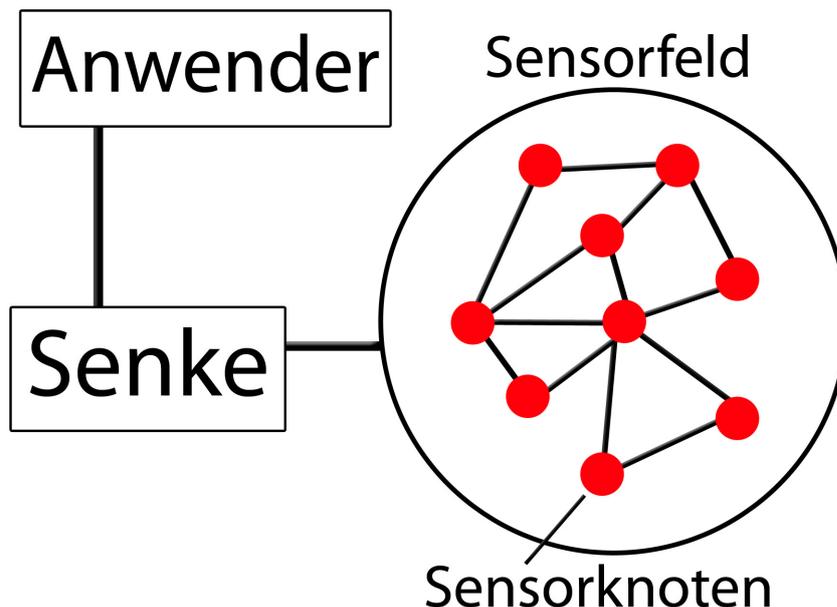


Fig. 5. Sensorfeld

Umgebung verteilt sind und Maschinen, Gefahrenbereiche etc. überwachen. Die Sensorknoten sind via Funk miteinander verbunden und bilden so das *Sensorfeld*. Die Daten, die die Sensoren messen, werden auf Routen aus dem Sensorfeld heraus geleitet zur *Senke*. Hierbei muss es nicht zwangsläufig nur eine Route vom Sensorfeld zur *Senke* geben. Die *Senke* kann vereinfacht als große Kontroll- und Steuereinheit aufgefasst werden. In ihr werden die erhaltenen Daten ausgewertet und gegebenenfalls gespeichert. Von der *Senke* aus existiert ebenfalls eine Verbindung zum *Anwender* des Systems. Die Verbindung kann hierbei sowohl direkt erfolgen durch Anschluss an Monitore oder auch indirekt über die Anbindung an das Internet.

3.2 Anforderungen

Bei dem Design von einem IDSN muss zunächst geklärt werden, welche Anforderungen zu erfüllen sind. Die Anzahl der unterschiedlichen Anforderungen ist sehr groß. Jedoch lassen sie sich auf die im Folgenden beschriebenen Hauptanforderungen reduzieren.

Hardwarebeschränkungen: Bedingt durch den Wunsch nach möglichst kleinen Sensoren, sind diese vor allem im Bereich der Batterien zur Energieversorgung als auch bei der Größe des Speichers sehr beschränkt. Als Folge dieses

Umstandes ist auch die allgemeine Verarbeitungsgeschwindigkeit oftmals sehr limitiert und die Funkreichweite relativ gering.[3][1]

Dynamische Topologie und schwierige Einsatzumgebung: In industrieller Umgebung kann es durchaus vorkommen, dass die Verbindung zu Sensorknoten abbricht, weil die Sensoren durch die Umgebungseffekte wie etwa Vibrationen, Schmutz, Hitze oder Funküberlagerungen gestört werden. Dieses führt zu einer sich verändernden Netzwerktopologie, auf die reagiert werden muss.[3]

Großflächige Verteilung und ad-hoc Architektur: Die Anzahl von Sensoren in IDSNs unterscheidet sich deutlich von Netz zu Netz. So sind in dem einen nur hundert, in einem anderen mehrere tausend zu finden. Gerade in IDSNs sind Sensorknotendichten von 20 Sensorknoten pro m^3 möglich[1]. Diese sind meist nicht strukturiert angeordnet, so dass jedes Netz in der Art seines Aufbaues einzigartig ist. Durch das Fehlen der festen Struktur ist die Steuerbarkeit des IDSNs dazu gezwungen, autonom das Netzwerk aufzubauen und zu erhalten.[3]

Quality-of-service(QoS) Anforderungen: Bedingt durch die vielen verschiedenen Einsatzgebiete von IDSNs, gibt es viele unterschiedliche QoS Anforderungen. Im Allgemeinen bezieht sich die QoS auf die Übereinstimmung der Daten, die an der Senke ankommen und dem, was in der industriellen Umgebung wirklich passiert. Für eine möglichst große Übereinstimmung müssen hier einige Schwierigkeiten überwunden werden. Dazu zählen:

Zuverlässigkeit: Die Zuverlässigkeit ist vor allem in einem drahtlosen Sensornetzwerk von entscheidender Bedeutung. Viele Faktoren (wie etwa Umgebungseinflüsse oder Beschädigung) können dafür sorgen, dass ein Sensor kurzzeitig oder gar dauerhaft nicht erreichbar ist oder auch falsche Daten übertragen werden. Bei IDSNs sind Bitfehler-raten von 10^{-2} bis 10^{-6} [3] möglich. Diese sollten jedoch das Sensornetzwerk insgesamt nicht komplett blockieren, so dass es weiterhin zuverlässig funktioniert.[1][6][4]

Latenz: In den meisten IDSNs sind die zu übertragenden Daten relativ klein, so dass auch mit einer geringen Datenübertragungsrate das Netzwerk schnell arbeitet. Der limitierende Faktor in dem Bereich der Latenz ist die Datenverarbeitungsrate. Dieses ist besonders in großen Netzwerken, in denen schnell viele Daten an der Senke ankommen, eine schwer zu erfüllende Anforderung.[4]

Sicherheit: Die Sicherheit ist eine weitere Hauptanforderung. Das Netzwerk muss vor Angriffen von Außen geschützt werden, so dass es nicht möglich ist

das komplette Netzwerk (und damit auch die komplette Umgebung in der dieses eingesetzt ist) zu blockieren. Aber auch die Daten der Sensoren sollten vor unautorisierten Zugriffen geschützt werden.[3][4]

Einbindung anderer Netzwerke: Besonders in IDSNs ist es notwendig, dass die Daten von überall und zu jederzeit abgerufen werden können. Dazu ist die Integration von anderen Netzen unabdingbar, insbesondere wird hier oft das Augenmerk auf die Einbindung der Internet Protokoll Architektur (IP-Protokolle) wert gelegt.[3][6]

4 Standards

Im Themenkomplex der IDSNs wurden bereits viele Standards erarbeitet, welche sich in vielen Bereichen stark unterscheiden. So gibt es zum einem Unterschiede bei der eingesetzten Übertragungstechnik und darin, wie konkret die Standards die IDSNs einschränken. Alle im folgenden vorgestellten Standards bauen auf den IEEE 802.15.4 Standard auf.

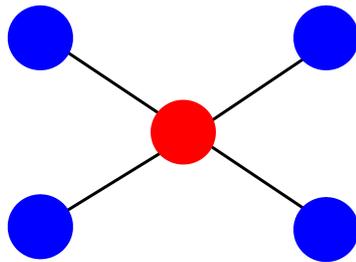


Fig. 6. Topologie Stern

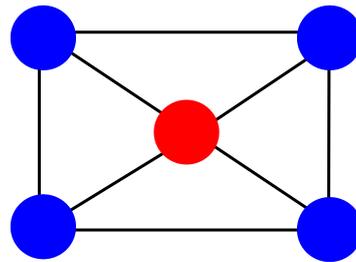


Fig. 7. Topologie Mesh

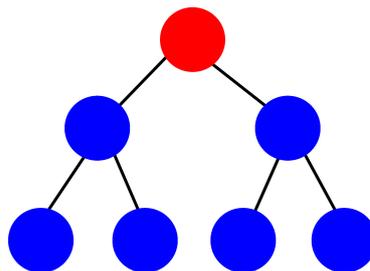


Fig. 8. Topologie Baum

Topologie: In IDSNs wird oft eine von drei Topologien verwendet: Die Stern-Topologie (Fig. 6), die Mesh-Topologie (Fig. 7) und die Baum-Topologie (Fig. 8). In allen Abbildungen repräsentiert der rot gefärbte Knoten die Senke, die blau gefärbten Knoten stellen die Sensorknoten dar.

Die beiden am meisten verwendeten Topologien sind Stern und Mesh. Bei beiden bildet die Senke den zentralen Knoten im Zentrum des Netzwerkes. Während die Stern-Topologie eine Kommunikation der Knoten nur in Richtung der Senke ermöglicht und eine Kommunikation der Sensorknoten untereinander nicht möglich ist, erweitert die Mesh-Topologie den Stern um weitere Verbindungen zwischen den verschiedenen Sensorknoten, so dass diese auch untereinander kommunizieren können.

Einen anderen Ansatz wählt die Baum-Topologie. In dieser bildet die Senke den Wurzelknoten. Die restlichen Knoten sind der Senke untergeordnet, jedoch kann die Senke nur direkt mit den beiden Kinderknoten kommunizieren. Wenn mit tiefer gelegenen Knoten kommuniziert werden soll, erfolgt dieses über die Kinderknoten.

IEEE 802.15.4: Der IEEE 802.15.4[11] Standard stellt die Grundlage für viele Standards im Bereich der IDSNs dar. Als besondere Merkmale hat er einen geringen Stromverbrauch, einen geringen Abstand der Knoten (ca. 10m) und eine eher geringe Datenübertragungsrate. So ist etwa im 2,45GHz Frequenzband, welches am meisten verwendet wird, eine Übertragungsrate von maximal 250 kBit/s möglich. 2007 wurde eine erweiterte Fassung der IEEE 802.15.4a Standard veröffentlicht. Dieser kann auch das Ultrabreitbandverfahren (UWB)[9] nutzen. Hierbei kann ein deutlich größerer Frequenzbereich genutzt werden, damit ist eine Datenrate von mindestens 110 MBit/s erreichbar. In einem Netzwerk, welches auf dem IEEE 802.15.4 Standard aufbaut, wird grundlegend unterschieden zwischen "Reduced Function Devices" (RFD) und den "Full Function Devices" (FFD). Dabei sind RFDs aufgrund der Tatsache, dass sie nur Teile des Protokolls implementieren nur in der Lage mit FFDs direkt zu kommunizieren, nicht jedoch mit anderen RFDs. In den meisten IDSNs werden Knoten allerdings fast ausschließlich als FFDs realisiert, um ein leichter zu handhabendes und leichter erweiterbares Netzwerk zu schaffen. Bei diesem Standard kann zwischen der Stern-Topologie, der Mesh-Topologie oder der Baum-Topologie gewählt werden. Aufgrund der Idee, einen einfachen Standard zu schaffen, wurden nur die beiden untersten Schichten des OSI-Schichtenmodells implementiert; die Bitübertragungsschicht und die Vermittlungsschicht. Dadurch obliegt es Standards welche auf dem IEEE 802.15.4 Standard aufbauen, die anderen Schichten zu implementieren und somit Routingverfahren und Ähnliches zu implementieren. Hierbei müssen sie auch mit einigen Besonderheiten umgehen können; so braucht ein Sensorknoten, welche Daten weiterleiten möchte, ca. 15ms, um den kompletten Datenübertragungsvorgang abzuschließen.

4.1 ZigBee

Der Zigbee Standard[5][2] setzt auf den IEEE 802.15.4 Standard auf und erweitert die beiden vorhandenen Schichten noch um eine Vermittlungsschicht und eine Anwendungsschicht. Die Vermittlungsschicht ist hierbei für das Organisieren und Routen in dem Multihopnetzwerk zuständig. Dabei wird auf die Funktionalitäten welche der IEEE 802.15.4 Standard zur Verfügung stellt, aufgebaut. In dieser Schicht wird auf die vom IEEE 802.15.4 vorgegebenen Klassen von Knoten aufgebaut und weiter verfeinert. Ein Sensorknoten wird meist als ein Endgerät (ZigBee End Device, ZED) definiert und kann als RFDs realisiert werden. Dadurch besitzt er nur Teilfunktionalitäten der ZigBee Protokolle. Knoten können auch Router sein (ZigBee Router, ZR), welche als FFDs implementiert werden müssen und für das Routing im ZigBee Netzwerk zuständig sind. Mit diesen beiden Knoten kann einer der drei geläufigen Topologien realisiert werden. Des Weiteren existiert ein Koordinator (ZigBee coordinator, ZC), welcher ebenfalls als FFD realisiert ist. Der Koordinator übernimmt die grundlegende Verwaltung des Netzes und ist auch für die Abgrenzung zu anderen Netzen zuständig. Wenn ein neuer Knoten in das Netz eintreten möchte, wird von der Vermittlungsschicht aus ein Prozess gestartet, welcher das Netzwerk erkundet. Nachdem dieser abgeschlossen ist, wird von einer höheren Schicht entschieden, welchem ZigBee Netzwerk sich der Knoten anschließen soll (es können sich mehrere ZigBee Netzwerke überlagern). Der Knoten wird dann in das entsprechende Netzwerk aufgenommen. In den meisten Fällen wird versucht, eine Baumstruktur zu erhalten mit dem Koordinator als Wurzel, dem Routern als inneren Knoten und den Endgeräten als Blättern. Hier gestaltet sich das Routing einfach, da die Router hierbei nur die Adresse von ihrem Vorgänger kennen und die beiden Adressen der Kinder. Komplizierte Strukturen wie Mesh-Netzwerk sind eher seltener anzutreffen und benötigen ein kompliziertes Routing unter zu Hilfenahme von Routingtabellen. Die Anwendungsschicht stellt ein Framework für die verteilte Anwendungsentwicklung und Kommunikation zur Verfügung. Zu diesem Zweck ist sie unterteilt in Anwendungsframework (Application Framework, AF), ZigBee Geräte Objekt (ZigBee Device Object, ZDO) und Anwendungsunterschicht (Application Sub layer APS).

4.2 WirelessHART

WirelessHART (WH)[7][10] baut ebenfalls auf dem IEEE 802.15.4 Standard auf und führt ein Zeitmultiplexverfahren (Time Division Multiple Access TDMA) zur Zugriffskontrolle ein, welches die Übertragung stabilisiert. Hierbei sind Zeitschlitze von 10 ms vorgegeben. Bei WH übernimmt ein Netzwerkmanager (NM) die Kontrolle über das Netzwerk und verteilt die Zeitschlitze für das Zugriffsverfahren. Der NM kann noch weiter unterteilt werden in ein Gateway, einen Security Manager und einen Network Manager. Da diese drei Systeme jedoch meist in einem Knoten vereint sind, wird meist nur vom NM gesprochen. Der NM ist weiterhin zuständig für die Aufnahme von Knoten in das Netzwerk. Im Gegensatz zum ZigBee Standard gibt es im WH nur Knoten, welche auch in

der Lage sind, aktiv am Routing teilzunehmen. Daher werden alle Knoten als FFDs realisiert. WH setzt grundsätzlich nur die Mesh Topologie und die Stern-Topologie ein. Um in ein Netzwerk beizutreten, muss ein neuer Knoten in WH dieses beim NM anmelden, welcher den Beitritt dann steuert. Dieses ermöglicht dem NM, das Netzwerk besser zu steuern, sorgt jedoch auch für Probleme bei der Knotenmobilität. Durch die Mesh Topologie existieren in einem WH Netzwerk meist mehrere Routen. So kann beim Ausfallen eines Knoten automatisch die Verbindung über einen anderen Knoten laufen. Im Bereich Sicherheit verfügt WH über gewisse Sicherheitsfunktionen, die stets aktiviert sind.

4.3 ISA100.11a

Auch der ISA100.11a Standard[10] baut auf dem IEEE 802.15.4 Standard auf und nutzt TDMA. Im Gegensatz zu WH, existiert in diesem Netzwerk eine Trennung zwischen Knoten, welche Router sind (also als FFDs realisiert sind) und den Knoten, welche nur als Ein/Ausgabeknoten definiert sind (also durch RFDs realisiert werden können) und nur Endpunkte im Routing sind. Aus Gründen der Erweiterbarkeit werden in den meisten ISNs jedoch fast alle Knoten als Router implementiert. Identisch mit WH, ist der Netzwerk/System Manager (NM), welcher erneut theoretisch in die drei Unterkomponenten Gateway, Security Manager und System Manager unterteilt werden kann. Die Einfügung eines neuen Knotens in das Netzwerk erfolgt ebenfalls wie bei WH, zentral gesteuert. Als typische Topologie sind auch hier Stern oder Mesh vorzufinden. Im Vergleich zum WH, findet der Anwender zwar ein ähnliches Netzwerk wie bei WH vor, hat jedoch mehr Freiheiten. So sind die Zeitschlitze für das TDMA frei wählbar; im Bereich der Sicherheit kann er selber Einstellungen vornehmen.

4.4 6LoWPAN

IPv6 over Low power Wireless Personal Area Network (6LoWPAN)[8] ist ein Standard, der auf dem IEEE 802.15.4 Standard aufbaut und versucht, viele Eigenschaften des IP-Protokolls zu adaptieren. Als besonderes Merkmal kann 6LoWPAN IPv6-Pakete sehr effektiv versenden, was allerdings den 6LoWPAN-Adaption-Layer in der Vermittlungsschicht nötig macht, da der IEEE 802.15.4 Unterbau eigentlich nur kleine 127Byte große Datenpakete ermöglicht, IPv6 jedoch mindestens 1280Bytes große Pakete benötigt. Bei 6LoWPAN ist es möglich das Routing auf zwei verschiedenen Ebenen vorzunehmen: So existiert zum einem die Möglichkeit auf der Sicherungsschicht Routing zu betreiben, dem sogenannten Mesh-Routing. Als Alternative erfolgt das Routing auf der Vermittlungsschicht, hierbei wird auch vom IP-Routing gesprochen. Dabei wird die Abstammung vom IP-Protokoll ausgenutzt, um Algorithmen, welche aus diesem Protokoll bereits bekannt sind, zu benutzen. Bei 6LoWPAN besteht des Weiteren die Möglichkeit sowohl auf der Sicherungsschicht als auch auf der Vermittlungsschicht einen Knoten in das Netzwerk aufzunehmen.

5 Bewertung

5.1 Knotenmobilität in industriellen drahtlosen Sensornetzwerken

Die Knotenmobilität in IDSNs spielt eine große Rolle: So sind Knoten durch Interferenzen der Umgebung von schwankenden Signalstärken betroffen und können stellenweise sogar kurzzeitig nicht erreichbar sein. Durch die Art der Netzwerke wird in diesen meist eine Behandlung der Knotenmobilität durch Techniken aus dem Bereich des Routings benutzt. Dieses geschieht, weil die meisten Standards, mit welchen IDSNs aufgebaut werden, sich grob an dem OSI-Schichtenmodell orientieren. Aber auch Techniken, welche aus dem Bereich Handoff stammen, sind anwendbar.

Um gewisse Anforderungen wie etwa die Zuverlässigkeit, optimal bereit zu stellen, sind Handoff Strategien optimal, da diese beim Wechsel zu einer neuen Basisstation ohne Verbindungsabbruch arbeiten. On-demand Routingverfahren hingegen können nicht für eine Route garantieren, so dass es möglich ist, dass zu einem Zeitpunkt keine Route gefunden wird. Das hat einen Verbindungsabbruch zur Folge. Allerdings ist nicht jedes Verfahren aus dem Segment des Handoffs immer geeignet. So benötigt der Mobile-Assisted Handoff eine relativ lange Zeitspanne (bis zu einer Sekunde), um den kompletten Handoff abzuschließen. Jedoch ist nicht für jedes Netzwerk gewährleistet, dass diese Sekunde zur Verfügung steht. Auch sind nicht alle Standards, welche im Bereich der IDSN geläufig sind, in der Lage alle Verfahren aus dem Feld der Knotenmobilität anzuwenden aufgrund ihres Konzeptes oder der technischen Limitierung. Oftmals wird bei der Planung eines IDSNs überprüft, welche Verfahren eingesetzt werden können und mit Hilfe einer Kostenfunktion ein Verfahren ausgewählt, welches für das Netzwerk das optimale ist. Diese Kostenfunktion ist variable und hängt insbesondere von den QoS-Anforderungen des Netzwerkes ab.

5.2 Bewertung der Standards

Um die Standards in Hinsicht auf Knotenmobilität gut bewerten zu können, empfiehlt es sich diese in Konzepte einzuteilen.

Das erste Konzept beinhaltet eine zentrale Steuerung des Netzwerkes durch einen Knoten. Sowohl WH als auch ISA100.11a setzen auf dieses Konzept. Durch den zentralen NM kann es in beiden Standards zu Problemen kommen, wenn ein neuer Knoten in das Netzwerk aufgenommen werden soll. So entsteht bei beiden eine erhöhte Netzwerklast, da sich ein neuer Knoten zunächst beim NM melden muss um in das Netzwerk aufgenommen zu werden. Da der NM fast immer der Senke entspricht, ist hier eine erhöhte Netzwerklast nicht wünschenswert. Allgemein ist die Aufnahme eines neuen Knoten in das Netzwerk bei diesem Konzept sehr umständlich. Auch abseits des Einfügens eines neuen Knotens sind Verfahren mit zentralem NM nicht für alle Verfahren der Knotenmobilität geeignet. Dadurch dass der NM stets sämtliche Informationen über das Netzwerk besitzen soll, sind bei diesem Konzept on-demand Routingverfahren nicht möglich, da sie eine dezentrale Organisation benötigen, in welcher die Knoten eine gewisse

Selbstständigkeit besitzen.

Das zweite Konzept nutzt eine dezentrale Steuerung mit einem zentralen Steuerknoten, der jedoch nur passiv im Netzwerk tätig ist. ZigBee ist ein Verfahren, das diese dezentrale Organisation zu einem gewissen Grad implementiert. Bei ZigBee ist der ZC nur passiv im Routing eingebunden, da dieser nur weiß, welche Knoten im Netzwerk sind. Bei diesem Konzept übernehmen die Router das Routing, daher ist hier die Aufnahme eines neuen Knoten in das Netzwerk meist leicht möglich. Soll jedoch ein Knoten in ein anderes Netzwerk überwechseln, so kann dieses hier nur mit einem Verbindungsabbruch realisiert werden.

Das letzte Konzept beinhaltet ebenfalls eine dezentrale Steuerung, allerdings entfällt hier der zentrale Knoten, welcher passiv im Netzwerk ist. Hier ist also erstmals ein wirkliches dezentral gesteuertes Netzwerk vorhanden. 6LoWPAN gehört (ähnlich wie ZigBee) zu einem Verfahren ohne zentrale Steuerung. Im Unterschied zu ZigBee ist 6LoWPAN jedoch noch um einiges offener und ohne zentralen Knoten gestaltet. Dieses ist primär durch die große Offenheit im Bereich des Routings gewährleistet. Durch das Routing auf zwei verschiedenen Ebenen existiert vor allem auf der Vermittlungsschicht die Möglichkeit Verfahren und Informationen zu nutzen, die auf den tiefer gelegenen Ebenen nicht vorhanden oder nutzbar sind.

Die Konzepte lassen sich in Bezug auf Knotenmobilität unterschiedlich bewerten. So ist im zentral gesteuerten Konzept die Knotenmobilität nur mit Methoden aus dem Bereich proaktives Routing behandelbar, dafür mit diesen jedoch sehr gut. Methoden aus dem Bereich des on-demand Routings sind nicht nutzbar. Auch lassen sich beim Handoff nicht alle Verfahren, wie etwa der Mobile-Controlled-Handoff, einsetzen.

Bei Konzepten mit einer dezentralen Steuerung, bei denen jedoch ein passiver zentraler Knoten vorhanden ist, lassen sich on-demand Verfahren einsetzen. Hier ist allerdings der Verbindungsabbruch beim Wechsel des Routers ein großes Hindernis, so dass auch hier das on-demand Routing schwer zu realisieren ist. Handoff Verfahren lassen sich bei diesem Konzept im Allgemeinen realisieren. In diesem Konzept ist die Knotenmobilität meist eher schwerer zu behandeln, da eine zentrale Übersicht über das Netzwerk fehlt.

Ebenfalls gut mit Knotenmobilität umgehen können Verfahren aus dem Konzept mit einem dezentral gesteuertem Netzwerk. Bei diesem Konzept existiert nahezu keine Einschränkung. Daher ist dieses Konzepte für proaktives als auch on-demand Routing geeignet, sowie für Handoff Verfahren. Bei Standards aus diesem Konzept obliegt es also dem Entwickler zu entscheiden, welches Verfahren er nutzen möchte, um die Knotenmobilität zu kompensieren. Zu beachten ist jedoch der höhere Einrichtungsaufwand im Vergleich zu den anderen Konzepten.

6 Fazit

In der Arbeit wurde zunächst die Knotenmobilität genauer betrachtet. Hierbei wurden zwei verschiedene Ansätze vorgestellt um die Knotenmobilität zu be-

handeln. Dem Handoff, ein eher statischer Ansatz und dem on-demand Routing, welcher als eine dynamische Methode angesehen werden kann. Weiterhin wurden die industriellen drahtlosen Sensornetzwerke genauer, in Hinblick auf Aufbau und Anforderungen die an diese gestellt werden, betrachtet. Es stellte sich heraus, dass jedes Netzwerk jeweils eigene Schwerpunkte besitzt, sich jedoch im Allgemeinen alle Anforderungen auf einige Grundlegende reduzieren lassen. Im Anschluss folgte eine kurze Vorstellung der verschiedenen Standards die im Bereich der industriellen drahtlosen Sensornetzwerke genutzt werden. Dabei zeigte sich dass viele dieser Standards auf den IEEE 802.15.4 Standard aufbauen und daher einige Gemeinsamkeiten aber auch Unterschiede haben. Am Ende wurden diese Standards in 3 Konzepte eingeteilt und die Konzepte auf ihre Möglichkeiten hin, mit Knotenmobilität umzugehen, bewertet.

Abschließend lässt sich feststellen, dass die Knotenmobilität in nahezu allen IDSNs eine Rolle spielt, jedoch gibt es bestimmte Netzwerke, wie etwa zu Beginn in Szenario 1 (mobile Roboterplattform) beschrieben, in denen diese Knotenmobilität stärker ausgeprägt ist. In dem gegebenen Fall wäre ein Verfahren, welches dem zweiten Konzept entstammt, sicherlich keine gute Wahl, denn hier hat besonders ZigBee durch die zwangsweise hohe Anzahl an Routerwechseln einen häufigen Verbindungsabbruch, welcher nicht erwünscht ist. Ein Standard aus der ersten oder dritten Konzeptgruppe wäre hier gut anwendbar, zum Beispiel der WH Standard, welcher durch seine redundanten Routen einen weiteren Schutz gegen Verbindungsabbrüche besitzt. Netzwerke, in denen die Knotenmobilität nur in geringem Maße auftritt, wie in Szenario 2 (Schweißroboter) skizziert, lassen sich effektiv mit allen Konzepten realisieren. Besonders der ZigBee Standard kann hier seine Stärken im Bereich des einfachen Routings und der geringen Latenzen ausspielen und wäre für diese Szenario eine mögliche Wahl. Werden die Erfahrung aus den beiden Szenarien zusammengefasst, kann resümiert werden, dass für jedes Netzwerk selbst analysiert werden muss, was für ein Grad an Knotenmobilität zu erwarten ist. Aufgrund dieser Analyse lassen sich meist einige Standards bereits ausschließen und der optimale wird daraufhin oftmals mittels einer Kostenfunktion bestimmt.

Wie sich diese Thematik in Zukunft weiterentwickelt, lässt sich nur schwer abschätzen. Klar ist jedoch, dass auch neue Standards im Bereich der IDSNs die Knotenmobilität nicht ignorieren können. Daher wird es stets einen Kompromiss geben zwischen der Behandlung der Knotenmobilität und dem Erhalt der Effizienz in einem IDSN.

References

1. Ian F Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. Wireless sensor networks: a survey. *Computer networks*, 38(4):393–422, 2002.
2. Paolo Baronti, Prashant Pillai, Vince WC Chook, Stefano Chessa, Alberto Gotta, and Y Fun Hu. Wireless sensor networks: A survey on the state of the art and the 802.15. 4 and zigbee standards. *Computer communications*, 30(7):1655–1695, 2007.

3. Vehbi C Gungor and Gerhard P Hancke. Industrial wireless sensor networks: Challenges, design principles, and technical approaches. *Industrial Electronics, IEEE Transactions on*, 56(10):4258–4265, 2009.
4. Ivan Howitt, Wayne W Manges, Phani Teja Kuruganti, Glenn Allgood, José A Gutierrez, and James M Conrad. Wireless industrial sensor networks: Framework for qos assessment and qos management. *ISA transactions*, 45(3):347–359, 2006.
5. Patrick Kinney et al. Zigbee technology: Wireless control that simply works. In *Communications design conference*, volume 2, 2003.
6. Lakshman Krishnamurthy, Robert Adler, Phil Buonadonna, Jasmeet Chhabra, Mick Flanigan, Nandakishore Kushalnagar, Lama Nachman, and Mark Yarvis. Design and deployment of industrial sensor networks: experiences from a semiconductor plant and the north sea. In *Proceedings of the 3rd international conference on Embedded networked sensor systems*, pages 64–75. ACM, 2005.
7. Sergio Montero, Javier Gozalvez, Miguel Sepulcre, and Gonzalo Prieto. Impact of mobility on the management and performance of wireless industrial communications. In *Emerging Technologies & Factory Automation (ETFA), 2012 IEEE 17th Conference on*, pages 1–4. IEEE, 2012.
8. Geoff Mulligan. The 6lowpan architecture. In *Proceedings of the 4th workshop on Embedded networked sensors*, pages 78–82. ACM, 2007.
9. Ian Oppermann, Matti Hämäläinen, and Jari Rinatti. *UWB: theory and applications*. John Wiley & Sons, 2005.
10. Stig Petersen and Simon Carlsen. Wireless versus isa100. 11a: the format war hits the factory floor. *Industrial Electronics Magazine, IEEE*, 5(4):23–34, 2011.
11. IEEE Computer Society. *IEEE Standard for Local and metropolitan area networks Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)*, volume 2011. IEEE Computer Society, 2011.
12. Ivan Stojmenovic. *Handbook of wireless networks and mobile computing*, volume 27. John Wiley & Sons, 2003.

Privatsphäre und Datenschutz beim Participatory Sensing

Christoph Klebsattel*

Betreuer: Anja Bachmann[†]

Karlsruher Institut für Technologie (KIT)
Pervasive Computing Systems – TECO

*christoph.klebsattel@student.kit.edu

[†]bachmann@teco.edu

Zusammenfassung. Diese Arbeit bietet eine grundlegende Einführung in das Themengebiet „Mobile Phone Sensing“. Der Schwerpunkt liegt hierbei auf „Participatory Sensing“ und den dort herrschenden Problemen bezüglich der Privatsphäre und des Datenschutzes der Nutzer. Die ausschlaggebende Frage ist, wie diese anonym bleiben können, obwohl sie (mindestens teilweise) sensible Daten über sich und ihre Umwelt mit anderen teilen. Wird diese Frage von einem Projekt nicht bedacht bzw. keine passende Lösung für das Problem gefunden, so können Gefahren für die Nutzer entstehen. Über Standort-Daten kann z.B. ein Wohnort ermittelt werden. Am Ende werden deshalb drei elementare Lösungsmöglichkeiten zum Begegnen dieser Gefahren aufgezeigt: Maßgeschneidertes Sensing und Nutzereinstellungen, anonymisierte Berichterstattung von Daten sowie das Überprüfen, Löschen, Speichern und Behalten von Daten als Kontrollmöglichkeit für den Nutzer.

Schlüsselwörter: Privacy, Data Protection, Participatory Sensing, Activity Recognition

1 Einleitung und Motivation

Smartphones wurden in den letzten Jahren und Monaten mit immer besserer Hardware ausgestattet, und dieser Trend wird vermutlich anhalten. In allen aktuellen Modellen namhafter Hersteller findet man so eine große Auswahl an Sensoren, unter anderem einen Beschleunigungsmesser, einen Kompass, ein Gyroskop, einen GPS-Sensor, ein Mikrofon und eine Kamera [26, p. 140].

Ihre Verwendung zur Erhebung von Messwerten, z.B. die Lärmbelastung an unterschiedlichen Orten [30], wurde hierdurch deutlich einfacher und macht das sog. „Mobile Phone Sensing“ attraktiver - sowohl für die Industrie [39] als auch für Forschungsprojekte [43,11]. Gleichzeitig lassen sich dank der mittlerweile riesigen Anzahl an Smartphones [27, p. 477], die weltweit im Betrieb sind, eine gleichfalls große Menge an Daten durch „Participatory Sensing“, einer speziellen Form des „Crowdsourcing“, erheben.

Trotz aller Vorteile beim Sammeln und Verarbeiten von Sensorwerten dürfen Projekte, die diese nutzen möchten, jedoch nicht die Bereiche Privatsphäre und Datenschutz ignorieren. Zum einen müssen gesetzliche Vorgaben beachtet werden. Zum anderen haben, je nach Projekt, auch die Teilnehmer selbst Ansprüche wie Anonymität, die jedoch wiederum die Qualität der Daten verringern können [6, p. 1942].

Im zweiten Abschnitt dieser Arbeit geht es daher zuerst einmal um die grundlegenden Begriffe und Konzepte zur Erhebung und Verarbeitung von Sensorwerten mit dem Fokus auf mobilen Geräten. Stellvertretend dafür liegt hier der Fokus auf Smartphones. Im dritten Abschnitt wird daraufhin das „Participatory Sensing“ anhand einzelner Anwendungsgebiete genauer betrachtet. Danach werden grundlegende Fragen der Privatsphäre und des Datenschutzes besprochen. Im letzten Abschnitt wird ein Fazit der Arbeit gezogen und ein Ausblick auf ungeklärte Fragen bzw. Probleme gegeben.

2 Grundlagen/Stand der Technik

Der Hauptteil dieser Arbeit beschäftigt sich mit Problemen der Privatsphäre und des Datenschutzes beim Participatory Sensing, wobei der Fokus auf dem Erfassen von Daten über Smartphone-Sensoren und deren Verarbeitung liegt. Einleitend werden deshalb grundlegende Begriffe des „Mobile Phone Sensing“ sowie des „Participatory Sensing“ besprochen und Funktionsweisen der einzelnen dort üblichen Verfahren erläutert.

2.1 Mobile Phone Sensing

2.1.1 Sensoren Schaut man ein paar Jahre zurück in die Vergangenheit, so war damals noch spezielle Hardware wie z.B. die „Mobile Sensing Platform“ (MSP) [5] notwendig, um Aktivitätserkennung zu betreiben. Durch die heute produzierten Smartphones wurde diese Schwierigkeit im Prinzip vollständig beseitigt.

Zum einen enthalten Smartphones eine reichhaltige Auswahl an Sensoren, unter anderem einen Beschleunigungsmesser, einen Kompass, ein Gyroskop, einen GPS-Sensor, ein Mikrofon und eine Kamera [26, p. 140]. Zum anderen bietet ihr jeweiliges Betriebssystem in der Regel Schnittstellen („APIs“), über die Apps auf Werte der Sensoren zugreifen können [26, p. 145]. Dank dieser Kombination ist es möglich, die Erkennung von Nutzeraktivitäten relativ einfach zu realisieren. Beispielhafte Projekte und passende Apps existieren unter anderem in den Bereichen Gesundheit [7], soziale Netzwerke [33] und Umweltüberwachung [36]. Die Verteilung von Apps zur Aktivitätserkennung wurde in den letzten Jahren ebenfalls stark verbessert. Große Anbieter wie Apple, Google und Microsoft bieten mit ihren „App Stores“ Plattformen zum einfachen Verwalten und Verteilen der kleinen Anwendungen an viele Menschen weltweit [26, p. 140].

2.1.2 Continuous Sensing Hinzu kommt, dass sich der Aufwand, der durch Rechenaufgaben entsteht, auf Cloud-Server verlagern lässt [26, p. 144]. Hierdurch lassen sich Ressourcen des Smartphones wie CPU-Zeit und Stromverbrauch sparen und ein kontinuierliches Erheben von Messwerten („Continuous Sensing“) ermöglichen. Allgemein gilt jedoch, dass sich Entwickler beim Entwurf ihrer App den Trade-off zwischen Continuous Sensing und Grenzen der Anwendergeräte klar machen und angemessen entscheiden müssen [26, p. 146]. Grundfunktionen wie das Telefonieren, Versenden von Nachrichten sowie Surfen dürfen nicht durch das Sensing eingeschränkt werden - nicht zuletzt, weil sich ein Anwender vermutlich sonst vom Projekt abwenden wird.

2.1.3 Gerätekontext Ebenfalls wichtig zur Aktivitätserkennung beim Mobile Phone Sensing ist der Kontext des mobilen Gerätes. Wo befindet sich das Gerät gerade - in einer Großstadt, in einer Tasche, oder in einem Automobil? Ist die Umgebung eher leise oder laut? Wer hat das Gerät aktuell in Besitz? Wer spricht gerade? Bei diesen Fragen haben statistische Modelle oft Probleme. Daher gibt es zur Lösung des sog. „Kontextproblems“ andere Ansätze [26, p. 146].

Ein Ansatz ist es, Daten von Smartphones aus der Umgebung zusammen auszuwerten, um Effekte von Rauschen bei der Erhebung zu verringern [17]. In einem anderen Ansatz werden jene Daten ignoriert, deren Qualität nicht ausreichend hoch genug ist. Hierdurch können zusätzlich Ressourcen gespart und Fehler reduziert werden (Beispiel: SoundSense [29]). Ganz allgemein ist es wichtig, die Interpretation der Sensorwerte durch Training mit Daten unterschiedlicher Szenarien robust gegen Fehler und unerwartete Situationen zu machen.

Eine vergleichbare Problemstellung ist, wie genau und gut menschliches Verhalten durch ein mobiles Gerät nur aufgrund von Sensorwerten interpretiert werden kann. Selten sind die rohen Daten ohne Interpretation etwas wert, wenn sie z.B. dem Nutzer präsentiert werden sollen. Beim Projekt „Nike+“ erhalten Nutzer zwecks Fitness-Training Informationen darüber, wie weit sie gelaufen sind, und können diese Angaben mit denen von Freunden vergleichen [26, p. 147]. Ohne genaue Aktivitätserkennung der einzelnen Nutzer wären die Sensordaten wertlos.

2.1.4 Personalisierung vs. Anonymität Dieser Form der Personalisierung, um dem Nutzer Feedback geben zu können, stehen Datenschutz und Anonymität entgegen. Wie kann ein Nutzer anonym bleiben, obwohl er u.U. ununterbrochen Daten an ein Projekt sendet? Wie können personenspezifische Daten (vorab) herausgefiltert werden? Wie anonym sind Nutzer, wenn Sensoren, die in naher Umgebung sind, Daten miteinander teilen?

Ein grundlegender Ansatz, der z.B. von SoundSense [29] verfolgt wird und für ein wenig Datenschutz sorgt, ist es, Rohdaten direkt auf dem mobilen Gerät zu verarbeiten, dort jedoch nicht zu speichern und erst die verarbeiteten Daten an Cloud-Server zu schicken.

2.1.5 Zusammenfassung zum Mobile Phone Sensing Zusammenfassend lässt sich sagen, dass die primären Hindernisse beim Mobile Phone Sensing nicht infrastrukturelle Einschränkungen, sondern technische Barrieren sind (vgl. [26, p. 149]). Diese beziehen sich auf Probleme, die zum einen bezüglich Ressourceneinschränkungen und zum anderen aufgrund von Ansprüchen der Privatsphäre und des Datenschutzes entstehen. Gleichzeitig müssen die erhobenen Daten trotz Rauschens und Kontextabhängigkeit richtig interpretiert und dem Nutzer in sinnvoller Weise als Feedback zurückgegeben werden.

2.2 Crowdsourcing

Bevor im nächsten Kapitel auf das „Participatory Sensing“ eingegangen wird, beschäftigt sich dieses mit dem Crowdsourcing. Da das „Participatory Sensing“ eine Sonderform des Crowdsourcing ist, gelten viele der hier genannten Herausforderungen auch dort, zusätzlich zu weiteren, spezielleren.

2.2.1 Crowdsourcing Ein Verfahren, das sich in den letzten Jahren zum verteilten Lösen von Problemen immer mehr durchgesetzt und bewährt hat, ist das sog. „Crowdsourcing“ [19]. Hierdurch können Unternehmen zum einen Produktionskosten sparen und zum anderen ihre Arbeit effizienter erledigen.

Beim Crowdsourcing wird ein Problem oder eine Aufgabe, die traditionell von einem Angestellten eines Unternehmens gelöst bzw. erledigt wird, an eine größere Gruppe zur Bearbeitung abgegeben, wobei prinzipiell jeder zu dieser Gruppe gehören kann (vgl. [18]).

Typischerweise handelt es sich dabei um solche Probleme, die für Computer schwierig, für Menschen jedoch relativ einfach zu lösen sind. Um möglichst viele Personen zu erreichen und diesen das Lösen der Aufgaben zu erleichtern, basieren diverse Crowdsourcing-Projekte auf Websites.

Ein Beispiel ist der Dienst „Amazon Mechanical Turk“ (MTurk) [2]. Dort kann prinzipiell jeder, der möchte, Aufgaben einstellen oder Aufgaben anderer bearbeiten, z.B. solche, bei denen es um das Labeln von Daten geht oder auch um Umfragen, bei denen abgestimmt werden muss.

Ein anderes Beispiel ist die Online-Enzyklopädie Wikipedia [50], bei der jeder neue Artikel verfassen und existierende bearbeiten kann.

Andere Formen von Crowdsourcing-Projekten sind, neben Abstimmungs- und Informationsaustauschsystemen, „Social Games“, bei denen während des Spielens nützliche Daten als Nebenprodukt entstehen, sowie kreative Systeme, bei denen die menschliche Kreativität zum Erfüllen einer Aufgabe entscheidend ist.

2.2.2 Ansporn und Entlohnung Unabhängig von der Form des Projekts ist es, einen Ansporn für Personen zu schaffen, bei einem Projekt teilzunehmen. Die Gründe, warum jemand bei einem Crowdsourcing-Projekt teilnehmen möchte,

können dabei sehr unterschiedlich sein. Für die meisten Teilnehmer ist der finanzielle Anreiz entscheidend. Diese nutzen die Projekte teilweise als Nebenjob, zu einem großen Teil jedoch auch als primäre Einnahmequelle [47].

Doch nicht jedes Projekt basiert auf monetärer Entlohnung. Beim Videoportal YouTube [51] z.B. sind Aufmerksamkeit, gemessen in Klick- und Abonnentenzahlen, der wichtigste Antrieb für hochladende Nutzer [21]. Andere häufige Gründe sind Altruismus, die Möglichkeit etwas zu lernen sowie Kompetenz [37]. Mozilla beispielsweise hatte vor der Veröffentlichung seines Browsers Firefox in Version 1.0 einen kleinen Pool talentierter Entwickler. Diese arbeiteten nicht regelmäßig bei dem Projekt mit, sondern behoben einzelne Bugs oder fügten eine neue Funktion, die sie haben wollten, selbst ein [24].

2.2.3 Qualitätsmanagement und „Cheating“ Trotz eines Ansporns muss jedoch sowohl auf Qualität als auch auf Betrugsversuche („Cheating“) geachtet werden. Die Qualität der Lösungen für Aufgaben hängt nicht nur sehr von den Teilnehmern ab, sondern u.a. auch von der Entlohnung. Ein größerer finanzieller Anreiz kann zu einer besseren Qualität der Lösungen führen [15]. Gleichzeitig ist es jedoch auch möglich, dass Teilnehmer für sich Quantität über Qualität stellen, um mehr zu verdienen [32]. Dieser Effekt kann jedoch auch bei zu geringer Entlohnung auftreten, wenn z.B. zu viele unbrauchbare Lösungen gesammelt werden [23].

Zum gleichen, unerwünschten Ergebnis können betrügende Teilnehmer führen, die z.B. bei Umfragen zufällige Antworten auswählen oder falsche Angaben machen, um ihren Lohn zu maximieren. Prinzipiell basieren alle Verfahren, die diesem Problem entgegenwirken sollen, auf einer der zwei folgenden Ideen: Zum einen können Kontrollfragen gestellt werden, die automatisch überprüft werden, zum anderen können abgegebene Lösungen manuell kontrolliert werden [52, p. 770]. Ein sowohl kostengünstiges als auch zuverlässiges und einfach umzusetzendes Verfahren ist das gruppenbasierte Überprüfen der Lösungen [52, p. 771]. Hier gilt: Wenn die Mehrzahl der Personen aus einer Kontrollgruppe eine Lösung für korrekt hält, wird die Lösung als korrekt bewertet [16].

Diese Kontrollmöglichkeiten machen das Crowdsourcing trotz aller möglichen Probleme zu einem mächtigen Verfahren zum einfachen und kostengünstigen Sammeln von Daten.

2.3 Participatory Sensing

Wird in dieser Arbeit von „Participatory Sensing“ gesprochen, so sind Anwendungen („Apps“) für Smartphones gemeint, die deren Sensoren nutzen, um Daten zu sammeln. Die Besitzer der Smartphones geben dabei die Sensordaten freiwillig zu ihrem eigenen Nutzen und/oder dem Nutzen anderer ab (vgl. [6, p. 1928]).

Im Folgenden werden Anwendungsgebiete anhand beispielhafter Projekte vorgestellt.

2.3.1 Der Mensch im Mittelpunkt der Anwendung Eine Gruppe von Anwendungen beim Participatory Sensing beschäftigt sich mit dem Menschen und sammelt Daten über ihn. Diese Anwendungen werden deshalb auch „people-centric“ genannt. Die folgenden Anwendungsgebiete sind dabei unter anderem denkbar: Persönliche Gesundheitsüberwachung, Berechnung der Auswirkungen auf die Umwelt, Überwachung und Dokumentation von sportlichen Leistungen, Verbesserung der sozialen Medien und Preisprüfungen (vgl. [6, p. 1929ff]).

Bei der persönlichen Gesundheitsüberwachung werden eingebettete oder externe Sensoren benutzt, um den physiologischen Zustand und die Gesundheit von „Patienten“/Teilnehmern zu überwachen. Ein Beispiel ist das Projekt *DietSense* [41], bei dem Teilnehmer beim Abnehmen unterstützt werden. Hierfür werden Entscheidungen bezüglich der Ernährung über Bilder und Tonbeispiele dokumentiert, die von einem Smartphone aufgezeichnet werden, das um den Hals getragen wird. Gleichzeitig wird der Kontext der Mahlzeit, z.B. Tageszeit und Standort, gespeichert, um Situationen bzw. Aktivitäten unterscheiden zu können: Handelt es sich bei der Mahlzeit um ein Mittagessen im Restaurant oder um einen Chips-Abendsnack auf dem Sofa? Alle erhobenen Daten werden in personenbezogene Datenbanken hochgeladen und können dann vom Nutzer eingesehen werden. Dieser kann außerdem Informationen auswählen, die er mit anderen, z.B. seinen Ärzten, teilen möchte.

Das Projekt *PEIR (Personal Environment Impact Report)* [36] beschäftigt sich mit der Berechnung der eigenen Belastungen der Umwelt durch eine Person. Mit PEIR können Nutzer über ihr Smartphone herausfinden, wie sehr sie zu jedem Zeitpunkt die Umwelt belasten. Hierzu werden Standort und Fortbewegungsmittel (Bus, Auto, ...) über den GPS-Sensor sowie den Beschleunigungsmesser bestimmt und an einen zentralen Server übertragen. Dieser wertet die erhaltenen Aktivitätsdaten aus und zeigt dem Nutzer dann u.a. passende Informationen zur eigenen Belastung der Umwelt durch die momentane „Reise“ an.

Bei *BikeNet* [13,12] werden sportliche Aktivitäten der Teilnehmer, genauer ihr Fahrradfahren, erfasst. Erhoben werden der aktuelle Standort, die Geschwindigkeit, die Anzahl der bei der Fahrt verbrannten Kalorien sowie die galvanische Hautreaktion. Unterschiedliche Sensoren werden hierfür eingesetzt: ein Mikrofon, ein Magnetometer, ein Geschwindigkeitssensor an den Pedalen, ein Neigungsmesser, ein Belastungsmonitor, ein Tachometer, ein Kilometerzähler sowie ein Sensor für die CO₂-Konzentration. Die externen Sensoren bilden hierbei ein eigenes Netzwerk und kommunizieren drahtlos mit dem Smartphone. Der Teilnehmer kann die gesammelten Daten dann selbst überprüfen oder mit anderen Teilnehmern teilen bzw. ggf. sogar mit weiteren Parametern wie Daten zur Luftqualität oder zur Verkehrslage kombinieren, um eine Karte mit empfehlenswerten Strecken für andere Fahrradfahrer zu entwerfen.

2.3.2 Die Umgebung im Mittelpunkt der Anwendung Bei Anwendungen, bei denen die Umgebung im Mittelpunkt steht („environment-centric“), werden primär Daten der Umgebung und nicht vom Anwender selbst betrachtet. Die

Ergebnisse des Sensings dienen dementsprechend in den meisten Fällen anderen Nutzern, z.B. den Machern der Anwendung, und werden für größere Projekte genutzt. Folgende Anwendungsgebiete sind zum Beispiel vorstellbar: Überwachung der Luftqualität, Überwachung thermischer Säulen, Überwachung von Lärm und Ambiente und Überwachung von Straßen- und Verkehrsbedingungen.

Ein Projekt, das sich mit der Luftqualität beschäftigt, ist *Haze Watch* [6, p. 1931]. Hier sammeln Smartphones über einen zusätzlichen externen Sensor Messwerte der Konzentrationen von Kohlenmonoxid, Ozon, Schwefeldioxid und Stickstoffdioxid in der Luft. Zwar besteht ihr Nachteil gegenüber Messstationen in einer geringeren Genauigkeit der Messwerte. Ein enormer Vorteil ist jedoch dafür die Mobilität. So können Smartphones z.B. auch bei unerwarteten Ereignissen wie der Luftverschmutzung durch Unfälle zur Messung eingesetzt werden. Allgemein ist es mit ihnen möglich, eine größere Fläche mit Messwerten abzudecken.

Mit der „Verschmutzung“ durch Lärm beschäftigen sich ebenfalls diverse Projekte wie *NoiseTube* [31], *Ear-phone* [40] und *NoiseSpy* [22]. Die in Smartphones eingebauten Mikrophone erfassen bei diesen den Lärmpegel, um die Lärmbelastung an unterschiedlichen Orten zu ermitteln und eine Karte hierüber zu erstellen. Hintergrund hierfür sind die Auswirkungen des Lärmes auf das menschliche Hören und Verhalten.

Bei beiden Projekttypen kann es relevant sein, die aktuelle Aktivität des Nutzers zu erkennen. Die Messwerte der Luftqualität hängen z.B. stark davon ab, ob man sich aktuell im Freien oder in einem Gebäude befindet. Auch die Frage, ob sich das Smartphone in einer Tasche oder am Ohr des Besitzers befindet, kann für die Werte entscheidend sein, sowohl bezüglich der Luftqualität als auch des Lärms.

Ein drittes Projekt, das sich mit der Umgebung des Teilnehmers beschäftigt, ist *Nericell* [34]. Ziel dieses ist es, die Zustände von Straßen sowie Verkehrsbedingungen an unterschiedlichen Orten zu erfassen. Hierzu werden der Beschleunigungsmesser, das Mikrophon sowie Ortungssysteme (GPS-Sensor und GSM-Funk) zur Erkennung von Schlaglöchern und Unebenheiten sowie von Bremssituationen und Hupgeräuschen genutzt. Die erhaltenen Informationen zur Oberflächenbeschaffenheit der Straßen, zu Umgebungsgeräuschen sowie zu Verkehrsbedingungen werden dann in eine öffentlich zugängliche Karte eingezeichnet.

3 Datenschutz beim Participatory Sensing

3.1 Schlüsselfragen

In Tabelle 1 sind alle Zugriffe auf Daten der im vorherigen Kapitel genannten Anwendungen zusammengefasst. Man erkennt schnell, dass es für ein Participatory Sensing-Projekt notwendig ist, dass Teilnehmer viele, teils sensible Informationen über sich preisgeben. Dank dieser Daten kann es durchaus möglich, sein Rückschlüsse auf das private Leben, Gewohnheiten und Beziehungen der

Anwendung	Zeit	Ort	Bilder	Ton	Verschmutzung	andere
DietSense	x	x	x	x		
PEIR	x	x				
Haze Watch	x	x			x	
NoiseTube	x	x		x		
Ear-phone	x	x		x		
NoiseSpy	x	x		x		
BikeNet	x	x		x	x	x
Nericell	x	x		x		x

Tabelle 1. Datenzugriff einzelner Anwendungen [6, p. 1934]

Personen zu ziehen. Anhand der Standort-Daten z.B. können Wohn- und Arbeitsort sowie typische Routen bestimmt, anhand von Uhrzeiten, zu denen das Smartphone aktiv war, zusätzlich ein Tagesrhythmus geschätzt werden.

Geht es in diesem Kapitel um Privatsphäre und Datenschutz, so bezieht sich dies stets auf den Umgang mit Sensordaten der einzelnen Teilnehmer. Der Datenschutz muss sowohl für diese Daten als auch für solche, die ein Teilnehmer mit dem Participatory Sensing-System teilt, garantieren, dass die Kontrolle über die Veröffentlichung beim „Besitzer“ bleibt [6, p. 1934].

Übertragen auf die obigen Beispiele könnte dies z.B. bedeuten, dass Nutzer Daten, die an ganz bestimmten Orten erfasst wurden, nicht weitergeben müssen, sondern entfernen können. Das bedeutet jedoch auch, dass jeder Nutzer aktiv über seine Privatsphäre bestimmen kann und muss - im Optimalfall unterstützt vom System, z.B. durch Vorschläge, welche Daten entfernt werden sollten [6, p. 1934].

Hierfür muss das Participatory Sensing-System jedoch soziokulturelle und kontextuelle Unterschiede beachten, denn diese haben eine starke, für jeden Menschen individuelle Auswirkung auf die Wahrnehmung der Sensibilität von Daten. Mit wie vielen Personen werden die Daten geteilt? Welche Typen Menschen sind dies? So zeigte sich z.B., dass Teilnehmer pragmatisch entscheiden, ob sie ihren Standort mit einer einzelnen Person teilen sollen. Sobald es jedoch um eine größere Gruppe Personen geht, kommen weitere Parameter wie der Hang, Aufmerksamkeit auf sich zu ziehen, sowie der Wille, sich selbst darzustellen, zur Entscheidungsfindung hinzu [49].

Man stelle sich ein Participatory Sensing-System vor, das die folgenden Daten sammeln kann bzw. möchte: Bilder, Tonaufnahmen, Beschleunigungen, Verschmutzungsdaten und Luftdruckwerte - jeweils kombiniert mit Daten zur Zeit und zum Ort. Dann sind die folgenden Fragen zur Analyse der Privatsphäre und des Datenschutzes entscheidend [6, p. 1935]:

- Wer sammelt die Sensordaten?
- Wer analysiert die Sensordaten?
- Wer hat Zugriff auf die analysierten Daten?

Im Folgenden werden beispielhaft für das oben beschriebene System Antworten auf die Fragen gegeben.

Die erhobenen Sensordaten werden an einen Server geschickt, der in der Regel vom Projektleiter bzw. von den Projektleitern betrieben wird. Dementsprechend haben diese einen direkten Zugriff auf die Daten [6, p. 1935]. Die übertragenen Daten können jedoch noch sensible Informationen enthalten, wenn diese nicht vorab auf dem Smartphone des Teilnehmers herausgefiltert werden. Was „sensibel“ ist kann nur und muss der Teilnehmer, wie oben beschrieben, selbst bestimmen. Die Smartphone-Anwendung des Projekts sollte daher mit ihren Einstellungsmöglichkeiten versuchen, möglichst alle denkbaren Nutzerwünsche zu Beschränkungen bezüglich der Datenerfassung, -weitergabe und -vorverarbeitung abzudecken [9].

Die Daten werden analysiert, sobald sie an den Server des Projektes übertragen wurden. Dort führen implementierte Prozesse noch vor der Analyse eine Vorverarbeitung durch. Bei dieser werden die Daten ggf. zusätzlich zur Vorverarbeitung auf dem Smartphone des Teilnehmers noch einmal anonymisiert, was jedoch im Allgemeinen nur in der Theorie ohne überbleibende sensible Daten funktioniert. Interpretiert werden die Daten dann von unterschiedlichen Gruppen, abhängig vom Projekt z.B. vom Teilnehmer selbst (siehe *DietSense*) oder von den Projektleitern, von Ärzten oder Wissenschaftlern [6, p. 1935].

Mit wem die Daten geteilt werden, ist für die Privatsphäre und den Datenschutz der Teilnehmer entscheidend. Sie wissen, dass diese „vom Projekt“ verarbeitet und verwendet werden. Geht es jedoch z.B. um Informationen zur Gesundheit, so halten es die Teilnehmer zwar sicher für angemessen, diese mit (ihren) Ärzten zu teilen, unter Umständen aber für unangemessen, wenn sie mit anderen Personen wie dem Projektleiter geteilt werden [38]. Dass diese allerdings trotzdem Zugriff darauf haben, wird von den Teilnehmern unter Umständen ignoriert [6, p. 1935].

Um so wichtiger ist es, eine angemessene Zugriffskontrolle auf die Daten umzusetzen, um das Vertrauen der Teilnehmer nicht zu gefährden. Theoretisch ist es schließlich möglich, dass jeder mit Zugriffsrechten Daten veröffentlichen, also frei zugänglich machen, kann. Es kann daher sinnvoll sein, nicht nur den Zugriff möglichst granular zu verteilen, sondern auch die Daten bzw. deren Analyseergebnisse je nach zukünftigem Verwendungszweck zu trennen.

Zusammenfassend kann man sagen, dass Datenschutz und Privatsphäre der Teilnehmer in der Hand der Projektleiter liegen. Diese haben einen direkten und uneingeschränkten Zugriff auf gesammelte Daten, und die Teilnehmer müssen ihnen daher vertrauen. Eine angemessene Verteilung bzw. Veröffentlichung der Daten durch die Projektleiter ist hierfür grundlegend, genauso wie effiziente Mechanismen zum Entfernen privater Daten zur Prävention von Verletzungen der Privatsphäre.

3.2 Gefahren

Angenommen das Worst-Case-Szenario tritt ein und Projektleiter brechen das Vertrauen ihrer Nutzer. Wie würde sich das auswirken? Um dies zu verdeutlichen werden im Folgenden mögliche Gefahren durch die unangebrachte Veröffentlichung von Informationen zu Zeit und Ort eines Teilnehmers stellvertretend für weitere Informationen (siehe Tabelle 1) betrachtet.

Wie Tabelle 1 vermuten lässt, sind Zeit und Ort für die meisten Anwendungen grundlegende Informationen, die gesammelt werden - egal, ob der Nutzer selbst im Mittelpunkt steht oder dessen Umwelt. Die in Smartphones verbauten GPS-Sensoren ermöglichen mittlerweile eine sehr akurate Bestimmung des Standorts. Fehlen diese Sensorwerte, weil z.B. der Sensor deaktiviert wurde oder die Abdeckung durch GPS-Satelliten zu gering ist, kann außerdem auf Standortbestimmung mittels WLAN- oder Mobilfunknetze umgestiegen werden [25]. Alternativ zur physikalischen Standortbestimmung (Längen- und Breitengrad) kann auch ein logischer Ort bestimmt werden, z.B. über Umgebungsgeräusche oder Licht und Farben der Umgebung, die mittels Kamera und Mikrophon erfasst werden [3].

Die Veröffentlichung von Daten zu Zeit und Ort würde dazu führen, dass dank der enthaltenen sensiblen Informationen eines Teilnehmers u.a. dessen Wohn- und Arbeitsort sowie Gewohnheiten ermittelt werden können [45]. Häufige Besuche eines Krankenhauses könnten z.B. auf einen schlechten medizinischen Zustand hinweisen; regelmäßige Teilnahmen an politischen Veranstaltungen lassen eine gute Einschätzung der politischen Gesinnung einer Person zu [28].

Insgesamt gilt, dass die Veröffentlichung von Standortdaten ohne Schutzmechanismen zu ernsthaften Konsequenzen führen und damit sowohl soziale Gefahren als auch Gefahren für die Sicherheit eines Teilnehmers darstellen kann [45]. Hierzu kommt, dass diese Gefahren sogar dann eintreten können, wenn die Daten vorher anonymisiert bzw. anonym beigetragen wurden, da anhand von Standortinformationen Rückschlüsse auf Personen möglich sind [36].

Durch Daten weiterer Sensoren können diese Gefahren noch erweitert bzw. verstärkt werden. Im Fall des DietSense-Systems z.B. werden Fotos, die aufgenommen werden, mit dem jeweiligen Standort und Aufnahmezeitpunkt markiert und nur mit ausgewählten Personen wie Ernährungsberatern geteilt [41]. Zwar kennen sich die Teilnehmer und Berater im Normalfall, jedoch könnte die Privatsphäre der Teilnehmer immer noch dadurch bedroht werden, dass Gesichter anderer und/oder Gegenstände auf den Bildern erkannt werden können [6, p. 1936].

3.3 Gegenmaßnahmen

Um möglichen Gefahren zu begegnen, haben sich einige Lösungen als passend herausgestellt. Je nach ihrer grundlegenden Idee haben die Lösungen unterschiedliche Angriffspunkte auf dem Weg vom Smartphone eines Teilnehmers bis zur Veröffentlichung von Daten für eine geplante Zielgruppe.

Die folgenden Bereiche werden hier besprochen:

- Maßgeschneidertes Sensing und Nutzereinstellungen
- Anonymisierte Berichterstattung von Daten
- Überprüfen, Löschen, Speichern und Behalten von Daten

Maßgeschneidertes Sensing und Nutzereinstellungen Wie bereits in Kapitel 3.1 beschrieben, ist es ein erster Schritt zugunsten der Privatsphäre und des Datenschutzes, wenn ein Teilnehmer in den Einstellungen der App seine Vorstellungen davon umsetzen kann. Eine binäre Umsetzung der Form „Sensing aktiviert“/„Sensing deaktiviert“ ist dabei jedoch nicht zu empfehlen, da es dem Projekt nichts nutzt, wenn es keine Daten erhält. Trotzdem haben einzelne Lösungen diese Idee umgesetzt [33].

Sinnvoller ist es, unterschiedliche Zwischenstufen anzubieten, z.B. das Erheben von Sensorwerten abhängig von unterschiedlichen Faktoren [8]. Das Sensing könnte z.B. an sensiblen Orten wie Zuhause oder bei der Anwesenheit weiterer Personen pausiert werden. Auch feingranularere Stufen sind vorstellbar [42]. Beim Erheben von Daten könnte ein Nutzer z.B. das Intervall festlegen wollen - alle 15 Sekunden vs. einmal pro Stunde -, bei Standortdaten könnte eine Präzision eingestellt werden - exakte Position vs. Stadt.

Anonymisierte Berichterstattung von Daten Eine Möglichkeit, die Privatsphäre auch beim bzw. nach dem Übertragen von Sensordaten an einen zentralen Server gewährleisten zu können, ist die Verwendung von Pseudonymen. Statt des echten Namens eines Teilnehmers wird bei allen Interaktionen ein Pseudonym verwendet [46,45,10]. Dies sorgt zusätzlich für Sicherheit beim Nutzer, und er wird seine Daten eher teilen, da er sich hinter seinem Pseudonym sicher fühlt [46].

Pseudonyme helfen natürlich nicht, wenn wie im vorherigen Abschnitt beschrieben anhand der Daten selbst eine Person bestimmbar ist. Zum Beispiel könnte über häufige Aufenthaltsorte ein möglicher Wohnort und damit der echte Name eines Teilnehmers bestimmt werden.

Eine andere Möglichkeit der anonymisierten Berichterstattung ist das sog. „spatial cloaking“, z.B. über das Modell der „k-Anonymität“ [48,20]. Hierbei werden Daten mehrerer Nutzer bei einem Dritten gespeichert, dort gemeinsam verarbeitet und dann das Ergebnis statt der einzelnen Datensätze an den zentralen Server zur Weiterverarbeitung gesendet. Bei Standortdaten z.B. könnte statt der einzelnen, präzisen Koordinaten der Durchschnitt von Teilnehmern innerhalb eines vorher definierten Clusters übermittelt werden. Wichtig ist dabei jedoch neben der Vorverarbeitung vor der Übermittlung an den Server, dass die Teilnehmer dem Dritten vertrauen.

Weitere Möglichkeiten der Anonymisierung sind das Hinzufügen von künstlichem Rauschen („data perturbation“) [14] sowie die Datenaggregation [44]: Bei diesem Ansatz werden gesammelte Daten teilweise an Teilnehmer, die sich in der Umgebung befinden, verteilt und dann von diesen an den zentralen Server gesendet. Es ist damit nicht mehr unmittelbar möglich, Daten einer Person zuzuordnen.

Überprüfen, Löschen, Speichern und Behalten von Daten Nach dem Sammeln von Daten können Teilnehmer bei den meisten Projekten diese überprüfen und ggf. sensible Daten wie Bilder mit Gesichtern oder besondere Standorte herausfiltern, also löschen, bevor diese an den Server übertragen werden [1]. Gerade das Übertragen an einen Server führt jedoch zu einem Verlust von Kontrolle beim Teilnehmer. Nach dem Hochladen muss dieser den Projektleitern vertrauen, dass Daten nicht mit unauthorisierten Personen geteilt und beim Löschen auch wirklich vom Server entfernt werden.

Ein denkbarer, aber eher schlechter Ansatz ist es, die Daten statt auf einem Server auf jedem Smartphone selbst zu speichern. Ein sinnvolles Arbeiten mit erhobenen Daten von Nutzern weltweit ist damit kaum mehr möglich - nicht zuletzt, weil einmal mehr die Geräte natürliche Schranken wie Speicherplatz und Stromverbrauch haben [6, p. 1941].

Eine bessere Lösung ist der Einsatz von „persönlichen Datentresoren“ („personal data vaults“) [35]. Teilnehmerdaten werden bei dieser Methode in individuell kontrollierten, sicheren Datenspeichern gelagert, auf die nur von ihrem Besitzer zugegriffen werden kann. Dieser kann wiederum seine Daten mit anderen teilen.

Eine Alternative dazu sind persönliche virtuelle Maschinen, die als individuelle Server verwendet werden und in die ein Teilnehmer seine Daten hochladen kann [4].

Nachteil all dieser Lösungen ist ein zusätzlicher Verwaltungsaufwand.

4 Fazit und Ausblick

Dank der Möglichkeiten des Participatory Sensings ist es sowohl für die Industrie als auch für Forschungsprojekte einfacher geworden, schnell und unkompliziert an Daten aus aller Welt zu gelangen. Über mit Sensoren ausgestattete Smartphones können Apps betrieben werden, die sowohl zum Nutzen der Projektteilnehmer als auch zum Nutzen der Projektleiter gesammelt und später auf Servern ausgewertet werden.

Problematisch dabei ist, dass diese Daten persönliche, teils sensible Daten enthalten können. Um durch Verletzung des Datenschutzes und der Privatsphäre entstehende Gefahren für Teilnehmer zu begegnen müssen deshalb angemessene Maßnahmen getroffen werden.

Hierfür wurden in dieser Arbeit die folgenden Ideen vorgestellt:

- Das Anbieten von Einstellungsmöglichkeiten, wann und welche Daten erfasst werden
- Das Anonymisieren bzw. Pseudonymisieren von erfassten Daten
- Eine Zugriffskontrolle über „persönliche Datentresore“

Da es trotz dieser Gegenmaßnahmen jedoch weiterhin möglich ist, sensible Informationen aus den vorverarbeiteten Daten zu ziehen, und die jeweiligen Umsetzungen teilweise einen deutlich erhöhten Verwaltungsaufwand bedeuten, können sich zukünftige Arbeiten mit besseren Lösungen beschäftigen.

Thema weiterer Arbeiten können außerdem Nutzerstudien sein. Durch diese kann z.B. analysiert werden, welche Daten Nutzer bereit sind, freizugeben, wer auf die Daten zugreifen können soll und welche Ängste sie bei der Teilnahme bei einem Participatory Sensing-Projekt haben.

Eine andere Möglichkeit wäre es, sich damit zu beschäftigen, wie sich die an ein Participatory Sensing-Projekt übermittelten Daten durch Verschlüsselung so schützen lassen, dass die Projektleiter trotzdem bestimmte Möglichkeiten haben, mit ihnen zu arbeiten.

Literatur

1. Abdelzaher, T., Anokwa, Y., Boda, P., Burke, J., Estrin, D., Guibas, L., Kansal, A., Madden, S., Reich, J.: Mobiscopes for human spaces. *IEEE Pervasive Computing* 6(2), 20–29 (Apr 2007), <http://dx.doi.org/10.1109/MPRV.2007.38>
2. Amazon: Amazon mechanical turk (Jan 2014), <https://www.mturk.com/>
3. Azizyan, M., Constandache, I., Roy Choudhury, R.: Surroundsense: Mobile phone localization via ambience fingerprinting. In: *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking*. pp. 261–272. *MobiCom '09*, ACM, New York, NY, USA (2009), <http://doi.acm.org/10.1145/1614320.1614350>
4. Cáceres, R., Cox, L., Lim, H., Shakimov, A., Varshavsky, A.: Virtual individual servers as privacy-preserving proxies for mobile devices. In: *Proceedings of the 1st ACM Workshop on Networking, Systems, and Applications for Mobile Handhelds*. pp. 37–42. *MobiHeld '09*, ACM, New York, NY, USA (2009), <http://doi.acm.org/10.1145/1592606.1592616>
5. Choudhury, T., Consolvo, S., Harrison, B., Hightower, J., LaMarca, A., Legrand, L., Rahimi, A., Rea, A., Bordello, G., Hemingway, B., Klasnja, P., Koscher, K., Landay, J., Lester, J., Wyatt, D., Haehnel, D.: The mobile sensing platform: An embedded activity recognition system. *Pervasive Computing, IEEE* 7(2), 32–41 (2008)
6. Christin, D., Reinhardt, A., Kanhere, S.S., Hollick, M.: A survey on privacy in mobile participatory sensing applications. *J. Syst. Softw.* 84(11), 1928–1946 (Nov 2011), <http://dx.doi.org/10.1016/j.jss.2011.06.073>
7. Consolvo, S., McDonald, D.W., Toscos, T., Chen, M.Y., Froehlich, J., Harrison, B., Klasnja, P., LaMarca, A., LeGrand, L., Libby, R., Smith, I., Landay, J.A.: Activity sensing in the wild: A field trial of ubifit garden. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. pp. 1797–1806. *CHI '08*, ACM, New York, NY, USA (2008), <http://doi.acm.org/10.1145/1357054.1357335>
8. Das, T., Mohan, P., Padmanabhan, V.N., Ramjee, R., Sharma, A.: Prism: Platform for remote sensing using smartphones. In: *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*. pp. 63–76. *MobiSys '10*, ACM, New York, NY, USA (2010), <http://doi.acm.org/10.1145/1814433.1814442>
9. Debatin, B., Lovejoy, J.P., Horn, A.K., Hughes, B.N.: Facebook and online privacy: Attitudes, behaviors, and unintended consequences. *Journal of Computer-Mediated Communication* 15(1), 83–108 (2009), <http://dx.doi.org/10.1111/j.1083-6101.2009.01494.x>

10. Deng, L., Cox, L.P.: Livecompare: Grocery bargain hunting through participatory sensing. In: Proceedings of the 10th Workshop on Mobile Computing Systems and Applications. pp. 4:1–4:6. HotMobile '09, ACM, New York, NY, USA (2009), <http://doi.acm.org/10.1145/1514411.1514415>
11. Eagle, N., (Sandy) Pentland, A.: Reality mining: Sensing complex social systems. *Personal Ubiquitous Comput.* 10(4), 255–268 (Mar 2006), <http://dx.doi.org/10.1007/s00779-005-0046-3>
12. Eisenman, S.B., Miluzzo, E., Lane, N.D., Peterson, R.A., Ahn, G.S., Campbell, A.T.: The bikenet mobile sensing system for cyclist experience mapping. In: Proceedings of the 5th International Conference on Embedded Networked Sensor Systems. pp. 87–101. SenSys '07, ACM, New York, NY, USA (2007), <http://doi.acm.org/10.1145/1322263.1322273>
13. Eisenman, S.B., Miluzzo, E., Lane, N.D., Peterson, R.A., Ahn, G.S., Campbell, A.T.: Bikenet: A mobile sensing system for cyclist experience mapping. *ACM Trans. Sen. Netw.* 6(1), 6:1–6:39 (Jan 2010), <http://doi.acm.org/10.1145/1653760.1653766>
14. Ganti, R.K., Pham, N., Tsai, Y.E., Abdelzaher, T.F.: Poolview: Stream privacy for grassroots participatory sensing. In: Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems. pp. 281–294. SenSys '08, ACM, New York, NY, USA (2008), <http://doi.acm.org/10.1145/1460412.1460440>
15. Harris, C.G.: You're hired! an examination of crowdsourcing incentive models in human resource tasks. In: Proceedings of the Workshop on Crowdsourcing for Search and Data Mining (CSDM 2011). pp. 15–18. WSDM 2011, ACM, New York, NY, USA (2011)
16. Hirth, M., Hoßfeld, T., Tran-Gia, P.: Cheat-detection mechanisms for crowdsourcing. Technical Report 474, University of Würzburg (8 2010)
17. Honicky, R., Brewer, E.A., Paulos, E., White, R.: N-smarts: Networked suite of mobile atmospheric real-time sensors. In: Proceedings of the Second ACM SIGCOMM Workshop on Networked Systems for Developing Regions. pp. 25–30. NSDR '08, ACM, New York, NY, USA (2008), <http://doi.acm.org/10.1145/1397705.1397713>
18. Howe, J.: The Rise of Crowdsourcing. *wired* (2006), <http://www.wired.com/wired/archive/14.06/crowds.html>
19. Howe, J.: Crowdsourcing: Why the Power of the Crowd Is Driving the Future of Business. Crown Publishing Group, New York, NY, USA, 1 edn. (2008)
20. Huang, K.L., Kanhere, S.S., Hu, W.: Preserving privacy in participatory sensing systems. *Comput. Commun.* 33(11), 1266–1280 (Jul 2010), <http://dx.doi.org/10.1016/j.comcom.2009.08.012>
21. Huberman, B.A., Romero, D.M., Wu, F.: Crowdsourcing, attention and productivity. *J. Inf. Sci.* 35(6), 758–765 (Dec 2009), <http://dx.doi.org/10.1177/0165551509346786>
22. Kanjo, E., Bacon, J., Roberts, D., Landshoff, P.: Mobsens: Making smart phones smarter. *IEEE Pervasive Computing* 8(4), 50–57 (Oct 2009), <http://dx.doi.org/10.1109/MPRV.2009.79>
23. Kazai, G.: An exploration of the influence that task parameters have on the performance of crowds. In: Proceedings of the CrowdConf 2010. CrowdConf 2010 (2010)
24. Ko, A.J., Chilana, P.K.: How power users help and hinder open bug reporting. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 1665–1674. CHI '10, ACM, New York, NY, USA (2010), <http://doi.acm.org/10.1145/1753326.1753576>

25. LaMarca, A., Chawathe, Y., Consolvo, S., Hightower, J., Smith, I., Scott, J., Sohn, T., Howard, J., Hughes, J., Potter, F., Tabert, J., Powledge, P., Borriello, G., Schilit, B.: Place lab: Device positioning using radio beacons in the wild. In: Proceedings of the Third International Conference on Pervasive Computing. pp. 116–133. PERVASIVE'05, Springer-Verlag, Berlin, Heidelberg (2005), http://dx.doi.org/10.1007/11428572_8
26. Lane, N.D., Miluzzo, E., Lu, H., Peebles, D., Choudhury, T., Campbell, A.T.: A survey of mobile phone sensing. *Comm. Mag.* 48(9), 140–150 (Sep 2010), <http://dx.doi.org/10.1109/MCOM.2010.5560598>
27. Li, X., Ortiz, P., Browne, J., Franklin, D., Oliver, J., Geyer, R., Zhou, Y., Chong, F.: Smartphone evolution and reuse: Establishing a more sustainable model. In: Parallel Processing Workshops (ICPPW), 2010 39th International Conference on. pp. 476–484 (2010)
28. Liu, L.: From data privacy to location privacy: Models and algorithms. In: Proceedings of the 33rd International Conference on Very Large Data Bases. pp. 1429–1430. VLDB '07, VLDB Endowment (2007), <http://dl.acm.org/citation.cfm?id=1325851.1326035>
29. Lu, H., Pan, W., Lane, N.D., Choudhury, T., Campbell, A.T.: Soundsense: Scalable sound sensing for people-centric applications on mobile phones. In: Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services. pp. 165–178. MobiSys '09, ACM, New York, NY, USA (2009), <http://doi.acm.org/10.1145/1555816.1555834>
30. Maisonneuve, N., Stevens, M., Niessen, M.E., Hanappe, P., Steels, L.: Citizen noise pollution monitoring. In: DG.O '09: PROCEEDINGS OF THE 10TH ANNUAL INTERNATIONAL CONFERENCE ON DIGITAL GOVERNMENT RESEARCH. pp. 96–103. Digital Government Society of North America / ACM Press (2009)
31. Maisonneuve, N., Stevens, M., Niessen, M., Steels, L.: Noisetube: Measuring and mapping noise pollution with mobile phones. In: Proceedings of the 4th International Symposium on Information Technologies in Environmental Engineering (ITEE). pp. 215–228 (2009)
32. Mason, W., Watts, D.J.: Financial incentives and the "performance of crowds". *SIGKDD Explor. Newsl.* 11, 100–108 (May 2010)
33. Miluzzo, E., Lane, N.D., Fodor, K., Peterson, R., Lu, H., Musolesi, M., Eisenman, S.B., Zheng, X., Campbell, A.T.: Sensing meets mobile social networks: The design, implementation and evaluation of the cenceme application. In: Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems. pp. 337–350. SenSys '08, ACM, New York, NY, USA (2008), <http://doi.acm.org/10.1145/1460412.1460445>
34. Mohan, P., Padmanabhan, V.N., Ramjee, R.: Nericell: Rich monitoring of road and traffic conditions using mobile smartphones. In: Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems. pp. 323–336. SenSys '08, ACM, New York, NY, USA (2008), <http://doi.acm.org/10.1145/1460412.1460444>
35. Mun, M., Hao, S., Mishra, N., Shilton, K., Burke, J., Estrin, D., Hansen, M., Govindan, R.: Personal data vaults: A locus of control for personal data streams. In: Proceedings of the 6th International Conference. pp. 17:1–17:12. Co-NEXT '10, ACM, New York, NY, USA (2010), <http://doi.acm.org/10.1145/1921168.1921191>
36. Mun, M., Reddy, S., Shilton, K., Yau, N., Burke, J., Estrin, D., Hansen, M., Howard, E., West, R., Boda, P.: Peir, the personal environmental impact report, as a

- platform for participatory sensing systems research. In: Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services. pp. 55–68. MobiSys '09, ACM, New York, NY, USA (2009), <http://doi.acm.org/10.1145/1555816.1555823>
37. Nam, K.K., Ackerman, M.S., Adamic, L.A.: Questions in, knowledge in?: A study of naver's question answering community. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 779–788. CHI '09, ACM, New York, NY, USA (2009), <http://doi.acm.org/10.1145/1518701.1518821>
 38. Nissenbaum, H.: Privacy as contextual integrity. *Washington Law Review* 79, 101–139 (2004)
 39. Nokia: Workshop on large-scale sensor networks and applications (Feb 2005)
 40. Rana, R.K., Chou, C.T., Kanhere, S.S., Bulusu, N., Hu, W.: Ear-phone: An end-to-end participatory urban noise mapping system. In: Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks. pp. 105–116. IPSN '10, ACM, New York, NY, USA (2010), <http://doi.acm.org/10.1145/1791212.1791226>
 41. Reddy, S., Parker, A., Hyman, J., Burke, J., Estrin, D., Hansen, M.: Image browsing, processing, and clustering for participatory sensing: Lessons from a dietsense prototype. In: Proceedings of the 4th Workshop on Embedded Networked Sensors. pp. 13–17. EmNets '07, ACM, New York, NY, USA (2007), <http://doi.acm.org/10.1145/1278972.1278975>
 42. Schilit, B.N., LaMarca, A., Borriello, G., Griswold, W.G., McDonald, D., Lazowska, E., Balachandran, A., Hong, J., Iverson, V.: Challenge: Ubiquitous location-aware computing and the "place lab" initiative. In: Proceedings of the 1st ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots. pp. 29–35. WMASH '03, ACM, New York, NY, USA (2003), <http://doi.acm.org/10.1145/941326.941331>
 43. Schmidt, A., Aidoo, K.A., Takaluoma, A., Tuomela, U., Laerhoven, K.V., Velde, W.V.d.: Advanced interaction in context. In: Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing. pp. 89–101. HUC '99, Springer-Verlag, London, UK, UK (1999), <http://dl.acm.org/citation.cfm?id=647985.743849>
 44. Shi, J., Zhang, Y., Liu, Y., Zhang, Y.: Prisense: Privacy-preserving data aggregation in people-centric urban sensing systems. In: Proceedings of the 29th IEEE International Conference on Computer Communications (INFOCOM). pp. 1–9 (2010)
 45. Shilton, K.: Four billion little brothers?: Privacy, mobile phones, and ubiquitous data collection. *Commun. ACM* 52(11), 48–53 (Nov 2009), <http://doi.acm.org/10.1145/1592761.1592778>
 46. Shilton, K., Burke, J.A., Estrin, D., Hansen, M., Srivastava, M.: Participatory privacy in urban sensing. In: Proceedings of the International Workshop on Mobile Devices and Urban Sensing (MODUS). pp. 1–7 (2008)
 47. Silberman, M.S., Irani, L., Ross, J.: Ethics and tactics of professional crowdwork. *XRDS* 17(2), 39–43 (Dec 2010), <http://doi.acm.org/10.1145/1869086.1869100>
 48. Sweeney, L.: K-anonymity: A model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* 10(5), 557–570 (Oct 2002), <http://dx.doi.org/10.1142/S0218488502001648>
 49. Tang, K.P., Lin, J., Hong, J.I., Siewiorek, D.P., Sadeh, N.: Rethinking location sharing: Exploring the implications of social-driven vs. purpose-driven location sharing. In: Proceedings of the 12th ACM International Conference on Ubiquitous

- Computing. pp. 85–94. Ubicomp '10, ACM, New York, NY, USA (2010), <http://doi.acm.org/10.1145/1864349.1864363>
50. Wikipedia: Wikipedia (Jan 2014), <http://www.wikipedia.org/>
 51. YouTube: Youtube (Jan 2014), <http://www.youtube.com/>
 52. Yuen, M.C., King, I., Leung, K.S.: A survey of crowdsourcing systems. In: Privacy, security, risk and trust (passat), 2011 iee third international conference on and 2011 iee third international conference on social computing (socialcom). pp. 766–773 (2011)

Power System Analysis

Johannes Bleier*

Betreuer: Yong Ding[†]

Karlsruher Institut für Technologie (KIT)

Pervasive Computing Systems – TECO

*johannes.bleier@student.kit.edu

[†]ding@teco.edu

Zusammenfassung. Stromnetze sind mehr als nur die Ohm'schen und Kirchhoff'schen Gesetze für Generatoren, Leitungen und Verbraucher. Man muss mehr betrachten, zum Beispiel den maximalen Strom, den ein Stromnetz transportieren kann oder wie sich ein Stromausfall an einem Punkt auf das gesamte Stromnetz auswirkt.

Dazu muss die Funktionsweise der Regelungsmechanismen analysiert werden. Für die einzelnen Eigenschaften gibt es verschiedene Analyseverfahren. Diese Ausarbeitung beschreibt, welche Analyseverfahren dies sind, welche Software es für das jeweilige Verfahren gibt und welche Techniken dabei verwendet werden können. Dabei wird auch auf weitere Unterscheidungsmerkmale der verschiedenen Softwarepakete eingegangen. Außerdem wird gezeigt, dass sich die Analyse nicht nur auf die Berechnung von physikalischer Größen beschränkt, sondern, dass auch die Preisberechnung in einem Stromnetz nicht so einfach ist. In diesem Zusammenhang wird das Location Marginal Pricing vorgestellt, ein standardisiertes Verfahren für die Preisberechnung in Stromnetzen.

1 Einführung in das Thema

Stromnetze sind allgegenwärtig, bei uns in Europa wie auch in anderen Teilen der Welt ist praktisch jeder Haushalt an ein Stromnetz angeschlossen, das sich über große Teile des Kontinents erstreckt. Solche großen Stromnetze kommen nicht nur bei der öffentlichen Energieversorgung, sondern zum Beispiel auch beim Oberleitungsnetz der Eisenbahn vor.

Bei der Planung eines Stromnetzes müssen Schutzmechanismen eingebaut werden, die dafür sorgen, dass nicht die nötige Spannung fehlt, wenn zu viele Verbraucher am Netz hängen, oder die Spannung zu hoch ist, da nicht genug Verbraucher vorhanden sind. Denn in jedem dieser Fälle könnte es passieren, dass das komplette Stromnetz ausfällt. Ob die gewählten Mechanismen auch funktionieren, kann mit Hilfe von Simulationen und mathematischen Methoden analysiert werden. Manchmal müssen sich die Stromanbieter gegenseitig auszuweichen, um die gewünschte Spannung erhalten zu können. Dies erfordert einen Markt, an dem der Strom gehandelt werden kann. Dazu benötigen die Stromanbieter aber ein Verfahren zur Strompreisbildung, das flexibel ist und auch den Weg, den der Strom transportiert werden muss, berücksichtigt.

1.1 Physikalische Grundlagen

Ein Stromnetz besteht aus vielen Komponenten, deren physikalische Eigenschaften bei der Analyse eines solchen Netzes betrachtet werden müssen. Da sind zum einen die Generatoren, die den Strom erzeugen, und die Verbraucher, zum anderen aber auch die zwischen den Generatoren und Verbrauchern liegenden Kabel und Transformatoren. Diese dürfen bei der Analyse nicht vernachlässigt werden, da beim Transformieren auf eine andere Spannung und beim Transportieren des Stromes selbst Energie verbraucht wird.

Die physikalische Beschreibung eines Stromnetzes benutzt hauptsächlich die Ohm'schen Gesetze und Kirchhoff'schen Regeln, die Anwendung derselben auf ein Stromnetz resultieren in einem Gleichungssystem, das für jedes Kabel und jeden Transformator eine Gleichung beinhaltet. Dabei muss man zwischen Gleich- und Drehstromnetzen unterscheiden.

Gleichstrom ist ein Verfahren, bei dem an den Leitungen eine konstante Spannung anliegt. Da für die Beschreibung eines Gleichstromnetzes die oben beschriebenen Gesetze ausreichen, können solche Netze als lineare Gleichungssysteme dargestellt werden. Gleichstrom wird bei der Energieübertragung über weite Entfernungen verwendet, da der elektrische Verlust ab einer gewissen Leitungslänge geringer ist als bei Drehstrom.

Drehstrom ist ein Verfahren, bei dem die elektrische Energie mit Hilfe von drei Wechselstromleitungen übertragen wird, die jeweils mit einer sinusförmigen Spannung betrieben werden und deren Phase gegeneinander verschoben sind. Da bei der Beschreibung des Netzes auch die Phasenverschiebungen berücksichtigt werden müssen, enthält das resultierende Gleichungssystem trigonometrische Funktionen und ist damit nicht mehr linear. Die meisten Stromnetze werden mit Drehstrom betrieben, da dies mehrere Vorteile hat, die aber nicht Teil dieser Ausarbeitung sein sollen.

Grundsätzlich kommt bei einem Stromnetz noch die Möglichkeit dazu, einzelne Generatoren, Verbraucher und Leitungen zu- oder abzuschalten, was entweder zu zusätzlichen, diskreten Variablen im Gleichungssystem führt, oder eben zu je einem Gleichungssystem für jede mögliche Konfiguration.

Das in Abbildung 1 dargestellte Drehstromnetz resultiert zu folgendem Gleichungssystem. Dabei stehen die \bar{v}_k für die Spannung und \bar{i}_k für die Stromstärke an der jeweiligen Stelle im Schaltnetz. jx_{kl} bzw. z_k sind die jeweiligen Wechselwiderstände und x_{kl} die dazugehörigen Blindwiderstände der Spulen, die als Approximation der langen Leitungen dienen.

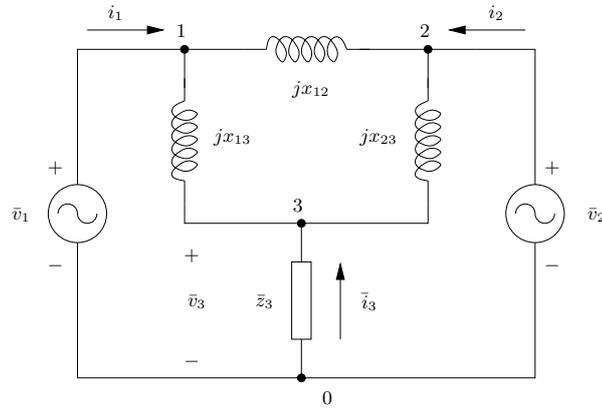


Abb. 1. Beispiel eines einfachen Stromnetzes.[9].

$$0 = \frac{\bar{v}_1 - \bar{v}_2}{jx_{12}} + \frac{\bar{v}_1 - \bar{v}_3}{jx_{13}} - \bar{i}_1 \quad (1)$$

$$0 = \frac{\bar{v}_2 - \bar{v}_1}{jx_{12}} + \frac{\bar{v}_2 - \bar{v}_3}{jx_{23}} - \bar{i}_2 \quad (2)$$

$$0 = \frac{\bar{v}_3 - \bar{v}_1}{jx_{13}} + \frac{\bar{v}_3 - \bar{v}_2}{jx_{23}} - \bar{i}_3 \quad (3)$$

$$\bar{i}_3 = -\frac{\bar{v}_3}{\bar{z}_3} \quad (4)$$

Das oben aufgeführte Beispiel ist aber nur ein sehr kleines. Praxisrelevante Beispiele und deren Gleichungssysteme sind um einiges größer, was sich auch auf die Laufzeit des simulierenden Programms auswirkt. Wie man dieses lineare Gleichungssystem im Zusammenhang mit Stromnetzen möglichst effizient lösen kann, wird im Abschnitt 4 erläutert.

Das Ziel, das man bei der Modellierung eines Stromnetzes erreichen will, ist das Erhalten eines konstanten Spannungslevels. Die dafür notwendige Regelung wird dabei typischer Weise von den Strom erzeugenden Elementen des Stromnetzes, den Generatoren vorgenommen, indem deren Leistung erhöht oder verringert wird. Alternativ gibt es auch die Möglichkeit den Verbrauch zu regeln, das ist zum Beispiel durch das Dimmen von Lampen oder Zu- bzw. Abschalten einzelner Verbraucher möglich.

Die große Schwierigkeit beim Regeln von Stromnetzen ist, dass die Energie die eingespeist wird auch verbraucht werden muss. Es gibt keine Möglichkeit „überflüssigen Strom“ aufzufangen und zu Speichern, deshalb muss auf einen erhöhten bzw. verringerten Stromverbrauch in Echtzeit mit einer erhöhten bzw. verringerten Stromproduktion reagiert werden[9].

Um in einem Stromnetz eine stabile Spannung zu halten gibt es zwei Regelungsverfahren. Für beide Verfahren verwendet man elektrische Bauteile, die bei Simulationen berücksichtigt werden müssen.

Da gibt es zum einen das FACTS (Flexible-AC-Transmission-System) für Drehstromnetze und das HVDC (High-Voltage Direct-Current) für Gleichstromnetze[9].

2 Software für die Simulation von Stromnetzen

Um die physikalischen Eigenschaften eines Stromnetzes zu analysieren, gibt es diverse darauf spezialisierte Software. Grundsätzlich lässt sich die Software in diesem Anwendungsfall, wie dies auch sonst der Fall ist zwischen Open-Source- und kommerziell vertriebene Closed-Software unterschieden. Auf die grundlegenden Unterschiede dieser Softwaregruppen soll hier nicht eingegangen werden, aber es gibt ein paar Kriterien, die bei der Auswahl der Software speziell beim Thema Stromflussberechnung eine Rolle spielen.

3 Software für die Simulation von Stromnetzen

Um die physikalischen Eigenschaften eines Stromnetzes zu analysieren, gibt es diverse darauf spezialisierte Software. Grundsätzlich lässt sich die Software in diesem Anwendungsfall, wie dies auch sonst der Fall ist zwischen Open-Source- und kommerziell vertriebene Closed-Software unterschieden. Auf die grundlegenden Unterschiede dieser Softwaregruppen soll hier nicht eingegangen werden, aber es gibt ein paar Kriterien die bei der Auswahl der Software speziell beim Thema Stromflussberechnung eine Rolle spielen.

Open-Source-Software Der große Vorteil der Verwendung von Open-Source-Software im Bereich der Analyse von Stromnetzen ist der, dass die verwendeten Algorithmen zum Testen von Alternativen modifiziert und auch ersetzt werden können. Das spielt vor allem in der Forschung eine große Rolle, zum Beispiel, wenn man seinen selbst erdachten oder optimierten Algorithmus testen oder mit den schon vorhandenen Algorithmen vergleichen will. Open-Source-Programme bringen meistens auch Implementierungen aller gängiger Algorithmen von Haus aus mit[9].

Closed-Source-Software Gegenüber dem Nachteil, dass man bei Closed-Source Software die Algorithmen nicht anpassen kann und im Zweifel nicht so viele Algorithmen oder Varianten von Algorithmen zur Verfügung stehen, hat man meist den Vorteil, dass die Implementierung der Algorithmen etwas effektiver und damit die Laufzeit im Allgemeinen etwas besser ist[9].

Ein weiterer Unterschied ist die Art der Bedienung. Manche Softwarepakete lassen sich mit einer Commandline bedienen, andere mit Hilfe einer GUI, außerdem gibt es Softwarepakete, die über beide Schnittstellen verfügen[9].

Commandline Die einfachste Art der Kommunikation mit einem Programm ist die Commandline, besonders aus Sicht des Entwicklers.

Graphical User Interface (GUI) Einige Programme bieten auch eine grafische Oberfläche an. Dies bietet neben der Möglichkeit sich Diagramme zum Beispiel über den zeitlichen Verlauf anzeigen zu lassen auch die Möglichkeit das Schaltnetz in Form eines One-Line-Diagramms anzuschauen. Damit lassen sich Fehler in der Eingabe einfacher finden. Wirklich benutzerfreundlich wird die Bedienung einer GUI aber nur, wenn das Programm auch die Möglichkeit bietet, das Modell des zu simulierenden Stromnetzes mit Hilfe eines CAD (*computer-aided design*) zu erstellen, denn damit ist es auch möglich den Schaltplan mit Hilfe von gegebenen Bauteilen direkt am PC zu zeichnen. Die meisten Programme besitzen keine Eigenentwicklung für die GUI und das CAD. Dies wäre zu aufwändig, stattdessen benutzen sie eine Schnittstelle für das zu Matlab gehörende Simulink. Das hat außerdem den Vorteil, dass man auf eine ausgereifte Oberfläche aufbaut. Dafür steht diese nicht zur Verfügung, wenn man, falls möglich, Octave statt Matlab verwendet. Ein Beispiel für eine GUI ist in Abbildung 2 zu sehen[9].

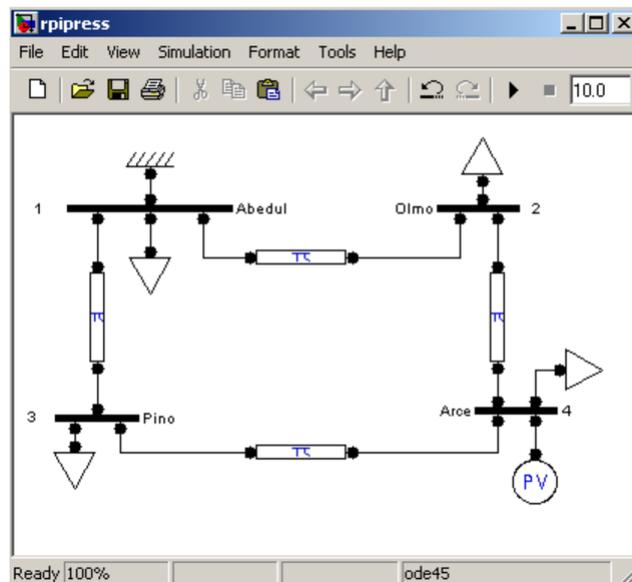


Abb. 2. GUI-Beispiel von PSAT[7]

Neben der Interaktion des Programmes gibt es aber noch einen weiteren Aspekt, der unter die Kategorie Bedienung fällt, nämlich die Speicherung der Daten und der Austausch zwischen verschiedenen Programmen. Grob gesagt

bietet jedes Softwarepaket sein eigenes Dateiformat an, indem die jeweilige Konfiguration des Stromnetzes und die anderen zu speichernden Daten gesichert werden. Dies macht den Austausch der Daten unter den Programmen schwierig, die Kompatibilität wäre aber wünschenswert, da nicht jedes Programm jedes Simulationsverfahren unterstützt und man aber mehrere Probleme simulieren will. Um dieses Problem zu lösen, bieten viele der aufgeführten Programme die Möglichkeit an, die Daten des eigenen Formats in ein anderes Format zu konvertieren.

Viele der im Folgenden beschriebenen Programme basieren auf Matlab. Das hat den Vorteil, dass man auf die umfangreichen Funktionen aus dem Bereich der Mathematik, vor allem der Vektoralgebra, zugreifen kann und diese nicht selbst implementieren muss.

3.1 Eine Auswahl verschiedener Softwarepakete

Im Folgenden werden ein paar der wichtigsten Open-Source-Softwarepakete, die es für die Analyse von Stromnetzen gibt, vorgestellt. Diese Liste ist bei weitem nicht vollständig, soll aber einen guten Überblick über die Unterschiede geben, die es bei der existierenden Software gibt. Welche konkrete Funktionalität die einzelnen Softwarepakete zur Analyse von Stromnetzen mitbringen, wird in Abschnitt 4 behandelt. Außerdem ist im Anhang eine Tabelle, die eine Übersicht über die Funktionalitäten der verschiedenen Softwarepakete liefert.

MatEMTP ist ein Außenseiter unter den hier aufgeführten Programmen. Es ist ausschließlich für die Berechnung der *Electro-Magnetic Transient Analysis* konzipiert und bietet damit für keine der anderen hier aufgeführten Problemstellungen einen Lösungsalgorithmus. *MatEMTP* bietet dem Benutzer eine GUI mit CAD[6].

Power System Analysis Toolbox (PSAT) ist besonders für kleine und mittlere Stromnetze geeignet. Für große Stromnetze ist die Software zu langsam um mit anderen mithalten zu können. Dafür ist *PSAT* nicht nur auf eine spezielle Problemstellung fixiert, sondern bietet als einziges der hier aufgeführten Programme für fast alle Analyseverfahren, die im Folgenden erläutert werden, eine Implementierung an. Das macht es zu einem sehr vielfältigen Tool.

Für die mathematischen Berechnungen, die hinter den Algorithmen stecken, verwendet *PSAT* zwar auch Matlab, aber bei der Entwicklung von *PSAT* wird auch auf die Kompatibilität zu Octave, einem Open-Source-Nachbau von Matlab, das ein Großteil der Matlab-Syntax unterstützt, geachtet. Damit kann *PSAT* auch ohne proprietäre Abhängigkeiten benutzt werden.

Außerdem bietet *PSAT* zur Ein- und Ausgabe eine GUI mit CAD an[8].

MATPOWER benutzt wie auch *PSAT* Matlab im Hintergrund und ist auch zu Octave kompatibel. *MATPOWER* ist bei der einfachen Power Flow-Berechnung eines der schnellsten Open-Source-Programme, das macht es im Gegensatz zu *PSAT* besonders geeignet für große Stromnetze. Es besteht

aus einer Sammlung von Matlab-M-Files, das macht es zu einer Library für Matlab bzw. Octave. Deshalb gibt es für *MATPOWER* auch keine GUI. Eine Besonderheit von *MATPOWER* sind die vielen automatischen Testfälle für den Programmcode, die ein gewisses Maß an Korrektheit des Codes garantieren[14].

UWPflow ist ein Programm für die Forschung, das während der Berechnung viele Werte wie die Jacobi-Matrix, Eigenwerte usw. für die weitere Analyse ausgibt. Bei der Verwendung von *UWPflow* wird ausschließlich auf die Commandline gesetzt[9].

Voltage Stability Toolbox for Power System Education and Research (VST) ist ein von der Drexel University, Philadelphia, PA entwickeltes Programm. Gedacht ist es als Simulationsprogramm für die Lehre. Daher bietet *VST* auch eine GUI zum Ausgeben von Diagrammen an, was die berechneten Ergebnisse anschaulicher macht[1].

4 Algorithmen für die Analyse von Stromnetzen

Um die im Abschnitt 1.1 angesprochenen physikalischen Eigenschaften eines konkreten Stromnetzes zu analysieren, gibt es verschiedene Verfahren, die im Folgenden erläutert werden sollen. Alle diese Verfahren bestimmen die jeweilige Lösung nur näherungsweise, da für praxisrelevante Netze die Bestimmung der exakten Lösung nicht praktikabel ist.

4.1 Power Flow Study

Die grundlegendste zu analysierende Eigenschaft in einem Stromnetz ist die Power Flow Study. Sie beschreibt das statische Verhalten eines Stromnetzes zu einem konkreten Zeitpunkt, wobei für jeden Generator die Leistung zu diesem Zeitpunkt gegeben ist.

Annähern von linearen Gleichungssystemen Zum näherungsweisen Bestimmen einer Lösung eines linearen Gleichungssystems gibt es zwei wichtige Verfahren:

Gauß-Seidel-Verfahren ist ein iterativer Algorithmus zum Annähern der Lösung eines LGS, der sehr schnell gegen die tatsächliche Lösung konvergiert. Gauß-Seidel ist der gängigste Algorithmus zum näherungsweisen Lösen eines Linearen Gleichungssystem. Er hat aber auch den Nachteil, dass er nicht die schnellen Vektoren- bzw. Matrizen-Operationen von auf Vektoralgebra spezialisierte Software wie Matlab verwenden kann, was auch bedeutet, dass er nicht parallelisiert werden kann[9].

Jacobi-Verfahren ist genau so wie das Gauß-Seidel-Verfahren ein iterativer Algorithmus. Er konvergiert zwar langsamer als Gauß-Seidel, hat aber gegenüber diesem den Vorteil, dass für einen Iterationsschritt Vektoralgebra zum Beispiel von Matlab benutzt werden kann, was auch bedeutet, dass man den Schleifendurchlauf innerhalb einer Iteration parallelisieren kann[9].

Annähern von nichtlinearen Gleichungssystemen Eine Möglichkeit, eine Lösung für ein nichtlineares Gleichungssystem anzunähern, ist die Linearisierung desselben. Das gängigste Verfahren dazu ist das Newton-Verfahren, unter anderem auch Newton-Raphson-Verfahren genannt. Dieses funktioniert im Prinzip so, dass für die gegebenen Gleichungen alle partiellen Ableitungen berechnet werden. Damit lässt sich die Jacobi-Matrix aufstellen. Mit deren Hilfe lässt sich ein Iterationsverfahren konstruieren.

Das Newton-Verfahren konvergiert für gut konditionierte Fälle in vier bis fünf Iterationen. Es kann aber auch vorkommen, dass die Lösung trotz weiterer Iterationen nicht erreicht wird. Um auch diese Fälle lösen zu können, muss man das Verfahren leicht modifizieren[9].

Alle im vorigen Abschnitt beschriebenen Softwarepakete implementieren die Power Flow Study mit Ausnahme von MatEMTP.

4.2 Eigenvalue Analysis

Benannt nach den Eigenwerten aus der Linearen Algebra hat dieses Verfahren nicht den Namen des Problems, sondern den der Lösung. Ziel ist es, das dynamische Verhalten eines Stromnetzes über die Zeit zu untersuchen. Gesucht ist, ob das Regelsystem einen Punkt erreicht, an dem sich die Spannung einpendelt. Wenn solch ein Punkt erreicht wird, spricht man von einem stabilen System.

Die Frage ob ein System stabil ist, lässt sich mit Hilfe der Eigenwerte der Matrizen, die sich aus den physikalischen Eigenschaften des Systems ergeben, bestimmen. Sind zum Beispiel die Realteile aller Eigenwerte kleiner Null, so ist das System stabil. Eigentlich ist es für die Berechnung der Eigenwerte das Gram-Schmidt-Verfahren notwendig, dieses ist aber sehr rechenzeitintensiv. Daher greift man zu vereinfachten Methoden, die nur einen Teil der Eigenwerte berechnen, was in diesem Anwendungsfall auch ausreicht, da es für die Stabilitätsbestimmung mit Hilfe der z -Transformation, einer Integraltransformation, nur wichtig zu wissen ist, ob der Betrag jedes Eigenwerts kleiner 1 ist[9,11].

Ein einfaches Verfahren zur Bestimmung des betragsgrößten Eigenwerts ist die Potenzmethode. Bei dieser multipliziert man den i -ten Iterationsvektor auf die aus dem Gleichungssystem resultierende Matrix und teilt das Ergebnis durch die Norm des Ergebnis. Das Resultat verwendet man für den $i+1$ Iterationsvektor[9].

PSAT und *VST* bringen eine Implementierung der Eigenvalue Analysis mit.

4.3 Continuation Power Flow

Die Continuation Power Flow beschäftigt sich mit der Maximalleistung eines Stromnetzes. Ziel ist die Berechnung des Punktes, an dem das Netz wegen einer Überlastung zusammenbrechen würde.

Mathematisch gesehen sucht man die maximal anlegbare Spannung, für die das Gleichungssystem noch eine Lösung liefert.

Berechnen kann man diesen Punkt indem man einzelne Spannungswerte ausprobiert. Dieses Verfahren ist aber nicht besonders performant, ein weitere Möglichkeit ist die Verwendung eines angepassten Newton-Verfahrens, es gibt aber auch noch andere Verfahren, die das Continuation Power Flow Problem lösen[9,2].

PSAT, *UWPflow* und *VST* beinhalten eine Implementierung von Continuation Power Flow.

4.4 Time Domain Analysis

Die Time Domain Analysis betrachtet, wie auch die Eigenvalue Analysis, die Stabilität eines Stromnetzes. Aber im Gegensatz zum Eigenvalue Analysis liegt der Fokus der Time Domain Analysis auf kurzzeitigen Änderungen und der Frage, ob diese die Stabilität des Systems in der Zukunft gefährden.

Zur Berechnung der Stabilitätskriterien wird bei der Time Domain Analysis meistens eine numerische Integration verwendet. Es gibt zwar noch eine andere Methode, *Lyapunov-Methode* genannt. Diese lässt sich zwar schneller berechnen, ist aber weniger genau und durch die Weiterentwicklung der Hardware in den letzten Jahren ist eine numerische Integration mittlerweile praktikabel geworden[9].

Die Time Domain Analysis lässt sich in zwei Teilbereiche aufteilen:

Electro-Magnetic Transient Analysis Diese behandelt Änderungen im Stromnetz, wie zum Beispiel das Zu- und Abschalten von Leitungen.

Das Standard-Integrationsverfahren für Electronic-Magnetic Transient Analysis ist die Trapez-Methode, ein numerisches Integrationsverfahren[9].

MatEMTP ist das einzige der in dieser Ausarbeitung aufgeführten Programme, dass die Electro-Magnetic Transient Analysis implementiert hat.

Transient Stability Analysis Große Störungen im Netz wie z. B. Stromausfälle oder Kurzschlüsse werden mit Hilfe der Transient Stability Analysis analysiert. Eine Problemstellung in diesem Bereich ist, dass sich das Netz innerhalb einer kritischen Zeitspanne, der *critical clearing time* wieder beruhigt haben muss, um die Stabilität gewährleisten zu können. Dies spielt bei der Berechnung der Stabilität in diesem Zusammenhang letzter Zeit immer weniger eine Rolle, da die Reaktionszeiten aktueller Schutzmechanismen schnell genug reagieren, sodass diese schon garantieren, dass die kritische Zeitspanne nicht überschritten wird. Z. B. liegt die *critical clearing time*, bei einem 50Hz-Netz bei 200ms, der Schutzmechanismus greift aber schon bei 80ms[13].

Eine Lösung der Transient Stability Analysis ist in *PSAT* und *VST* implementiert.

4.5 Optimal Power Flow

Verschiedene Arten von Generatoren verursachen unterschiedlich hohe Kosten, außerdem sind die Verluste durch die Stromleitungen zum Beispiel durch deren unterschiedliche Längen verschieden. Da ein Stromnetz an sich selten voll ausgelastet ist, gibt es hier eine Möglichkeit, den Preis durch die Generatoren und durch den Verlust durch die einzelnen Leitungen möglichst gering zu halten. Dieses Problem nennt man Optimal Power Flow.

Optimal Power Flow ist ein nichtlineares Optimierungsproblem. Um dieses zu lösen bietet sich u.a. die Gradientenmethode an[9].

Programme, die das Optimal Power Flow unterstützen sind *PSAT* und *MATPOWER*, wobei bei *MATPOWER* besonders hervorzuheben ist, dass es bei der Optimal Power Flow-Berechnung viele Möglichkeiten für den Benutzer zur Variierung des Problems bietet. Zum Beispiel bietet es die Möglichkeit die Gewichtung der einzelnen Parameter zu variieren und das Problem um zusätzliche Größen zu erweitern[14].

5 Anbindung eines Stromnetzes an den Markt

In Bereichen wie zum Beispiel der öffentlichen Energieversorgung spielt nicht nur die Stabilität der Energieversorgung eine Rolle. Da in einer Region nicht immer der Strom erzeugt werden kann, der auch benötigt wird, ist es auch manchmal nötig, dass sich die Energieversorger gegenseitig aushelfen und sich Strom verkaufen. Dies wird immer wichtiger im Zuge der Umstellung auf erneuerbare Energien, da zum Beispiel Windräder und Solarzellen nicht auf Knopfdruck ihre Leistung erhöhen können. Außerdem möchte man deren Strom, falls dieser verfügbar ist, also zum Beispiel der Wind weht, auch verwenden. Das bedeutet dann, diesen zu verkaufen, falls man ihn selbst nicht verwenden kann.

Betrachten wir ein Stromnetz, das der Einfachheit halber aus mehreren Stromerzeugern und einem Verbraucher besteht. Dabei sind die Stromerzeuger über eine Fläche verteilt und die Leitungen zu den jeweiligen Erzeugern verbrauchen beim Transport unterschiedlich viel Strom, der vom transportierten Strom angezogen wird. Wenn nun der Verbraucher mehr Strom benötigt, kann er sich entscheiden, wie er seinen Bedarf auf die einzelnen Stromerzeuger verteilt. Dieser möchte natürlich einen möglichst geringen Preis bezahlen und da die Erzeuger ihren Preis frei wählen können wird er womöglich den Günstigsten wählen. Doch dabei gibt es zwei Probleme. Zum einen muss der Verbraucher dabei die maximale Kapazität des gewählten Stromerzeugers berücksichtigen. Dieser kann vielleicht gar nicht den kompletten Strom liefern, und nicht nur das, auch die Stromleitungen haben ihre Kapazitäten, die nicht überschritten werden dürfen. Dies alleine kann schon dazu führen, dass, wenn der Verbraucher erst den Günstigsten und dann den nächst Günstigeren wählt und auf diese beiden seinen Bedarf verteilt, dieser nicht mehr am billigsten davon kommt.

Zum anderen hat auch jede Leitung ihren Verlust, der bei der Preisberechnung mit berücksichtigt werden muss. Allein das Problem in Formeln zu fassen ist nicht so einfach, das korrekte Problem ist ein Teil des Optimal Power Flow Problems, bei dem die finanziellen Kosten der Stromerzeugung sowie dem Verlust durch die Leitungen zu berücksichtigen sind. Jedoch ist dieses Problem für große Netze nicht so ohne Weiteres in akzeptabler Zeit zu berechnen. Erschwerend kommt noch dazu, dass der Strompreis durch den Verlust an den Leitungen an jedem Verzweigungspunkt des Netzes ein anderer ist und für jeden dieser Punkte das Optimum neu berechnet werden muss. Eine Vereinfachung dieses Optimierungsproblems ist das Location Marginal Pricing[4].

5.1 Location Marginal Pricing (LMP)

Beim Location Marginal Pricing wird das nichtlineare Optimierungsproblem der Stromkosten an einem bestimmten Knoten mit einem linearen Optimierungsproblem angenähert. Der erste Schritt dazu ist die Wahl eines Referenzknotens. Nun wird jeder Preis relativ zu diesem Knoten angegeben.

Grundlage des LMP ist eine Formel, die beschreibt, welche Kosten für eine bestimmte Konfiguration entstehen[4].

$$\lambda_i = \lambda - LF_i \cdot \lambda + \sum_{k=1}^K S_{ik} \cdot \mu_k \quad (5)$$

Zuerst wird ein Referenzknoten berechnet, mit dessen Hilfe man dann die Preisunterschiede an verschiedenen Knoten berechnen kann. Das in der Formel vorkommende λ steht für den Preis am Referenzbus, die LF_i beschreiben den Stromverlust am Knoten i im Verhältnis zum Referenzbus und die Summe beschreibt die Beschränkungen. Dabei steht μ_k für den Schattenpreis der jeweiligen Leitung k und S_{ik} für die Beschränkung des Einflusses einer Stromeinspeisung an Knoten i auf die Leitung k . Als Ergebnis bekommen wir den Schattenpreis λ_i des Knotens i im Verhältnis zum Referenzbus.

Wenn man nun die Kosten aller Generatoren (6) und aller Verbraucher (7) aufsummiert und mit deren jeweiligen erzeugenden (P_i) bzw. verbrauchenden (L_i) Strommengen multipliziert, erhält man folgende Gleichungen:

$$-\sum_i \lambda_i \cdot P_i = -\sum_i (\lambda - LF_i \cdot \lambda + \sum_k S_{ik} \cdot \mu_k) \cdot P_i \quad (6)$$

$$\sum_i \lambda_i \cdot L_i = \sum_i (\lambda - LF_i \cdot \lambda + \sum_k S_{ik} \cdot \mu_k) \cdot L_i \quad (7)$$

Durch Addieren von (6) und (7) ergibt sich das Gesamtergebnis:

$$-\lambda \cdot \sum_i (P_i - L_i) + \lambda \sum_i LF_i \cdot (P_i - L_i) - \sum_k \mu_k \sum_i (P_i - L_i) \cdot S_{ik} \quad (8)$$

$$= -\lambda \cdot Loss + \lambda \cdot Loss_{marg} - \sum_k \mu_k T_k^{\max} \quad (9)$$

Dabei beschreibt $\sum_k \mu_k T_k^{\max}$ die Beschränkungen des Netzes durch die Kapazitäten T_k^{\max} der jeweiligen Leitungen k . $Loss$ steht für den Verlust durch den Transport des Stroms und $Loss_{marg}$ den Grenzverlust[5,4].

Um das so erhaltene lineare Optimierungsproblem zu lösen bietet sich das Simplex-Verfahren an[10]. Eine Möglichkeit, die Strompreisbildung mit Hilfe von Location Marginal Pricing zu Simulieren, bietet die Open-Source-Software *AMES Wholesale Power Market Test Bed*[3].

Doch mit dem LMP ist es noch nicht getan, es beschreibt erst das Optimierungsproblem, das erst noch gelöst bzw. angenähert werden muss, um die günstigste Möglichkeit, den Strom zu kaufen, auch annähernd bestimmen zu können.

5.2 Weiterführendes

Dieses Verfahren ist in Bezug auf den Preis erst einmal statisch. Der Anbieter setzt den Preis und der Käufer kauft ein oder eben auch nicht. Das ist vielleicht etwas unbefriedigend, wie in anderen Großmärkten auch, möchte man auch im Strommarkt mit Auktionen arbeiten, die den Preis variabel gestalten, um direkter auf Angebot und Nachfrage eingehen zu können. Nur hat der Strommarkt eine Tücke, da kein Strom gespeichert werden kann, muss diese in kürzester Zeit verkauft, bzw. zugekauft werden. Konkret stehen dafür 15 Minuten zur Verfügung, wenn in dieser Zeit die Spannung nicht reguliert wird, kann es zum Beispiel bei Leitungen zu einer Überbelastung und damit zu einem Stromausfall kommen. Dafür ist eine „klassische“ Auktion zu langwierig, sodass man sich eine neue Form der Auktion einfallen lassen muss. Eine in der Praxis in Deutschland verwendete Methode ist eine automatisierte Auktion die auf ein sogenanntes Agentensystem, *DEZENT* genannt, aufbaut. Dabei existiert für jeden Knoten zu einem anderen Anbieter ein elektronischer Agent, der sich mit den Agenten der anderen Anbieter mit Hilfe von Algorithmen auf einen Preis einigt.[12]

6 Fazit

Für die Analyse von Stromnetzen muss mehr beachtet werden wie die Ohm'schen und Kirchhoff'schen Gesetze. Eine stabile Regelung und das Verhalten bei einer Störung spielt genau so eine Rolle.

Um diese Fälle zu analysieren, gibt viele verschiedene Softwarepakete. Es gibt Software, die auf die Lösung weniger Teilprobleme der Stromflussberechnung spezialisiert ist, wie z. B. *MatEMTP*, und welche, die für möglichst viele

Problemstellungen Lösungen anbieten, wie z. B. *PSAT*. Diese unterscheiden sich aber nicht nur in ihrem Funktionsumfang was Stromnetze angeht, sondern auch in der Art ihrer Bedienung. Ein Teil wird mit der Kommandozeile bedient, andere haben eine graphische Oberfläche. Außerdem eignen sich manche auch für besonders große Netze, andere sind dafür zu langsam.

Außerdem wurde gezeigt, dass sich die Analyse nicht nur auf die Berechnung von physikalischer Größen beschränkt, sondern, dass auch die Preisberechnung in einem Stromnetz nicht so einfach ist. In diesem Zusammenhang wurde das Location Marginal Pricing vorgestellt. Es ist das Standardverfahren für die Preisberechnung in Stromnetzen. Um das Location Marginal Pricing in einem Stromnetz zu berechnen, wurde die Software *AMES Wholesale Power Market Test Bed* kurz vorgestellt.

Die Analyse von Stromnetzen wird in der Zukunft eine immer wichtigere Rolle spielen, da durch die größere Verbreitung der erneuerbaren Energieerzeugung die Komplexität solcher Netze zunimmt. Die Stromerzeugung zum Beispiel in Windparks und Solarkraftwerken lässt sich nicht auf Knopfdruck erhöhen, daher wird es in Zukunft immer wichtiger, Strom auch über weite Strecken zu transportieren. Dies wird auch den Strommarkt beleben, da nicht jeder Energieversorger auch überall auf dem Kontinent eigene Kraftwerke unterhalten kann. Außerdem müssen Stromnetze in Zukunft flexibler sein, da der Strom nicht immer vom gleichen Kraftwerk kommt, sondern zum Beispiel je nach Wetterlage aus anderen Regionen.

Damit wachsen die Anforderungen an die Analyse von Stromnetzen, was es auch in Zukunft nötig macht, in diesem Bereich zu forschen und gegebenenfalls auch leistungsfähigere Softwaretools zu entwickeln.

7 Anlagen

Tabelle 1. Funktionalität der einzelnen Softwarepakete[9,8,14,1,6]

Name	Power Flow Study	Optimal Power Flow	Eigenvalue Analysis	Continuation Power Flow	Time Domain Analysis	GUI
MatEMTP					EMT	✓
Matpower	✓	✓				
PSAT	✓	✓	✓	✓	TSA	✓
UWPflow	✓			✓		
VST	✓		✓	✓	TSA	✓

Literatur

1. Aasun, S., Nwankpa, C.O., Kwatny, H.G.: Voltage stability toolbox for power system education and research. IEEE Transactions on Education (2006)
2. Ajarapu, V., Christy, C.: The continuation power flow: a tool for steady state voltage stability analysis. IEEE Transactions on Power Systems (1992)
3. Li, H., Tesfatsion, L.: Development of open source software for power market research: The ames test bed, http://www2.econ.iastate.edu/tesfatsi/OSS_AMES.2009.pdf
4. Litvinov, E.: Power system & Imp fundamentals, <http://www2.econ.iastate.edu/classes/econ458/tesfatsion/Imp.AdvancedWPM.ELitvinovWEM301.pdf>
5. Litvinov, E., Zheng, T., Rosenwald, G., Shamsollahi, P.: Marginal loss modeling in Imp calculation. IEEE Transactions on Power Systems (2004)
6. Mashedjian, J., Alvarado, F.: Creating an electromagnetic transients program in matlab: Matemp. IEEE Transactions on Power Delivery (1997)
7. Milano, F.: A brief introduction to the power system analysis toolbox (psat) for power system analysis undergraduate courses, <http://faraday1.ucd.ie/archive/psat/brieftutorial.pdf>
8. Milano, F.: An opensource power system analysis toolbox. IEEE Transactions on Power Systems (2005)
9. Milano, F.: Power System Modelling and Scripting. Springer Verlag London Limited (2010)
10. Nelder, J.A., Mead, R.: A simplex method for function minimization. The Computer Journal Vol. 7 (1965)
11. Puente León, F., Kiencke, U., Jägel, H.: Signale und Systeme. Oldenbourg Wissenschaftsverlag Berlin, Boston (2011)
12. Struß, O.: Open-source-modellierung und auktionsorientierte regulierung dezentraler netze (2012)
13. Varaiya, P., Wu, F.F., Chen, R.L.: Direct methods for transient stability analysis of power systems: Recent results. Proceedings of the IEEE (1985)
14. Zimmerman, D., Murillo-Sánchez, E., Thomas, R.J.: Matpower: Steady-state operations, planning, and analysis tools for power systems research and education. IEEE Transactions on Power Systems (2011)

Sensor Emulation with Machine Learning Methods

Philipp Schmurr *

Advisor: Anja Bachmann †

Karlsruhe Institute of Technology (KIT)
Pervasive Computing Systems - TECO
* philipp.schmurr@student.kit.edu
† bachmann@teco.edu

Abstract. The number of sensors in daily life has been increasing rapidly over the last years. Everyone carries a whole set of sensors around with his smartphone or other smart devices. Moreover robot platforms become overloaded with different kinds of sensors due to the fact that the cost and weight of sensors is decreasing continuously. Also a new kind of sensor usage has arisen with the distribution of battery driven sensor networks querying all wanted data from an area. With this tremendously increasing number of sensors used, the problems, sensors are bound to, are more present than ever before. Especially because some sensors are that mass produced that quality testing became luxury, resulting in more faulty sensors. But computer science is facing every kind of problem that occurs with always new algorithms and optimizations. One kind of these approaches is sensor emulation, which is the topic for this paper. Therefore three different approaches will be presented and evaluated here. Namely sensor modelling, sensor correlation and sensor virtualization. In this article I am not going to present my own work. This paper only acclaims to be a summary of existing work and is created as part of my studies on computer science at Karlsruhe Institute of Technology.

Keywords: Sensor Emulation, Sensor Correlation, Machine Learning, Sensor Networks

1 Introduction

This paper will give an overview on some kinds of sensor emulation which are further explained and motivated in the following subsections. Later on, exemplary use cases are explained more detailed. For the start, the topic sensor emulation will be motivated in a general manner. To do so, the core reasons for sensor emulation will be investigated. This process is repeated for each of the presented approaches.

General reasons are problems in terms of sensor usage. Some of these problems will be introduced in this paper and an approach to encounter such a problem will be given. One group of problems is the unreliability of the sensor itself and its data. Emulation is then an attempt to make sensors more useable.

1.1 Sensor model

The first approach is to encapsulate the sensor into a model which masks the real sensor and is able to apply calculations on measured values and handle possibly provided external data appropriately. Problems that can be addressed by this step are low sampling rates. The sensor then needs to be emulated between its real measurements to fulfill the user's conditions, this is well-known as up-sampling. The next problem is that sensors can be damaged after a period of usage and it is not possible to replace it, e.g. pressure sensors in the walls of a water dam. Now the model is required to have learned from old data to continue its work. The cooperation with other sensors within the area is also a possibility [1, 9]. The advantage of a model replacing the sensor is that the user must not care about sensor defects. Of course, the user should be notified that the presented data is no real measurement any more. In some cases the objective might be to stop querying the sensor very frequently and use a model that has been trained with previous data [1, 9, 10]. This addresses the problem of limited energy supplies on some sensor platforms. In general, basic calculations are much more energy-efficient than querying a sensor.

1.2 Sensor Correlation

If multiple sensors exist in a system, guessing one sensor's value only from the data of the others can be used to reduce energy costs again. Still, the generated values shall fulfil the requirement of a low variance to the real value. It is necessary that the system detects correlations between sensors [1]. The already mentioned problem of replacement is encountered by this sensor cooperation. These days, there are many approaches of such sensor networks with low-priced battery driven nodes, expiring when their energy budget is exhausted [1, 10]. Energy cost is different on every sensor type. Hence, knowledge about correlations between different sensor types helps to save energy by using the more efficient acquisition mode. For example the battery voltage of a sensor node with a temperature sensor can be used to emulate the value of the temperature sensor, because these sensors are correlated [1]. This is based on the fact that battery voltage is temperature-dependent and increases with higher temperature. How correlations can be detected and used in such networks is explained in Chapter 3, which is mainly based on [1]. There is presented a query processor for such networks that is capable of this detection and their usage.

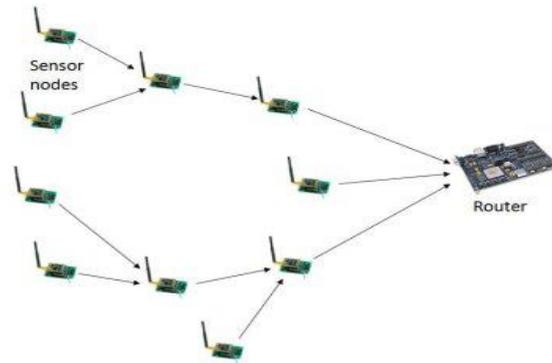


Fig. 1. Sensor Network example.

1.3 Virtual Sensor

Another problem that encourages sensor emulation is the variance in sensor measurements called noise [2]. Under special circumstances the real sensor value can be totally occluded by this noise. For example a compass sensor is not useable at all if it is placed close to a magnetic material or any kind of magnetic field. This requires preprocessing on the data to reduce noise and make it useable. This can be done with a learning model that receives noisy input and creates valuable output information. This approach is called a virtual sensor [2] and is dedicated to problems where sensor correlation is not an option, for example in real time projects like mobile robots, where a high sampling rate and reliable sensor values are inevitable. This is due to the fact that the process of robot action planning is very complex itself [2], even with good conditioned input data. The approach of a virtual sensor uses trainable models to map the often very high dimensional, noisy input data e.g. the picture taken by a camera on-board the robot, to noise free virtual values. This approach is mainly based on a paper by Shridhar Mahadevan et al. [2] about speeding up the learning process of mobile robots using this technique. The realization and usage will be explained in Chapter 4.

2 Modeled Sensor

There are multiple ways of creating a model for a sensor. But for every use case another approach is to be favored. The first important point for this decision is whether the sensor can get external data to work with, or not. In most cases time can be measured by the sensor devices themselves so this can be considered not to be external data. Then it is possible without the external input of time data to query the sensor in fixed time intervals.

2.1 Closed box model

If no data can be given to the sensor, then the only possibility for emulation is keeping a set of last measurements. Due to limited memory the oldest value becomes dropped, if the capacity is exceeded to create new space. If the data of the sensor can be considered linear then using a regression method [8] to obtain a function approximating the real value is an option. If the data is more complex signal processing methods can be used to generate a more complex function, for example a Fourier row [4]. Depending on the type of sensor it may also be possible to analyze the probability distribution of the sensor values. This can be done by statistical methods like the Chi-squared test. Then the expected value of the distribution can be used as an emulated sensor value [8]. All these approaches are only applicable if the sensors' data is relatively stable and tolerances to the real value are not a problem. Moreover real readings are required to update the stored data to react to changes the sensor can experience. So these techniques may be used to save battery or handle low sampling rates.

2.2 Input handling model

If the sensor is capable of receiving external data, it is to be distinguished if the kind of this data is known by the sensor model, or not. Unknown data formats can, for example, be given to an artificial neural network with output neurons for the sensor's data ranges, or any other appropriate learning model [2, 3, 7]. During the training period the sensor uses its measured values to adapt the model for the given input and real sensor value. Input data in a known format can be used more directly. For the case that this input data consists of values of another sensor, the next chapter of this paper will give some answers. The remaining possibilities for input data are for example user inputs and more abstract data, like the season of the year, what would be an informative input for a temperature sensor. In cases where the sensor is connected to a major computation device, such a modelling with input data will be done on the superior system where all the information is stored and processed. Only in cases where the central system is low on computation power and most things are outsourced to the smaller components these concepts will be in use.

3 Using Sensor Correlation

This chapter summarizes the results acquired from the already mentioned article about sensor networks. The authors have created a new kind of database system in the style of TinyDB [9], but more powerful in some cases. Especially in terms of energy efficiency, because they use a probability model instead [1].

3.1 Overview

At first general information on the system will be given. It is called BBQ, which abbreviates the name “Barbie-Q: A Tiny-Model Query System”. This model is used for generating approximations for sensor network queries from a user. Therefore it uses probabilistic confidences for each sensor value that could be queried and generates an optimal plan for acquiring real data from sensors. The model can detect faulty sensor data and extrapolate missing sensor values. Hence the user does not see these problems which occurrence has already been mentioned before and is even more present while working with sensor networks, due to the fact that more sensors exist. The user of this query system can now focus on achieving his goals without the need of taking care.

3.2 Basic functionality

To lower energy cost in a radical way it is necessary to only query sensors increasing the probability of the predicted value to a maximum. That is, if a request for a sensor value is given by the user, the system has to check whether the probability of this value is already high enough, which results in immediate return of the value. Otherwise it has to query for some sensor data, balancing between energy cost and maximum increase of the required probability. Therefore the system can use data from other sensors that are known to be correlated with the requested sensor and have a lower energy cost to acquire data. For example proximate temperature sensors are going to be correlated very likely. The system is capable of learning which sensors are correlated. This is done in BBQ by using multivariate time-varying Gaussians and Kalman Filters [1, 5, 6, 8]. They store a probability density function (pdf) that assigns a probability for every sensor attribute in the network which is going to decrease when time passes without sensor readings. For every sensor attribute the Gaussian gains an additional dimension, so this can be very high dimensional. As the values returned are only approximate values the user is given the possibility to pass error tolerances and confidence bounds to the system together with his query. That way he is able to adjust how much uncertainty he is willing to accept.

3.3 Mathematical Background

To begin explaining the backgrounds the Gaussian or Normal distribution is mentioned first. In the one dimensional case this is a well-known probability distribution

with a density function that takes two parameters: the mean value μ , also known as the expected value and the variance σ^2 . For the multivariate counterpart [1, 5, 8] of dimension n , μ becomes an n -dimensional vector. But σ^2 is going to be a matrix Σ with n rows and n columns. On the diagonal of this matrix the typical variances of each dimension are stored. For all fields not seated on the diagonal line the values indicate the covariance between the two dimensions. It is noticeable that this value has to be the same for both sides of the diagonal, so this matrix is symmetric. Covariance can be used to express correlation between two dimensions. The higher the absolute covariance the higher the correlation. This can be expressed with the Pearson Correlation Coefficient [8]:

$$\rho(X, Y) = \frac{C(X, Y)}{\sqrt{V(X) * V(Y)}}$$

Where C is the covariance between X and Y , while V is the corresponding variance. A value not equal zero is called positive or negative correlation between X and Y respectively. When querying for a specific sensor value the system first checks the expected value stored in the Gaussian. In this case this is just selecting the corresponding row in the vector of mean values μ . Then it has to be calculated whether the probability of this value is still high enough. Therefore the probability of the mean value is calculated by inserting its value in the marginalized density function for this dimension. Usually a level of error tolerance is given, so the system has to check for:

$$\mathcal{P}(X \in [\mu - \epsilon, \mu + \epsilon])$$

Which is the probability that the value is close to our expectations. This is not solvable by a closed formula [1]. To solve this, such an integration has to be performed:

$$\int_a^b p(x_i) dx_i$$

This is known as the error function, for which many good approximations exist. The resulting probability indicates if the corresponding mean value is satisfying the request or querying new sensor data is required.

Marginalization has just been used to create the density function required to calculate the probability. This is very simple using Gaussian distributions. The given set of attributes is called \mathbf{Y} in the following. From the matrix Σ and the vector μ all columns and rows not having a corresponding entry in \mathbf{Y} are dropped. This leads to a matrix $\Sigma_{\mathbf{Y}\mathbf{Y}}$ and a vector $\mu_{\mathbf{Y}}$. Then the calculations can be made in a dimension-reduced system, in the easiest case only one dimension remains.

3.4 Working with observations

Once a lack of confidence was detected it is unavoidable to query the sensor network for some sensor values to increase the confidence. Therefore it is necessary to work with conditioned probability density functions [1, 5].

$$p(X_1, \dots, X_{j-1}, X_{j+1}, \dots, X_n | x_j)$$

This can be modified by the Bayesian rule to:

$$\frac{p(X_1, \dots, X_{j-1}, x_j, X_{j+1}, \dots, X_n)}{p(x_j)}$$

The conditioned density function is also called posterior density function. Now the probability of the sensor request can be computed again, but using:

$$\mathcal{P}(X_i \in [a_i, b_i] | x_j)$$

The advantage of Gaussians is that the conditioned probability density function is again a Gaussian distribution. So this can be repeated with additional sensor queries. But in between the matrix Σ and the vector μ have to be updated correctly. This can be done by matrix operations. Considered the observations of the attributes \mathbf{O} containing the values \mathbf{o} have been made. Then \mathbf{Y} is defined to be the remaining attributes. Then the new matrix $\Sigma_{Y|o}$ can be calculated by:

$$\Sigma_{Y|o} = \Sigma_{YY} - \Sigma_{YO} \Sigma_{OO}^{-1} \Sigma_{OY}$$

Here the first index is used to indicate which rows are to be selected and the second index is used for the columns to be selected. The inverse is always calculated after selecting the reduced dimensional matrix, not the values from the inverse matrix are picked. The new matrix does not rely on the actual values observed \mathbf{o} . Now still the mean vector μ has to be updated. This is done by:

$$\mu_{Y|o} = \mu_Y + \Sigma_{YO} \Sigma_{OO}^{-1} (o - \mu_O)$$

If a correlation existed between sensors in \mathbf{O} and \mathbf{Y} their corresponding vector entries might become changed.

$$\mu = \begin{pmatrix} 13 \\ 15 \\ 9 \end{pmatrix}, \Sigma = \begin{pmatrix} 1 & 0,8 & 0,5 \\ 0,8 & 2 & 1,2 \\ 0,5 & 1,2 & 3 \end{pmatrix}$$

$$\mu_{Y|o} = \begin{pmatrix} 13 \\ 9 \end{pmatrix} + \begin{pmatrix} 0,8 \\ 1,2 \end{pmatrix} * 0,5 * (16 - 15) = \begin{pmatrix} 13,4 \\ 9,6 \end{pmatrix}$$

$$\Sigma_{Y|o} = \begin{pmatrix} 1 & 0,5 \\ 0,5 & 3 \end{pmatrix} - \begin{pmatrix} 0,8 \\ 1,2 \end{pmatrix} * 0,5 * (0,8 \quad 1,2) = \begin{pmatrix} 0,68 & 0,02 \\ 0,02 & 2,28 \end{pmatrix}$$

Fig. 2. Example showing the modifications when Sensor 2 becomes measured a value of 16. Notice the variance reduction of the correlated sensors 1 and 3.

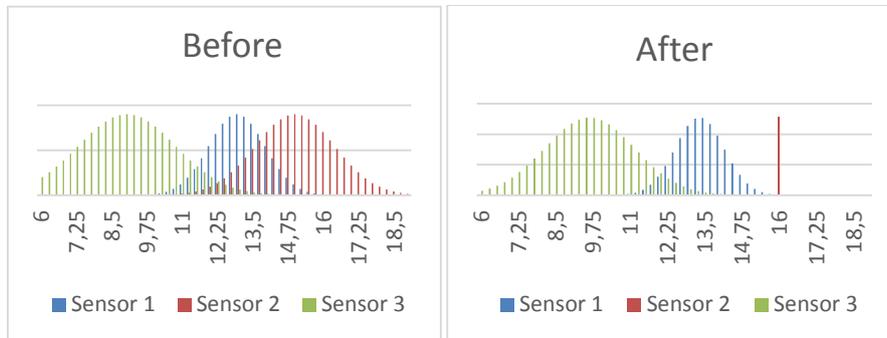


Fig. 3. Change of the distributions of the Sensors to the example in Fig. 2.

3.5 Dynamic Temporal Model

At this point the definition of the system does not apply any changes over time and correlations cannot be detected yet. Until now, this is a time-static model where all calculations are limited to be executed within a discrete time step to fulfill user queries. But all changes done to the matrices are dropped as soon as the system reaches the next time step. Although it is important to keep all sensor queries made in that time step including their value. As soon as a sensor was queried within a time step, the static system recognizes it to be true. Further on, it will be explained what happens if a new time step is reached. In this case a so-called Kalman Filter is required [1, 6]. This is a way to enable discrete temporal steps for the used multivariate Gaussian. Kalman Filters require some matrices to be defined by the user. The first is the state transition model which effects the values the filter will predict. For test cases an identity matrix can be used, so that the system estimates the last vector of means mentioned earlier. This is the mean vector to the beginning of the time step, not the one modified by the sensor lookups and conditional probabilities during the time step. The values of this matrix may be enhanced with additional knowledge about the sensor network, for example if a sensor network that is equipped with temperature sensors is observed, knowledge that the temperature will increase after sunrise and decrease after sunset can be used. The system can be advised to use another matrix in different parts of the day. This can be done by using a modulo operation on the currently valid timestamp to select from a pool of matrices [1]. So the matrix used in the morning would have values greater one, because it is likely that the next measured value will be higher than the last one. In the evening the values should be below one to have the opposite effect. The next matrix is the control input model, it maps a vector of values directly to the state predictions the system will do. In the case here, this is not needed because the system does not require special values to be inserted. Matrix number three is the covariance matrix of another normal distribution. It is used to express a noise that is applied to sensor reading covariance estimations done by the system. Its values tell the system how strong a sensor reading will impact the estimations created by the system. The last matrix that has to be given is again a covariance matrix of yet another normal distribution. This time it is the system's noise. Or in other words, the

entries in this matrix tell the system how strong the elapsed time decreases the confidence on the sensor values estimated within the system. In the next subsection it will be explained how calculations are done with these matrices.

3.6 Kalman Filters

In the operations that are executed by this filter there are two major steps [6]. First the system predicts data concerning the system's state like the next mean vector, the next sensor value and the covariance matrices for both of those. Therefore the system has to know which sensors have been queried during the last time step. Their actual value comes into play just after this, in the second major part of a Kalman Filter, to verify the predictions made in the first step.

There are two major predictions. The mean vector prediction and the prediction of a new covariance matrix. Both of them are based on the last time step's data, combined with the already mentioned state transition model matrix. Additionally there is added some other data. In the case of the mean vector this would be the control input if needed and the covariance matrix is updated with the system's noise matrix. A third prediction is the sensor measurement covariance prediction. It is generated with the covariance prediction and the measurement noise matrix. Its purpose is to define the influences of new measured values on the next time step.

Out of all data acquired till now the so-called optimal Kalman gain, the core of the filter, can be calculated. It is then used to create the final matrix and vector for the next time step. Therefore it is combined with the predictions and a vector of actually measured sensor values. These calculations are able to manipulate the covariance between sensors and detect correlations.

Finally the system is able to do what the base idea of sensor correlation was, so it can be used to emulate values of sensors with only knowledge of another sensors' values. Until the next time step operations are performed by the model just using the static approach again to satisfy all incoming requests within a time step, operating on the new generated matrices.

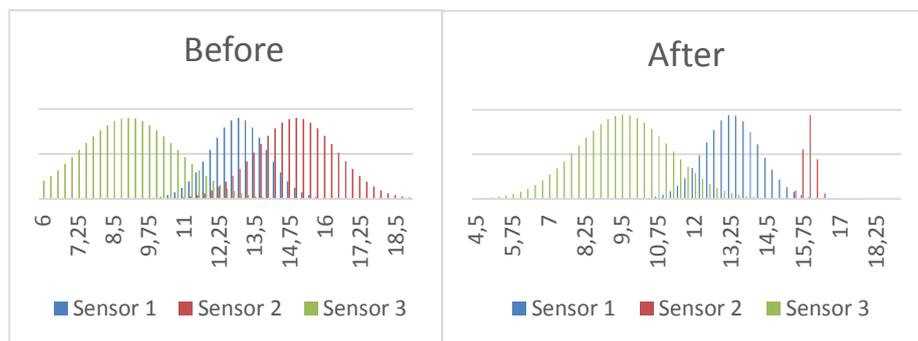


Fig. 4. Example distribution before and after applying a Kalman filter on the example of Fig. 2.

3.7 Summary

As mentioned this approach is capable of facing the problems of energy cost and any kind of sensor failure. But there are still some limitations that will now be given here. First of all: the starting point. The first probability density function has to be acquired by already given data sets. So if no data is given at the beginning this approach will not work properly. Also does this model rely on a spatial-static network with only little changes to a sensor's position [1]. That is why a usage in a mobile system can be ruled out. Overall, this approach has a high potential for working on sensor networks and may be combined with another system focusing on energy efficiency but targeting the network communication itself [10, 13]

4 Implementing Virtual Sensors

This chapter will explain how the concept of virtual sensors can be used. Therefore the scenario taken from the paper "Rapid concept learning" [2] by Sridhar Mahadevan et al. will be explained.

4.1 Scenario

The virtual sensor approach is applied here to help a robot to get more distinct data from its real hardware. The robot used is a small wheel driven platform which is equipped with a camera for the detection of trash cans and a ring of 16 ultrasonic and IR sensors to navigate. All of these sensor types have problem with irregularities. The camera may be dazzled by bright light and the ultrasonic sensors reflect on flat surfaces so the measured value is greater than in reality. So what is needed here is something capable of taking the set of input data and decide what the data represents. In the case of [2] this is simplified by splitting the input data into several parts, like the ultrasonic sensors placed on the front side of the robot, before giving it to the input interface of their virtual sensor. In fact the sensor data of the ultrasonic sensors gets integrated into a 32x32 pixel local occupancy grid, with the values black (if the pixel is occupied by a wall), white (if it is free space) and grey (if the data is unknown). Then for every direction of the robot, namely front, back, left and right the corresponding half of this grid is passed to the virtual sensor's input interface. The output generated by the sensor is one of the following percepts: wall, opening, door and unknown. With this view on the robots position it is much easier to plan next steps.

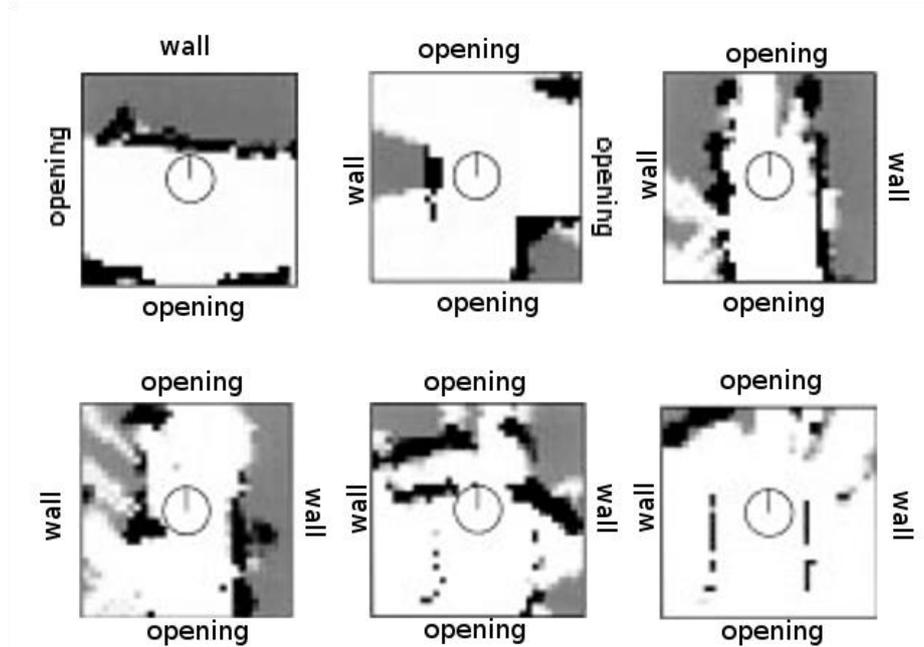


Fig. 5. Some example grids generated from ultrasonic sensor data, labeled with the wanted result. Such 32x32 pixel grids are used as input for the sensor interface.

4.2 Artificial Neural Network

The just mentioned virtual sensor calculates its data with an artificial neural net [2] using an optimized back propagation algorithm called “quick propagation method” [3]. The net has 512 inputs for the pixels of the local occupancy grid, 32 hidden neurons and 4 outputs representing the above mentioned output percepts. Although many of the readers of this paper will already know the concept of artificial neural networks, a short summary on the functionality of such a structure will be given [14]. Like described above the network has three layers, namely input, output and the so called hidden layer. Every layer consists of a given number of artificial neurons. These neurons take an input vector, then multiply this vector with a weight vector, which is stored in the neuron and sum up the values of the target vectors dimensions. If the value shall be greater than a certain threshold, this threshold is subtracted from the calculated value. This single dimension real value is given to an activation function. In most cases this is a logistic function, also called sigmoid function. This is the case because the other activation functions are not differentiable. So the input data is given to the input layer neurons which perform their action on the input. The neurons of each layer are connected to all of the neurons in the next layer. Sometimes the factor of such a connection might be zero, which is represented by a zero value in one of the weight vector dimensions. Then this connection can be considered to be not exist-

ent. After the first layer has finished calculations, the results become the input vector for the next layer, performing same actions again until the output layer is reached and the output vector is created. The resulting vector has always the same number of dimensions as the number of neurons in the last layer. As explained the network generates a result vector for the input data. In the beginning this will not give any good results, because the network has to be trained first. Therefore the back propagation algorithm will be explained here. To train a network it is necessary to have a set of training data with expected results. So one after the other an input vector is taken from the sets and evaluated. This is called forward propagation. Once the result is calculated it can be compared to the wanted output vector. This is done by subtracting the expected vector from the received vector and square the resulting error vector. Using the error vector changes in the weight vectors for every node can be calculated. So the process back propagates through the net to the beginning, changing all the weights by its specific value.

Now this knowledge can be applied to the robot example. After a period of training the artificial neural network will compute very good values for any input grid making the intentions of this approach possible: easier concept learning for the robot's planning layer. And this was achieved by solving the noise problem the sensors are bound to.

4.3 Summary

Now it shall be discussed when this technique is good practice and when better not to use it. First of all a trainer or supervisor is needed, who tells the network what is correct to change the weights appropriately [2]. Also exist cases where the approach would simply be not applicable due to an extreme amount of work for the supervisor. For example a model of pressure sensors in a water dam where every of maybe more than 1000 sensors has to be told its specific values. It is way better if an approach using sensor correlation is used, because the process is equal in every sensor, but the absolute values might differ. On very small devices neural networks may not be applicable, because they require a lot of computations, what would exceed a small devices' capacities. On the other side the neural network approach is solid and has very good results. It is used in many projects, for example in a project that monitors changes in a forest by processing pictures taken from the area [11]. Another example is a project using an artificial neural network to detect different kinds of gases in their environment [12].

5 Conclusion

To finish this paper a more general view on the applicability of the given approaches shall be acquired. Therefore a set of sensor types will be observed on their capabilities of being used for sensor emulation and which approach matches best.

In general all sensors have to be partitioned in two categories. On the one hand the group of sensors used in industrial processes where high precision and quick reactions to changing values are required. On the other hand the group of sensors with higher error tolerances in their usage areas. Only for the second category emulation is a good approach. In the following all sensors mentioned are considered to be in this second group.

At first temperature sensors will be evaluated, because they are important in the correlation chapter. Temperature usually evolves slowly throughout the day but can be very different in different seasons of the year. The temperature data is often not required to be absolutely exact, which results in a comfortable variance. This makes sensor emulation applicable and the best way to do this can be considered a correlation detecting system. If no other sensors are available the approach of a modeled sensor is also applicable.

The second sensor type mentioned includes distance sensors. Here all sensors that are bound to irregularities are a good field of application for some of the methods mentioned. Distance sensors are used to detect major changes in their measurements, e.g. if something moves into the observed area. That is the reason why correlation detection cannot be applied here, because the sensors measuring another direction will not have a decreasing value just because another has. The virtual sensor approach is perfect for this type of sensors, when multiple are available, because it handles all of the just mentioned cases correctly and reduces the noise in the data received. For single sensors the modeled sensor is not applicable, due to the mentioned fact that this type of sensors is usually used to detect differences in the measurements to react appropriately.

At last the sensors mounted on a smart device will be evaluated for their applicability for sensor emulation. Nowadays these devices are equipped with temperature sensors, acceleration sensors and pressure sensors. On some devices even more sensors will be installed, but the focus lies on the just presented ones. First of all a smart device has limited energy supplies. Second installing high-end components like industrial sensors is too expensive for the end user market. These facts acclaim to be a perfect environment to apply the presented approaches. On a single device it still is difficult to apply the approaches, because no reference sensors are available and no supervisor tells the sensors what the current values might consist of. But close to everyone carries such a device around. If all these devices would cooperate this could be considered the greatest sensor network in the world.

References

1. A. Desphande, C. Guestrin, S. R. Madden, J. M. Hellerstein and W. Hong. Model-Driven Data Acquisition in Sensor Networks. 2004.
2. S. Mahadevan, G. Theodorou and N. Khaleeli. Rapid Concept Learning for Mobile Robots. 1998.
3. S. E. Fahlman. Faster-learning variations on back-propagation: An empirical study. 1988
4. F. Puente León, U. Kiencke and H. Jäkel. Signale und Systeme. 2011.
5. K. B. Petersen and M. S. Pedersen. The Matrix Cookbook. November 2012.
6. L. Kleeman. Understanding and applying Kalman filtering. January 26, 1996.
7. Y. Bengio. Markovian models for sequential data. 1999.
8. C. R. Shalizi. Advanced Data Analysis from an Elementary Point of View. 2012.
9. S. R. Madden, M. J. Franklin, J. M. Hellerstein and W. Hong. TinyDB: an acquisitional query processing system for sensor networks. 2005.
10. A. Waada, S. Olariu, L. Wilson, M. Eltoweissy and K. Jones. Training a Wireless Sensor Network. 2005
11. S. Gopal and C. Woodcock. Remote Sensing of forest change using artificial neural networks. 1996.
12. JC. Patra, G. Panda and R. Baliarsingh. Artificial neural network-based nonlinearity estimation of Pressure Sensors. 1994.
13. Y. Yao and J. Gehrke. Query Processing in Sensor Networks. 2003.
14. T. Mitchell. Machine Learning. 1997

Methoden der Anomalieerkennung mit besonderem Hinblick auf Sensornetzwerke

David Vogelbacher*

Betreuer: Anja Bachmann[†]

Karlsruher Institut für Technologie (KIT)
Pervasive Computing Systems – TECO

*uaghg@student.kit.edu

[†]bachmann@teco.edu

Zusammenfassung. Anomalieerkennung spielt in vielen verschiedenen Bereichen eine bedeutende Rolle. Dementsprechend wurden schon viele Methoden dafür entwickelt und untersucht. In diesem Übersichtspaper wird ein knapper Überblick über verschiedene Methoden gegeben. Dabei wird im Speziellen auch darauf eingegangen, welche Methoden sich für Sensornetzwerke eignen. Sensornetzwerke sind in der heutigen ubiquitären Welt immer wichtiger und in vielen Anwendungsbereichen von Sensornetzwerken spielt auch die Anomalieerkennung eine große Rolle. Deshalb werden Methoden benötigt, die die besonderen Eigenschaften von Sensornetzwerken berücksichtigen.

Schlüsselwörter: Anomalieerkennung, maschinelles Lernen, Sensornetzwerke

1 Einleitung

Anomalieerkennung ist in vielen Anwendungsgebieten sehr wichtig und wurde daher auch schon intensiv erforscht. Anomalien sind unerwartete Ereignisse, wie zum Beispiel ein hoher Messwert in einer Serie von sonst niedrigen Messwerten. Andere auftretende Bezeichnungen für Anomalien sind „Ausreißer, Ausnahmen, Abweichungen, Überraschungen, Besonderheiten oder Kontaminationen“ (frei übersetzt aus [4]). Solche Anomalien zu erkennen ist oft von großer Bedeutung, da in vielen Anwendungsgebieten Anomalien in den Daten mit wichtigen, oft kritischen Ereignissen korrespondieren, auf die reagiert werden muss [4]. Zum Beispiel kann eine Anomalie abnormale Betriebsbedingungen anzeigen, was ohne Gegenmaßnahmen zu starken Performance-Einbrüchen führen kann, wie beispielsweise ein Flussproblem in einer Pipeline [10]. Daher wurden auch schon viele Methoden der Anomalieerkennung für die verschiedensten Gebiete entwickelt [4]. So kann Anomalieerkennung eingesetzt werden, um Kreditkartenbetrüger zu entlarven, um Einbrecher in IT-Systeme zu entdecken, um fehlerhafte Sensoren zu entdecken oder auch zur militärischen Überwachung feindlicher Aktivitäten [4]. Weitere mögliche Anwendungsgebiete sind auch das Auffinden

von Neuheiten in Bildern, das Überwachen der medizinischen Werte von Patienten, das Auffinden von unerwarteten Einträgen in Datenbanken oder auch das Überwachen von Netzwerk-Performance (beispielsweise zum Auffinden von Engstellen) [10]. Entstehen können Anomalien unter anderem durch „menschliche Fehler, Instrumentenfehler, natürliche Abweichungen in Populationen, betrügerisches Verhalten, Änderungen im Verhalten von Systemen oder Fehler in Systemen“ (frei übersetzt aus [10]).

Bei manchen Anwendungsgebieten reicht es aus zur Anomalieerkennung einfache Schwellwerte vorzugeben. Beispielsweise könnte man sagen, es wäre eine Anomalie, wenn im Winter die Außentemperatur in Deutschland bei 30°C liegt. In vielen Anwendungsgebieten jedoch ist die Erkennung schwieriger. *Chandola et al* [4] erläutern in ihrem Übersichtspaper einige Faktoren, welche die Anomalieerkennung sehr komplex machen können. Dazu zählt unter anderem das Auftreten von Rauschen, welches dem Aussehen von Anomalien ähneln kann und schwer davon zu unterscheiden ist [4]. Weiterhin zählt dazu die Schwierigkeit, eine Schwelle zwischen anomalem Verhalten und normalem Verhalten zu finden und zu definieren, da die Grenze oft nicht präzise ist, sowie die Tatsache, dass Anomalien, die aus böswilligen Aktionen resultieren, oft so getarnt werden, dass sie fast wie normales Verhalten aussehen [4].

In der heutigen Welt gibt es immer mehr Sensoren und Sensornetzwerke. Deshalb ist die Anomalieerkennung in Sensornetzwerken ein bedeutendes Thema. Ein Beispiel wäre ein Sensornetzwerk bei einem Staudamm, das alle möglichen Werte wie Druck des Wassers auf die Staumauer, Wassertemperatur etc. misst. Hier ist Anomalieerkennung zum einen wichtig, um fehlerhafte Sensoren zu erkennen, die falsche Werte liefern, damit diese Sensoren schnell ersetzt werden können. Zum anderen ist hier Anomalieerkennung wichtig, um Anomalien in den Messdaten zu erkennen wie beispielsweise ein erhöhter Druck auf den Staudamm, damit rechtzeitig Maßnahmen ergriffen werden können um ein Brechen des Staudammes zu verhindern. Sensoren sind oft eingeschränkt bei Energie, Speicher und Rechenleistung [23, 26]. Weiterhin erläutern *Branch et al* [2], dass drahtlose Sensornetzwerke oft sehr anfällig für Anomalien sind. Gerade hier ist es daher oft wichtig, dass die Methoden zur Anomalieerkennung wenig Kommunikation und Rechenleistung benötigen, aber trotzdem gute Ergebnisse liefern [23]. In diesem Paper wird ein strukturierter Überblick über Methoden der Anomalieerkennung bereitgestellt. Im Speziellen wird darauf eingegangen, in welchem Maße die verschiedenen Techniken sich für Sensornetzwerke eignen.

2 Grundlagen und Übersicht

Definitionen von Anomalien gibt es viele verschiedene, eine mögliche Definition ist die von *Barnett und Lewis* [1], die auch von *Hodge et al* [10] zitiert wurde:

„Eine Beobachtung (oder eine Teilmenge von Beobachtungen), die inkonsistent mit dem Rest des Datensatzes ist.“ (frei übersetzt)

Es gibt viele verschiedene Techniken, die auf ihre Eignung zur Anomalieerkennung untersucht wurden. Welche Methoden geeignet sind hängt stark vom jeweiligen Anwendungsfall ab. Wichtig ist auch welche Daten schon vorher vorhanden sind. So gibt es Methoden, die gelabelte Datensätze zum Training benötigen. Gelabelte Daten sind Daten, deren Klassen und Klassenzugehörigkeiten bekannt sind. Im Kontext der Anomalieerkennung bedeutet dies, dass zu jeder Dateninstanz bekannt ist, ob sie einer Anomalie entspricht oder normalem Verhalten. Methoden, die gelabelte Datensätze zum Training verwenden, werden auch *Überwachte Methoden* genannt [10]. Methoden, die Datensätze zum Training benötigen, bei denen nur Instanzen normalen Verhaltens (oder auch nur Instanzen anomalen Verhaltens) gelabelt sein müssen, werden *Semi-Überwachte Methoden* genannt [10]. Methoden, die keine Datensätze zum Training benötigen, werden als *Nicht-Überwachte Methoden* bezeichnet [10]. Nicht-Überwachte Methoden können daher unabhängig davon, ob schon Datensätze vorhanden sind, eingesetzt werden. Überwachte und Semi-Überwachte Methoden können bessere Ergebnisse liefern, allerdings ist es nicht immer einfach Datensätze zum Training zu erhalten und das Labeln der Daten ist oft sehr teuer und aufwändig. Weiterhin gibt es Methoden, die als Ergebnis nur ja/nein-Entscheidungen darüber liefern, ob eine Dateninstanz eine Anomalie ist, und es gibt auch Methoden, die als Ergebnis für jede Dateninstanz einen Anomaliegrad liefern, der aussagt, wie wahrscheinlich es ist, dass diese Dateninstanz eine Anomalie ist [4]. Methoden, welche einen Anomaliegrad liefern, können mittels eines Schwellwerts, ab dem die Dateninstanzen als Anomalien deklariert werden, leicht in Methoden umgewandelt werden, die nur eine ja/nein-Entscheidung treffen [4].

Chandola et al [4] unterteilen die verschiedenen Techniken der Anomalieerkennung in die folgenden Kategorien:

1. Klassifikation
2. Nearest-Neighbor
3. Clustering
4. Statistik
5. Informationstheoretische Anomalieerkennung

Nachfolgend werden verschiedene Arten der Anomalieerkennung entsprechend ihrer Kategorie (basierend auf [4]) vorgestellt.

2.1 Klassifikationsbasierte Anomalieerkennung

Klassifikation ist eine Grundtechnik des maschinellen Lernens. Die grundsätzliche Vorgehensweise ist folgende [4] : Zuerst wird mithilfe von gelabelten Trainingsdatensätzen, deren Klassen also bekannt sind, ein Klassifikator trainiert, der die Klassen von Dateninstanzen möglichst gut bestimmen können soll. Mithilfe dieses Klassifikators können dann neue Daten klassifiziert werden. Im Kontext der Anomalieerkennung bedeutet dies, dass Trainingsdatensätze benötigt werden, deren Klassen (Anomalie oder normal) bekannt sind. Mit dem damit

trainierten Klassifikator kann dann für neue Dateninstanzen bestimmt werden, ob es sich um Anomalien oder um normales Verhalten handelt. Im Folgenden werden verschiedene Methoden vorgestellt werden, die benutzt werden können um Klassifikatoren zu trainieren.

Neuronale Netze Neuronale Netze bestehen aus künstlichen Neuronen und Verbindungen zwischen diesen Neuronen. Diese Verbindungen haben verschiedene Gewichte. Es gibt Eingabeneuronen, an die Dateninstanzen angelegt werden können und Ausgabeneuronen, an denen das Ergebnis der Klassifikation abgelesen werden kann. Trainiert werden neuronale Netze, indem man die Trainingsdatensätze an den Eingabeneuronen anlegt und dann die Gewichte entsprechend des Fehlers an den Ausgabeneuronen ändert. Es gibt hier viele verschiedene Techniken, um eine gute Änderung der Gewichte zu bestimmen. Nach [10] können Neuronale Netze sich gut auf verborgene Muster einstellen und sind in der Lage komplexe Grenzen zwischen verschiedenen Klassen zu erlernen. Die benötigte Zeit zum Training ist sehr groß, da alle Daten mehrmals durchlaufen werden müssen, damit das Netz sich stabilisiert und die Daten korrekt modelliert werden [10]. Neuronale Netze werden beispielsweise von *Ryan et al.* [25] zur Einbruchserkennung („Intrusion Detection“) genutzt.

Support-Vector-Maschinen Der Ansatz von Support-Vector-Maschinen ist es den Datensatz in einen höherdimensionalen Raum zu projizieren, in welchem die verschiedenen Klassen durch eine Hyperebene getrennt werden können [9]. Der grundlegende Ansatz zur Anomalieerkennung ist eine Support-Vector-Maschine anhand von gegebenen Datensätzen zu trainieren [4]. Um dann eine neue Dateninstanz zu klassifizieren schaut man, auf welcher Seite der Hyperebene sie liegt [4]. *Xu et al.* [32] benutzen Support-Vector-Maschinen zur Anomalieerkennung in drahtlosen Sensornetzwerken.

Regelbasierte Erkennung In der Regelbasierten Anomalieerkennung benutzt der Klassifikator Regeln, um die Klasse einer Dateninstanz zu bestimmen [22]. Diese Regeln können beispielsweise anhand von schon gegebenen Daten erkannt werden oder auch von Expertenwissen oder anderen Quellen herrühren. Wenn eine Regel, die anomales Verhalten definiert, auf die Dateninstanz zutrifft, wird sie als Anomalie deklariert [22]. Regelbasierte Anomalieerkennung wird beispielsweise von *Wong et al.* [31] verwendet um frühzeitig Ausbrüche von Krankheiten anhand einer Datenbank, welche in Notaufnahmen registrierte Krankheitsfälle beschreibt, zu erkennen.

2.2 Nearest-Neighbor-basierte Anomalieerkennung

Nearest-Neighbor-Techniken gehen von der Annahme aus, dass normale Daten in Regionen mit hoher Dichte vorkommen, während Anomalien in Regionen mit geringer Dichte vorkommen [4]. Um Nearest-Neighbor-Techniken anwenden zu

können, muss es also möglich sein die Distanz zwischen verschiedenen Dateninstanzen zu berechnen [4]. Wenn die Dateninstanzen als Vektoren des \mathbb{R}^n aufgefasst werden können, kann hier beispielsweise die euklidische Distanz verwendet werden. Aber auch andere Metriken sind denkbar.

Chandola et al [4] beschreiben die Basistechnik folgendermaßen: Der Anomaliegrad einer Dateninstanz wird festgesetzt auf die Distanz der Dateninstanz zu ihrem k -ten Nachbarn. Mittels eines Schwellwerts können dann Anomalien von normalen Dateninstanzen unterschieden werden. Die Komplexität dieser Basistechnik liegt in $O(n^2)$, was bei großen Datensätzen nicht sehr effizient ist. Daher wurden Varianten vorgeschlagen um die Effizienz zu verbessern [4]. So gibt es Varianten, die den Suchraum verringern, indem sie bestimmte Dateninstanzen ignorieren, die keinesfalls Anomalien sein können [4]. Beispielsweise kann der Merkmalsraum aufgeteilt werden in ein Grid von Hyperwürfeln [4]. Die Dateninstanzen, die in Hyperwürfeln liegen, in denen auch viele andere Dateninstanzen liegen, können ignoriert werden, da sie höchstwahrscheinlich keine Anomalien sind [4]. Weitere vorgeschlagene Varianten sind nach *Chandola et al* [4] Methoden, in denen der Anomaliegrad anders berechnet wird. Beispielsweise benutzen *Zhang und Wang* [34] die Summe der Distanzen der k nächsten Nachbarn um den Anomaliegrad zu berechnen. Eine andere Möglichkeit den Anomaliegrad zu berechnen ist, die Dichte in der Region einer Dateninstanz abzuschätzen als die Anzahl von Nachbarn die eine kleinere Distanz haben als ein bestimmter Wert d [4, 13, 14, 15, 16]. Als Anomaliegrad kann dann das Inverse dieser Dichte verwendet werden. Es gibt auch Varianten, die andere Distanzmetriken verwenden [4]. Beispielsweise verwenden *Cho et al.* [7] die Mahalanobis-Distanz zur Anomalieerkennung beim Tippverhalten („Keystroke dynamics“) [11]. Nearest-Neighbor-basierte Anomalieerkennung wird auch eingesetzt um Landminen mittels Satellitenbildern zu entdecken [3, 4]. Weiterhin benutzen *Liao et al.* [19] einen k -Nearest-Neighbor-Klassifikator zur Einbruchserkennung („Intrusion Detection“).

2.3 Clusterbasierte Anomalieerkennung

Clustering befasst sich mit dem Entdecken von Clustern in den Daten. Ein Cluster ist dabei eine Gruppe von Daten, die zusammengehören. Unter der Annahme, dass „Normale Dateninstanzen zu Clustern gehören, während Anomalien nicht zu Clustern gehören“ (frei übersetzt aus [4]), können bekannte Clustering-Algorithmen auf die Daten angewandt werden und diejenigen Dateninstanzen als Anomalien deklariert werden, die nicht zu einem Cluster gehören. Dabei müssen Cluster-Algorithmen benutzt werden, die nicht erzwingen, dass jede Instanz zu einem Cluster gehören muss [4]. Eine weitere Annahme, die verwendet werden kann ist die Folgende: „Normale Dateninstanzen liegen nahe am Clusterzentrum, während anomale Dateninstanzen weit entfernt vom Clusterzentrum liegen“ (frei übersetzt aus [4]). Anomalien lassen sich anschließend wie folgt finden [4]: Zuerst wird ein Clustering-Algorithmus benutzt, um die Daten zu clustern. Dann wird für jede Dateninstanz die Distanz zu dem Zentrum ihres Clusters berechnet.

Diese Distanz kann dann als Anomaliegrad benutzt werden. Eine mögliche Distanzmetrik ist hierbei die euklidische Distanz, aber auch andere Metriken sind denkbar.

Wenn die Anomalien selber Cluster bilden sind die obigen beiden Ansätze ungeeignet [4]. Beim ersten Ansatz werden die Anomalien nicht gefunden, weil sie selber Cluster bilden [4]. Beim zweiten Ansatz werden die Anomalien oft nicht gefunden, weil die anomalen Werte dann eine geringe Distanz zum Clusterzentrum haben, da die Cluster anomaler Werte eher klein sind (da Anomalien nicht so häufig auftreten wie normales Verhalten) [4]. Für Daten, in denen die Anomalien auch Cluster bilden, ist deshalb unter Umständen die folgende Annahme besser geeignet: „Normale Dateninstanzen gehören zu großen Cluster, während anomale Instanzen zu kleinen Clustern gehören“ (frei übersetzt aus [4]). Unter dieser Annahme können dann Anomalien gefunden werden, indem zuerst einen Clustering-Algorithmus benutzt wird, um die Cluster der Daten zu bestimmen. Darauf werden diejenigen Dateninstanzen als Anomalien deklariert, die zu kleinen Clustern gehören [4].

Beim Verwenden von Clusterbasierten Techniken ist es also sehr wichtig geeignete Annahmen über die Daten zu treffen und dann dementsprechend den geeigneten Ansatz zu wählen.

2.4 Statistische Anomalieerkennung

Die Annahme bei statistischen Techniken ist, dass „Normale Dateninstanzen in Regionen hoher Wahrscheinlichkeit eines statistischen Modells anzutreffen sind, während Anomalien in Regionen niedriger Wahrscheinlichkeit dieses Modells anzutreffen sind“ (frei übersetzt aus [4]). Bei statistischen Techniken wird zuerst ein statistisches Modell basierend auf schon gegeben Daten erstellt. Neue Dateninstanzen können dann klassifiziert werden (als Anomalie oder normales Verhalten), je nachdem ob sie mit hoher Wahrscheinlichkeit von dem Modell generiert wurden oder nicht [4]. Zu unterscheiden sind hier parametrisierte und nicht-parametrisierte Methoden [10]. Parametrisierte Techniken gehen von einer vorgegebenen Wahrscheinlichkeitsverteilung aus und schätzen dann mithilfe der vorgegebenen Daten nur noch die Parameter dieser Verteilung [4]. Nicht-parametrisierte Techniken hingegen treffen keine Annahmen über eine zugrunde liegende Wahrscheinlichkeitsverteilung [4]. Im Folgenden werden parametrisierte Techniken und nicht-parametrisierte Techniken genauer vorgestellt:

Parametrisierte Techniken

Gaußverteilung und andere Wahrscheinlichkeitsverteilungen Eine beliebte statistische Modellierung ist die Gaußverteilung. Hierbei wird angenommen, dass die Verteilung der Daten näherungsweise einer Gaußverteilung entspricht. Die Parameter der Verteilung können dann mittels Maximum-Likelihood-Methoden geschätzt werden [4]. Als Anomaliegrad einer Dateninstanz x kann das Inverse $(f(x))^{-1}$ der Dichtefunktion f benutzt werden [4]. Alternativ können auch statistische Hypothesentest genutzt werden [4]. Das gleiche Verfahren kann auch bei

anderen Wahrscheinlichkeitsverteilungen angewandt werden. Auch ein Mischen von verschiedenen Wahrscheinlichkeitsverteilungen kann sinnvoll sein [4]. Diese Methode ist sehr vielversprechend, wenn die zugrunde liegenden Daten auch wirklich der Verteilung entsprechen. *Ye et al.* [33] verwenden beispielsweise Chi-Quadrat-Statistiken um Einbrüche in IT-Systeme zu entdecken.

Regressionsanalyse Bei der Regressionsanalyse wird versucht die Abhängigkeiten zwischen verschiedenen Eingabevariablen durch ein Modell zu beschreiben [5]. Dieses Modell soll beschreiben, welche Kombinationen von Eingabevariablen gültig sind und welche nicht [5]. Das Modell kann eine Funktion sein, welche die verschiedenen Eingabevariablen in Beziehung setzt. Ein Modell wäre beispielsweise, dass $f(x_1, \dots, x_n) = 0$ gilt für alle gültigen Kombinationen von Eingabevariablen x_1, \dots, x_n und eine geeignete Funktion f . Zu gegebenen Eingabevariablen wird dann die Differenz zwischen dem tatsächlichen Wert und dem durch das Modell generierten Wert als Residuum bezeichnet. Beim obigen Modell wäre das Residuum also gerade $f(x_1, \dots, x_n)$. Der grundlegende Ansatz der in [4] beschrieben wird, ist, dass zuerst ein Regressionmodell anhand gegebener Daten bestimmt wird. In einem zweiten Schritt kann dann das Residuum einer Dateninstanz als Anomaliegrad benutzt werden.

Nicht-parametrisierte Techniken

Histogrammbasiert Um ein Histogramm zu erhalten, muss der Merkmalsraum der Daten in verschiedene Klassen eingeteilt werden [4]. Zum Beispiel könnte man bei einer Temperaturmessung sagen, dass 0°C bis 10°C eine Klasse ist, 10°C bis 20°C eine weitere, usw.. Ein Histogramm erhält man dann mittels eines Datensatzes, indem pro Klasse gezählt wird wie viele Dateninstanzen darin liegen [4]. Der Basisansatz zu Anomalieerkennung wird in *Chandola et al* [4] folgendermaßen beschrieben: Zuerst wird ein Histogramm gebildet, aufbauend auf den Werten eines gegebenen Datensatzes. Um dann zu testen, ob eine neue Dateninstanz eine Anomalie ist, wird gezählt, wie viele andere Daten in der gleichen Klasse des Histogramms sind. Wenn diese Anzahl niedriger ist als ein gewisser Schwellwert, dann wird die Dateninstanz als Anomalie deklariert. Die Größe der Klassen des Histogramms ist hier sehr wichtig [4]. Bei zu kleinen Klassen, werden auch häufig normale Werte als Anomalien erkannt [4]. Bei zu großen Klassen hingegen, können viele Anomalien übersehen werden [4]. Histogrammbasierte Anomalieerkennung wird unter anderem von *Kind et al.* [12] benutzt um Anomalien im Datenverkehr von Netzwerken zu erkennen.

Kern-Dichteschätzer Die Kern-Dichteschätzung wird verwendet um eine unbekannte Wahrscheinlichkeitsverteilung anhand von Stichproben zu schätzen [21]. Im Gegensatz zu Histogrammen, welche eine diskrete Schätzung liefern, kann die Kern-Dichteschätzung eine stetige Schätzung erzeugen [21]. Die Anomalieerkennung kann hier analog zur Anomalieerkennung bei den Parametrisierten Techniken, die eine bekannte Wahrscheinlichkeitsverteilung zugrunde legen,

durchgeführt werden. Nach *Laxhammar et al.* [18] werden Kern-Dichteschätzer beispielsweise zur Anomalieerkennung im Seeverkehr benutzt.

2.5 Informationstheoretische Anomalieerkennung

Die Informationstheorie beschäftigt sich unter anderem mit dem Informationsgehalt von Daten. Unter der Annahme, dass „durch Anomalien in den Daten eine Irregularität im Informationsgehalt des Datensatzes hervorgerufen wird“ (frei übersetzt aus [4]), kann Informationstheorie zur Anomalieerkennung benutzt werden. *Chandola et al.* [4] beschreiben den grundlegenden Ansatz folgendermaßen: Finde die minimale Teilmenge S eines gegebenen Datensatzes, so dass die Änderung der Komplexität des Datensatzes durch Entfernen von S maximal ist. Die Dateninstanzen, die zu S gehören, werden dann als Anomalien deklariert. Eine mögliche Metrik zum Messen der Komplexität der Daten ist die Kolmogorow-Komplexität [4]. Die Kolmogorow-Komplexität eines Datums ist dabei die Länge des kürzesten Programms, welches dieses Datum erzeugt [30]. Unter anderem wird Informationstheorie, im Speziellen die Kolmogorow-Komplexität, von *Kulkarni et al.* [17] benutzt um Denial-of-Service-Attacks zu entdecken.

3 Sensornetzwerke

Anomalieerkennung in Sensornetzwerken ist wichtig zur Fehlererkennung (fehlerhafte Sensoren, fehlerhafte Übertragung), Angriffserkennung und zur Erkennung von Anomalien in den gemessenen Daten [22]. Welche Methoden für die Anomalieerkennung anwendbar sind hängt in großem Maße von der Art des Sensornetzwerkes ab. So können die Sensoren beispielsweise verschiedene Messgrößen erfassen oder sie messen alle die gleiche Größe, aber die Werte sind unterschiedlich verteilt. Es ist auch möglich, dass mehrere Sensoren die gleiche Größe mit der gleichen Verteilung messen. Wenn die Messgrößen der Sensoren unterschiedlich und nicht korreliert sind, dann muss für jede Messgröße die Anomalieerkennung separat ausgeführt werden, da die Daten untereinander nicht in Bezug stehen. Wenn die Messgrößen der Sensoren unterschiedlich, aber trotzdem in irgendeiner Form korreliert sind, dann könnte man versuchen dies bei der Anomalieerkennung auszunutzen. Während meiner Recherche habe ich dazu allerdings keine Literatur gefunden. Wenn die Messgrößen gleich sind, aber mit verschiedener Verteilung, kann versucht werden die Daten so zu skalieren, dass alle Verteilungen gleich sind. Die Anomalieerkennung kann man dann auf den skalierten Daten durchführen oder man betrachtet die einzelnen verschiedenen Verteilungen wieder separat.

Der am einfachsten zu behandelnde Fall ist, dass die verschiedenen Messgrößen und Verteilungen übereinstimmen. Dies scheint auch der einzige, bisher intensiv in der Literatur erforschte, Fall zu sein. Man könnte natürlich auch hier wieder an jedem Sensor separat eine Anomalieerkennung durchführen. Dabei würde

allerdings viel Potential verschenkt werden, da bei weniger Daten auch die Anomalieerkennung ungenauer ist und fehlerhafte Sensoren (die einen verzerrten Wert messen) unter Umständen gar nicht bemerkt werden würden. Also liegt es nahe die Anomalieerkennung auf den gesamten Daten aller Sensoren auszuführen. Der naive Ansatz ist hier alle Sensordaten an einen zentralen Knoten zu schicken und an diesem Knoten dann die Anomalieerkennung mit gewöhnlichen Mitteln, wie sie im vorherigen Kapitel vorgestellt wurden, auszuführen [22]. Dies erfordert allerdings einen hohen Kommunikationsaufwand und je nach Bandbreite auch hohen Zeitaufwand.

Es ist in Sensornetzwerken auch eine Herausforderung, dass die Algorithmen oft online arbeiten sollten [4]. Daher müssen die Algorithmen effizient sein, besonders da Sensoren oft eingeschränkt sind bei Energie, Speicher und Rechenleistung [23, 26]. Ein vielversprechender Ansatz ist es gewöhnliche Methoden der Anomalieerkennung, wie sie im vorherigen Kapitel vorgestellt wurden, für Sensornetzwerke zu optimieren, indem man die Algorithmen, die normalerweise alle Daten zentral verarbeiten, ändert, so dass sie verteilt arbeiten [22]. Gerade in drahtlosen Sensornetzwerken ist es auch wichtig die Kommunikation zwischen den Sensoren gering zu halten, da sie viel Energie benötigt und die Lebensdauer des Netzwerkes verringern kann [23]. Nach [22, 35] liegt beispielsweise bei Sensoria-Sensoren und Berkeley-Motes das Verhältnis zwischen dem Energieverbrauch bei der Kommunikation und dem Verbrauch beim Rechnen zwischen 10^3 und 10^4 . Gerade hier ist es also sehr wichtig den Kommunikationsaufwand gering zu halten. In diesem Kapitel werden spezielle Techniken zur Anomalieerkennung in Sensornetzwerken vorgestellt.

3.1 Klassifikationsbasierte Anomalieerkennung

Support-Vector-Maschinen In [24] wird ein verteilter Algorithmus vorgeschlagen, der eine Hypersphäre findet, die die Mehrheit der Datenvektoren im Merkmalsraum an jedem Sensor erfasst [22]. Die Minderheit der Daten, die nicht in der Sphäre enthalten sind, werden zu Anomalien erklärt [22]. *Rajasegarar et al* [22] fassen die Methode folgendermaßen zusammen: Die Sensoren kommunizieren nur um den Radius der Sphäre festzulegen. Anhand dieses Radius' führt dann jeder Sensor eine Anomalieerkennung durch. Dadurch entsteht nur ein geringer Kommunikationsaufwand. Die Komplexität dieser Methode ist allerdings im Allgemeinen eher hoch [22].

3.2 Regelbasierte Ansätze

Regelbasierte Ansätze lassen sich in Sensornetzwerken gut einsetzen, vorausgesetzt, dass geeignete Regeln erstellt werden können, da sie normalerweise keinen großen Kommunikationsaufwand benötigen. Weiterhin ist auch die Komplexität von Regelbasierten Techniken im Allgemeinen sehr niedrig [22]. In [27] wird ein Regelbasiertes System zur Angriffserkennung in Wireless-Sensornetzwerken vorgestellt, das Monitorknoten benutzt auf denen das Angriffserkennung-System

implementiert ist [22]. Dieses System wird nachfolgend beispielhaft genauer vorgestellt:

Ein Monitorknoten ist ein spezieller Knoten, welcher folgende Informationen sammelt [22]: Er sammelt alle Nachrichten, auch wenn sie nicht an ihn adressiert sind (promiscuous mode). Weiterhin entdeckt er Nachrichten-Kollisionen, wenn er eine Nachricht versendet. Es werden die folgenden Regeln verwendet [27]:

1. Integritätsregel: Die Größe des Datenteils einer Nachricht sollte sich während der Übertragung zwischen den einzelnen Knoten nicht ändern. Auf diese Weise können Attacks, bei denen die Daten verändert werden (und damit im Normalfall auch die Größe der Daten) erkannt werden.
2. Verspätungsregel: Das Weitersenden einer Nachricht durch einen Nachbarn muss innerhalb einer bestimmten Zeit passieren. Dadurch wird einem Angreifer nur wenig Zeit gelassen, um Manipulationen an der Nachricht durchzuführen.
3. Wiederholungsregel: Es gibt eine obere Grenze wie oft das gleiche Paket von einem Knoten versendet werden darf. Dadurch können Attacks erkannt werden, bei denen der Angreifer das Netz mit Paketen überflutet (Denial-of-Service).
4. Radio-Übertragungsbereich: Nachrichten, die an einem Monitorknoten ankommen, müssen von einem der Nachbarn des Monitorknoten versendet worden sein. Dadurch können Attacks erkannt werden, bei denen der Angreifer versucht aus großer Entfernung mittels eines starken Senders Nachrichten zu verschicken.
5. Jamming-Regel: Die Anzahl an Kollisionen einer von einem Monitorknoten versendeten Nachricht darf ein bestimmtes Limit nicht überschreiten. Hierdurch können Attacks erkannt werden, in denen ein Knoten Rauschen in den Kommunikationskanal einschleust, um die Kommunikation zu stören.

Der Algorithmus, um Angriffe zu erkennen, besteht nun aus den folgenden drei Phasen [22]: Zuerst werden durch die Monitorknoten Informationen gesammelt. Dann werden die Regeln angewandt, um Attacks zu entdecken. Wenn die Anzahl an entdeckten Attacks ein bestimmtes Limit überschreitet, wird ein Alarm ausgelöst. Dieses Limit sollte so gewählt sein, dass gelegentliche normale Fehler wie Datenverlust und Übertragungsfehler noch keinen Alarm auslösen.

3.3 Clusterbasierte Anomalieerkennung

Das Problem bei den gewöhnlichen Clustering-Techniken ist, dass sie zentral arbeiten. In [23] wird eine verteilte Clustering-Technik für hierarchische Sensornetzwerke vorgeschlagen: Zuerst werden die Messdaten an jedem Sensor geclustert. Daten über die Cluster werden dann an den in der Hierarchie übergeordneten Sensorknoten geschickt. Dieser vereint die Daten über die Cluster seiner Kindknoten mit seinen eigenen Clustern und ermittelt dann die Cluster dieser vereinigten Menge. Dann schickt er die Cluster-Daten wieder an seinen

Vaterknoten. Dies geht so lange, bis der höchste Sensorknoten in der Hierarchie erreicht ist. Der Kommunikationsaufwand wird hier erheblich verringert gegenüber einem zentralen Clustering-Ansatz [23]. Bei Daten von dem *Great Duck Island project* [29] zeigt er aber trotzdem eine vergleichbare Genauigkeit [23]. Die Komplexität dieser Methode ist im Allgemeinen allerdings eher hoch [22].

3.4 Dichtebasierte Anomalieerkennung

In [22, 28] wird folgende Methode vorgestellt, um in einem hierarchischen Sensornetzwerk die Dichte der Daten zu schätzen: Jeder Sensor schätzt die eigene lokale Dichte in einem gleitenden Fenster mittels Kern-Dichte-Schätzern. Die globale Dichte kann dann geschätzt werden, indem die Kern-Funktionen von allen Datenpunkten aufsummiert werden.

Rajasegarar et al. [22] beschreiben weiterhin folgende Methode, um normale Dichtebasierte Methoden der Anomalieerkennung an Sensornetzwerke anzupassen: An jedem Sensorknoten werden lokale Anomalien basierend auf der lokalen Dichte ermittelt. Diese lokalen Anomalien werden dann an Vaterknoten geschickt, zusammen mit den lokalen Dichte-Parametern. Diese Parameter beinhalten ausgewählte lokale Datenvektoren und einen Kern-Parameter. Die Vaterknoten entdecken dann, ausgehend von den lokalen Anomalien globale Anomalien, indem sie die lokalen Dichte-Parameter kombinieren. Wenn nur wenige Anomalien auftreten, wird der Kommunikationsaufwand hier gering gehalten. Wenn allerdings viele Anomalien auftreten, ist diese Methode nicht sehr effizient.

3.5 Statistische Anomalieerkennung

Parametrisierte Techniken Wenn die Verteilung der Daten gut bekannt ist, dann können parametrisierte statistische Techniken gut geeignet sein zur Anomalieerkennung [22]. Beispielsweise benutzen *Ngai et al.* [20] eine Technik basierend auf einem Chi-Quadrat-Test um Sinkhole-Attacken in einem drahtlosen Sensornetzwerk festzustellen [22]. Sinkhole-Attacken sind hierbei „Attacken, in denen der Eindringling ein hochenergetisches Gerät benutzt um sich als der Sensorknoten auszugeben, der den kürzesten Pfad zur Basisstation hat. Die Sensorknoten, die diese Nachricht bekommen, lenken dann ihren Traffic über den Eindringling“ (frei übersetzt aus [22]).

3.6 Generischer Ansatz von *Branch et al.* [2]

Branch et al. [2] haben einen generischen Ansatz zur Anomalieerkennung entworfen, der flexibel ist in Bezug auf die zugrunde liegende Technik zur Anomalieerkennung (Techniken wie sie im vorherigen Kapitel vorgestellt wurden). Außerdem benötigt er wenig Bandbreite und hat einen geringen Energieverbrauch, da die Anomalieerkennung dezentral im Netzwerk stattfindet. Dieser Ansatz wird im Folgenden genauer vorgestellt werden [2]:

Sei hierzu \mathbb{D} der Merkmalsraum der Daten und $R(x, D)$ eine Funktion, die einer Dateninstanz x einen Anomaliegrad innerhalb dem Datensatz D zuordnet. R muss dabei die zwei folgenden Bedingungen bei gegebener Dateninstanz x erfüllen:

1. „Wenn $Q_1 \subseteq Q_2 \subseteq \mathbb{D}$ gilt, dann muss $R(x, Q_1) \geq R(x, Q_2)$ sein (Anti-Monotonie)“ (frei übersetzt).
2. „Wenn $R(x, Q_1) > R(x, Q_2)$ gilt, dann existiert ein $z \in Q_2 \setminus Q_1$, so dass $R(x, Q_1) > R(x, Q_1 \cup \{z\})$ (Glätte)“ (frei übersetzt).

Beispielsweise erfüllt die k-Nearest-Neighbor-Metrik (als Anomaliegrad von x wird die Distanz des k-ten Nachbarn von x genommen) diese Bedingungen. Sei nun weiterhin mit $O_n(D)$ die Menge der Dateninstanzen des Datensatzes D mit den n höchsten Anomaliegraden bezeichnet. Eine genaue Beschreibung des Algorithmus' wäre hier zu aufwändig. Daher folgt hier nur eine grobe Beschreibung: Jeder Sensor p_i wartet asynchron auf ein Ereignis (ein neuer Sensorwert wird gemessen oder eine Nachricht eines Nachbarn trifft ein). Beim Auftreten eines solchen Ereignisses berechnet p_i die eigene Schätzung $O_n(P_i)$ (P_i sind die Daten, die der Sensor im Moment hält. Dies sind eigene Daten oder Daten anderer Sensoren, die ihm geschickt wurden) neu und sendet an jeden Nachbarknoten p_j diejenigen Punkte P_i und sonstige Informationen, die die Schätzung $O_n(p_j)$ verändern könnten. *Branch et al.* haben bewiesen, dass, wenn die Daten an den Sensorknoten sich für eine ausreichende Zeit nicht ändern, der generische Algorithmus terminiert und $O_n(P_i) = O_n(D)$ für jeden Sensorknoten gilt, wobei D die Gesamtheit aller an den Sensoren erfassten Daten ist (also die Datenmenge, die ein simpler zentraler Algorithmus zur Anomalieerkennung benutzen würde). *Branch et al.* haben weiterhin mittels des SENSE Simulators für drahtlose Netzwerke [6] die Performance ihres Algorithmus' mit der k-Nearest-Neighbor-Metrik evaluiert hinsichtlich Genauigkeit und Energieverbrauch und kamen zu guten Ergebnissen. So lieferte der Algorithmus in 99% der Fälle das korrekte Ergebnis und *Branch et al.* vermuten, dass der Fehler von 1% durch verlorengegangene Nachrichten entsteht, da der Algorithmus an sich als korrekt bewiesen wurde. Der Platzbedarf des Algorithmus' wird in [2] leider nicht genauer untersucht. Wenn viele Anomalien auftreten, dann könnte dies ein Problem werden, da alle Sensoren speichern müssen, welche Informationen sie versendet und empfangen haben.

3.7 Anomalieerkennung von Positionsdaten in Wireless-Sensornetzwerken

In Wireless-Sensornetzwerken ist es oft wichtig, dass jeder Sensor seine Position kennt [8]. Zum Beispiel, damit im Falle von Anomalien oder anderen wichtigen Messungen bekannt ist, an welcher Position sie auftraten [8]. „Beispielsweise kann ein Rettungsteam seine Aufgabe nur dann Durchführen, wenn die Position des gefährlichen Vorganges (der durch Sensoren gemeldet wird), bekannt ist“ (frei übersetzt aus [8]). *Du et al* [8] beschreiben folgende Problemlösungen:

Man kann das Problem mittels GPS-Empfänger an jedem Sensorknoten lösen. Dies ist allerdings sehr teuer. Es ist daher besser nur wenige Sensoren mit GPS-Empfängern (sogenannte Beacon-Nodes) auszustatten und dann die Position der anderen Sensoren mittels geeigneter Algorithmen zu bestimmen [8]. Beispielsweise kann ein Sensor anhand der ihm übermittelten Positionsdaten der Beacon-Nodes und anderen Daten wie Ankunftszeit (Time of Arrival) oder Stärke der empfangenen Signale die eigene Position errechnen [8]. Wenn die Sensoren in einer freundlichen Umgebung sind, ist dies weiter kein Problem. Wenn allerdings Angreifer versuchen könnten die Positionsdaten zu verfälschen, indem sie einzelne Sensorknoten kompromittieren oder die Kommunikation stören [8], dann kann Anomalieerkennung helfen, dies zu entdecken. Deshalb schlagen *Du et al.* [8] ein Schema vor, um solche Angriffe zu erkennen, indem das Wissen über die Verteilung der Sensoren genutzt wird. Wenn zum Beispiel die Sensoren aus einem Flugzeug geworfen werden, dann kann man schon vorher die Abwurfpunkte in den Sensoren speichern, zusammen mit einer Wahrscheinlichkeitsverteilung, die den Auftreffpunkt auf der Erde des Sensors bei gegebenem Abwurfpunkt modelliert [8].

4 Fazit

In diesem Paper wurden verschiedene Methoden der Anomalieerkennung erläutert. Die vorgestellten Methoden wurden kategorisiert und es wurden auch Beispielanwendungen der jeweiligen Ansätze genannt. Danach wurden im Speziellen Methoden der Anomalieerkennung für Sensornetzwerke betrachtet. Sensornetzwerke haben viele Einschränkungen wie beispielsweise Rechenleistung und Energie, die bei der Anomalieerkennung beachtet werden müssen. Es wurden die Kategorien des vorangegangenen Kapitels aufgegriffen und Beispielansätze im Kontext der Anomalieerkennung in Sensornetzwerken genannt. Die Wahl einer geeigneten Technik hängt sehr stark vom Anwendungsfall ab und muss gut bedacht werden. Regelbasierte Anomalieerkennung eignet sich beispielsweise sehr gut für Sensornetzwerke, da der Aufwand dieser Methode sehr gering ist. Allerdings ist es meist schwierig geeignete Regeln zu finden, weshalb man andere Methoden benutzen muss wie beispielsweise Support-Vector-Maschinen, die mehr Rechenleistung benötigen. Zu Sensornetzwerken finden sich bisher in der Literatur hauptsächlich Verfahren, bei denen die Messgrößen der Sensoren alle die gleiche Verteilung besitzen müssen. Eine vielversprechende Richtung für zukünftige Forschung kann daher sein, Methoden zu entwickeln, die für Sensoren mit unterschiedlich verteilten Messgrößen (die aber trotzdem in irgendeiner Form korreliert sind) geeignet sind. Ein weiterer Ansatz für künftige Forschung könnte darin bestehen, zu untersuchen, inwiefern die Kombination von verschiedenen Anomalieerkennungstechniken in Sensornetzwerken nützlich sein kann.

Literaturverzeichnis

- [1] Barnett, V., Lewis, T.: Outliers in statistical data, vol. 3. Wiley New York (1994)
- [2] Branch, J., Giannella, C., Szymanski, B., Wolff, R., Kargupta, H.: In-network outlier detection in wireless sensor networks. *Knowledge and Information Systems* 34(1), 23–54 (2013), <http://dx.doi.org/10.1007/s10115-011-0474-5>
- [3] Byers, S., Raftery, A.E.: Nearest-neighbor clutter removal for estimating features in spatial point processes. *Journal of the American Statistical Association* 93(442), 577–584 (1998), <http://amstat.tandfonline.com/doi/abs/10.1080/01621459.1998.10473711>
- [4] Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. *ACM Comput. Surv.* 41(3), 15:1–15:58 (Jul 2009), <http://doi.acm.org/10.1145/1541880.1541882>
- [5] Chatterjee, S., Hadi, A.S.: Regression analysis by example. John Wiley & Sons (2013)
- [6] Chen, G., Branch, J., Pflug, M., Zhu, L., Szymanski, B.: Sense: a wireless sensor network simulator. In: *Advances in pervasive computing and networking*, pp. 249–267. Springer (2005)
- [7] Cho, S., Han, C., Han, D.H., Kim, H.I.: Web-based keystroke dynamics identity verification using neural network. *Journal of organizational computing and electronic commerce* 10(4), 295–307 (2000)
- [8] Du, W., Fang, L., Peng, N.: Lad: Localization anomaly detection for wireless sensor networks. *Journal of Parallel and Distributed Computing* 66(7), 874 – 886 (2006), <http://www.sciencedirect.com/science/article/pii/S0743731506000670>, <ce:title>Special Issue 19th International Parallel and Distributed Processing Symposium - {IPDPS} 2005</ce:title>
- [9] Hearst, M.A., Dumais, S., Osman, E., Platt, J., Scholkopf, B.: Support vector machines. *Intelligent Systems and their Applications, IEEE* 13(4), 18–28 (1998)
- [10] Hodge, V., Austin, J.: A survey of outlier detection methodologies. *Artificial Intelligence Review* 22(2), 85–126 (2004), <http://dx.doi.org/10.1007/s10462-004-4304-y>
- [11] Killourhy, K.S., Maxion, R.A.: Comparing anomaly-detection algorithms for keystroke dynamics. In: *Dependable Systems & Networks, 2009. DSN'09. IEEE/IFIP International Conference on*. pp. 125–134. IEEE (2009)
- [12] Kind, A., Stoecklin, M., Dimitropoulos, X.: Histogram-based traffic anomaly detection. *Network and Service Management, IEEE Transactions on* 6(2), 110–121 (2009)
- [13] Knorr, E.M., Ng, R.T.: A unified approach for mining outliers. In: *Proceedings of the 1997 Conference of the Centre for Advanced Studies on Collaborative Research*. pp. 11–. CASCON '97, IBM Press (1997), <http://dl.acm.org/citation.cfm?id=782010.782021>

- [14] Knorr, E.M., Ng, R.T.: Finding intensional knowledge of distance-based outliers. In: VLDB. vol. 99, pp. 211–222 (1999)
- [15] Knorr, E.M., Ng, R.T., Tucakov, V.: Distance-based outliers: algorithms and applications. *The VLDB Journal—The International Journal on Very Large Data Bases* 8(3-4), 237–253 (2000)
- [16] Knox, E.M., Ng, R.T.: Algorithms for mining distance-based outliers in large datasets. In: *Proceedings of the International Conference on Very Large Data Bases*. Citeseer (1998)
- [17] Kulkarni, A., Bush, S.: Detecting distributed denial-of-service attacks using kolmogorov complexity metrics. *Journal of Network and Systems Management* 14(1), 69–80 (2006), <http://dx.doi.org/10.1007/s10922-005-9016-3>
- [18] Laxhammar, R., Falkman, G., Sviestins, E.: Anomaly detection in sea traffic - a comparison of the gaussian mixture model and the kernel density estimator. In: *Information Fusion, 2009. FUSION '09. 12th International Conference on*. pp. 756–763 (2009)
- [19] Liao, Y., Vemuri, V.R.: Use of k-nearest neighbor classifier for intrusion detection. *Computers & Security* 21(5), 439–448 (2002)
- [20] Ngai, E.C., Liu, J., Lyu, M.R.: On the intruder detection for sinkhole attack in wireless sensor networks. In: *Communications, 2006. ICC'06. IEEE International Conference on*. vol. 8, pp. 3383–3389. IEEE (2006)
- [21] Parzen, E., et al.: On estimation of a probability density function and mode. *Annals of mathematical statistics* 33(3), 1065–1076 (1962)
- [22] Rajasegarar, S., Leckie, C., Palaniswami, M.: Anomaly detection in wireless sensor networks. *Wireless Communications, IEEE* 15(4), 34–40 (2008)
- [23] Rajasegarar, S., Leckie, C., Palaniswami, M., Bezdek, J.: Distributed anomaly detection in wireless sensor networks. In: *Communication systems, 2006. ICCS 2006. 10th IEEE Singapore International Conference on*. pp. 1–5 (2006)
- [24] Rajasegarar, S., Leckie, C., Palaniswami, M., Bezdek, J.: Quarter sphere based distributed anomaly detection in wireless sensor networks. In: *Communications, 2007. ICC '07. IEEE International Conference on*. pp. 3864–3869 (2007)
- [25] Ryan, J., Lin, M.J., Miikkulainen, R.: Intrusion detection with neural networks. In: *Advances in neural information processing systems*. pp. 943–949. MORGAN KAUFMANN PUBLISHERS (1998)
- [26] da Silva, A.P.R., Martins, M.H.T., Rocha, B.P.S., Loureiro, A.A.F., Ruiz, L.B., Wong, H.C.: Decentralized intrusion detection in wireless sensor networks. In: *Proceedings of the 1st ACM International Workshop on Quality of Service & Security in Wireless and Mobile Networks*. pp. 16–23. Q2SWinet '05, ACM, New York, NY, USA (2005), <http://doi.acm.org/10.1145/1089761.1089765>
- [27] da Silva, A.P.R., Martins, M.H., Rocha, B.P., Loureiro, A.A., Ruiz, L.B., Wong, H.C.: Decentralized intrusion detection in wireless sensor networks. In: *Proceedings of the 1st ACM international workshop on Quality of service & security in wireless and mobile networks*. pp. 16–23. ACM (2005)

- [28] Subramaniam, S., Palpanas, T., Papadopoulos, D., Kalogeraki, V., Gunopoulos, D.: Online outlier detection in sensor data using non-parametric models. In: Proceedings of the 32nd international conference on Very large data bases. pp. 187–198. VLDB Endowment (2006)
- [29] Szewczyk, R., Mainwaring, A., Polastre, J., Anderson, J., Culler, D.: An analysis of a large scale habitat monitoring application. In: Proceedings of the 2nd international conference on Embedded networked sensor systems. pp. 214–226. ACM (2004)
- [30] Wallace, C.S., Dowe, D.L.: Minimum message length and kolmogorov complexity. *The Computer Journal* 42(4), 270–283 (1999)
- [31] Wong, W.K., Moore, A., Cooper, G., Wagner, M.: Rule-based anomaly pattern detection for detecting disease outbreaks. In: AAI/IAAI. pp. 217–223 (2002)
- [32] Xu, S., Hu, C., Wang, L., Zhang, G.: Support vector machines based on k nearest neighbor algorithm for outlier detection in wsns. In: Wireless Communications, Networking and Mobile Computing (WiCOM), 2012 8th International Conference on. pp. 1–4. IEEE (2012)
- [33] Ye, N., Chen, Q.: An anomaly detection technique based on a chi-square statistic for detecting intrusions into information systems. *Quality and Reliability Engineering International* 17(2), 105–112 (2001), <http://dx.doi.org/10.1002/qre.392>
- [34] Zhang, J., Wang, H.: Detecting outlying subspaces for high-dimensional data: the new task, algorithms, and performance. *Knowledge and Information Systems* 10(3), 333–355 (2006), <http://dx.doi.org/10.1007/s10115-006-0020-z>
- [35] Zhao, F., Liu, J., Liu, J., Guibas, L., Reich, J.: Collaborative signal and information processing: an information-directed approach. *Proceedings of the IEEE* 91(8), 1199–1209 (2003)

Literaturüberblick über Methoden zur expliziten Interaktion in intelligenten Umgebungen

Christopher Bazant
Christopher.Bazant@student.kit.edu

Karlsruhe Institute of Technology (KIT)
Institute of Telematics - Pervasive Computing Systems - TecO
Vincenz-Prießnitz-Str. 1, 76131 Karlsruhe, Germany <http://www.teco.edu>
Betreuer: Matthias Berning

Zusammenfassung. Die voranschreitende Computertechnologie ermöglicht heutzutage die Modifizierung von Gegenständen wie Lampen, Klimaanlage oder Küchengegenständen. Diese Appliances verlangen nach einer neuen intuitiven Art der Steuerung. In den letzten Jahren ist daher die Erforschung neuer Interaktionsmöglichkeiten in den Vordergrund gerückt. Dazu gehören u.a. die Steuerung per Gesten, Sprache oder mit Hilfe eines Touchscreens. Diese Seminararbeit gibt einen Überblick über diese Steuerungsarten. Einschlägigen Artikel der letzten Jahre werden der jeweiligen Interaktionsart zugeordnet, sowie die Konzepte und Ansätze zusammengefasst. Es werden Vor- und Nachteile der Artikel aufgelistet, sowie Hardware-Voraussetzungen zur Anwendung der Konzepte genannt.

Key Words: Smart Home, Ambient Intelligence, Gesture, Gaze based, Multimodal, Touch, TUI, Spoken-Dialogue, Voice, categorize, user interface, environmental control, smart environment

1 Einleitung und Grundlagen

1.1 Motivation

In den letzten Jahrzehnten hat die Computertechnologie einen Wandel erlebt. Anfangs waren Computer unhandlich, teuer und schwer zu bedienen. Ein einzelner Computer wurde von mehreren Menschen genutzt. Die Entwicklung des Personal Computers in den 80er Jahren brachte eine wesentliche Änderung: Der Computer wurde für jedermann erschwinglich und das Bedienkonzept von Tastatur und Maus war alltagstauglich. Die voranschreitende Entwicklung von Computersystemen in den letzten zwei Jahrzehnten ermöglicht heutzutage die Modifizierung von Alltagsgegenständen wie z.B. Lampen, Kühlschränke, Fernseher und mobilen Endgeräten. Man verwendet nun nicht mehr nur einen Computer, sondern eine Vielzahl von Computersystemen (siehe Abb. 1).

Durch die Vielzahl der Systeme rückt die komfortable und einfache Bedienung immer mehr in den Vordergrund. Auch im Hinblick auf den demografischen

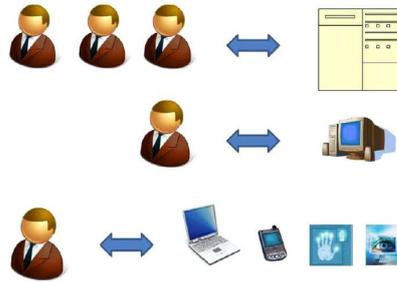


Abb. 1. Entwicklung des Verhältnis zwischen Mensch und Computer aus [1]

Wandel, der in vielen Ländern eine große Herausforderung darstellt. Laut dem Statistischen Bundesamt Deutschland wird der Anteil der über 60 jährigen im Jahre 2060 auf 39,2% anwachsen [2]. Durch den entstehenden höheren Bedarf an jungen Hilfskräften wird es ohne die Entwicklung geeigneter Technologien für ältere und bewegungseingeschränkte Menschen auf Dauer nicht möglich sein, diesen ein altersgerechtes Leben zu bieten. Um diesen Menschen aber ein unabhängiges Leben zu ermöglichen und sie in ihrem Alltag bestmöglich zu unterstützen bedarf es neuer Technologien, die einfach und intuitiv zu bedienen sind. Dabei sollen diese Technologien keinesfalls den Menschen ersetzen, ihn aber bestmöglich unterstützen. Die Bedienung dieser Technologie ist daher ein essentieller Bestandteil, der in den Fokus der Wissenschaft rückt.

1.2 Grundlagen

Im folgenden Kapitel werden grundlegende Begriffe erläutert, die für das Verständnis dieser Arbeit nötig sind. Es wird zuerst der Begriff *Ambient Intelligence* definiert, danach werden die Begriffe *explizite* und *implizite* Interaktion von einander abgegrenzt und die Begriffe *Modalität* sowie *Appliance* erläutert.

Ambient Intelligence Ambient Intelligence (Intelligente Umgebung) ist als ein Paradigma zu sehen, welches die verschiedene Konzepte von Ubiquitous (allgegenwärtig) Computing, Pervasive (durchdringend) Computing und der Mensch-Maschine Interaktion vereint. Der Begriff wurde erstmal von der European Commissions Information Technologies Advisory Group (ISTAG) eingeführt. Cook et. al haben ihrer Arbeit [1] eine Reihe von Aussagen in der gängigen Literatur über Ambient Intelligence gesammelt. Nach den Autoren ist eine intelligente Umgebung durch folgende Kriterien definiert:

1. *sensitive* Die Umgebung ist (kontext)-sensitiv.
2. *responsive* Die Umgebung reagiert auf den Menschen.
3. *adaptive* Die Umgebung passt sich den Personen, die sich in ihr befinden, an.

4. *transparent* Die Computer innerhalb der Umgebung sind für den Menschen nicht wahrnehmbar bzw. unsichtbar. Das Konzept geht auf Mark Weiser zurück, der dieses bereits 1991 einführte und in seinem Aufsatz schrieb: „The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it“. [3]
5. *ubiquitous* Der Begriff des „Ubiquitous Computing“ oder des „Pervasive Computing“ ist eng mit dem Konzept der Transparenz verzahnt. Das Konzept des „Ubiquitous Computing“ wurde ebenfalls von Mark Weiser [4] geprägt. Es besagt, dass Computer allgegenwärtig werden und sich in unsere Gesellschaft praktisch unsichtbar einfügen. Die Vision von Weiser ist in Abb. 2 dargestellt: jeder nutzt implizit eine Vielzahl von Computern.
6. *intelligent* Die Umgebung besitzt die Fähigkeit zu lernen und intelligent auf den Benutzer zu reagieren. Das heißt auch, dass das System hilft, wenn Hilfe gebraucht wird und ansonsten unsichtbar bleibt.

Eine weitere Eigenschaft einer intelligenten Umgebung ist die Vernetzung zwischen Geräten, die über verschiedene Sensoren Informationen sammeln, diese verarbeiten und über Aktuatoren auf die Umwelt und den Menschen reagieren. Abb. 2 zeigt den Paradigmenwechsel in der Interaktion und die Vernetzung der Geräte. Während bei der bisherigen Interaktion zwischen dem Menschen und den Objekten seiner Umwelt jedes Gerät einzeln angesprochen wird, so findet beim Konzept der Ambient Intelligence eine Interaktion mit allen Geräten gleichzeitig statt. Dabei werden Informationen zwischen den Geräten ausgetauscht um auf die Wünsche des Menschen reagieren zu können.

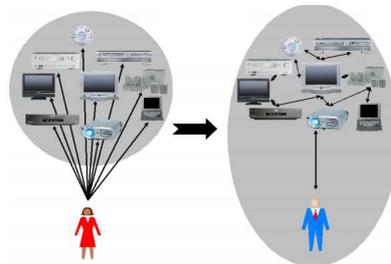


Abb. 2. Die Vision von M. Weiser: die Kommunikation mit allen Geräten gleichzeitig

Explizite und implizite Interaktion Ein wichtiger Bestandteil von Ambient Intelligence ist die Interaktion zwischen Mensch und Computer. Interaktion bezeichnet das „aufeinander bezogene Handeln zweier oder mehrerer Personen“ [5]. Im Kontext mit intelligenten Umgebungen bezeichnet der Begriff die Kommunikation zwischen Mensch und Maschine. Es gibt zwei Arten von Interaktionen: Bei einer **expliziten Interaktion** möchte der Nutzer gezielt etwas steuern und

nimmt dabei aktiv am Geschehen teil. Er verändert dadurch explizit seine Umgebung. Ein Beispiel ist das aktive Anschalten eines Gerätes oder das leiser bzw. lauter stellen des Radios.

Bei einer **impliziten Interaktion** interpretiert das System die Absicht des Nutzers kontextbezogen und reagiert entsprechend, indem es die Umwelt verändert. Ein Beispiel ist, das automatische Ausschalten der Heizung, sobald der Nutzer ein Fenster öffnet.

Die Konzepte führen zu einem Zwiespalt: Man möchte einerseits die Interaktion zwischen Mensch und Maschine verringern, um Menschen eine möglichst bequeme und implizite Nutzung solcher Systeme zu ermöglichen. Andererseits benötigt man explizite Interaktion um dem System die Bedürfnisse und Wünsche mitzuteilen, die nicht aus dem Kontext hervorgehen.

Modalitäten Als Modalität wird der Kanal zur Kommunikation zwischen Mensch und Maschine bezeichnet. Abbildung 3 zeigt Beispiele zu den menschlichen Modalitäten: Blick, Sprache und Gesten. Jede Modalität hat Vor- und Nachteile (z.B. ist die Sprache nicht gut geeignet um handwerkliche Arbeiten zu erledigen) und nicht jeder Mensch verfügt über alle Modalitäten (z.B. durch Behinderungen). Dies ist beim Design von Mensch-Maschine-Interfaces zu beachten.

Modalität	Beispiel
Visuell	Blick (Aufnahme von) Bilder(n)
Akustisch	Spracheingabe Akustische Ausgaben ohne Sprache
Haptisch/kinästhetisch	
Berührung	Druck, Oberflächenbeschaffenheit
Handbewegung	Zeichensprache, 3D-Bewegungen, Handschrift, Zeichnungen
Kopfbewegung	Lippenlesen/Mundbewegungen, Kopfbewegungen und Blickrichtung, Gesichtsausdrücke
Körperbewegung	Skeletontracking, Bewegungsanalyse

Abb. 3. Beispiele zu menschlichen Modalitäten aus [6]

Appliance Eine Appliance ist ein Gerät, welches aus einer Kombination von Hardware und eine speziell für diese Hardware entwickelte Software besteht. Eine Appliance erfüllt meist nur eine spezifische, eingeschränkte Funktion. Sie wird allein für diese Funktion entworfen und besitzt meistens nur wenige andere Funktionen. Sie ist somit funktional abgeschlossen. In vielen Appliances werden Funktionen daher hardwareseitig implementiert. Das schränkt zwar die Funktionalität ein, erhöht aber Arbeitsgeschwindigkeit und damit die Effizienz. Eine Appliance sollte idealerweise einfach zu bedienen und zu erlernen, trotzdem vielseitig in der Anwendung (z.B. spezifische Funktion in vielen Kontexten nutzbar) sein und Spaß machen. Ein Beispiel wäre ein Internetradio oder das iPad.

1.3 Ziele und Grenzen der Seminararbeit

Die heutige Technologie ermöglicht es mit relativ wenig Aufwand eine Ambient Intelligence zu schaffen, in der Appliances miteinander verbunden sind. Um mit diesen Appliances auf einfache und bequeme Art und Weise zu interagieren, spielt das User-Interface eine entscheidende Rolle. Gerade mit dem Aufschwung der Microsoft Kinect und mobilen Endgeräten wie Smartphones sind in den letzten Jahren eine Vielzahl neuer Interfaces zur Steuerung von Appliances entstanden. Diese Geräte erlauben eine einfache und intuitive Steuerung, sind günstig und können von jedermann erworben werden.

Im Hinblick auf die Vielzahl dieser Systeme wird in der Seminararbeit ein kontextbezogenen Literaturüberblick erstellt um interessante Ansätze der letzten Jahre vorzustellen. Es werden Artikel erläutert, die sich mit Interfaces mit expliziter Steuerung der Appliances beschäftigen. Dazu zählen: *Gestensteuerung*, *Sprachsteuerung*, *Blicksteuerung*, *Interfaces mit Touch*, *Tangible User-Interfaces (TUI)*, und *Multimodale Ansätze*. Es werden Konzepte der einzelnen Artikel vorgestellt. Es soll ein objektiver Literaturüberblick über die einzelnen Thematiken gegeben werden. Dabei wird keine subjektive Bewertung der einzelnen Artikel durchgeführt, es werden jedoch Vor- und Nachteile erläutert, sowie die Hardware-Voraussetzungen genannt.

Die Arbeit ist wie folgt gegliedert. In Kapitel 2 werden einschlägige Artikel kategorisiert. Die Artikel werden dazu in einer Tabelle nach den einzelnen Interaktionsarten übersichtlich aufgelistet. In den weiteren Kapiteln 3, 4, 5, 6 und 7 werden Artikel zur Gestensteuerung, Sprachsteuerung, Blicksteuerung, Steuerung per Berührung (Touch) und tangible Interfaces vorgestellt. Dabei werden zu jeder Interaktionsart einschlägige Studien vorgestellt, bevor mehrere Artikel zu einzelnen Ausprägungen jeder Interaktionsart vorgestellt werden. Kapitel 8 erläutert Konzepte zur multimodalen Steuerung von Appliances. Es werden Artikel zum Design der multimodalen Interfaces, sowie Artikel die verschiedene Modalitäten kombinieren vorgestellt. In Kapitel 9 werden die gesammelten Erkenntnisse zusammengefasst und eine Aussicht auf zukünftige Forschung gegeben. Weiterhin wird die neue Kinect 2.0 vorgestellt, die eine bessere Gestenerkennung bietet und somit die Flexibilität für solche Interfaces erhöht.

2 Kategorisierung der einzelnen Artikel

Tabelle 1 zeigt die Einordnung der Artikel in die verschiedenen Interaktionsarten. Es werden Artikel zur *Gesten-*, *Sprach-* und *Blicksteuerung* sowie *Steuerung per Berührung* und *Tangible User-Interfaces* eingeordnet. *Multimodale Ansätze* sind durch die Zuordnung in mehr als eine Kategorie gekennzeichnet. Bei der Einordnung wurden gleiche Konzepte zusammengefasst (z.B. Studien zur Gestensteuerung). Verweise auf Sekundärliteratur, sowie Artikel zu bestimmten Algorithmen, die bei den jeweiligen Steuerungsarten verwendet wurden, sind nicht in die Tabelle eingeordnet, um diese übersichtlich zu halten.

	Steuerung per				
	Geste	Sprache	Blick	Berührung	TUI
Kapitel 3: Steuerung per Geste					
[7,8,9]	X				
[10,11,12]	X				
[13,14,15,16,17,18]	X				
[19,20]	X				
[21,22,23]	X				
[24]	X				
Kapitel 4: Steuerung per Sprache					
[25,26,27,28,29,30]		X			
[31,32,33]		X			
[34,35]		X			
[36,37,38]		X			
[39]		X			
Kapitel 5: Steuerung per Blick					
[40,41,42]			X		
[43,44]			X		
[45,46,47]			X		
[48,49]			X		
Kapitel 6: Steuerung per Berührung					
[50,51,52]				X	
[53,54,55]				X	
[56]				X	
Kapitel 7: Steuerung per Tangible-User-Interfaces					
[57,58]					X
[59]					X
[60,61,62]					X
[63]					X
Kapitel 8: Multimodale Ansätze					
[64,65]	X	X			
[66,67,68]	X		X		
[69]			X	X	
[70]		X		X	
[71]	X			X	
[72]	X				X
[73]	X	X		X	

Tabelle 1. Artikel gegliedert in die verschiedenen Konzepte: Steuerung per Geste, Sprache, Blick und Berührung, sowie Tangible-User-Interfaces und Multimodale Ansätze

3 Steuerung per Geste

Kurtenbach und Hulteen definieren eine Geste wie folgt: „A gesture is a motion of the body that contains information. Waving goodbye is a gesture. Pressing a key on a keyboard is not a gesture because the motion of a finger on its way to hitting a key is neither observed nor significant. All that matters is which key was pressed“ [74].

Unter Gestensteuerung fallen die Geste von Händen oder einzelner Finger, des Gesichtsausdruckes oder des gesamten Körpers. Eine Geste kann statisch (z.B. ein Zeichen) oder dynamisch (eine Bewegung) sein und mit oder ohne ein weiteres Eingabegerät erfolgen (z.B. Smartphone oder WiiMote).

In der Literatur gibt bereits einige Artikel zur Gestenerkennung, z.B. [7] und [8], die ein sehr günstiges Gestenerkennungssystem entworfen haben. Oder [9], die ein Erkennungssystem als Web-Service anbieten. Im folgenden Abschnitt wird auf das Design von Gesten und Steuerungskonzepte in intelligenten Umgebungen eingegangen.

3.1 Design von Gesten

Ein wichtiger Bestandteil der Gestensteuerung ist das Design der Gesten selbst. Gesten sollten natürlich sein, leicht zu erlernen und leicht zu merken. Vatavu [10] führt in seiner Arbeit eine Studie durch, in der er Nutzer in den Designprozess von Gesten mit einbezieht und zwei verschiedene Eingabetechnologien evaluiert: (1) Gestensteuerung mit einer WiiMote und (2) Freihand mit der Xbox Kinect (siehe Abb. 4). Beide Ansätze werden miteinander verglichen und es werden eine Reihe von Gesten für beide Ansätze vorgeschlagen.

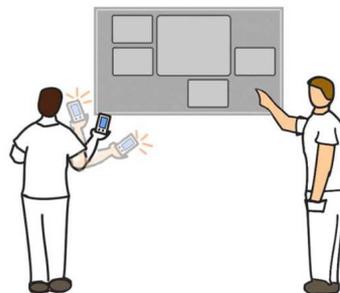


Abb. 4. Steuerung mit Eingabegerät vs. Freihandsteuerung aus [10]

In dem Artikel [11] stellen die Autoren eine Methodik vor, um Gesten zur Steuerung von Geräten via Smartphone in intelligenten Umgebungen zu definieren und zu evaluieren. Die Methodik wurde ursprünglich in [75] und [76] für Interfaces auf Oberflächen erstellt. Die Autoren adaptieren diese Methodik. Grundidee ist der Einbezug der Nutzer in den Designprozess. Es wird auch eine

Studie nach der vorgestellten Methodik durchgeführt. In einem praktischen Test wurde das Touch-Interface des Smartphone genutzt um Geräte via einer GUI auszuwählen. Die ausgewählten Geräte werden dann per Gesten gesteuert, die sich in der Studie als äußerst intuitiv herausgestellt hatten. (siehe Abb. 5). Die Gestenerkennung wird mit Fast Dynamic Time Warping [77] durchgeführt. In zukünftigen Arbeiten wollen die Autoren zur Auswahl der Geräte auf die GUI verzichten. Diese soll entweder per „Zeigefunktion“ oder durch Bilderkennung durch die Kamera im Smartphone erfolgen.

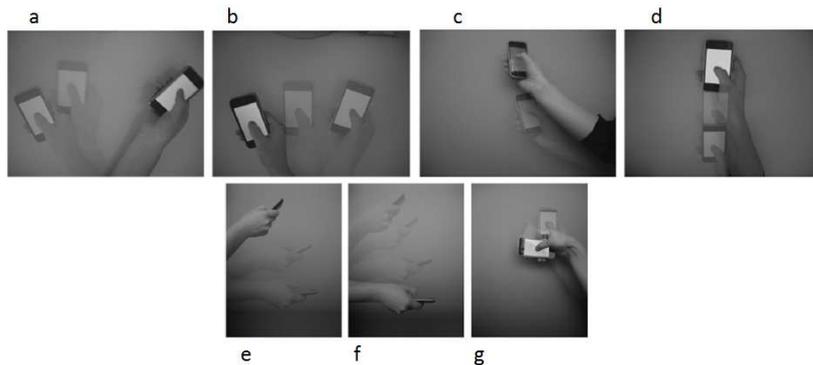


Abb. 5. Finale Gesten aus [11]: a) nach rechts bewegen; b) nach links bewegen; c) Auf- und Abwärts bewegen d) nach vorne zeigen; e) nach unten bewegen; f) nach oben bewegen; g) drehen

In [12] wird untersucht wie man mit weit entfernten Objekten, die z.B. auf eine Leinwand projiziert werden, interagieren kann. Dabei konzentrieren sich van de Camp et. al. auf die Untersuchung von Gesten, die mit einer Hand durchgeführt werden können und bei denen man kein weiteres Eingabegerät benötigt. Abb. 6 zeigt die Klassifizierung und Auswahl der Gesten für die Studie. Die Autoren führen zur Gestenerkennung eine Wizard-of-Oz Studie durch, um sich auf die Evaluierung konzentrieren zu können. Die Gesten wurden von den Teilnehmern in drei Kategorien bewertet: (1) Allgemeine Zufriedenheit, (2) Ermüdungsgrad und (3) Ausführungszeit der Gesten. Die Gesten „point“ und „airtap“ erzielten die besten Ergebnisse.

3.2 Freihanderkennung ohne Eingabegeräte

Die Kinect von Microsoft ermöglicht mit einem relativ einfachen, günstigen und zugleich hochmobilen Konzept, die Freihanderkennung und damit den intuitiven Einsatz von Gestensteuerung in intelligenten Umgebungen. Auch andere Kameras ermöglichen durch Tiefenanalyse die Möglichkeit Gesten zu erkennen. Eine ausführliche Literaturrecherche über Gestenerkennung mit Tiefenkameras ist in

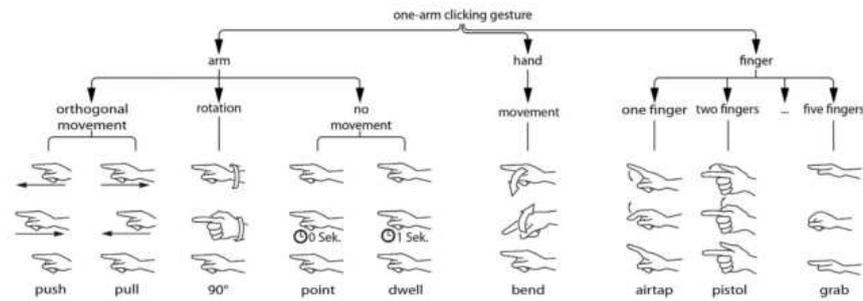


Abb. 6. Klassifizierung und Auswahl der untersuchten Gesten aus [12]

[78] zu finden.

In [13] wird ein System vorgestellt, welches ermöglicht, mit Zeigegesten Geräte wie Lampen und Ventilatoren zu steuern. Es unterstützt mehrere Nutzer und verfolgt die Bewegung der einzelnen Personen in Echtzeit. Um dies zu erreichen werden zwei Kinect-Kameras verbunden. Ein zentrales Modul berechnet die eingehenden Daten und erstellt ein 3D-Modell der Nutzer, des Raumes und der Appliances. Abb. 7 zeigt die Modellierung der Nutzer. Für die Steuerung der Objekte wird das NAIF Framework [79] benutzt. Das System erlaubt weiterhin unterschiedliche Aktionen, je nachdem ob der Nutzer sitzt oder steht. Im Praxistest erwies sich der Ansatz als äußerst intuitiv, jedoch ist die Anzahl der möglichen Gesten noch limitiert. Ein weiterer Nachteil ist die Kinect übliche Registrierung des Nutzers am Anfang.

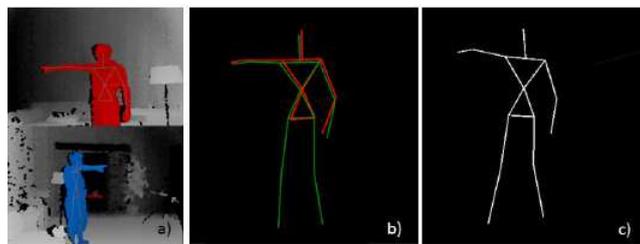


Abb. 7. Modellierung der Nutzerinformationen aus [13]: a) Nutzer aus Sicht von 2 Kinect-Kameras; b) 3D-Modell eines Nutzer aufgenommen von 2 Kameras; c) Zusammenlegung der Nutzerdaten zu einem Modell

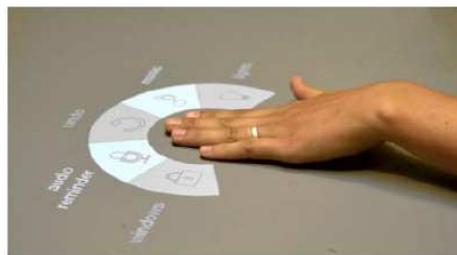
Kinect in Kitchen [14] überträgt die Idee der Kinect in die Küche und erlaubt die Steuerung einer Zeituhr, eines Musikplayers und das Blättern in einem digitalen Kochbuch. Der Nutzer kann durch eine Seitwärtsbewegung durch das Menü blättern und durch eine Vorwärtsbewegung eine Auswahl treffen. Dabei können die Nutzer nicht nur Arme, sondern auch Kopf und Beine benutzen. Die Autoren

haben noch mit unabsichtlicher Auswahl von Aktionen zu kämpfen, da der Nutzer dauerhaft registriert ist. Trotzdem befanden die meisten Nutzer es als sehr hilfreich und als intuitive Art der Steuerung. Die Autoren möchten in Zukunft eine automatische Menüsperre und Multi-User Erkennung implementieren.

Heidrich et. al. [15] entwickeln eine Steuerung für Appliances auf einem Interface, welches auf eine Oberfläche projiziert wird. Das Interface wird verwendet um Fernseher, Lichter und Rolläden zu steuern. Abb. 8 zeigt das System: An der Decke ist eine Kinect-Kamera (zur Handbewegungserfassung) und ein Projektor (Zur Anzeige des Interface auf der Oberfläche) angebracht. Um die Bewegung der Hand besser zu erfassen, ist eine weitere Kinect-Kamera in der Wand untergebracht. Eine Geste auf dem Tisch aktiviert das kreisförmige Menü und der Nutzer kann die entsprechende Appliance durch eine Handbewegung durch den Menüpunkt hindurch aktivieren. In einer durchgeführten Studie wird das Interface evaluiert. Die Nutzer tendierten intuitiv dazu die Menüpunkte per Touch zu aktivieren. Daher überlegen die Autoren, die Auswahl der Menüpunkte zu ändern.



(a) Aufbau des Interfaces: An der Decke sind Kinect-Kamera und Projektor in eine Lampe integriert



(b) Das kreisförmige Menü, welches auf die Oberfläche projiziert wird

Abb. 8. Überblick des Systems aus [15]

Nomadic Gestures [16] personalisiert das Gestenerkennungssystem. Anstatt für jedes System eine Reihe von Gesten fest zu definieren, kann der Nutzer seine eigene Gesten vor der Nutzung in das System via Smartphone hochladen (siehe Abb. 9). Das Interface passt sich also an den Nutzer an und nicht der Nutzer an das Interface. In der Arbeit wird das System an einem „Ambient Display“ getestet. Zur Gestenerkennung wird die Kinect eingesetzt. Nachteil ist, dass jeder Nutzer erst einmal eine Reihe von Gesten definiert haben muss und ein Smartphone besitzen sollte. Weiterhin muss die Übertragung der Gesten manuell erfolgen. In weiteren Arbeiten soll dieses Problem angegangen werden.

Bisher wurde nur Literatur zur Gestenerkennung mit der Kinect betrachtet. Einen anderen Ansatz verfolgen die Autoren von [17]. Sie stellen ein optisches

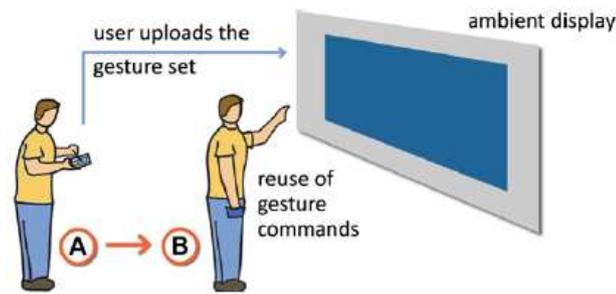


Abb. 9. Idee von *Nomadic Gestures* [16]: A) personalisierte Gesten werden in das System hochgeladen; B) pers. Gesten werden für Interaktion genutzt;

Bewegungserkennungssystem vor, dass nicht wie die meisten anderen Systeme 3D-Daten (horizontal, vertikale und Tiefeninformation) nutzt, sondern die Richtung und Geschwindigkeit sowie zusätzlich die Position des sich bewegenden Objektes liefert (Optical Flow System). Das System wurde selbst entwickelt und wird benutzt um einen Fernseher zu steuern (Navigation im Menü, Auswahl eines Filmes und Lautstärke verändern). Laut den Autoren ist es extrem günstig (weniger als 10\$), jedoch in der Gestenerkennung und Steuerung nur rudimentär gegenüber Systemen wie der Kinect. Abb.10 zeigt die Konfiguration des Systems.

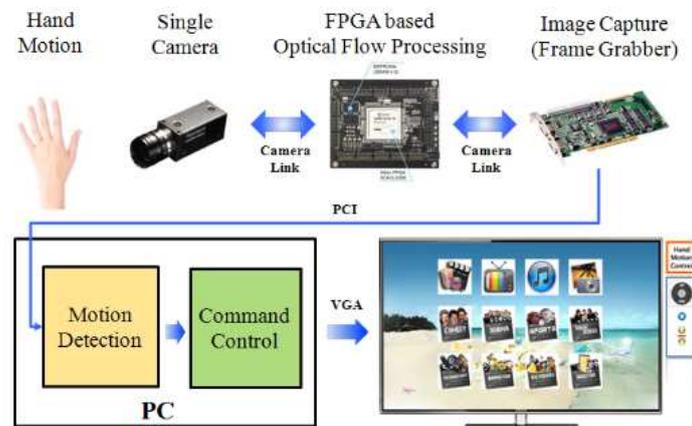


Abb. 10. Konfiguration des Optical Flow System aus [17]

In [18] wird durch einer handelsüblichen Kamera und Gesten einen Fernseher gesteuert (Senderwechsel, Lautstärke) kann. Die Autoren stellen ein Erken-

nungssystem vor, dass aus der Videoaufnahme in Echtzeit Bewegungen erkennt, interpretiert und so den Fernseher steuert. Die Echtzeitberechnung basiert auf MoSIFT, einer sehr rechenaufwändigen aber erfolgversprechenden Methode zur Erkennung von Bewegung. Um geringe Latenzzeiten zu ermöglichen modifizierten die Autoren diese Methode im zweiten Schritt. Für die Berechnung ist jedoch weiterhin eine hohe Rechenleistung nötig.

3.3 Gestenerkennung mit Smartphone

Nutzer bevorzugen eine Fernsteuerung, mit der sie global und ohne Verzögerung alle Objekte innerhalb einer intelligenten Umgebung steuern können [19]. Dies könnte z.B. durch ein Smartphone erreicht werden. Durch die integrierten Sensoren, das Touch-Interfaces und die heutzutage hohe Verfügbarkeit scheint das Smartphone ein perfektes Eingabegerät zu sein.

In [20] basiert die Implementation auf den verbauten Lage-, Richtungs- und Beschleunigungssensoren im Smartphone, welche die Richtung des Smartphones berechnen. Zusätzlich wird der Aufenthaltsort (auf den Raum genau) durch WiFi bestimmt. Der Nutzer zeigt auf das (vorher modifizierte) Zielobjekt, aktiviert es via Klick auf dem Smartphone und kann eine spezielle Geste aus. Zeigt der Nutzer beispielsweise auf eine Lampe so wird diese per Klick eingeschaltet. Führt er weiterhin eine Abwärtsbewegung durch, so wird diese dunkler. Da heutzutage in den meisten Wohnungen WiFi vorhanden ist, ist kaum zusätzliche Infrastruktur nötig. Lediglich ein Rechner, der als Steuerungs- und Vermittlungseinheit zwischen Smartphone und den Zielgeräten dient muss aufgebaut werden. Nachteil der Methode ist die Ungenauigkeit der Lokalisierung. Da mit nur einem WiFi keine Kreuzvalidierung der Position möglich ist, könnte es Probleme beim Ansprechen von dicht aneinander stehenden Geräten geben. Laut den Autoren ist jedoch keine exakte Positionsbestimmung mit durchschnittlich fünf Objekten nötig. Die Autoren haben auch ein Video veröffentlicht¹.

3.4 Gestenerkennung mit dem WiiMote-Controller

Auch der Controller der Nintendo Wii, der WiiMote, kann zur Erkennung von Gesten eingesetzt werden. Nazemi et al. [21] führten ein Experiment durch, bei dem die Teilnehmer Zoom- und Scroll-Gesten mit der WiiMote durchführen mussten. In [22] benutzen die Autoren die WiiMote um damit das Navigationsmenü eines Home-Entertainment System zu steuern.

In [23] wird die WiiMote-Controller für die Steuerung von Appliances genutzt (Abb. 11). Für die Gestenerkennung wird das Framework TaKG [80] benutzt. Das Framework stellt eine Reihe von Mechanismen, wie z.B. DynamicTimeWarping [81] bereit um die verschiedenen Signale der WiiMote zu verarbeiten und die Gestenerkennung zu ermöglichen. Weiterhin kann man dem System mit Hilfe des Framework Gesten für verschiedene Aktionen beibringen. Die WiiMote wird verwendet um eine Dunstabzugshaube, einen Fernseher und eine Lampe, die

¹ <http://www.youtube.com/watch?v=2AbnjOqz24c>

über eine Middleware basierend auf dem ISO 24752 Standard URC (Universal Remote Console) und UHC (Universal Control Hub)[82] erreichbar sind, per Geste zu steuern. Die Auswahl des Gerätes wird per Button auf der WiiMote (und Feedback via Audio) getroffen.



Abb. 11. Smart Kitchen aus [23]

Die Autoren untersuchen in ihrer Arbeit zwei Ansätze. Hintergrund dieser Konzepte ist, dass die Nutzer weniger Gesten lernen müssen, um die Geräte zu kontrollieren: (1) Gesture control in context: Eine Geste soll eine Funktionen bei einem Appliances auslösen. Dabei soll die gleiche Geste eine ähnliche Funktion bei verschiedenen Appliances bewirken. (2) Gesture triggered scenarios: eine Geste soll eine Kettenreaktion auslösen. Z.B. soll die Geste für „ins Bett gehen“ alle laufenden Geräte ausschalten. Die Konzepte wurden von 13 Teilnehmern evaluiert. 85% befanden den ersten Ansatz für gut, 70% den zweiten Ansatz. In zukünftigen Arbeiten möchten die Autoren, die Auswahl der Geräte per Blickerkennung oder durch eine „Zeigefunktion“ realisieren.

3.5 Gestenerkennung mit Hilfe von Handschuhen

Anstatt eines Controllers, können auch Handschuhe für die Unterstützung der Gestenerkennung dienen. In [83] wird eine Infrarot-Kamera und einen Handschuhe mit Infrarot-Emitter benutzt um die Handbewegung zu verfolgen (siehe Abbildung 12) für die Steuerung von Lampen oder eines Video-Players. Eine Lampe wird durch das Zeichnen des Buchstaben „L“ eingeschaltet, das Stoppen eines Filmes durch den Buchstaben „S“. Die Autoren haben auch eine Nutzerstudie durchgeführt. Das System hatte dabei eine Erfolgsrate bei der Identifizierung der Gesten von 97,25% und arbeitet unabhängig von den Lichtbedingungen im Raum. Die Teilnehmer fanden das System für die Steuerung der Geräte leicht erlernbar und die Gesten intuitiv.

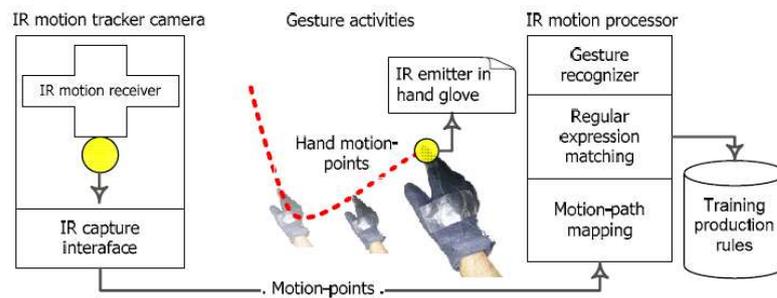


Abb. 12. Aufbau und Komponenten der Gestenerkennung aus [83]

Durch neue Technologien soll das Leben von älteren und bewegungseingeschränkten Menschen verbessert werden (Ambient Assisted Living). Die Komplexität von Gesten stellt ältere Menschen aber vor eine große Herausforderung. In [24] wird auf sog. „Baby Signs“ zurückgegriffen. Diese sind laut den Autoren auch für ältere Menschen einfach zu erlernen. Zur Gestenerkennung wird eine Kamera und farbige Handschuhe benutzt. Die Segmentierung der Hand zum Hintergrund erfolgt mit HSV Histogramme [84], für die Bewegungserkennung wurde ein HMM (Hidden Markov Model) adaptiert und das Hidden Markov Model Toolkit [85] verwendet. Die Erfolgsrate der Gestenerkennung betrug 94,33% für die in Abb. 13 gezeigten Gesten.

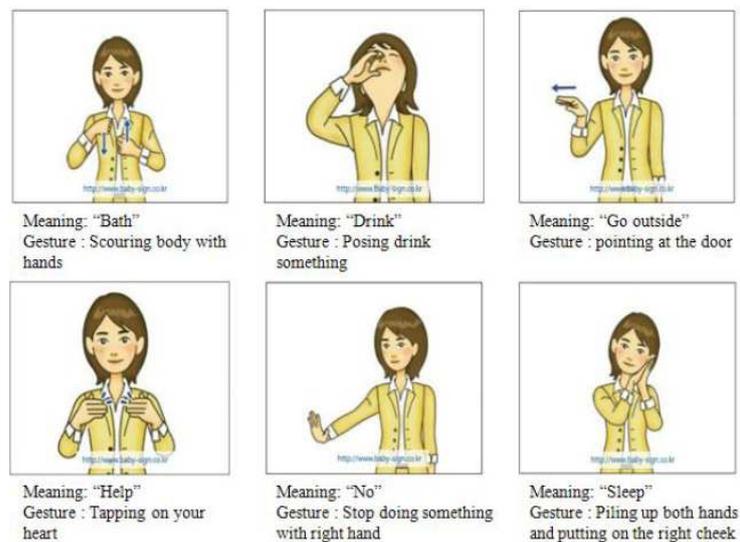


Abb. 13. Die sog. „Baby Signs“ aus [24]

4 Sprachsteuerung

Eine der natürlichsten Wege der Kommunikation zwischen Menschen ist die Sprache. Sprachsteuerung nimmt daher eine wichtige Rolle bei der Interaktion in intelligenten Umgebungen ein. Im folgenden Kapitel werden relevante Studien zu Sprachsteuerung vorgestellt. Danach werden Systeme für Sprachbefehle und Spoken-Dialogue-Systeme vorgestellt.

4.1 Studien zur Sprachsteuerung

Die Sprache hat ein großes Potential für die Steuerung in intelligenten Umgebungen. In [25] wurden verschiedene Interfaces (GUI, Sprache, Sprache mit Ortsangabe, Geste und Sprache etc.) zur Steuerung von Lampen in einer Wizard-of-Oz Studie verglichen, um festzustellen, welches bevorzugt wird. Die Autoren haben festgestellt, dass Sprache die bevorzugte Methode zur Steuerung von Lampen ist.

In einer weiteren Studie [26] wurde untersucht, ob Sprachsteuerung ein geeignetes Interface für ältere Menschen darstellt. Dazu wurden bei den Teilnehmern Lampen und ein weiteres Gerät nach Wahl angepasst, um dieses mit Sprache steuern zu können. Die Software-Realisierung erfolgte mit Open-Source Software. Als Hardware kam ein Insteon Powerline Controller und ein Laptop mit Mikrofon zum Einsatz. Die Teilnehmer konnten die Geräte drei Tage lang entweder manuell oder per Sprache steuern. Die Sprachsteuerung wurde trotz einiger Probleme sehr gut angenommen. Die Autoren merken jedoch an, dass es durch den kurzen Zeitraum zu Verfälschungen der Studie kommen könnte (z.B. durch Mehrnutzung neuer Interaktionsmöglichkeiten).

Die Stimme und der Sprachstil variieren von Mensch zu Mensch. Insbesondere groß ist der Unterschied zwischen jungen und alten Menschen. Möller [27], Georgila [28] und Vippera [29] untersuchen die Sprachstile, das Vokabular und die Interaktion mit dem Sprachsystem. Sie fanden erhebliche Unterschiede zwischen den beiden Gruppen. Während jüngere Menschen ihren Sprachstil und das Vokabular relativ schnell an das System anpassen konnten, tendieren ältere Menschen dazu, mit dem Sprachsystem wie mit einem Menschen zu reden: Sie gingen eine Konversation ein, bedankten sie sich oder gaben mehr Informationen als nötig wären. Sie benutzen eine höhere Anzahl an Wörtern und Synonyme für die gleichen Befehle als es die jüngere Menschen taten. [29] kommt zu dem Ergebnis, dass Systeme, die nur mit Daten von jüngeren Menschen implementiert wurden, erhebliche Probleme bei älteren Nutzern haben werden. Die Erkenntnisse sollten beim Design von Sprachsteuerungssystemen beachtet werden.

4.2 Steuerung per Sprachbefehl

Mit Sprachbefehlen kann auf einfache Art eine Appliance gesteuert werden. Das System gibt dabei kein oder nur wenig Feedback. In [31] schlagen die Autoren eine Architektur für ein Sprachsteuerungssystem zur Kontrolle von Appliances vor, das mit Standardtechnologien für Spracherkennung (VoiceXML, MRCP, SRGS

und SISR) implementiert ist. Als Hardware ist nur ein Computer je Raum und Peripherie wie Mikrophon nötig. Das System ist unabhängig von den genutzten Appliances und kann sich bei Änderungen der Appliances anpassen. Die Autoren führen einen Test durch, in welchem die Teilnehmer via Sprachbefehl Lampen an- und ausschalten und heller/dunkler stellen. Die Nutzer bewerten die Art der Steuerung als angenehm. Probleme gab es jedoch systemseitig bei der Erkennung und Interpretation der Sprachbefehle.

Zhu et. al. [32] stellen ein Sprachsteuerungssystem für Appliances vor, die via ZigBee vernetzt sind. Die Autoren verwenden zur Spacherkennung SI-ASR (Speaker-Independent Automatic Speech Recognition), was es erlaubt die Spacherkennung Nutzer unabhängig durchzuführen. Durch die Vernetzung der Spacherkennungsmodule ist es weiterhin möglich alle Geräte von einem Standort anzusteuern. Es können per Sprachbefehl sowohl Lampen an-/ausgeschaltet werden, ein WasserBoiler auf eine Temperatur eingestellt (die Messung erfolgt über Sensoren) oder auch ein Musikplayer (inkl. Songauswahl in einem Menü) angesteuert werden. Die Spacherkennung kann auf drei Arten erfolgen: (1) auf Knopfdruck, (2) Key-Word getriggert oder (3) dauerhaft aktiviert. Die Nutzer können wählen, wie das System angesteuert werden soll.



Abb. 14. benötigte Komponenten für das Spacherkennungssystem [32]

Soda et. al [33] entwickeln ein Sprachinterface auf Grundlage eines sog. Microphone Arrays-Network [86]. Der Einsatz von mehreren vernetzten Sub-Microphone-Arrays ermöglicht es, dass Nutzer von überall Sprachbefehle verwenden können. Weiterhin kann der Standort einer Person im Raum erkannt werden und zwischen mehreren Geräuschquellen unterschieden werden (siehe Abb. 15). Soda et. al. implementieren auch einen Prototypen ihres Systems. Dieser arbeitet nur mit einem *Sub-Microphone-Array* und die Spacherkennungsrate lag je nach Distanz zwischen 74%-94%.

Der Ansatz klingt vielversprechend, denn durch die Lokalisation einer Person können auch kontextbezogene Sprachbefehle (wie z.B. „Schalte das Licht über

mir an“) ausgeführt werden. Die Autoren planen die Erweiterung ihres Prototypen mit mehreren *Arrays*, um die Lokalisation zu ermöglichen und die allg. Performance zu verbessern.

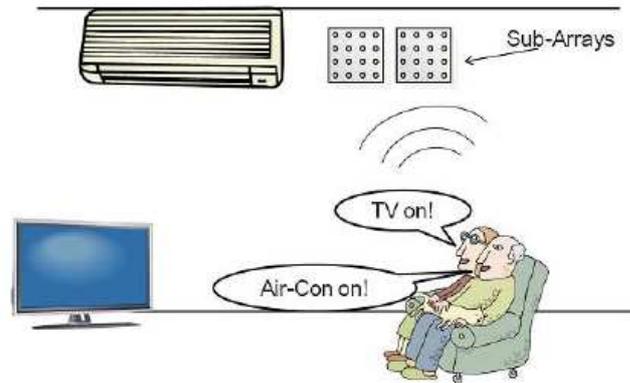


Abb. 15. Sprachbefehle ohne Nutzung eines tragbaren Mikrophones aus [33]

4.3 Konversationen mit Spoken-Dialogue Systeme

Spoken-Dialogue Systeme (SDS) sind ein wichtiger Bestandteil bei der Steuerung von Appliances. Der Computer interagiert mit dem Menschen durch Sprache, reagiert auf Befehle und gibt dem Nutzer ein Feedback. Einen ausführlichen Literaturüberblick über des Thema gibt Lopez et. al. [87]. im Folgenden werden zwei Systeme vorgestellt.

Aus den Artikeln von Minker et. al. [34] und Filipe et. al. [35] geht hervor, dass der Trend im Design von SDS in assistierende, sich den den Nutzer anpassende und proaktiv handelnde Systeme geht. Ein weiterer Trend ist das Dialog-Management und die Interaktion des Systems mit der Umwelt. Die Interaktion mit der Umwelt wird meist durch eine Middleware umgesetzt, die als Interface zwischen Mensch und SDS agiert. Filipe et. al. [35] präsentieren ein Konzept zur Steuerung von Küchengeräten, wie z.b. der Mikrowelle durch Sprachbefehle. Dazu erweiterten die Autoren das SDS um einen „Environment Interaction Manager“, der die Kommunikation zwischen dem SDS und den Appliances koordiniert.

Interact [36], [37] ist eine intelligente Umgebung die mit den Nutzer durch das Sprach-Interface *Odisea* interagieren kann. *Interact* enthält verschiedene Appliances, die durch Standard-Mechanismen (EIB [88] und SNMP [89]) mit dem SDS und untereinander kommunizieren. Die Nutzer können das Licht an-/ausschalten, das Radio-Programm wechseln oder Türen öffnen. Die Umgebung

wird dabei initial per XML-Datei beschrieben, die Informationen über die Appliances enthält. Mit der XML-Datei erstellt das System ein Blackboard [90], mit dessen Hilfe das Interface die Appliances ansprechen kann. Montoro et. al. entwickeln das System in [38] weiter.

4.4 Sweet-Home Projekt

Das *Sweet-Home Project* [91] hat das Ziel, älteren und bewegungseingeschränkten Menschen im Alltag in ihrer Wohnung zu unterstützen. Das Projekt hat sich dabei als Ziel gesetzt, hilfebedürftigen Menschen die Teilhabe am Alltag zu ermöglichen, indem diese verschiedene Appliances mit Sprachbefehlen steuern können und die Umgebung durch Sensoren überwacht werden, um Notfallsituationen festzustellen. Das Projekt versucht bereits entwickelte Standards (wie z.B. e-lio oder Visage) zu integrieren. Eine große Herausforderung ist die Trennung von Sprachbefehlen vs. Umgebungsgeräuschen. Zwar wurden hier schon mit dem entwickelten PATSH System [92] einige Fortschritte erzielt, jedoch funktioniert die Trennung noch nicht einwandfrei. Im Zuge des Projektes wurden auch (Wizard-of-Oz) Studien durchgeführt, um die Akzeptanz solcher Systeme zu untersuchen [30]. Die Studien belegen, dass Sprachsteuerung ein großes Potential zur Steuerung der Appliances besitzt. Gerade Sicherheitsrelevante Aspekte sind bei älteren Personen geschätzt, so z.B. das automatische Absetzen eines Notrufes, wenn die Person „Hilfe“ ruft.

Eine gute Übersicht über Komponenten und das verwendete Audiosystem im Sweet-Home Projekt ist in [39] zu finden.

5 Blicksteuerung

Durch das Sehvermögen können Menschen ihre Umwelt wahrnehmen und mit ihr z.B. durch Blickkontakt interagieren. Auch im Kontext zur Steuerung von intelligenten Systemen kann der Blickkontakt eingesetzt werden. Dabei ist es jedoch sehr schwierig zu unterscheiden, ob man ein Objekt nur anschaut oder mit ihm interagieren möchte. Man nennt dies das „Midas Touch Problem“ [93]. Es existieren einige Lösungen für das Problem, wie z.B. die sog. „dwell time“. Um eine Funktion auszulösen muss man ein Objekt für längere Zeit anschauen. Eine andere Variante ist *blink* (*Blinzeln*), bei welcher man durch blinzeln die Auswahl trifft. Eine weitere Variante ist die Kombination mit anderen Modalitäten wie z.B. Sprache oder Geste (siehe Kapitel 8).

Einen sehr preiswertes Blickerkennungssystem (ITU Gaze Tracker) haben Augustin et. al. in [40] entwickelt und in [41] evaluiert. Laut den Autoren erreicht das günstigste System gleichwertige Ergebnisse verglichen mit hochpreisigen kommerziellen Systemen. Einen allg. Übersichtsartikel zu Blickerkennungssysteme haben Hansen et. al. erstellt [42].

Ein wichtiges Projekt rund um Blicksteuerung ist *COGAIN*. Das *COGAIN* (<http://www.cogain.org/>) Projekt startete 2004 und wurde bis 2009 von der

Europäischen Kommission unterstützt. Bei dem Projekt arbeiten mehrere Forscher zusammen um neue Technologien rund um Blickerkennungs-systeme zu entwickeln. Im folgenden werden Studien zum Design von Gesten, sowie Artikel mit und ohne Head-Mounted Eye Tracker (HMET) vorgestellt. HMET sind Systeme welche am Kopf getragen werden und die Blickerkennung dauerhaft durchführen können.

5.1 Studien zur Blicksteuerung

In [43] hat MacKenzie zwei Experimente durchgeführt um die Blicksteuerung zu evaluieren: (1) „Dwell-Time vs. Key Selection (Tastatur, Maus)“ und (2) Dwell-Time vs. Blink Selection“. Die Teilnehmer mussten anschließend die Interaktionsmethoden bewerten. Neben dieser subjektiven Bewertung wurden auch Kriterien wie Fehlerrate und Genauigkeit gemessen. Obwohl die Maussteuerung am besten abschneidet, sieht der Autor Blicksteuerung als vielversprechende Eingabemethode.

Eine andere Studie [44] befasst sich mit der Auswahl der Blickgesten, vor allem im Kontext mit bewegungseingeschränkten Menschen, die auf eine Blicksteuerung angewiesen sind um sich im Alltag zurechtzufinden. Es wurden zwei empirische Studien durchgeführt: Die erste war eine Umfrage um Anforderungen und Wünsche der Nutzer herauszufinden. Die zweite Studie untersuchte die Auswahl von sog. „single stroke gaze gestures“ und „complex gaze gestures“.

5.2 Blicksteuerung ohne HMET

DOGeye [45] ist ein blickgesteuertes Interface, welches die mittelbare Steuerung von Appliances wie z.B. Fernseher oder Licht ermöglicht. Damit es für gängige Blickerkennungs-systeme und für Appliances nutzbar ist, wurde die Software des Interface nach den *ETU universal driver* Spezifikation implementiert. Die Schnittstelle und Interaktion zu den Appliances ist auf Basis des *Dog* (Domotic OSGi Gateway) aufgebaut. Als Blickerkennungs-system kam myTobii P10 zum Einsatz. Abb. 16 zeigt die Architektur. Der Fokus liegt auf der Blicksteuerung, das Interface kann jedoch auch per Touch gesteuert werden und erfüllt damit die COGAIN Richtlinien.

In Abb. 17 ist das Interface abgebildet. Am oberen Rand sind verschiedene Kategorien wie „Home Management“ oder „Entertainment“ zu sehen. Die Auswahl der Kategorie bzw. des Raumes erfolgt mit einem kurzen Blick. Danach kann ein Aktion mit einer längeren Blick (dwell Time) auf den Button „Enter the room“ ausgewählt werden. Die weitere Auswahl erfolgt nach dem gleichen Schema: Ein kurzer Blick für die Auswahl (bspw. die Lampe), ein langer Blick für die Aktion (einschalten). In einer Nutzerstudie wurde das Interface als sehr einfach und intuitiv bewertet.

In [46] stellen Hennessy et. al. System vor, um die Augenbewegungen einer Person über eine längere Distanz zu erfassen. Dabei wird das Augenerkennungs-system durch einen Schwenk- und Neigungsmechanismus ständig auf den Nutzer

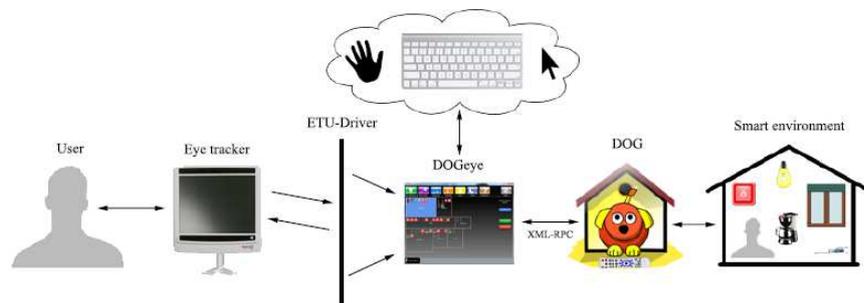


Abb. 16. Architektur von DOGeye aus [45]

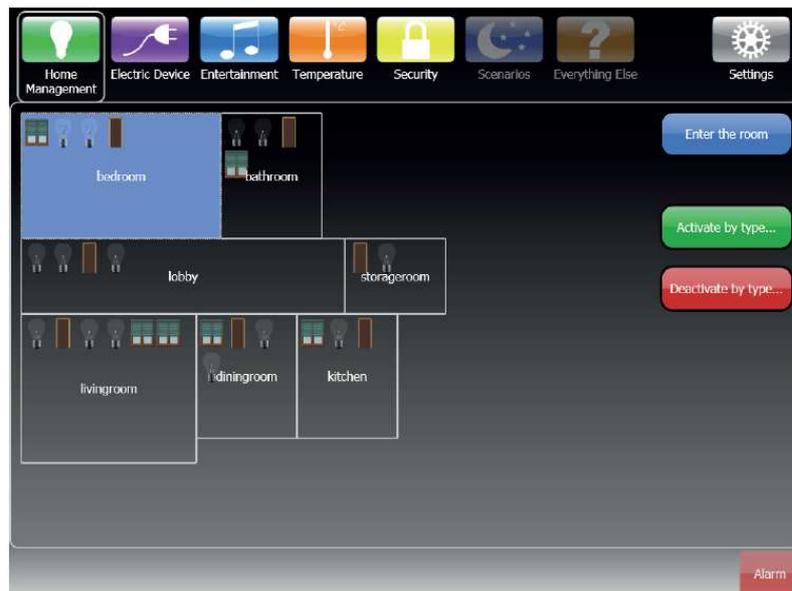


Abb. 17. Das DOGeye-Interface aus [45]

ausgerichtet. Die Erkennung des Gesichtes erfolgt dabei durch die Microsoft Kinect. Somit ist es möglich, dass der Nutzer seine Position ändert und die Blickerkennung trotzdem funktioniert. Das Augenerkennungssystem nutzt als Basis die Kamera von [94], erweitert diese aber um einen Zoom und eine Verstärkung des Infrarotlichtes um eine höhere Reichweite zu erreichen. Abb. 18 zeigt den Schwenk- und Neigungsmechanismus und den Aufbau des Gesamtsystems. In einer durchgeführten Studie wurde ein Fernseher aus ca. 3 Metern Entfernung gesteuert (Lauter/Leiser, Sender wechseln). Die Autoren vergleichen hierbei die

alleinige direkte Blicksteuerung mit einer Kombination aus Blick und Steuerung per Knopfdruck. Die Nutzer präferierten jedoch die Steuerung ohne weiteres Eingabegerät.

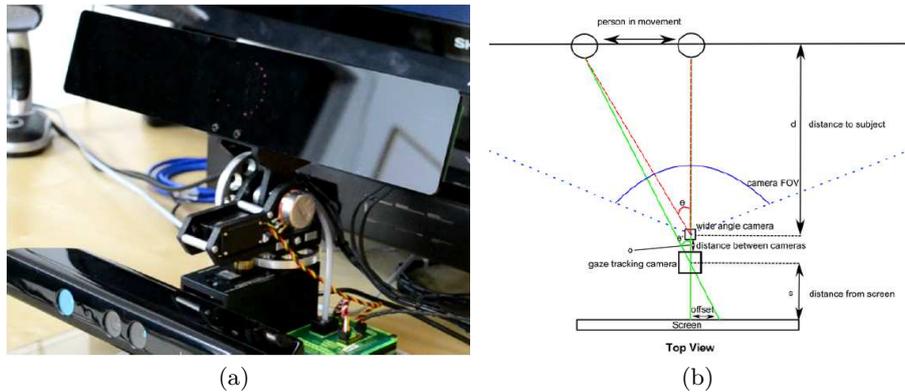


Abb. 18. a) Schwenk- und Neigungsmechanismus und b) Aufbau des Systems aus [46]

In [47] stellen die Autoren eine Blicksteuerung für bettlägerige Personen vor, die eine Interaktionen an einem großen projizierten Display ermöglicht. Abb. 19 zeigt den Aufbau des Systems. Die Blickerkennung wird mit einer Webcam und einem PC mit Open-Source Software (ITU Gaze Tracker) durchgeführt. Zur Evaluierung des Systems wurde eine Studie durchgeführt: Die Genauigkeit war zwar schlechter als in üblichen kommerziellen Systemen, jedoch wurde die Erkennung nicht mit einer High-End-Kamera durchgeführt und der Abstand zur Kamera war mit ca. zwei Metern deutlich größer. Ein signifikante Verbesserung wurde mit der Sitzposition erreicht und die Erkennungsrate war im unteren Bereich des Bildschirms deutlich besser als im oberen. Vorteil des vorgestellten Systems ist der platzsparende Aufbau und die Möglichkeit umstehende Personen am Geschehen teilhaben zu lassen.

5.3 Steuerung mit HMET

Head-Mounted Eye Trackers (HMET) sind Systeme welche am Kopf getragen werden und die Blickerkennung dauerhaft durchführen können. Diese Systeme haben meist zwei Kameras: eine für die Blickerkennung und eine für die Umgebungserkennung. Vorteil ist, dass der Nutzer sehr mobil ist. Erkauft wird dieser Vorteil durch den erhöhten Tragaufwand der Apparatur.

Mardanbegi et. al. stellen in [48] ein System vor, mit welchem man in der Lage ist, mit beliebig vielen unterschiedlichen Flachbildschirmen (oder anderen flachen Displays) per Blicksteuerung zu interagieren. Abb. 20 zeigt die Vorrichtung der Autoren und das Einsatzszenario: Der Nutzer ist in der Lage mit dem

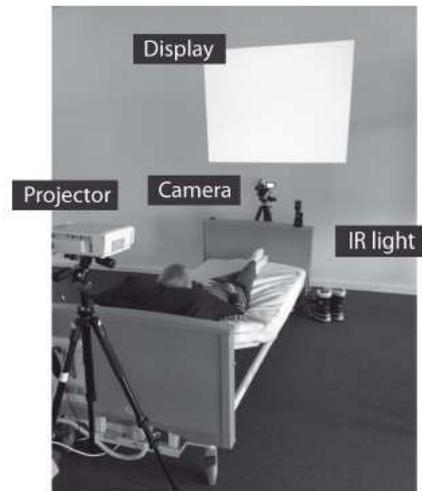


Abb. 19. Aufbau des Blicksteuerungssystems aus [47]: Ein Projektor zur Erzeugung des Displays, eine Kamera zur Blickerkennung und ein Infrarot-Licht

Smartphone, dem Laptop und dem Fernseher zu interagieren. Die Bildschirme sind via Netzwerk mit einem Server verbunden. Die Autoren erläutern weiterhin die Erkennung der unterschiedlichen Bildschirme und zeigen wie der Nutzer mit seiner Umgebung (Ventilator, Tür, Radio) interagieren kann (siehe Abb. 21). Die Autoren merken an, dass die Erkennung der Bildschirme noch nicht einwandfrei funktioniert.



Abb. 20. Die Vorrichtung der Autoren und das Einsatzszenario aus [48]

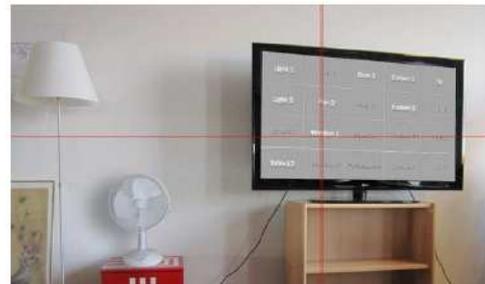


Abb. 21. Der Nutzer wählt ein Feld und blinzelt doppelt um die Auswahl zu treffen aus [48]

Attention Responsive Technology (ART) [49] ist ein System für die direkte Interaktion mit Appliances. Dabei wird ein HMET-System verwendet und die

zu steuernden Umgebungsobjekte werden mit dem SIFT-Algorithmus in echtzeit erkannt. Die Objekte wurden im vorherein in einer Datenbank gespeichert und per Netzwerk miteinander verbunden. Hat der Algorihtmus ein Objekt erkannt, so wird dieses automatisch in einem Interface angezeigt und kann damit an-/ausgeschaltet werden. Im Artikel ist hierzu ein Interface wie ein Smartphone/Tablet beschrieben. Meines Erachtens nach könnte man diesen Mechanismus aber auch z.B. durch *blink* oder durch eine *dwell time* erreichen.

6 Steuerung per Berührung

Die Touchsteuerung (Berührungssteuerung) bietet eine einfache und intuitive Art der Bedienung. Auch im Umfeld zur Steuerung von intelligenten Objekten kann die Touchsteuerung eingesetzt werden. Im folgenden werden zuerst Studien zur Touchsteuerung vorgestellt, danach werden einige Systeme zur Steuerung von Appliances mit Hilfe von Touchscreen (Tablets/Smartphones) vorgestellt. Im Anschluss wird ein Artikel erläutert, in dem ein virtuelles Display auf eine Oberfläche projiziert und damit ein Fernseher gesteuert wird.

6.1 Studien zur Touchsteuerung

Culen et. al. [50] fassen in ihrer Arbeit mehrere selbst durchgeführte Studien zusammen. Ziel war es, herauszufinden, ob und wie ältere Leute mit Touch-Interfaces zurechtkommen. Es sollten Herausforderungen und Möglichkeiten für das zukünftige Design für diese Interfaces in Bezug auf ältere Menschen identifiziert werden. So haben viele ältere Menschen Probleme die Symbole auf den kleinen Bildschirmen zu erkennen und es fehlt das Feingefühl einen kleinen Bereich auf dem Bildschirm zu drücken. Die Autoren kommen zum Ergebnis, dass das Interface individuell anpassbar sein muss, denn die Fähigkeiten älterer Leute können sich relativ schnell verändern. Weiterhin sollten die Nutzer mit in den Design-Prozess einbezogen werden, insbesondere wenn es um körperlich beeinträchtigte Menschen geht.

In [51] untersuchen die Autoren in einem Experiment die Perfomance von drei Eingabegeräte: (1) Finger, (2) Stylus und (3) eine klassische Maus (als Richtwert). Dazu mussten die Teilnehmer drei verschiedene Aktionen durchführen: (1) Tapping, (2) Dragging und (3) Radial Dragging (auf Flächen zeigen die kreisförmig um den Cursor angeordnet sind). Die Perfomance variiert je nach Aktion: Der Finger ist für das Tapping am schnellsten, aber relativ schlecht beim Dragging. Der Stylus ist sehr schnell beim Dragging.

Wobbrock et. al. [52] untersuchte in einer Studie Touch-Gesten auf Oberflächen, indem den Teilnehmern die Wirkung der Geste erklärt wurde und diese dann eine für sie passende Geste ausführen sollten. Wobbrock fand z.B. heraus, dass für die Teilnehmer die Anzahl der benutzten Finger nur eine kleine Rolle spielt und eine Hand zur Steuerung gegenüber zwei Händen bevorzugt wird. Er stellt weiterhin eine Reihe von Gesten für Aktionen vor, die für die Teilnehmer besonders leicht und intuitiv waren.

6.2 Steuerung mit Touchscreens

CRISTAL (Control of Remotely Interfaced Systems using Touch-based Actions in Living spaces) [53] ist ein Interface zur Kontrolle von Appliances via Touch (Abb. 22). Die Idee ist die Einbindung mehrere Nutzer in die Steuerung. Der Touchscreen ist gleichzeitig Interface und Tisch und zeigt die Umgebung von oben durch eine an der Decke befestigte Kamera. Dadurch erhält man ein Live-Bild, welches als Interface dient und mit dem es möglich ist, verschiedene Appliances wie Fernseher, Licht oder den Staubsauger zu steuern und ein direktes Feedback zu erhalten (z.B. Licht heller/dunkler stellen). Abb. 23 zeigt die HW-Architektur: Ein Computer steuert die Appliances, die über Firewire, USB oder Bluetooth mit diesem verbunden sind. In einer Studie wurde das System als intuitiv und als leicht bedienbar bewertet. Der Platzbedarf ist jedoch enorm, gerade im Umfeld eines Wohnzimmers.



Abb. 22. Laborraum für CRISTAL aus [53]

HouseGenie [54] (Abb. 24) ermöglicht die Steuerung und Überwachung von Appliances via Smartphone. Dazu wird auf dem Smartphone ein 2D-Ansicht der Wohnung gezeigt, auf welchem man die Geräte per Klick oder Drag&Drop direkt auswählen und steuern kann. Per Klick wird ein Gerät an- oder ausgeschaltet, per Drag&Drop eine Interaktion zwischen Geräten initiiert: (1) display sharing (Inhalt des einen Displays wird auf das andere gespiegelt) und (2) content-to-device: Dateien werden von einem Gerät zum anderen übertragen. Weiterhin unterstützt das System auch Sprachsteuerung.

Ein großer Nachteil der bisherigen Methoden ist die Einbindung neuer Appliances in die Umgebung, da dadurch oft eine Anpassung der Software nötig war. In [55] wird ein System zur automatisierten Einbindung neue Geräte vorgestellt. Diese werden automatisch erkannt und in das Interface eingebunden. Das System unterstützt gängige Standards wie Zigbee oder DMX-Controller und Standard-Protokolle. Für jede Appliance wird ein sog. Widget auf der GUI abgelegt. Über

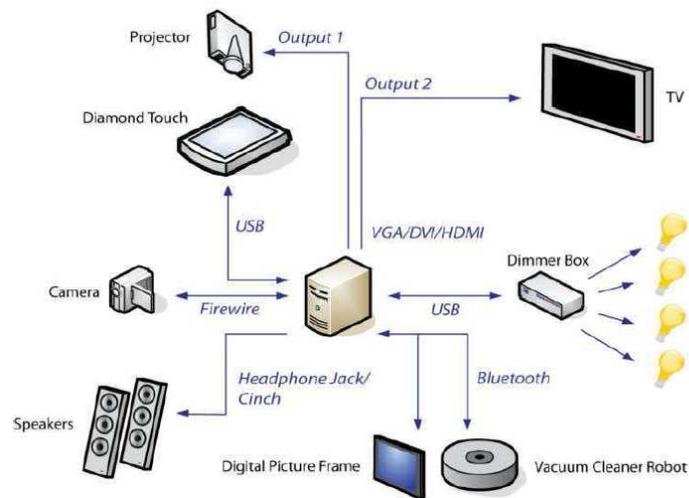


Abb. 23. Architektur von CRISTAL aus [53]



Abb. 24. Steuerung der Appliance mit *HouseGenie* [54]

dieses Widget kann die Appliance gesteuert werden. Es können auch Profile angelegt werden um verschiedene Einstellungen zu speichern. Das System wurde sowohl für große Touchscreens, als auch für Smartphones entwickelt und in einer Studie evaluiert.

6.3 Steuerung an Projektionen auf Oberflächen

Ein andere Ansatz ist in [56] zu finden. Der Nutzer kontrolliert die Appliances durch ein User-Interfaces, das auf eine Oberfläche (z.B. Couchtisch) projiziert wird. Somit ist es möglich auf dem projizierten Interface per Bewegung und Tippen der Finger die Appliances steuern. Abb. 25 zeigt den Überblick der Komponenten. Ein Projektor und eine Kamera sind an der Decke befestigt. Der Projektor projiziert die virtuelle Tastatur auf die Oberfläche, während die Kamera die Bewegung der Finger verfolgt. Die Bewegung und das Tippen der Finger

wird per Bewegungserkennung erkannt. In einem Experiment wurde einem modifizierten Fernseher [95] mit dem virtuellen Display bedient. Die Komponenten waren dabei an einem handelsüblichen Notebook angeschlossen, welcher auch die Gestenerkennung berechnet. Das System unterstützt auch komplizierte Touchgesten wie Drag&Drop oder rotierende Bewegungen.

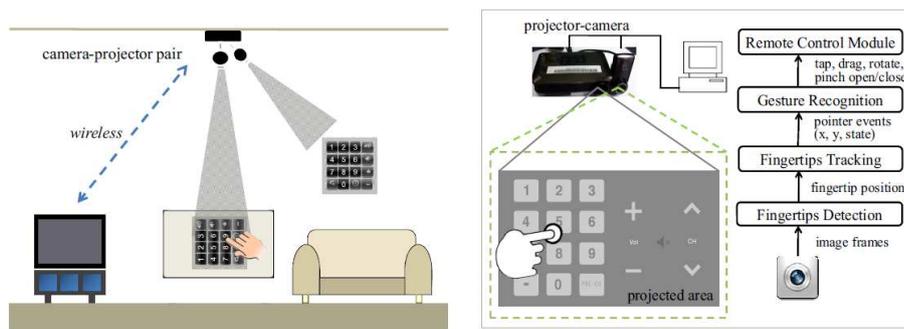


Abb. 25. Überblick des Systems aus [56]

7 Steuerung per Tangible-User-Interfaces

In *Tangiblen User Interfaces (TUI)* werden physische Objekte zur Steuerung von z.B. Information oder Appliances verwendet. Hornecker definiert TUI folgendermaßen: „Mit Tangible Interfaces agieren wir in unserer materiellen Umgebung und berühren das Interface selber“ [96]. Tangible Objekte stellen dabei nicht nur Input- sondern auch Outputmöglichkeiten dar. Die Form und räumliche Position kann so den Zustand eines Systems beeinflussen.

7.1 Studien zu Tangiblen User Interfaces

In [57] vergleichen Drossis et. al. die Interaktion von Touch, Gesten (Erkennung durch Microsoft Kinect) und tangiblen Interfaces miteinander (Versuchsaufbau siehe Abb. 26). Das Ziel war es einerseits herauszufinden bei welchen Eingabemethoden die Nutzer erfolgreicher eine Aufgabe lösen konnten und andererseits wie die Nutzer die Eingabemethoden allgemein bewerten. Die Teilnehmer mussten dazu verschiedene Aufgaben mit den Eingabemethoden lösen und sollten dabei ihre Gedanken laut äußern (Think-Aloud Methode). Als Ergebnis halten die Autoren fest, dass es auf den Kontext ankommt, welche Methode dies bessere ist: Touchsteuerung bietet eine gute Methode zur alltäglichen Steuerung der Umgebungsobjekte. Obwohl Gestensteuerung sehr ermüdend war, so bereitete diese

Steuerung den Teilnehmern viel Spaß, so dass Gestensteuerung im Kontext von Entertainment eingesetzt werden sollte. Tangible Steuerung führte zu gemischten Reaktionen, da die Steuerung komplexer Sachverhalte nur schwer möglich war.



Abb. 26. Aufbau der Interfaces: links die Gestensteuerung; rechts sowohl die Touch, als auch die Steuerung mit dem tangiblen Interface [57]

Contanza et. al. [58] evaluieren in ihrer Arbeit zwei tangible Interfaces, die sie zum Download im Internet bereit gestellt hatten. Die Interfaces konnten benutzt werden um damit Musik zu komponieren. Sie bestanden größtenteils aus Papiervorlagen, die sich die Nutzer selbst ausdrucken konnten. Die Erkennung der Lage der Objekte wurde durch eine einfache Webcam des Nutzers durchgeführt. (siehe Abb. 27. Sie untersuchten durch Logfiles, sowie durch die von Nutzer generierte Bilder und Musik, wie sie damit alltäglich umgehen. Der Artikel zeigt auch, dass man Studien über das Internet durchführen kann und dadurch Feedback erhält. Nachteil dieser Methode ist allerdings die Adressierung der Zielgruppe, denn nur stark interessierte Nutzer werden tatsächlich damit interagieren.

7.2 Steuerung Tangibles-User-Interface

In [59] wird ein Würfel zur Steuerung der Lautstärke eines PCs und zur Farbauswahl und Leuchtstärke einer Lampe verwendet. Der Würfel ist mit Lage- und Beschleunigungssensoren ausgestattet. Ist eine Lage mit einer Funktion verbunden, so wird dies durch farbige LEDs auf der entsprechenden Seite angezeigt. Durch Kipp- und Drehbewegung kann dann eine Aktion ausgelöst werden.

In [60] entwickeln die Autoren ein tangibles User-Interface, welche es ermöglicht verschiedene Appliances per WiFi oder Bluetooth miteinander zu verbinden. Das ermöglicht die einfache Kontrolle über Verbindung zwischen z.B. einem



Abb. 27. Aufbau des tangiblen Interfaces aus [58]

tragbaren MP3-Playern und der Stereoanlage (Die Autoren nennen dies Semantic Connections). Abb. 28 zeigt den *Interaction Tile*: Jede Appliance wird durch ein physisches Objekt dargestellt. Eine zentralisierte Steuereinheit fungiert als Vermittler zwischen den Appliances. Zum Herstellen einer Verbindung werden die kleinen Objekte an die zentrale Steuereinheit gekoppelt. Diese prüft ob Verbindungen überhaupt möglich sind und gibt via Lichtsignal Feedback über die Verbindung. Die Autoren haben auch ein Video veröffentlicht². Die einfache Methode Verbindungen zwischen verschiedenen Objekten und die entsprechende Visualisierung ist sicherlich ein Vorteil dieses Interfaces. Der Nachteil ist aber, dass jede Appliance durch ein physisches Objekt dargestellt werden muss.

In [61] untersuchen die Autoren den oben genannte *Interaction Tile* als zentralisierter Ansatz gegenüber sog. *Interaction Tabs* (Abb. 29) als dezentralen Ansatz. Die Idee ist auch bei den Interaction Tabs eine Verbindung zwischen unterschiedlichen Appliance auf einfache Art und Weise zu ermöglichen, indem man die Tabs miteinander verbindet. Eine Studie zeigte, dass der dezentrale Ansatz über die Interaction Tabs besser Abschnitt.

In [62] entwickeln und vergleichen die Autoren ein tangibles Interface und einen Augmented Reality Ansatz für die Verbindung von zwei Geräten und vergleichen diese Ansätze. In dem Augmented-Reality Ansatz signalisieren sichtbare

² <http://www.youtube.com/watch?v=vdZcjqfq8RQ>



Abb. 28. Interaction Tile aus [60]



Abb. 29. Interaction Tabs [61]

Linien (realisiert durch einen Projektor) eine Verbindungen zwischen den Geräten. Beim tangiblen Interface können Appliances markiert und gescannt werden um herauszufinden ob Verbindungen möglich sind, sowie um Verbindungen aufzubauen.

TANGerINE [63] ist ein tangibles Tischinterface, auf den man durch Benutzung eines Würfels [97], digitale Informationen angezeigt bekommt (siehe Abb. 30). Die Nutzer können mit dem Würfel auf dem Tisch (Active Context) oder in der näheren Umgebung interagieren (Nearby Context). Die Nutzer können den Würfel auf verschiedene Seite drehen und damit verschiedene Aktionen auszulösen. Mit Hilfe von Beschleunigungssensoren, verschiedenen LEDs an jeder Seite und einer Kamera an der Decke kann errechnet werden, welche Seite des Würfels nach oben zeigt. Die Autoren beschreiben ein Experiment: Ein Quiz, bei welchem der Würfel durch Drehung auf eine der sechs Seiten im NearbyContext zur Auswahl eines der Themen benutzt wird. Zur Beantwortung kann der Nutzer den Würfel dann auf den Tisch legen. Bei einer richtigen Antwort wird weitere Inhalt auf den Tisch (Active Context) mit Hilfe eines Projektor projiziert.



Abb. 30. TANGerINE aus [63]

8 Multimodale Interaktion

Ein Multimodales System nutzt mehrere Modalitäten als Input (z.B. Geste, Sprache oder Blick) zur Steuerung von z.B. Appliances. Als Input reicht es aber nicht aus auf einem Knopf zu drücken wie bei einem herkömmlichen PC via Maus und Tastatur. Vielmehr muss der Input eine Bedeutung haben (vgl. Definition einer Geste aus Kapitel 3). Basierend auf dieser Idee werden im folgenden zwei Artikel zum Design von multimodalen Interfaces, sowie mehrere Artikel zu multimodalen Steuerungskonzepten vorgestellt.

8.1 Design zu Multimodalen Interfaces

Um sich bei den Nutzern zu etablieren, sollte die Interaktion mit Interfaces in intelligenten Umgebungen einfach und intuitiv sein. Fernandez et. al. [98] schlagen in ihrer Arbeit eine Architektur für eine multimodales Interface vor, welches zwei Ideen verfolgt: (1) Die Nutzung des Model-View-Presenter Konzeptes und (2) die Nutzung des Service-Orientierten Ansatzes. Die Autoren implementieren mit der OSGi Service Plattform einen Prototypen ihres multimodalen Interfaces und führen eine Studie durch.

In [99] führen Sebe et. al. eine Literaturanalyse zu bestehenden Konzepten sowie zum Design von multimodalen Interfaces durch und identifizieren Schlüsselaspekte. Die Autoren kommen zu dem Schluss, dass für das Design eines multimodales Interface ein nutzerorientierter Ansatz zu benutzen sei und das multimodale Interfaces eine der Technologien sind, mit denen Nutzer in naher Zukunft mit Appliances interagieren.

8.2 Kombination von Gesten- und Sprachsteuerung

In [64] haben Carrino et. al. Zeige- und symbolische Geste mit Sprache kombiniert. Durch die Zeigegeste wird ein Gegenstand ausgewählt. Durch die Symbolische Geste oder durch Sprache wird mit dem Gegenstand interagiert (siehe Abb. 31). Die Autoren haben einen Prototypen entwickelt: eine Kamera gekoppelt mit einer WiiMote. Die Auswahl der Appliance (durch Zeigegeste) wird durch die Kamera mit Hilfe von Parallel Tracking and Multiple Mapping (PTAMM) realisiert. Die Aktivierung der Appliance (durch eine symbolisch Geste) erfolgt mit Hilfe der Sensoren in der WiiMote. Zur Spracherkennung wurde die Microsoft Speech API genutzt. Die Autoren haben ihren Prototypen in einer kleinen Studie evaluiert. Das System hat mit 95% eine sehr hohe Genauigkeit erzielt und die Teilnehmer bewerteten es als sehr gut oder gut.

Multimodal Interface for Smart Objects (MISO) [65] nutzt eine Kombination aus Zeigegesten und Fingerschnipsen zur Auswahl einer Appliance. Durch eine vordefinierte Geste (z.B. eine Kreisbewegung) wird eine Aktion bei der entsprechenden Appliance ausgelöst. Die Erkennung der Gesten erfolgt durch die Microsoft Kinect. Für die Erkennung des Fingerschnipsen haben die Autoren das System von Vesa et. al [100] modifiziert. *MISO* benötigt kein weiteres Eingabegerät zur Steuerung der Appliances. Jedoch merken die Autoren an, dass

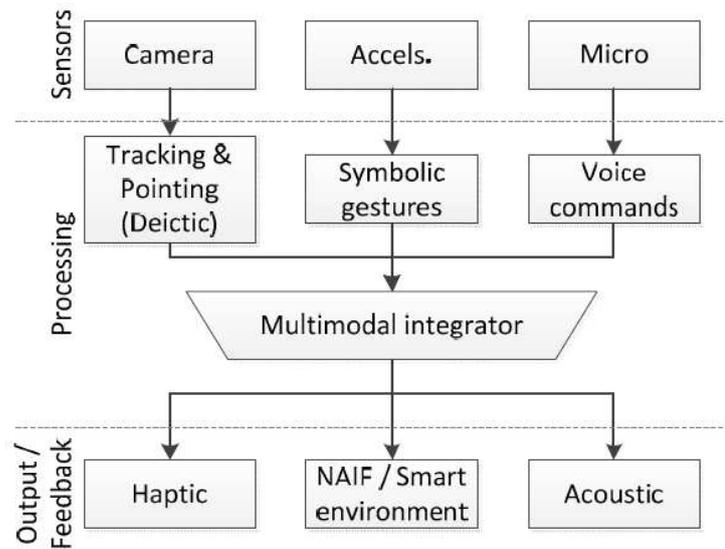


Abb. 31. Konzept des Systems aus [64]

das System gerade für die Erkennung des Fingerschnipsen eine große Anzahl an Testsamples benötigt. Weiterhin kann es zu einem Auswahlproblem kommen, wenn mehrere Appliances dicht nebeneinander stehen.



Abb. 32. Konzept des Systems aus [65]: Die Auswahl erfolgt per Zeigegeste, die Bestätigung per Geräusch

8.3 Kombination von Gesten- und Blicksteuerung

Leet et. al. [66] kombinieren Gesten- und Blicksteuerung um ein neues User-Interface zu schaffen. Die Blicksteuerung wird hierbei zur Fixierung eines Punktes benutzt, während eine Geste zur Aktionsauswahl dient. Für die Blickerken-

nung kombinieren die Autoren zwei Weitwinkel und zwei Kameras mit Zoomobjektiven um eine optimale Erkennung zu gewährleisten. Für die Gestenerkennung wird die Microsoft Kinect verwendet. Die Autoren zeigen das Interface u.a. in einem Ego-Shooter, indem man sich umschaun und einen Gegner fixieren kann. Führt man dann die Geste für „schießen“ aus, so wird ein Schuss ausgelöst. Man kann weiterhin die Umgebungskarte per Multi-Touch Geste vergrößern. Das zweite Beispiel ist die Steuerung eines interaktiven Fernsehgerätes (nur mit Blicksteuerung). Abb. 33 zeigt das Zusammenspiel der Steuerungsarten. Ein ähnliches Konzept wird von Hales et. al. [67] verwendet, die zur Blick- und Umgebungserkennung ein am Kopf befestigtes System verwenden, zur Gestenerkennung eine einfache Kamera, die die Umgebung aufnimmt. Sowohl die Gestenerkennung als auch die Blickerkennung werden mit Open-Source Software durchgeführt. Die Autoren testen ihr System u.a. an einem Roboter: Hier konnte man durch das Anschauen den Roboter auswählen. Durch Gesten die Fahrriichtung bestimmen und durch das Hochhalten von mehreren Fingern die Geschwindigkeit.

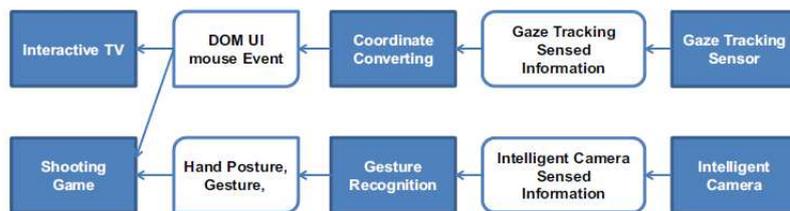


Abb. 33. Zusammenspiel der Gesten- und Blicksteuerung aus [66]

Ein weitere Kombination von Gesten- und Blicksteuerung ist in [68] zu finden. Die Autoren kombinieren Kopfgesten mit einer Blicksteuerung um eine freihändige Interaktion zu ermöglichen. Die Kopfgesten werden dabei in Echtzeit durch ein am Kopf befestigtes Blickerkennungssystem erfasst. Abb. 34 zeigt die Idee: Bei einer Bewegung des Kopfes und gleichzeitiger Fixierung eines Punktes bewegen sich die Augen nach unten bzw. nach oben. Das gleiche zeigt sich bei Seitwärtsbewegungen oder Kombinationen von beiden. Mardanbegi et. al. testen ihr aufgebautes System u.a. an einem Interface, auf welchem ein Rezept angezeigt wird (siehe Abb. 35). Durch das Blicken den Bereich der Bilder (A) und gleichzeitige Kopfbewegung nach rechts oder links die Bilder wechseln. Im Bereich (B) kann man einen Musikplayer steuern, in Bereich (C) die Lautstärke ändern. Die Autoren planen diese Methode auch zur Kontrolle von Appliances einzusetzen.

8.4 Kombination von Blick- und Touchsteuerung

Stellmach et. al. [69] stellen in ihrer Arbeit für die Aktionsauswahl an weiter entfernten Display eine Blicksteuerung kombiniert mit einem tragbaren Touchscreen

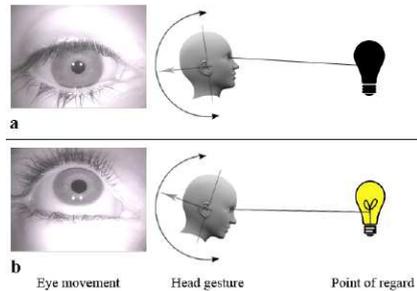


Abb. 34. Augenbewegung bei Fixierung eines Punktes und gleichzeitiger Bewegung des Kopfes aus [68]

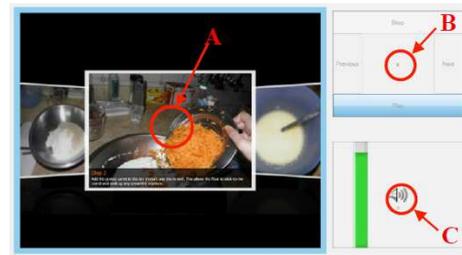


Abb. 35. Steuerung eines interaktiven Kochbuchs aus [68]

vor. Abb. 36 verdeutlicht das Konzept: Die Blicksteuerung dient zur Auswahl und Touchsteuerung zur Bestätigung dieser Auswahl. Die Autoren vergleichen verschiedene Konzepte zur Auswahl von diversen Aktionen auf dem Display, wie z.B. MAGIC touch, das die Steuerung des Cursor auch per Touch ermöglicht oder MAGIC tab, welches es ermöglicht auf dem Touchscreen durch eine Liste derjenigen Aktionen zu gehen, welche nah am Fokuspunkt der Blickrichtung liegt. Die Konzepte wurden in einer Studie evaluiert. Die Kombination von Blick- und Touchsteuerung umgeht das bereits vorgestellte Midas Touch-Problem.

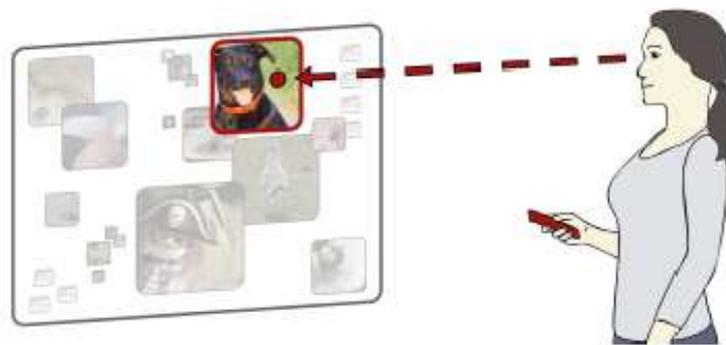


Abb. 36. Konzept der Blick- und Touchsteuerung: Die Auswahl erfolgt per Blick, die Bestätigung der Auswahl mit dem Eingabegerät [66]

8.5 Kombination von Sprach- und Touchsteuerung

In [70] stellen einen Ansatz zur Kombination von Sprach- und Touchsteuerung, welches für den Einsatz in einem Fahrzeug gedacht ist. Die Sprachsteuerung

wird zur Auswahl der Objekte verwendet, während die weiterführende Steuerung dieser Objekte dann per Touchgesten erfolgt. So kann man z.B. die Fenster per Sprachbefehl auswählen und diese dann per Touchsteuerung öffnen/schließen. In einem Experiment haben die Autoren ein Lenkrad mit einem Android-Tablet modifiziert um so ein Touch-Interface zu bieten. In einer ersten Studie wurde gezeigt, dass diese Art der Steuerung einfach und verständlich ist. Es wurden weiterhin erste Touchgesten für die Steuerung der Appliances gesammelt.

8.6 Kombination von Gesten- und Touchsteuerung

In *Point&Control* [71] wird ein Bedienkonzept vorgestellt, bei dem die Positions- und Richtungsbestimmung per Kinect durchgeführt wird, die Auswahlbestätigung der Appliance per Touchfunktion des Smartphones. Ein Server benutzt die Kinect um die Position und die Richtung zu bestimmen, in welche das Smartphone gezeigt wird. Dadurch kann per Kollisionsabfrage berechnet werden, welches Gerät angesprochen werden soll. Gibt es mehrere passende Geräte, so wird eine Liste der in Frage kommenden Geräte auf dem Smartphone angezeigt. Per Klick auf den Touchscreen des Smartphones wird das entsprechende Gerät aktiviert. Neue Appliances können beispielsweise per QR-Code registriert werden. Abb. 37 zeigt einen Überblick des Systems. Die Autoren haben auch ein Video veröffentlicht ³

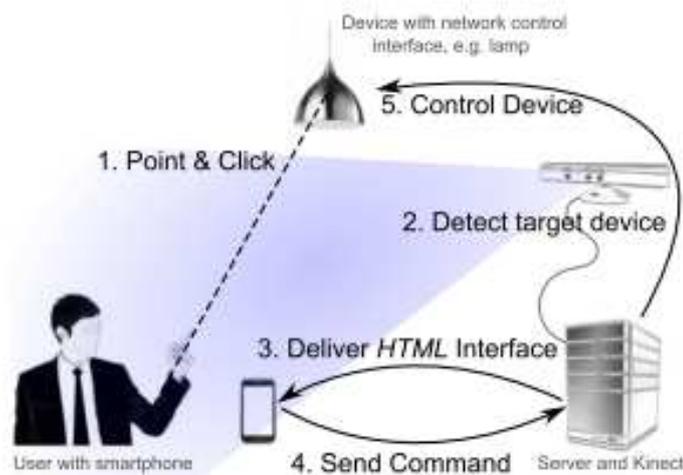


Abb. 37. Architektur von Point&Control aus [71]

³ <http://www.youtube.com/watch?v=RqlCYBIUMos>

8.7 Kombination von Gestensteuerung und TUI

Molyneaux et. al. [72] nutzt tangible Objekte als In- und Output Geräte. Mit Hilfe eines Projektors und einer Kamera werden Gegenstände erkannt und auf diese ein Interface projiziert. Dies erlaubt die Interaktion mit den Objekten und eine direkte Steuerung durch die Interfaces auf den Objekten. Molyneaux erschafft somit ein hybrides Interface, welches die Vorteile eines GUI mit den Vorteilen eines TUI verbindet. Sobald ein vordefiniertes Objekt in Reichweite der Kamera ist, gibt dieses ein Signal ab. Die Kamera erkennt das Objekt und der Projektor kann dann auf Wunsch ein Interface zur Verfügung stellen (siehe Abb. 38). Die Autoren führen ein Experiment durch, in welchem ein Photoalbum auf ein selbst entworfenes Objekt projiziert wird (siehe Abb. 39). Auf dem Objekt werden neben den Bildern auch kleine Buttons dargestellt, mit denen man nach rechts bzw. links blättern kann. Je nachdem ob das Buch geschlossen oder offen ist, werden unterschiedliche Funktionen ausgelöst: (1) ist das Buch geschlossen kann man eines von drei Fotoalben auswählen; (2) öffnet man nun das Buch, so kann man die Bilder des ausgewählten Albums anschauen. Die Erkennung, ob das Buch offen oder geschlossen ist, erfolgt durch vorher definierte 3D-Modelle.

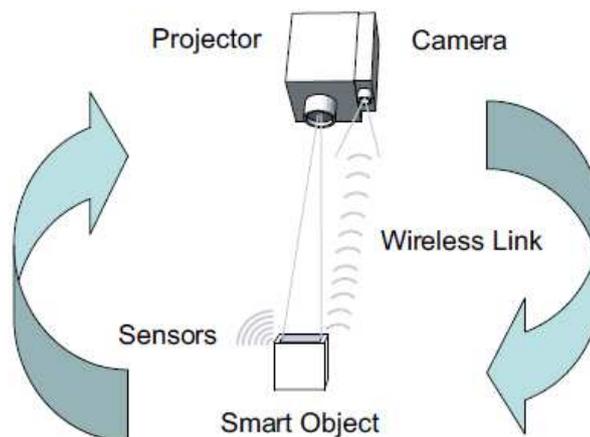


Abb. 38. Zusammenspiel der Komponenten aus [72]

8.8 Kombination von mehr als zwei Modalitäten

GeeAir [73] ist ein multimodales selbst entwickeltes Eingabegerät basierend auf einer Kombination der Nintendo WiiMote und dem Nunchuk-Controller (siehe Abb. 40). Es unterstützt die Auswahl und Steuerung von Appliances (u.a. Fernseher, Radio und Lampen) durch Sprache, Gesten, Knöpfe und einem Joystick. Die Auswahl der Appliance erfolgt dabei über den Joystick und Knopfdruck



Abb. 39. Links: Geschlossenes Buch und Auswahl des Fotoalbumes; Rechts: Offenes Buch und Auswahl der Fotos

oder durch Sprache. Eine Auswahlbestätigung erfolgt per Licht- oder Audiofeedback. Per vordefinierter Geste kann dann eine Appliance gesteuert werden. Die Erkennung der Gesten erfolgt durch den FDSVM-Algorithmus über eingebaute Bewegungssensoren. Vorteil von *GeeAir* ist, dass es für Sprach-, oder sehbehinderte Menschen eine Möglichkeit zur Steuerung von Appliances bietet und das der modifizierte Controller auch bei schlechten Lichtverhältnissen funktioniert.

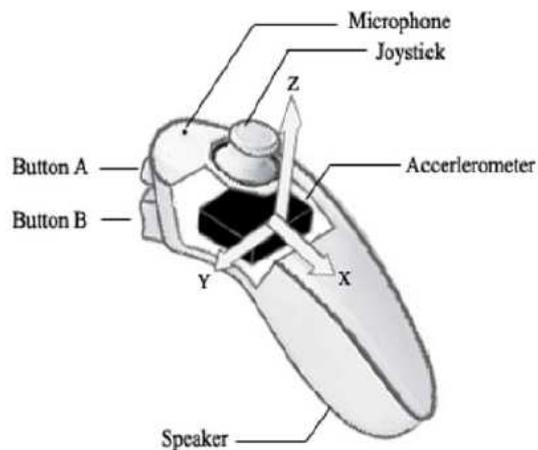


Abb. 40. Prototyp des Controllers aus [73]

9 Zusammenfassung und Ausblick

In diesem Kapitel wird eine Zusammenfassung gegeben und ein Fazit gezogen. Weiterhin gibt es einen Ausblick auf zukünftige Forschungen und Möglichkeiten, wie z.B. mit der neuen Microsoft Kinect 2.0.

9.1 Zusammenfassung

In den letzten Jahrzehnten hat die Computertechnologie einen Wandel erlebt. Von ersten Großrechnern in den 60er Jahren, zum Personal Computer in den 80er Jahren bis hin zu Tablets und Smartphones in der heutigen Zeit schreitet die Miniaturisierung voran. Die moderne Technologie ermöglicht heutzutage die Modifizierung von Alltagsgegenständen, sog. Appliances. Dabei ist die Bedienung dieser Appliances ein wesentlicher Bestandteil und entscheidet über Erfolg und Misserfolg der neuen Technologien. Komfortable, leicht erlernbare und intuitive Interfaces rücken daher immer weiter in den Vordergrund und in den Fokus der Wissenschaft. Ein weiterer Aspekt ist der demografische Wandel, der in den industrialisierten Ländern weiter voranschreitet. Durch den damit einhergehenden Bedarf an jungen Pflege- und Hilfskräften, welcher nur schwer gedeckt werden kann, werden Technologien benötigt, um diesen älteren und bewegungseingeschränkten Menschen ein unabhängiges und autarkes Leben zu ermöglichen. Die Bedienung dieser neuen Technologien stellt einen essentiellen Bestandteil dar.

In den letzten Jahren sind eine Vielzahl neuer Konzepte für Interfaces entstanden. In dieser Arbeit wurde daher ein Literaturüberblick über verschiedene Interfaces erstellt. Darin enthalten sind Interfaces zu Gesten-, Sprach- und Blicksteuerung, sowie die Steuerung per Berührung (Touch) und Tangible-User-Interfaces, sowie multimodale Interfaces. Es wurden interessante Aspekte (aus Sicht des Autors) der letzten Jahre gesammelt und zusammengefasst. Die Zusammenfassungen der Artikel sollen einen Eindruck der vorgestellten Konzepte inkl. deren Vor- und Nachteile vermitteln, sowie die benutzte Hardware aufzeigen. Für eine detaillierte Sicht, vor allem über die verwendeten Algorithmen oder die genaue Methodik bei durchgeführten Studien wird auf den jeweiligen Artikel verwiesen.

Es ist festzuhalten, dass jede Modalität über Vor- und Nachteile verfügt und nicht jede Modalität für jeden Typ Mensch, insbesondere im Hinblick auf körperlich beeinträchtigte Menschen, geeignet ist.

So kann die Gestensteuerung auf Dauer sehr ermüdend und die Gestenerkennung bei schlechten Lichtverhältnissen problematisch sein. Auch gibt es noch Auswahlprobleme bei dicht nebeneinanderstehenden Objekten (Unterscheidung zwischen Objekten schwierig), sowie offene Fragen bei der Auswahl der „richtigen“ Gesten für bestimmte Aktionen. Die Gestensteuerung stellt sich aber als sehr intuitiv und leicht erlernbar dar und bei Einsatz von Tiefenkameras ist beispielsweise kein weiteres Eingabegerät notwendig.

Sprachsteuerung ist ein sehr natürlicher, intuitiver und bequemer Weg zur Kommunikation, verwenden wir diese doch tagtäglich um uns untereinander zu verständigen. Die logische Konsequenz ist also die Übertragung der Sprache zur Interaktion mit Appliances. Jedoch kommt es noch häufig zu Problemen bei der

Differenzierung zwischen Sprachbefehlen und Umgebungsgeräuschen und auch die Erkennung einzelner Wörter bei Verwendung von Synonymen und Dialekten ist schwierig. Weiterhin ist die kontextbezogene Steuerung aufgrund noch bestehender Probleme bei der Lokalisierung schwierig.

Blicksteuerung erweist sich mit der Fokussierung eines Objektes mit den Augen ebenso als sehr intuitiv. Viele Techniken verwenden aber noch am Kopf befestigte Vorrichtungen (HMET) zur genauen Erkennung der Augenbewegungen. Durch den Einsatz eines HMET erhält man zwar eine erhöhte Mobilität, erkauft wird diese aber durch einen erhöhten Trageaufwand. Die dauerhafte Augenbewegungen wirken zudem ermüdend und die Erkennung bei schlechten Lichtverhältnissen stellt wie bei der Gestensteuerung ein Problem dar. Weiterhin muss das Midas-Touch Problem (Problem der ungewollten Aktionsauslösung), beachtet werden. Lösungen hierfür existieren zwar durch die sog. Dwell-Time oder z.B. durch doppeltes Blinzeln, doch kann es hier trotzdem zu ungewollten Aktionen kommen.

Die Steuerung per Berührung z.B. mit Multi-Touch-Displays von Smartphones oder Tablets bieten die komfortable Bedienung eines GUI, was zu einer präzisen Auswahl der zu steuernden Geräte und einer einfachen Mehrbenutzersteuerung führt. Durch den Einsatz von kleinen mobilen Endgeräten ist man zudem sehr mobil. Nachteil hierbei ist aber die Notwendigkeit eines Endgerätes, sowie der erhöhte Platzbedarf bei Einsatz von großen Touchscreens.

Tangible-User-Interfaces ermöglichen die Steuerung durch Manipulation von physischen Objekten, was zu einer Visualisierung der aktuellen Systemzustände führt. Für einfache Sachverhalte stellen TUIs eine leichtes Bedienkonzept dar, komplexe Steuerbefehle lassen sich damit aber nur schwer umsetzen.

Multimodale Interfaces nutzen mehrere Modalitäten als Input zur Steuerung von Appliances. Somit kann man die Vorteile der Modalitäten verbinden um Synergieeffekte zu schaffen. Die intuitive Steuerung per Zeigegeste kann so z.B. mit Sprachbefehlen verbunden werden. Die so erhaltenen Vorteile machen eine einfache und bequeme Steuerung möglich. Derzeit mangelt es aber noch an Studien zur multimodalen Interfaces, z.B. welche Kombinationen der Modalitäten in einem bestimmten Kontext von den Nutzern als gut bewertet und angenommen wird.

9.2 Ausblick

Ambient Intelligence steckt im Vergleich zu klassischen Forschungsgebieten (wie Physik oder Mathematik) noch in den Kinderschuhen. Die vorgestellten Techniken funktionieren zwar unter Laborbedingungen, sind aber von einer Alltagstauglichkeit teilweise noch weit entfernt. Probleme wie die zuverlässige Erkennung von Sprachbefehlen bei lauten Umgebungsgeräuschen, die Erkennung von Augenbewegungen auf große Entfernung oder die Gestensteuerung bei schlechten Lichtverhältnissen sind nur im Ansatz gelöst. Die immer weitergehende Entwicklung besserer und günstiger Technik hilft diese Probleme anzugehen. So hat z.B. Microsoft mit der Kinect 2.0 (Marktstart November 2013) ein günstiges

System zur Erkennung von Gesten weiter entwickelt. Bereits der Vorgänger erfreute sich auch unter Wissenschaftlern großer Beliebtheit. Mit der Kinect gibt es ein System zur Gestenerkennung, welches günstig und mobil einsetzbar ist. Mit der Kinect 2.0 entwickelt Microsoft dieses System weiter. Die Kinect erkennt gleichzeitig bis zu 6 Personen und deren Bewegungen, was z.B. zu einer Mehrbenutzersteuerung führen kann. Auch ist die Gestenerkennung genauer: So können z.B. Schulter- und Drehbewegungen erkannt werden, welches eine feingranulare Gestenerkennung ermöglicht. Durch die Integration einer verbesserten Infrarotkamera und der „Time-of-Flight“ Technologie kann mit der Kinect 2.0 auch bei schlechten Lichtverhältnissen eine Gestenerkennung von mehreren Personen durchgeführt werden. Weiterhin kann die Kinect 2.0 Sprachbefehle einzelner Personen zuordnen, was eine kontextbezogene Spracherkennung ermöglicht. Durch den größeren Erkennungswinkel der Kamera können Menschen aus näherer Distanz erkannt werden, was zum einen zu einer besseren Raumausnutzung führt und zum anderen die Gestenerkennung in kleineren Wohnräumen erlaubt. Durch die höhere Auflösung der Kamera ist es zudem möglich, Veränderungen der Hautfarbe im Gesicht, sowie Gesichtsausdrücke wahrzunehmen und somit auf den emotionalen Zustand der Person zu reagieren. So wäre es z.B. denkbar automatisch entsprechende Musik abzuspielen oder die Lichtstärke zu verändern. Kinect 2.0 erlaubt somit einen vielfältigeren Einsatz als die Vorgängerversion und liefert Technologien um einige der oben genannten Probleme zu lösen.

Doch was passiert, wenn diese technischen Probleme gelöst sind? Eine von vielen Möglichkeiten beschreibt Böhlen [101]. Er spricht von einer „second order ambient intelligence“ und formuliert einige Hypothesen, welche Richtung Ambient Intelligence einschlagen könnte und vor allem sollte.

Nach Meinung des Autors dieser Seminararbeit liegt kurz- und mittelfristig die Lösung vieler Probleme im Design von multimodalen Interfaces. Nicht jede Modalität ist für jeden Menschen gleich gut geeignet. So besitzen z.B. Menschen mit körperlichen Behinderungen nicht die Möglichkeit durch bestimmte Modalitäten zu interagieren. Mit multimodalen Interfaces kann man einige dieser Probleme lösen und erhält durch die Kombination der Vorteile der einzelnen Interaktionsarten ein leistungsstarkes, intuitives und adaptives Interface. Noch ist aber offen, wie diese Interfaces gestaltet werden sollen. Sollten Interfaces geschaffen werden, die dem Nutzer die Wahl zwischen einzelnen Modalitäten lassen oder sollten die Modalitäten kombiniert werden? Dann stellt sich konsequenterweise die Frage, welche Modalitäten kombiniert werden sollen, um ein möglichst effizientes Interface für bestimmte Aufgaben zu schaffen? Welche Kombinationen sind für welchen Typ Mensch gut oder schlecht? Es gibt bisher nur wenige Artikel, die sich mit dem Design von multimodalen Interfaces beschäftigen. Diese offenen Fragen sollten daher in weiteren Studien evaluiert werden. Dabei sollte der Grundsatz zum Design von Interfaces beachtet werden: Interfaces müssen sich dem Menschen anpassen, nicht der Mensch dem Interface.

Literatur

1. Cook, D.J., Augusto, J.C., Jakkula, V.R.: Ambient intelligence: Technologies, applications, and opportunities. *Pervasive and Mobile Computing* **5**(4) (2009) 277–298
2. Deutschland, S.B.: 12. koordinierte bevoelkerungsvorausberechnung: Bevoelkerung deutschlands bis 2060 (2011) <https://www.destatis.de/>.
3. Weiser, M.: The computer for the twenty-first century. *Scientific American* **165** (1991) 94–104
4. Weiser, M.: Hot topics-ubiquitous computing. *Computer* **26**(10) (1993) 71–72
5. : Duden – Deutsches Universalwoerterbuch. 6 edn. Bibliographisches Institut, Mannheim (October 2006)
6. Blattner, M.M., Glinert, E.P.: Multimodal integration. *Multimedia, IEEE* **3**(4) (1996) 14–24
7. Wobbrock, J.O., Wilson, A.D., Li, Y.: Gestures without libraries, toolkits or training: a \$1 recognizer for user interface prototypes. In: *Proceedings of the 20th annual ACM symposium on User interface software and technology, ACM* (2007) 159–168
8. Kratz, S., Rohs, M.: A \$3 gesture recognizer: simple gesture recognition for devices equipped with 3d acceleration sensors. In: *Proceedings of the 15th international conference on Intelligent user interfaces, ACM* (2010) 341–344
9. Vatavu, R.D., Chera, C.M., Tsai, W.T.: Gesture profile for web services: an event-driven architecture to support gestural interfaces for smart environments. In: *Ambient Intelligence. Springer* (2012) 161–176
10. Vatavu, R.D.: A comparative study of user-defined handheld vs. freehand gestures for home entertainment environments. *Journal of Ambient Intelligence and Smart Environments* **5**(2) (2013) 187–211
11. Kühnel, C., Westermann, T., Hemmert, F., , S., Müller, A., Möller, S.: I’m home: Defining and evaluating a gesture set for smart-home control. *International Journal of Human-Computer Studies* **69**(11) (2011) 693–704
12. Camp, F., Schick, A., Stiefelhagen, R.: How to click in mid-air. In: Streitz, N., Stephanidis, C., eds.: *Distributed, Ambient, and Pervasive Interactions. Volume 8028 of Lecture Notes in Computer Science. Springer Berlin Heidelberg* (2013) 78–86
13. Caon, M., Yue, Y., Tscherrig, J., Mugellini, E., Abou Khaled, O.: Context-aware 3d gesture interaction based on multiple kinects. In: *AMBIENT 2011, The First International Conference on Ambient Computing, Applications, Services and Technologies. (2011) 7–12*
14. Panger, G.: Kinect in the kitchen: testing depth camera interactions in practical home environments. In: *CHI’12 Extended Abstracts on Human Factors in Computing Systems, ACM* (2012) 1985–1990
15. Heidrich, F., Golod, I., Russell, P., Ziefle, M.: Device-free interaction in smart domestic environments. In: *Proceedings of the 4th Augmented Human International Conference, ACM* (2013) 65–68
16. Vatavu, R.D.: Nomadic gestures: A technique for reusing gesture commands for frequent ambient interactions. *Journal of Ambient Intelligence and Smart Environments* **4**(2) (2012) 79–93
17. Jeong, S., Song, T., Kwon, K., Jeon, J.: Tv remote control using human hand motion based on optical flow system. In: Murgante, B., Gervasi, O., Misra, S., Nedjah, N., Rocha, A., Taniar, D., Apduhan, B., eds.: *Computational Science*

- and Its Applications - ICCSA 2012. Volume 7335 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2012) 311–323
18. Chen, M.y., Mummert, L., Pillai, P., Hauptmann, A., Sukthankar, R.: Controlling your tv with gestures. In: Proceedings of the international conference on Multimedia information retrieval, ACM (2010) 405–408
 19. Koskela, T., Väänänen-Vainio-Mattila, K.: Evolution towards smart home environments: empirical evaluation of three user interfaces. *Personal and Ubiquitous Computing* **8**(3-4) (2004) 234–240
 20. Peters, S., Loftness, V., Hartkopf, V.: The intuitive control of smart home and office environments. In: Proceedings of the 10th SIGPLAN symposium on New ideas, new paradigms, and reflections on programming and software, ACM (2011) 113–114
 21. Nazemi, K., Burkhardt, D., Stab, C., Breyer, M., Wichert, R., Fellner, D.W.: Natural gesture interaction with accelerometer-based devices in ambient assisted environments. In: *Ambient Assisted Living*. Springer (2011) 75–90
 22. Schreiber, M., Wilamowitz-Moellendorff, M., Bruder, R.: New interaction concepts by using the wii remote. In Jacko, J., ed.: *Human-Computer Interaction. Novel Interaction Methods and Techniques*. Volume 5611 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2009) 261–270
 23. Neßelrath, R., Lu, C., Schulz, C.H., Frey, J., Alexandersson, J.: A gesture based system for context-sensitive interaction with smart homes. In: *Ambient Assisted Living*. Springer (2011) 209–219
 24. Lee, T.Y., Kim, H.H., Park, K.H.: Gesture-based interface using baby signs for the elderly and people with mobility impairment in a smart house environment. In: *Aging Friendly Technology for Health and Independence*. Springer (2010) 234–237
 25. Brumitt, B., Cadiz, J., Cadiz, J.: "let there be light" examining interfaces for homes of the future. In: in: *Proceedings of Interact 01*, Citeseer (2001)
 26. Lockwood, S., Cook, D.J.: *Computer, light on!* (2008)
 27. Möller, S., Göttsche, F., Wolters, M.: A corpus analysis of spoken smart-home interactions with older users. In: in: *Proceedings of the 6th International Conference on Language Resources and Evaluation* (2008). (2008)
 28. Georgila, K., Wolters, M., Karaiskos, V., Kronenthal, M., Logie, R., Mayo, N., Moore, J., Watson, M.: A fully annotated corpus for studying the effect of cognitive ageing on users' interactions with spoken dialogue systems. (2008)
 29. Vipperla, R., Wolters, M., Georgila, K., Renals, S.: Speech input from older users in smart environments: Challenges and perspectives. In: *Universal Access in Human-Computer Interaction. Intelligent and Ubiquitous Interaction Environments*. Springer (2009) 117–126
 30. Portet, F., Vacher, M., Golanski, C., Roux, C., Meillon, B.: Design and evaluation of a smart home voice interface for the elderly: acceptability and objection aspects. *Personal and Ubiquitous Computing* **17**(1) (2013) 127–144
 31. López, G., Peláez, V., González, R., Lobato, V.: Voice control in smart homes using distant microphones: a voicexml-based approach. In: *Ambient Intelligence*. Springer (2011) 172–181
 32. Zhu, J., Gao, X., Yang, Y., Li, H., Ai, Z., Cui, X.: Developing a voice control system for zigbee-based home automation networks. In: *Network Infrastructure and Digital Content, 2010 2nd IEEE International Conference on*, IEEE (2010) 737–741

33. Soda, S., Nakamura, M., Matsumoto, S., Izumi, S., Kawaguchi, H., Yoshimoto, M.: Handsfree voice interface for home network service using a microphone array network. In: *Networking and Computing (ICNC), 2012 Third International Conference on*, IEEE (2012) 195–200
34. Minker, W., López-Cózar, R., Mctear, M.: The role of spoken language dialogue interaction in intelligent environments. *Journal of Ambient Intelligence and Smart Environments* **1**(1) (2009) 31–36
35. Filipe, P., Mamede, N.: Ambient intelligence interaction via dialogue systems (2010)
36. Montoro, G., Alamán, X., Haya, P.A.: Spoken interaction in intelligent environments: a working system. *Advances in Pervasive Computing* (2004) 217–222
37. Montoro, G., Haya, P.A., Alamán, X.: Context adaptive interaction with an automatically created spoken interface for intelligent environments. In: *Intelligence in Communication Systems*. Springer (2004) 120–127
38. Montoro, G., Haya, P.A., Alaman, X.: A dynamic spoken dialogue interface for ambient intelligence interaction. *IJACI* (2010) 24–51
39. Vacher, M., Portet, F., Lecouteux, B., Golanski, C., et al.: Speech analysis for ambient assisted living: technical and user design of a vocal order system. *Telhealthcare Computing and Engineering: Principles and Design* (2013) 607–638
40. San Agustin, J., Skovsgaard, H., Hansen, J.P., Hansen, D.W.: Low-cost gaze interaction: ready to deliver the promises. In: *CHI'09 Extended Abstracts on Human Factors in Computing Systems*, ACM (2009) 4453–4458
41. San Agustin, J., Skovsgaard, H., Mollenbach, E., Barret, M., Tall, M., Hansen, D.W., Hansen, J.P.: Evaluation of a low-cost open-source gaze tracker. In: *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, ACM (2010) 77–80
42. Hansen, D.W., Ji, Q.: In the eye of the beholder: A survey of models for eyes and gaze. *Pattern Analysis and Machine Intelligence*, *IEEE Transactions on* **32**(3) (2010) 478–500
43. MacKenzie, I.S., Majaranta, P., Aoki, H., Donegan, M., Hansen, D., Hansen, J., Hyrskykari, A., Rähkä, K.: Evaluating eye tracking systems for computer input. *Gaze Interaction and Applications of Eye Tracking: Advances in Assistive Technologies*, edited by P. Majaranta, H. Aoki, M. Donegan, DW Hansen, JP Hansen, A. Hyrskykari, and K.-J. Rähkä (2012) 205–25
44. Mollenbach, E.: Selection strategies in gaze interaction. (2010)
45. Bonino, D., Castellina, E., Corno, F., De Russis, L.: Dogeye: Controlling your home with eye interaction. *Interacting with Computers* **23**(5) (2011) 484–498
46. Hennessey, C., Fiset, J.: Long range eye tracking: bringing eye tracking into the living room. In: *Proceedings of the Symposium on Eye Tracking Research and Applications*, ACM (2012) 249–252
47. Hansen, J.P., Agustin, J.S., Skovsgaard, H.: Gaze interaction from bed. In: *Proceedings of the 1st Conference on Novel Gaze-Controlled Applications*, ACM (2011) 11
48. Mardanbegi, D., Hansen, D.W.: Mobile gaze-based screen interaction in 3d environments. In: *Proceedings of the 1st Conference on Novel Gaze-Controlled Applications*, ACM (2011) 2
49. Shi, F., Gale, A.G., Purdy, K.: Direct gaze based environmental controls. (2006)
50. Culén, A.L., Bratteteig, T.: Touch-screens and elderly users: A perfect match? In: *ACHI 2013, The Sixth International Conference on Advances in Computer-Human Interactions*. (2013) 460–465

51. Cockburn, A., Ahlström, D., Gutwin, C.: Understanding performance in touch selections: Tap, drag and radial pointing drag with finger, stylus and mouse. *International Journal of Human-Computer Studies* **70**(3) (2012) 218–233
52. Wobbrock, J.O., Morris, M.R., Wilson, A.D.: User-defined gestures for surface computing. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM (2009) 1083–1092
53. Seifried, T., Haller, M., Scott, S.D., Perteneder, F., Rendl, C., Sakamoto, D., Inami, M.: Cristal: a collaborative home media and device controller based on a multi-touch display. In: *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ACM (2009) 33–40
54. Suo, Y., Wu, C., Qin, Y., Yu, C., Zhong, Y., Shi, Y.: Housegenie: Universal monitor and controller of networked devices on touchscreen phone in smart home. In: *Ubiquitous Intelligence & Computing and 7th International Conference on Autonomic & Trusted Computing (UIC/ATC)*, 2010 7th International Conference on, IEEE (2010) 487–489
55. Kartakis, S., Antona, M., Stephanidis, C.: Control smart homes easily with simple touch. In: *Proceedings of the 2011 international ACM workshop on Ubiquitous meta user interfaces*, ACM (2011) 1–6
56. Lin, C.Y., Lin, Y.B.: Projection-based user interface for smart home environments. In: *Computer Software and Applications Conference Workshops (COMPSACW)*, 2013 IEEE 37th Annual, IEEE (2013) 546–549
57. Drossis, G., Grammenos, D., Bouhli, M., Adami, I., Stephanidis, C.: Comparative evaluation among diverse interaction techniques in three dimensional environments. In: *Distributed, Ambient, and Pervasive Interactions*. Springer (2013) 3–12
58. Costanza, E., Giaccone, M., Kueng, O., Shelley, S., Huang, J.: Ubicomp to the masses: a large-scale study of two tangible interfaces for download. In: *Proceedings of the 12th ACM international conference on Ubiquitous computing*, ACM (2010) 173–182
59. Gregor, S., Rahimi, M., Vogt, M., Schulz, T., Luck, K.: *Tangible interaction-vom konzept zur realisierung1*
60. Van der Vlist, B., Niezen, G., Hu, J., Feijs, L.: Design semantics of connections in a smart home environment. *Design and Semantics of Form and Movement (DeSForM 2010)*, Lucerne, Switzerland (2010) 48–56
61. Kwak, M., Niezen, G., Van Der Vlist, B., Hu, J., Feijs, L.: Tangible interfaces to digital connections, centralized versus decentralized. In: *Transactions on edutainment V*. Springer (2011) 132–146
62. van der Vlist, B., Niezen, G., Rapp, S., Hu, J., Feijs, L.: Controlling smart home environments with semantic connections: a tangible and an ar approach. *Design and semantics of form and movement (DeSForM)* (2012) 160–169
63. Baraldi, S., Del Bimbo, A., Landucci, L., Torpei, N., Cafini, O., Farella, E., Pieracci, A., Benini, L.: Introducing tangerine: a tangible interactive natural environment. In: *Proceedings of the 15th international conference on Multimedia*, ACM (2007) 831–834
64. Carrino, S., Péclat, A., Mugellini, E., Abou Khaled, O., Ingold, R.: Humans and smart environments: a novel multimodal interaction approach. In: *Proceedings of the 13th international conference on multimodal interfaces*, ACM (2011) 105–112
65. Fleer, D., Leichsenring, C.: Miso: a context-sensitive multimodal interface for smart objects based on hand gestures and finger snaps. In: *Adjunct proceedings of the 25th annual ACM symposium on User interface software and technology*, ACM (2012) 93–94

66. Lee, H., Lim, S.Y., Lee, I., Cha, J., Cho, D.C., Cho, S.: Multi-modal user interaction method based on gaze tracking and gesture recognition. *Signal Processing: Image Communication* (2012)
67. Hales, J., Rozado, D., Mardanbegi, D.: Interacting with objects in the environment by gaze and hand gestures
68. Mardanbegi, D., Hansen, D.W., Pederson, T.: Eye-based head gestures. In: *Proceedings of the Symposium on Eye Tracking Research and Applications*, ACM (2012) 139–146
69. Stellmach, S., Dachsel, R.: Look & touch: gaze-supported target acquisition. In: *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, ACM (2012) 2981–2990
70. Pflöging, B., Kienast, M., Schmidt, A., Döring, T., et al.: Speet: A multimodal interaction style combining speech and touch interaction in automotive environments. In: *Adjunct Proceedings of the 3rd International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, *AutomotiveUI*. Volume 11. (2011)
71. Budde, M., Berning, M., Baumgärtner, C., Kinn, F., Kopf, T., Ochs, S., Reiche, F., Riedel, T., Beigl, M.: Point & control–interaction in smart environments: you only click twice. In: *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*, ACM (2013) 303–306
72. Molyneaux, D., Gellersen, H.: Projected interfaces: enabling serendipitous interaction with smart tangible objects. In: *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction*, ACM (2009) 385–392
73. Pan, G., Wu, J., Zhang, D., Wu, Z., Yang, Y., Li, S.: Geeair: a universal multimodal remote control device for home appliances. *Personal and Ubiquitous Computing* **14**(8) (2010) 723–735
74. Kurtenbach, G., Hultheen, E.A.: Gestures in human-computer communication. *The art of human-computer interface design* (1990) 309–317
75. Morris, M.R., Wobbrock, J.O., Wilson, A.D.: Understanding users’ preferences for surface gestures. In: *Proceedings of graphics interface 2010*, Canadian Information Processing Society (2010) 261–268
76. Nielsen, M., Störing, M., Moeslund, T.B., Granum, E.: A procedure for developing intuitive and ergonomic gesture interfaces for hci. In: *Gesture-Based Communication in Human-Computer Interaction*. Springer (2004) 409–420
77. Salvador, S., Chan, P.: Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis* **11**(5) (2007) 561–580
78. Suarez, J., Murphy, R.R.: Hand gesture recognition with depth images: A review. In: *RO-MAN, 2012 IEEE*, IEEE (2012) 411–417
79. Perroud, D., Barras, F., Pierroz, S., Mugellini, E., Khaled, O.A.: Framework for development of a smart environment: Conception and use of the naif framework. In: *New Technologies of Distributed Systems (NOTERE), 2011 11th Annual International Conference on*, IEEE (2011) 1–7
80. Nesselrath, R.: TaKG - Ein Toolkit zur automatischen Klassifikation von Gesten. *Mastersthesis, DFKI* (3 2008)
81. Ten Holt, G., Reinders, M., Hendriks, E.: Multi-dimensional dynamic time warping for gesture recognition. In: *Thirteenth annual conference of the Advanced School for Computing and Imaging*. Volume 300. (2007)
82. Zimmermann, G., Vanderheiden, G.: The universal control hub: An open platform for remote user interfaces in the digital home. In: *Human-Computer Interaction. Interaction Platforms and Techniques*. Springer (2007) 1040–1049

83. Rahman, A., Hossain, M.A., Parra, J., El Saddik, A.: Motion-path based gesture interaction with smart home services. In: Proceedings of the 17th ACM international conference on Multimedia, ACM (2009) 761–764
84. Sigal, L., Sclaroff, S., Athitsos, V.: Skin color-based video segmentation under time-varying illumination. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **26**(7) (2004) 862–877
85. Young, S., Woodland, P., Byrne, W.: Htk: Hidden markov model toolkit (1993)
86. Takagi, T., Noguchi, H., Kugata, K., Yoshimoto, M., Kawaguchi, H.: Microphone array network for ubiquitous sound acquisition. In: Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on, IEEE (2010) 1474–1477
87. López-Cózar, R., Callejas, Z.: Multimodal dialogue for ambient intelligence and smart environments. In: Handbook of ambient intelligence and smart environments. Springer (2010) 559–579
88. Montoro, G., Alam-Ájn, X., Haya, P.: A plug and play spoken dialogue interface for smart environments. In Gelbukh, A., ed.: Computational Linguistics and Intelligent Text Processing. Volume 2945 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2004) 360–370
89. Martínez, A.E., Cabello, R., Gómez, F.J., Martínez, J.: Interact-ddm: a solution for the integration of domestic devices on network management platforms. In: Integrated Network Management, 2003. IFIP/IEEE Eighth International Symposium on, IEEE (2003) 485–488
90. Englemore, R.: Blackboard systems: edited by Robert Englemore, Tony Morgan. Addison-Wesley (1988)
91. d’Informatique de Grenoble, L.: Sweet-home projekt (2009) <http://sweet-home.imag.fr/>.
92. Vacher, M., Istrate, D., Portet, F., Joubert, T., Chevalier, T., Smidtas, S., Meillon, B., Lecouteux, B., Sehili, M., Chahuara, P., et al.: The sweet-home project: Audio technology in smart homes to improve well-being and reliance. In: Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE, IEEE (2011) 5291–5294
93. Jacob, R.J.: What you look at is what you get: eye movement-based interaction techniques. In: Proceedings of the SIGCHI conference on Human factors in computing systems, ACM (1990) 11–18
94. Hennessey, C., Nouredin, B., Lawrence, P.: A single camera eye-gaze tracking system with free head motion. In: Proceedings of the 2006 symposium on Eye tracking research & applications, ACM (2006) 87–94
95. Lin, C.L., Hung, Y.H., Chen, H.Y., Chu, S.L.: Content-aware smart remote control for android-based tv. In: Consumer Electronics (ICCE), 2012 IEEE International Conference on, IEEE (2012) 678–679
96. Hornecker, E.: Die rückkehr des sensorischen: Tangible interfaces und tangible interaction. Engpass Mensch-Computer-Interface. Historische, aktuelle und zukünftige Lösungsansätze für die Computerbedienung. Transcript Verlag (2008) 235–256
97. Cafini, O., Farella, E., Benini, L., Baraldi, S., Torpei, N., Landucci, L., Del Bimbo, A.: Tangerine smcube: a smart device for human computer interaction. In: Proc. of IEEE European Conference on Smart Sensing and Context. (2008)
98. Fernández, M.Á., Peláez, V., López, G., Carus, J.L., Lobato, V.: Multimodal interfaces for the smart home: findings in the process from architectural design to user evaluation. In: Ubiquitous Computing and Ambient Intelligence. Springer (2012) 173–180

99. Sebe, N.: Multimodal interfaces: Challenges and perspectives. *Journal of Ambient Intelligence and smart environments* **1**(1) (2009) 23–30
100. Vesa, S., Lokki, T.: An eyes-free user interface controlled by finger snaps. In: *Proc. of the 8th Int. Conference on Digital Audio Effects (DAFx. 05)*. (2005) 262–265
101. Böhlen, M.: Second order ambient intelligence. *Journal of Ambient Intelligence and Smart Environments* **1**(1) (2009) 63–67

Zugriff auf lokale und verteilte Daten im Kontext der Android Plattform

Robert Zetzsche*

Betreuer: Anja Bachmann[†]

Karlsruher Institut für Technologie (KIT)
Pervasive Computing Systems – TECO

*robert.zetzsche@student.kit.edu

[†]bachmann@teco.edu

Zusammenfassung. Die nachfolgende Ausarbeitung behandelt den Zugriff auf lokale und verteilte Informationen aus der mobilen Cloud mit Hilfe eines Android-Smartphones. Dabei erfolgt eine Machbarkeitsstudie anhand eines Prototypen, bei dem verschiedene Sensoren und Cloud-Dienste implementiert und abgegriffen werden. Übergeordnetes Ziel der Untersuchung ist das Nachweisen der Machbarkeit und die Bewertung der Nützlichkeit der Informationen für eine Aktivitäts- oder Kontextererkennung. Es wird ein Überblick über vorhandene Dienste und deren Zugriffsmöglichkeiten gegeben. Es hat sich gezeigt, dass viele Dienste bzw. Informationen abgegriffen werden können, dies jedoch eine aktive Zustimmung (Berechtigung) des Benutzers voraussetzt.

Schlüsselwörter: Android, Aktivitätserkennung, Kontextererkennung, Mobile Cloud

1 Einleitung

Smartphones, Wearables und Tablets. Dies sind die Trends der letzten 3 Jahre, die kaum einen Abbruch nehmen. Die Verbreitung dieser Geräte nimmt auch in der allgemeinen Bevölkerung stark zu, während z.B. der PC-Markt nach wie vor einbüßt¹. Dabei ist Android das bestimmende mobile Betriebssystem und gewinnt seit Jahren hinzu. Allein im dritten Quartal 2013 liefen 81% der verkauften Smartphones mit dem von Google bzw. der OHA entwickelten Betriebssystem Android². Alle diese Geräte sind mit mobiler Sensorik und mobilen Internet ausgestattet und bieten somit eine gute Schnittstelle, um Informationen über den Nutzer des Gerätes zu gewinnen. Diese Informationen können anschließend verknüpft werden und lassen dadurch Aussagen zu der Aktivität und dem Kontext des Anwenders zu. Dadurch ist es möglich smarte und intelligente Applikationen für den Nutzer zu erstellen. Der Zugriff auf lokale Daten und die Sensorik, als auch der Zugriff auf Cloud-Dienste spielt dabei eine wichtige Rolle.

¹ <http://www.golem.de/news/marktforschung-tablet-boom-wandelt-sich-pcs-weiter-in-der-krise-1403-105412.html>

² <http://zdnet.de/88175756/smartphonemarkt-android-erreicht-81-prozent>

Das nachfolgende Paper liefert einen Überblick über den Zugriff auf lokale Daten und auf Cloud-Dienste im Rahmen des Android-Betriebssystems. Dabei werden zunächst in Kapitel 2 die Grundlagen und Begrifflichkeiten eingeführt, anschließend folgt in Kapitel 3 eine Übersicht über diverse Dienste, auf die von einem Smartphone aus zugegriffen werden kann. Dabei werden auch die Unterschiede zwischen verschiedenen technischen Zugriffsmöglichkeiten gezeigt. Außerdem erfolgt eine kurze Analyse der Aussagekraft der Daten für eine evtl. später erfolgende Aktivitäts- bzw. Kontexterkenkung. In Kapitel 4 wird eine Beispiel-Applikation bzw. ein Proof of Concept vorgestellt, welche ein Großteil der in Kapitel 3 vorgestellten Dienste implementiert. Es wird das Interface gezeigt und die Implementierung erklärt. Zum Abschluss wird in Kapitel 5 ein Fazit gezogen und ein kleiner Ausblick präsentiert.

2 Grundlagen

Im folgenden Kapitel werden die wichtigsten Begrifflichkeiten der nachfolgenden Arbeit definiert. Dabei stellen die Begriffe den Rahmen und das Ziel des Papers dar. Eine Kombination von Daten des Smartphones und Daten aus der mobile Cloud soll schlussendlich zu einer schlüssigen Aktivitäts- und Kontexterkenkung führen.

2.1 Mobile Cloud

Mobile Cloud Computing ist eine spezielle Form des Cloud Computing. Dabei erfüllt sie alle Charakteristiken [9] des normalen Cloud Computings:

- **On-Demand Self-Service** Ein Nutzer kann Ressourcen wie Rechenkapazitäten und Netzwerkspeicher reservieren und konsumieren. Dies geschieht dabei automatisch und ohne menschliche Interaktion mit jedem einzelnen Service-Provider.
- **Broad Network Access** Funktionen sind über das Netzwerk verfügbar. Dabei wird auf sie über standardisierte Mechanismen zugegriffen.
- **Resource pooling** Rechenressourcen des Providers werden gesammelt, um verschiedenen Nutzern dynamisch virtuelle und physikalische Ressourcen zuzuweisen. Dabei sind die Ressourcen nur bedingt ortsabhängig. Der Nutzer weiß nicht, wo seine zuteilten Ressourcen sich befinden, außer in einem spezifischen Kontext (z.B. Land oder Kontinent).
- **Rapid elasticity** Ressourcen werden, teilweise automatisch, elastisch bereitgestellt und freigegeben, um auf Bedarf von innen und außen rapide zu skalieren. Dabei scheint es für den Nutzer, als seien unbegrenzt viele Ressourcen existent, auf die jeder Zeit zugegriffen werden kann.
- **Measured service** Cloud Systeme steuern und optimieren automatisch den Ressourcenverbrauch über ein Dosierbarkeitssystem. Die Ressourcennutzung wird dabei überwacht, kontrolliert und reported, sodass eine Transparenz sowohl für den Nutzer als auch den Anbieter entsteht.

Dabei ist beim Mobile Cloud Computing der Kontext der mobilen Geräte ein besonderer. Es gibt dabei Unterschiede in der Art der Vernetzung, als auch der Ressourcen selbst, da mobile Geräte sowohl über wenig Speicher als auch knappe Rechenkapazitäten verfügen. Dies führt dazu, dass nicht alle Prozesse und Tasks auf dem mobilen Gerät selbst abgearbeitet werden können, sondern in die Cloud verlagert werden müssen. Sei es um bspw. Sprachsynthese durchzuführen oder Videospiele zu berechnen. Dabei existieren im einzelnen verschiedene Definitionen von Cloud Computing [4]:

- Mobile Cloud Computing als die Verbindung von mehreren mobilen Endgeräten selbst, die zusammen gewisse Vorhaben berechnen oder Sensordaten bereitstellen. Dabei treten sie über eine Peer-to-Peer-Verbindung in Kontakt.
- Mobile Cloud Computing als das Ausführen von Applikationen, die auf Cloud-Daten zugreifen, z.B. das Ausführen der Google-Mail-Anwendung auf einem Android-Smartphone. Dabei interagiert das mobile Endgerät als Thin-Client, welches Kontakt zu einem Google-Server (Rich Server) aufnimmt. Diese Ausarbeitung basiert auf dieser Definition von mobile Cloud Computing.

2.2 Mobile Sensing

Mobile Endgeräte sind heute keine reinen Kommunikations- und Rechengenäte mehr, sondern verfügen über eine Vielzahl von Sensoren auf die zugegriffen werden kann. Dies geht über die Standard-Sensoren wie Lagesensor (z.B. GPS), Näherungssensor und Beschleunigungssensor hin zu Temperatursensoren, Gyroskopen oder digitalen Kompassen. Dabei zählen auch die Kamera und das Mikrofon zu Sensoren die über das Smartphone ausgewertet werden können. Das Sammeln und Auswerten dieser Sensoren wird Mobile Sensing genannt. Es öffnet den Raum für neue Applikationen und Anwendungsfälle. Dabei ist auch die Anwendungsgruppe unterschiedlich wählbar, sei es eine Einzelperson oder eine Gruppe von Personen, die sich in der Nähe befinden. Diese Informationen, die beim Mobile Sensing anfallen und ausgewertet werden können, können anschließend in die Aktivitäts- oder Kontextererkennung einfließen.[8]

2.3 Aktivitäts- und Kontextererkennung

Bei der Aktivitäts- und Kontextererkennung handelt es sich um einen sich teils überschneidenden und häufig als Synonym verwendeten Begriff, welcher auf dem mobile Sensing fußt. Dabei ist jedoch die Aktivitätserkennung die Grundlage der Kontextererkennung. Bei der Aktivitätserkennung werden verschiedene Sensoren und weitere Daten eines, typischerweise mobilen, Endgeräts ausgewertet. Dabei spielen häufig der Lagesensor und der Beschleunigungssensor eine wichtige Rolle. Aus diesen Daten kann eine Aktivität, wie z.B. Laufen oder Stehen, ermittelt werden. Aufbauend auf dem Erkennen von Aktivitäten existiert die Kontextererkennung. Bei dem Kontext handelt es sich um die nächst größere Abstraktionsebene, welche meist auf dem unterschiedlichen Erkennen von Aktivitäten erfolgt.

Beispielsweise kann die Abfolge der Aktivitäten Laufen, Stehen und Schießen auf den Kontext des Fußballspielens abgeleitet werden. Der Kontext verbindet also die erkannten Aktivitäten zu einem größeren Konstrukt.[11] [10]

3 Dienste

Das Kapitel Dienste stellt einen groben Überblick über die Dienste dar, auf welche von einem mobilen Gerät aus zugegriffen werden kann. Dabei handelt es sich sowohl um den Zugriff auf Daten, welche lokal vorliegen oder ermittelt werden können, als auch um Daten, welche über eine Internetverbindung und einem zugehörigen Account ermittelt werden können. Alle diese Daten sind durch bestimmte Berechtigungen geschützt, können jedoch nach der Zustimmung vollkommen problemlos abgegriffen werden. Im Folgenden werden die Möglichkeiten im Kontext des Android-Betriebssystems beleuchtet. Eine Abbildung auf ein anderes mobiles Betriebssystem ist jedoch ohne Problem möglich. Zuerst folgt das Aufzeigen von Möglichkeiten im lokalen Kontext, wie Sensoren oder Systemfunktionen, anschließend die Auswertung von Daten aus Social Networks. Danach kommt die Analyse der Google-Dienste, welchen eine besondere Bedeutung beim Android-Betriebssystem zukommt. Diese werden bei Google-zertifizierten Geräten grundsätzlich mit ausgeliefert und können nach der Registrierung direkt genutzt werden. Dazu gehört auch der vereinfachte Zugriff für Entwickler auf diese Daten. Zum Schluss werden noch einige wichtige Dienste erwähnt, welche in dieser Auflistung fehlen.

3.1 Social Networks

Social Networks gehören heute zu unserem täglichen Leben dazu. Sie spielen in allen Aspekten eine Rolle und werden oft täglich konsultiert, daher kommt ihnen eine Große Rolle bei dem Zugriff auf Daten und Informationen zu. Auch im Sinne der Aktivitätserkennung spielen sie eine wichtige Rolle, da Statusupdates, Likes und Kommentare täglich ausgetauscht werden. Außerdem bieten alle Social Networks die Möglichkeit Orte zu teilen oder zu bestimmen, auch wenn die Lokations-Provider wie GPS ausgeschaltet sind. Im Folgenden wird sich die Arbeit auf den Zugriff auf die drei größten Social Networks konzentrieren auf denen die meisten aktiven Nutzer zugegen sind [2], um Facebook, Google+ und Twitter. Das Zugreifen auf Daten ist jedoch bei allen Social Networks ähnlich und kann auf andere Genre-Vertreter angewendet werden, vorausgesetzt es existiert eine API.

Facebook Facebook ist das meistgenutzte Social Network in ganz Deutschland, dies betrifft nicht nur die Anzahl der Nutzer sondern auch die Aktivitätsrate der Nutzer [2]. Dadurch spielt es in diesem Bereich die größte Rolle und bietet ebenfalls eine große Vielfalt von Zugriffsmechanismen an. Diese sind für alle

Plattformen anwendbar, da sowohl eine normale REST³-API existiert als auch ein spezielles Android SDK zum Zugriff auf Facebook, welches z.B. die Kontenverwaltung von Android unterstützt. Dabei gibt es verschiedene Stufen eines Zugriffs auf Facebook. Der öffentliche Zugriff auf Facebook-Daten, welcher das öffentliche Profil beinhaltet. Dies ist je nach Privatsphäre-Einstellungen des jeweiligen Nutzers recht ergiebig. Prinzipiell erhält dies die ID, den Namen (Vor-, Nach- und Gesamtnamen), das Geschlecht, die Herkunft (begrenzt auf das Land) und den Nutzernamen auf Facebook. Diese sind jedoch nur über die Graph-API verfügbar, dann jedoch ohne Anmeldung (falls die User-ID oder der Facebook-Nutzername bekannt ist). Dabei erfolgt eine Anfrage an die API per:

```
https://graph.facebook.com/ID
```

Dabei bekommt werden alle Informationen in einem JSON-Response zurückgeliefert. Um Zugriff auf das Bild zu bekommen, muss folgender Request ausgeführt werden:

```
https://graph.facebook.com/ID/picture
```

Ein weiterer Zugriff ist auch mit der Graph-API möglich [3], dann wird ebenfalls eine Authentifizierung über ein oAuth⁴-Token benötigt. Diesen Vorgang bietet jedoch auch das Facebook SDK für Android. Dieses muss von <https://developers.facebook.com/docs/android> heruntergeladen werden, außerdem muss eine Facebook-Anwendung als Entwickler erstellt werden. Anschließend wird eine Application-ID generiert und angezeigt, die für den Zugriff benötigt wird. Seine Android SHA-1 Hash als auch das zugreifende Package der Android-Anwendung muss in die Facebook-Anwendung eingetragen werden. Sind diese Start-Bedingungen alle erfüllt, kann über das Einbinden des Facebook-Buttons des SDKs eine Anmeldung angesteuert werden. Dabei muss zwingend die App-ID in dem Login-Button gesetzt werden. Über diese Anmeldung wird initial ebenfalls nur der Zugriff auf die öffentlichen Daten und die Freundesliste möglich.

³ Bei Representational State Transfer (REST) handelt es sich um ein Programmierparadigma für Web-Anwendungen, welches von Roy Thomas Fielding entwickelt wurde. Es wird eine Art und Weise beschrieben, wie Ressourcen im Web adressiert und übertragen werden können. Seine Hauptkriterien sind zustandsloser und uniformer Zugriff auf Ressourcen in einer Client-Server-basierten Architektur. Die häufigste Ausprägung ist im World Wide Web zu finden, bei der die einzelnen Ressourcen eindeutig über eine URL gekennzeichnet sind und über HTTP abgegriffen werden. Dabei müssen alle relevanten Daten dem HTTP Request übergeben werden. Ausgetauscht werden Daten in einem bestimmten Transferformat wie XML, JSON oder HTML. [5]

⁴ Bei oAuth handelt es sich um ein Protokoll, welches es ermöglicht geschützte Ressourcen einer bestimmten Webanwendung (dem Service Provider) einer oder mehreren anderen Anwendungen (Consumern) zugänglich zu machen, ohne dabei sein Passwort oder den Nutzernamen preiszugeben. Dies funktioniert über den Austausch von sogenannten Tokens, nach einer erfolgreichen Authentifizierung gegenüber dem Service Provider. In diesen Tokens ist bspw. der zugreifende Account und für welche Ressourcen die Autorisierung gilt festgelegt.[1]

Dabei können nun initial neue Berechtigungen angefragt werden oder während der Laufzeit dynamisch angefragt werden. Die Berechtigungen von Facebook teilen sich dabei in 4 verschiedene Bereiche ein:

- **Public Profile and Friend List:** Hierbei handelt es sich um die oben genannten öffentlichen Informationen, als auch um den Zugriff auf die Freundesliste. Diese werden automatisch gewährt, wenn eine Facebookanmeldung erfolgt.
- **Email Permissions:** Dabei handelt es sich ausschließlich um die E-Mail Permission, welche gesondert behandelt wird. Sie ermöglicht ausschließlich das Auslesen der Email-Adresse, mit welcher sich der Nutzer angemeldet hat.
- **Extended Profile Properties:** Hierin sind alle Rechte gebündelt, welche einen erweiterten Zugriff auf das angemeldete Profil zulassen. Dazu gehören z.B. die Berechtigungen für den Zugriff auf die Standorte, die Likes oder die Status-Updates.
- **Extended Permissions:** Unter diesen Berechtigungen zählen z.B. alle Schreib- bzw. Publizierrechte und der Lesezugriff auf erweiterte Rechte, welche unabhängig vom eigenen Profil sind, z.B. dem aktuellen Stream.
- **Open Graph Permissions:** Diese Rechte sind Sonderberechtigungen für die Nutzung des Open-Graphs von Facebook, bspw. das Nutzen der Spiele-Plattform von Facebook. Außerdem beinhaltet diese Gruppe ebenfalls das Publizieren von Facebook-Actions. Hierunter werden z.B. Posts, Likes und Kommentare verstanden.
- **Page Permissions:** Die Seiten-Berechtigung ist eine besondere Berechtigung für das Verwalten von Facebook-Seiten, die mit dem jeweiligen Nutzer verbunden sind.

Das Absetzen der zugehörigen Requests erfolgt anschließend über die Request-Klasse des Facebook-SDKs. Dabei wird intern jedoch ein normaler Aufruf zur Graph-API getätigt.

Facebook Informationen haben insgesamt eine schwankende Aussagekraft über die jeweilige Aktivität des Benutzers. Der Standort der bei Facebook gepostet werden kann, hat eine sehr hohe Aussagekraft auf die Aktivität des Nutzers, da hier auch ohne angeschalteten Lagesensor ein Standort festgelegt werden kann. Außerdem lässt sich durch häufiges Posten ebenfalls gut auf eine Aktivität schließen. Die anderen Informationen haben eine sehr geringe Aussagekraft, wenn sie nicht mit anderen Informationen verknüpft werden (z.B. angegebener Wohnort in Facebook und ermittelter Standort des Lagesensors lässt sich auf die Aktivitäten „zu Hause“ oder „nicht zu Hause“ abbilden)

Google+ Der Zugriff auf Google+ stellt eine Besonderheit dar, da Google+ eine enge Verzahnung zu allen anderen Google-Diensten besitzt. Dabei ist es sowohl möglich den Google-Account selbst, als auch den Google+-Account als Grundlage für die Authentifizierung zu benutzen. Der direkte Zugriff auf Google+ wird im Kapitel Google-Dienste weiter behandelt.

Twitter Twitter gehört zu den großen aufstrebenden Social Networks, belegt in Deutschland jedoch nur Platz 7 der Top Social Networks. Die tägliche Nutzerrate erreicht jedoch annähernd die 50%, wodurch Twitter zu einem sehr interessantem Feld für Entwickler wird [2]. Dennoch existiert kein direktes Twitter SDK für Android. Die Kommunikation muss also über die REST-API erfolgen, oder über API's die teilweise nicht von Twitter veröffentlicht wurden⁵. Zudem fährt Twitter eine sehr restriktive Entwickler-Politik, bei der maximal 100.000 sogenannte Access-Tokens vergeben werden können⁶. Ein Access-Token wird benötigt um auf Twitter zuzugreifen, dieses wird pro Nutzer und registriertem Gerät erstellt und zurückgeliefert. Dadurch ist es quasi nicht möglich, große Applikationen mit Twitter-Integration zu bauen. Daher spielt Twitter hier, als auch in der späteren Beispielimplementierung, keine Rolle. Dennoch kann Twitter einen sehr hohen Einfluss auf die Aktivitätsrate haben, da hier häufiges Posten teil des Nutzungs-Modells ist.

3.2 Lokale Aktivität

Ein Smartphone ist ein Wunder der Technik, in der viele Sensoren, Prozessoren u.ä. verbaut sind. Zudem läuft der Großteil der menschlichen Kommunikation heute über ein Smartphone. Aus diesem Grund ist das Abgreifen von lokalen Datenquellen ein wichtiger Bestandteil bei der Aktivitätsbestimmung. Dabei gibt es viele verschiedene auszulesende Faktoren, welche im Nachfolgenden nicht vollständig genannt werden können. Das Kapitel konzentriert sich auf die wichtigsten Bestandteile und Datenquellen des Smartphones.

Sensorik In einem Smartphone sind heutzutage sehr vielfältige Sensoren verbaut. Dies fängt bei den drei klassischen Sensoren (Lage-, Beschleunigungs- und Näherungssensoren) an und endet bei den speziell eingebauten Sensoren, wie Druck-, Temperatur- oder Geomagnetischen Sensoren, die bspw. im Samsung Galaxy S4 verbaut sind⁷. Sensorik spielt bei der Bestimmung von Aktivitäten eine große Rolle und bietet zudem den Vorteil, dass sie lokal verbaut sind und es somit keinem Internetzugriff bedarf.

Der Zugriff auf die Android-Sensoren ist prinzipiell ähnlich. Es wird ein `SensorEventListener` implementiert, in dem die Methode `onSensorChanged` implementiert wird. Anschließend wird dieses Event bei jeder Änderung des Sensors aufgerufen. Dabei lässt sich das jeweilige Aktualisierungsintervall über 4 Konstanten anpassen, die auf die jeweiligen Bedürfnisse von Anwendungen angepasst sind [6]. Diese sind jedoch lediglich Richtwerte und garantieren nur die jeweilige Zugriffszeit, unterbieten sie jedoch meistens. Über einen so initialisierten Service lassen sich nun konstant oder zeitabhängig Daten aufzeichnen. Über diesen Zugriff lassen sich eine Vielzahl von Sensoren in Android ansteuern (Bspw.

⁵ <https://dev.twitter.com/docs/twitter-libraries>

⁶ <https://blog.twitter.com/2012/changes-coming-to-twitter-api>

⁷ <http://www.chip.de/ii/1/9/2/9/9/2/3/0/samsugn-644e2592f536439b.jpg>

Näherungs- und Beschleunigungssensor). Dabei existieren jedoch gewisse Einschränkungen bzw. Besonderheiten, z.B. bei folgenden Sensoren:

- **Näherungssensor:** Der Näherungssensor liefert nur die Information, ob er verdeckt wird oder nicht. Nicht jedoch die direkte Entfernung zum Gegenstand selbst. Diese wird weitergegeben, indem ein Wert auf der Z-Wert-Achse zurückgegeben wird, der entweder den Wert 0 oder einen anderen Wert, meist im Minusbereich, annimmt. Dabei steht der Wert 0 für die Verdeckung des Sensors, ein anderer Wert für die offene Lage des Sensors. Es ist druchaus möglich, an diese Information weitere Aktivitäten zu knüpfen.
- **Beschleunigungssensor:** Beim Beschleunigungssensor besteht die Besonderheit darin, dass die Erdbeschleunigung ebenfalls eine Auswirkung auf den Beschleunigungssensor hat. Um die reine Beschleunigung zu messen, bedarf es daher eines Algorithmuses zur Entfernung dieses Störfaktors [6]. Zudem ist die Aktualisierungsrate des Beschleunigungssensor sehr hoch, diese sollte daher von Hand auf die jeweiligen Bedürfnisse abgestimmt werden.

Die Implementierung des Lagesensors gestaltet sich im Gegensatz zum Beschleunigungs- oder Näherungssensor etwas anders. Die Lage kann prinzipiell über verschiedene Möglichkeiten ermittelt werden. Dabei wird nicht zwingend das GPS benutzt, sondern auch die eingewählte Telefonzelle, WLAN-Netzwerke oder andere Quellen. Auf dem Android-Betriebssystem wird die Nutzung der Google-Play-Services empfohlen. Dabei bietet Android drei verschieden granulare Möglichkeiten zur Auswahl für den Nutzer:

- **Hohe Genauigkeit:** Dabei werden alle Möglichkeiten zur Positionsbestimmung (GPS, WLAN, Mobilfunk) genutzt.
- **Energiesparmodus:** Dabei werden WLAN und Mobilfunk zur Standortbestimmung genutzt, diese sind erheblich Stromsparender da sich der GPS-Sensor nicht verbindet.
- **Nur Gerät:** Dabei wird lediglich das GPS zur Positionsbestimmung genutzt.

Dabei setzt Google standartmäßig den Wert auf hohe Genauigkeit und setzt zudem in den Erweiterten WLAN Optionen die Erkennungsfunktion bei ausgeschaltetem WLAN. Durch diese Option kann das WLAN, auch wenn es manuell deaktiviert wurde, dafür genutzt werden, um die umliegenden WLAN-Netze zu scannen und somit eine Standorterkennung möglich zu machen.

Nun kann der Entwickler über die Google-Play-Services auf den Standort zugreifen. Durch diese neuen Einstellungen wird nicht mehr dem Entwickler überlassen, was für eine Genauigkeit er benötigt, sondern dem Anwender selber. Der Entwickler erhält nur eine Location, mit der er anschließend weiterarbeiten kann. Die Standortbestimmung funktioniert jedoch auch über das normale Android-SDK, in welchem der Entwickler mit der Rechtevergabe auf die gewünschte Granularität zugreifen kann.[6]

Die Sensorik hat im Durchschnitt eine sehr hohe Aussagekraft über die aktuelle Aktivität des Nutzers. Über den Lagesensor können sowohl Standort als auch

die Bewegung ermittelt werden, dieses Prozedere kann zudem vom Beschleunigungssensor unterstützt werden (z.B. der Unterschied in den Beschleunigungen von rennen zu laufen). Der Näherungssensor kann lediglich zurückgeben, ob das Smartphone verdeckt ist oder nicht. Der Näherungssensor ist dadurch eine unterstützende Informationsquelle.

Touchscreenbewegung Der Touchscreen ist die physische Hauptbenutzerschnittstelle eines Smartphones. Daher wird dieser ständig frequentiert und lässt somit Aussagen auf Nutzerverhalten zu. In welcher Art und Weise der Touchscreen genutzt wird, kann daher einen großen Einfluss auf die Aktivität haben (Singletouch/Multitouch). Zudem hat er eine hohe Bedeutung bei der Optimierung von Apps. Die Touchscreenbewegung kann über theoretisch drei verschiedene Mechanismen aufgezeichnet werden:

- **Accessibility Service:** Hier wird ein Accessibility Service erstellt, der eigentlich den Zugriff für behinderte Menschen erleichtern soll. Hierbei wird anschließend auf die Events „TYPE_TOUCH_INTERACTION_START“ und „TYPE_TOUCH_INTERACTION_END“ gewartet und diese ausgewertet werden. Diese Touchevents werden jedoch bisher noch nicht ausgelöst, wodurch sich diese Möglichkeit selber wegrationalisiert.
- **Overlay:** Es wird ein Overlay über die Android-Oberfläche gelegt, welche Touchevents abfangen kann bzw. erhält. Dabei besteht jedoch das Problem, das anschließend das Android-System nicht mehr nutzbar ist, weil die Touchevents nicht mehr weitergereicht werden. Eine weitere Möglichkeit ist das Overlay auf 1x1 Pixel darzustellen und anschließend auf das Event „MotionEvent.ACTION_OUTSIDE“ zu hören. Diese Möglichkeit existierte noch bis Android 2.3.3, ab Android 4.0 werden jetzt nur noch die Koordinaten $(x,y) = (0,0)$ zurückgeliefert. Damit disqualifiziert sich auch diese Möglichkeit.
- **Aktivitätsbasis:** Bei der Touchscreenbewegungserkennung über eine Activity kann die dispatchTouchEvent-Methode überschrieben werden. Dadurch kann die Bewegung über das übergebene Event abgegriffen werden. Dies funktioniert jedoch lediglich auf allen selber implementierten Activities, jedoch nicht im gesamten System oder anderen Anwendungen.

Ein Abgreifen aller Touchevents ist aktuell nicht möglich und auch nicht vorgesehen, da es als Tapjacking-Schutz implementiert wurde.

Die Touchscreenbewegung hätte eine hohe Aussagekraft über die Tatsache wie der Nutzer das Smartphone benutzt und welche Apps er genau startet. Dies kann ebenfalls für eine Aktivitätserkennung nützlich sein, jedoch lediglich im Rahmen der aktiven Nutzung des Smartphones.

Displayaktivität Die Displayaktivität spielt eine wichtige Rolle, da sie eine solide Aussage zur aktuellen Interaktion mit dem Smartphone zulässt. Es ist daher relativ wahrscheinlich, dass wenn das Display angeht bzw. angeschaltet wird, der Nutzer in diesem Moment mit dem Smartphone interagiert. Diese

Möglichkeit wird vom Android System ohne Probleme unterstützt. Es bedarf dazu keiner zusätzlichen Berechtigung im Android-Manifest. Dabei wird ein einfacher BroadcastReceiver implementiert und registriert, der auf die Events „Intent.ACTION_SCREEN_OFF“ und „Intent.ACTION_SCREEN_ON“ lauscht und anschließend den jeweiligen Code ausführt. Hierbei gibt es vielfältige Möglichkeiten der Nutzung dieses Receivers.

Der Zugriff auf die bisherigen Displayzeiten bzw. Aktivitätszeiten des Smartphones ist seit Android 4.4 nicht mehr ohne Root-Rechte möglich, obwohl diese Statistik in den Einstellungen von Android nach wie vor vom Nutzer eingesehen werden kann. Daher kann auf diese Daten nicht mehr zugegriffen werden. Sie hätten Aussage über die allgemeine Nutzung des Smartphones und der Nutzerkategorie (z.B. Heavy-User) geben können.⁸

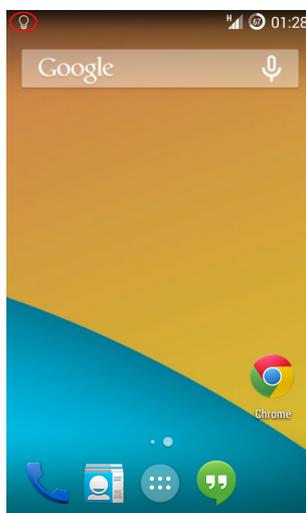


Abb. 1. Benachrichtigung in Android 4.4 (sh. links oben)

Die Displayaktivität hat ebenfalls eine sehr spezifische Aussagekraft über die Aktivität, die ebenfalls im Kontext der Smartphonebenutzung liegt. Dabei lässt sich jedoch valide feststellen wann und wie lange der Nutzer das Smartphone nutzt und ob er es zu diesem Zeitpunkt gerade nutzt. Diese Informationen können in einer Verknüpfung mit anderen Informationen ebenfalls zu einer Aktivitätserkennung beitragen.

Benachrichtigungen Benachrichtigungen sind ein zentrales Element eines Smartphone-Betriebssystems. Diese geben dem Nutzer die Möglichkeit, mit Apps zu interagieren, ohne dass die App geöffnet werden muss. Diese werden bei

⁸ <http://www.androidnext.de/news/android-4-4-kitkat-akkufresser-apps-mit-wakelocks-aufspueren-nicht-mehr-moeglich>

Android innerhalb der Statusbar angezeigt (sh. Bild 1). Der Zugriff auf die Benachrichtigungen lässt oft Schlüsse auf die Frequentierung und die eigentliche Nutzung des Smartphones zu. Die Möglichkeit, auf diese zuzugreifen, besteht allerdings erst seit Android 4.3. In diesem dritten Jelly Bean Update wurde der Zugriff auf alle Benachrichtigungen ohne Hilfe eines Accessibility Services ermöglicht. In der neuesten Android Version wurde zudem der Zugriff auf die Werte der jeweiligen Benachrichtigung deutlich erleichtert. Um auf alle Benachrichtigungen lesenden Zugriff zu erhalten, muss jedoch die Berechtigung „android.permission.BIND_NOTIFICATION_LISTENER_SERVICE“ gesetzt werden. Anschließend kann über das Implementieren eines NotificationListener-Service und eines dazugehörigen BroadcastReceivers auf die Benachrichtigungen zugegriffen werden und diese anschließend zu lesen oder löschen. Zudem können Methoden implementiert werden, die aufgerufen werden, wenn neue Notifications gepostet oder gelöscht werden. Diese lesende Fähigkeit ergänzt die Möglichkeit, selber Notifications zu posten.

Die Benachrichtigungen haben nur eine Teilaussagekraft bei der Aktivitätserkennung, hieraus lässt sich ableiten, wie häufig der Nutzer auf dem Smartphone frequentiert wird und welche Art von Benachrichtigungen ihn erreichen.

Zeit Die Zeit ist bei der Nutzung des Smartphones und allgemein der Aktivitätserkennung ebenfalls eine wichtige Komponente. Da ein Großteil der Menschen einen festen Rythmus hat, ist die Aktivitätserkennung 3 Uhr nachts weniger sinnvoll als 10 Uhr morgens. Der Zugriff auf die Zeit in Android gestaltet sich sehr einfach. Dabei wird eine Instanz von `java.util.Calendar` geholt und mit Hilfe dieser auf die wichtigen Daten der Zeit (z.B. „Calendar.HOUR_OF_DAY“, „Calendar.MINUTE“ oder „Calendar.SECOND“) zugegriffen. Eine Einordnung kann anhand dieser zurückgegebenen Werte anschließend erfolgen.

Die Zeit ist ein substanzieller Bestandteil, der mit anderen Informationen verbunden zur Aktivitätserkennung beitragen kann. Etwa kann eine lang gemessene Zeit zwischen zwei Displayaktivitäten zwischen 0 Uhr und 8 Uhr auf die Aktivität „Schlaf“ abgebildet werden.

3.3 Google-Dienste

Sowohl auf dem Android Smartphone, als auch auf anderen mobilen Plattformen ist es möglich, auf verschiedene Google-Dienste zuzugreifen. Dabei gibt es entweder speziell bereitgestellte Google SDKs für die jeweils darunterliegende Plattform, oder eine allgemeingültige API, bei der über einfache HTTP-Requests und Posts JSON Daten ausgetauscht werden können. Da in Kapitel 4 die beispielhafte Umsetzung einer Android-Applikation geschildert wird, wird im Folgenden verstärkt das Android SDK gezeigt. Ein Zugriff über eine normale API kann dennoch ebenfalls erfolgen. Um auf die nachfolgend genannten Google-Services zuzugreifen, muss eine Authentifizierung und Autorisierung erfolgen. Dabei muss ein Projekt unter `developer.google.com` erstellt werden, die notwendigen APIs aktiviert werden und die Android-Anwendung als zugreifender Client spezifiziert

werden. Der genaue Ablauf der Authentifizierung und Autorisierung in Android wird in Kapitel 4 ausführlich erläutert.

Google Calender Beim Google Calender handelt es sich um den integrierten Kalender, welcher direkt zu den Google-Diensten gehört, falls ein Google Konto erstellt wird. Zudem gehört er zu der Basisaustattung eines jeden Android-Gerätes, falls es sich um ein von Google zertifiziertes Gerät handelt. Durch diese Tatsachen gibt es drei verschiedene Möglichkeiten des Zugriffs:

- **Lokaler Zugriff:** Der lokale Zugriff auf den Google Calender geschieht über den Content-Provider der installierten Calender-App. Dieser Content-Provider kann über den String „content://com.android.calendar“ geholt werden. Auf diesen Content-Provider können anschließend SQL artige Befehle ausgeführt werden, welche anschließend die jeweiligen Events/Instances zurückliefern. Der große Vorteil dieser Methode ist das fehlende Anfragen der REST-API. Es kommt somit also nicht zu einem direkten Netzwerkverkehr. Dabei wird sich die Synchronisierung der Google-Services zu Nutze gemacht. Dadurch ist zudem direkt ersichtlich, dass über die App auf den Google Calender zugegriffen wird, da die Android Berechtigungen „android.permission.READ_CALENDAR“ benötigt wird. Von Nachteil ist diese Methode genau dann, wenn der Nutzer die automatische Synchronisation des Calenders abgestellt hat, da dann nicht die aktuellsten Daten zurückgegeben werden. Zudem muss vorher überprüft werden, ob der Google Calender überhaupt installiert ist.
- **Google Calender API für Java:** Die Google Calender API existiert für verschiedene Programmiersprachen, u. a. auch für Java. Diese nutzt intern die normale REST-API, vereinfacht jedoch den Zugriff auf diese. Dazu muss die Google Calender API im jeweiligen Google Developer Project explizit eingeschaltet werden. Für einen Zugriff aus einer Android-Applikation heraus müssen zudem die Berechtigungen „android.permission.INTERNET“ angefragt werden (falls vorher erfolgreich authentifiziert wurde). Der Vorteil dieser Methode ist die garantierte Aktualität der zurückgelieferten Accounts. Problematisch ist diese Methode bei vielen Zugriffen auf diese API, da die maximale kostenlose Nutzung nicht 100.000 Requests am Tag überschreiten darf. Falls mehr nötig sind, muss eine preislich nicht bestimmte Gebühr an Google entrichtet werden [7].
- **REST-API:** Die REST-API kann ebenfalls von jedem Gerät aus und von jeder Programmiersprache genutzt werden. Hier werden ebenfalls lediglich die Internet-Berechtigungen benötigt. Die Anwendung muss dennoch auf `developer.google.com` eingetragen werden. Anschließend ist es möglich, direkt POST, GET, PUT, oder DELETE Befehle über HTTP auszuführen. Diese Möglichkeit des Zugriffs ist jedoch eher dann vonnöten, wenn keine spezielle API für die verwendete Programmiersprache existiert.

Der Zugriff auf den Google Calender wurde in Kapitel 4 über den lokalen Zugriff realisiert, da sie auf der offiziellen Android Developer-Website [6] als Guideli-

ne empfohlen wird und keine maximale Anfragehäufigkeit bei dieser Methode existiert.

Der Google Calendar hat bei einem hohen Nutzungsgrad und dem eventuellen einpflegen von Labels und Terminstandorten eine sehr hohe Aussagekraft auf die aktuelle Aktivität des Nutzers. Dieses Vorgehen kann bspw. bei Google Now beobachtet werden, der den Kalender als Hauptbestandteil mit einbezieht.

Google+ Bei Google+ handelt es sich um das von Google initiierte Social-Network, welches in Deutschland und der Welt nach wie vor steigende Mitglieder Zahlen hat⁹. Der Zugriff auf Google+ kann ebenfalls über drei verschiedene Möglichkeiten implementiert werden (Google+ API muss unter `developer.google.com` aktiviert sein):

- **Zugriff über die Google-Play-Services:** Die Verbindung und der Zugriff auf Google+ kann über die Google-Play-Services geschehen. Dabei wird ein `PlusClient` initialisiert und auf diesem die `connect`-Methode aufgerufen. Dabei muss zudem überprüft werden, ob die `Connection` erfolgreich aufgebaut wurde oder nicht. Dazu müssen in der jeweiligen Klasse die Interfaces `ConnectionCallback` und `OnConnectionFailed` implementiert werden. Dieses Verfahren gleicht daher dem Zugriff auf die Facebook-Services des Facebook Android SDKs. Hier werden jedoch die Google-Play-Services vorausgesetzt (sind jedoch standardmäßig installiert). Dies ist das aktuellste Verfahren und wird auf der offiziellen Android Entwickler Seite empfohlen [6] die beispielhafte Implementierung unter Kapitel 4 wurde jedoch mit Hilfe der Google+ API implementiert.
- **Zugriff über die Google+ API:** Die Google+ API ist nur eine Kapselung der REST-API und stellt einen einfacheren Zugriff auf diese dar. Dabei wird statt einem `PlusClient` ein `Plus`-Objekt erzeugt. Über dieses können anschließend die Daten abgerufen werden. Hierbei muss kein weiteres Interface implementiert werden, hat also einen geringeren Implementierungsaufwand aber eine geringere Fehlerbehandlungsmöglichkeit. Über dieses Objekt können anschließend verschiedene Daten abgerufen werden, dabei müssen die Rechte jedoch beachtet werden.
- **Zugriff über die REST-API:** Wie auch beim Google Calendar Zugriff, können auch hierüber Daten abgerufen werden. Dabei ist jedoch nur ein GET möglich. Im Gegensatz zur Facebook-Graph-API gibt es hier ohne eine eigene Anmeldung keine Möglichkeit, Requests abzusetzen.

Google+ bietet insgesamt lediglich vier verschiedene Berechtigungen an [7]:

- **PLUS_LOGIN:** Ist die Standard-Einstellung, mit der die Anmeldung erfolgen sollte. Diese beinhaltet den Zugriff auf das Profil (auch `PLUS_ME`) und den Zugriff auf die Personen in den eigenen Kreisen.

⁹ <http://www.socialmediastatistik.de/300-millionen-nutzer-im-stream-google-macht-grossen-sprung/>

- **PLUS_ME:** Beinhaltet die Berechtigung, zu erkennen, wer auf Google+ zugreift. Dies beinhaltet die Ausführung von Requests mit dem Keyword „me“. Außerdem können mit dieser Berechtigung auch die öffentlichen Posts des Nutzers ausgelesen werden.
- **USERINFO_EMAIL:** Ermöglicht den Zugriff auf die Email-Adresse.
- **USERINFO_PROFILE:** Ermöglicht das Abfragen von Basisinformationen des Profils. Dies ist equivalent zur alten „Profile“ Berechtigung. Diese existiert zudem aus Kompatibilitätsgründen zu Google Profiles [7].

Durch diese Berechtigungen ist festzuhalten, dass Google sehr wenige Informationen für Anwendungsentwickler bereitstellt. Diese kommen nicht annähernd an die Informationsdichte der Facebook-Informationen heran. Zum Beispiel ist ein Auslesen der privaten Posts oder des Streams mit Google+ nicht möglich. Dadruch ist Google+ im Gegensatz zu Facebook und Twitter für eine Aktivitätserkennung eher ungeeignet, da weder viele Informationen ausgelesen werden können, noch besonders viele Personen öffentlich kommunizieren. Die Verknüpfung von privaten Profilverinformationen zu anderen Informationen ist jedoch genau so gut möglich wie bei Facebook oder Twitter.

Google Tasks Bei Google Tasks handelt es sich um die Aufgabenverwaltung von Google, welche direkt in den Google Calendar integriert ist. Sie dient lediglich dafür, Aufgaben für ein bestimmtes Datum zu erstellen und sie als erledigbar zu kennzeichnen. Google Tasks ist ein Teil der Google-Services, der zwar nach wie vor existiert, aber schon länger kein Update mehr erhalten hat. Die Google Tasks werden vmtl. demnächst durch Google Keep und die Erinnerungsfunktion von Google Now ersetzt. Der Zugriff auf die Google Tasks kann sowohl über eine Java-API ausgeführt werden oder über eine REST-API.

Der Zugriff erfolgt im Android Kontext über die Java-API. Daber wird ein Tasks-Objekt erzeugt, über welches anschließend die fälligen Tasks abgerufen oder erstellt werden können. Dabei gibt es die zwei Berechtigungen TASKS und TASKS_READONLY, welche den lesenden oder schreibenden Zugriff ermöglichen.

Google Tasks kann eine hohe Aussagekraft für die aktuelle Aktivität haben, vorausgesetzt Google Tasks wird oft benutzt. Dennoch fehlt es Google Tasks an vielen wichtigen Informationen, da die Aufgaben lediglich Titel und Beschreibung sowie einen Termin (lediglich in Tagesgranularität) besitzen. Das Festlegen von Standorten oder anderen Personen ist nicht möglich, wodurch der Einsatzradius relativ beschränkt ist.

Google Mail Bei Google Mail handelt es sich um einen der ersten Services, die von Google angeboten wurden. Google Mail wurde im April 2004 angekündigt und ging 2007 in eine offene Betaphase, bevor es 2009 endgültig veröffentlicht wurde¹⁰. Google Mail ist oft der zentrale Einstiegsdienst in die Welt der Google

¹⁰ <http://googlepress.blogspot.de/2004/04/google-gets-message-launches-gmail.html>

Services. Der Zugriff auf Mails ist über einen Content-Provider möglich, der genauso funktioniert wie der von Kapitel 3.3. Eine normale API existiert jedoch nicht, sodass der Zugriff ausschließlich auf Android Smartphones möglich ist.

E-Mails können starke Aussagekraft auf die Aktivität haben, da E-Mail zu einem der zentralen Kommunikationsmitteln gehört. Über das Auslesen der Sendungshäufigkeit oder der Labels lassen sich Aussagen über die Art und Weise der Smartphone-Nutzung treffen (z.B. Label Arbeit oder Studium).

Google Drive Google Drive ist der Personal Cloud Storage Dienst von Google und wurde aus Google Docs herausentwickelt bzw. zusammengeführt. Dadurch lassen sich neue Dateien erstellen und im Web bearbeiten. Außerdem existieren Desktop-Tools, die es ermöglichen, einen speziellen Ordner in Google Drive zu synchronisieren. Die Relevanz für die Aktivitätserkennung ist jedoch nicht so hoch, da sich weniger Aussagen über die jeweilige Aktivität des Nutzers treffen lassen. Die Relevanz könnte jedoch demnächst steigen, da durch eine neue Google Drive Version ein Aktivitätsstream hinzugekommen ist¹¹, der es ermöglicht, die letzten Bearbeitungen von Dateien einzusehen. Falls eine API veröffentlicht wird, die Zugriff auf dieses Feature gewährleistet, wäre dies für die Aktivitätserkennung durchaus von Bedeutung. Im Rahmen dieses Updates wurde auch eine neue Google Drive API als Developer Preview vorgestellt¹². Dadurch gehört die Google Drive API nun auch zu den Google-Play-Services und kann über das Nutzen des Google-Play-Services SDK auch für die Anwendungsentwicklung genutzt werden. Vorher wurde der Zugriff über eine eigens entwickelte API geregelt.

Andere Google-Dienste Neben den bereits oben genannten Google-Diensten stehen noch viele weitere Google-Dienste zur Verfügung. Teile davon können mit einer API oder einer anderen Zugriffsmöglichkeit, wie dem Content Provider, abgegriffen werden und bieten somit ebenfalls den Zugriff auf Daten. Dabei sind jedoch einige Hauptdienste wie Youtube, Google Maps oder Google Contacts nicht von einer großen Bedeutung für die Aktivitätserkennung. Diese wurden somit auch nicht in der Beispielimplementierung aus Kapitel 4 berücksichtigt.

Andere zentrale Dienste können bisher noch nicht über eine API angesteuert werden. Dazu gehören unter anderen Google Now, den persönlichen Assistenten auf jedem Android Smartphone, und Google Keep, der Notizen-Anwendung von Google. Dies sind Dienste, die erst vor Kurzem von Google herausgegeben wurden und für die daher noch keine öffentliche API existiert. Es existieren jedoch teilweise experimentelle APIs in einschlägigen Entwicklerforen, welche jedoch Root benötigen und zudem das Installieren spezieller Anwendungen erfordern¹³. Dadurch sind sie nicht massentauglich. Nichtsdestotrotz werden nach und nach

¹¹ http://business.chip.de/news/Google-Drive-Jetzt-auch-mit-Activity-Stream_66591489.html

¹² <http://googledevelopers.blogspot.de/2014/01/introducing-google-drive-android-api.html>

¹³ <http://forum.xda-developers.com/showthread.php?t=2554173>

neue APIs zu den dazugehörigen Dienste veröffentlicht, sodass demnächst auch Zugriff auf diese Dienste ermöglicht werden könnte.

Zudem gibt es noch einige Services von Google, die nicht zu den Google Services gehören und bisher nur vom Nutzer eingesehen werden können. Dazu gehören unter anderen der Google Suchverlauf und der Google Standortverlauf, welcher von Google Now zum Anzeigen von Karten benutzt wird. Zugriff auf diese Verläufe könnte massiven Einfluss auf eine Aktivitätserkennung haben.

3.4 Andere Dienste

Neben den Social Networks und den Google Services existieren viele weitere Cloud-Dienste, die ebenfalls auf dem Smartphone abgegriffen werden können. Es gibt eine Vielzahl von Wetter-Services, Sport-Services und weiteren Diensten, die Aussage über die Aktivität bringen können. Prinzipiell lassen sich alle Dienste ansteuern, die eine API bieten, egal ob REST-API oder eine spezielle Programmiersprache bzw. Platform-API. Im Nachfolgenden werden zwei Dienste vorgestellt, über die Daten ausgelesen werden können. Dies lässt sich auf alle anderen Dienste mit APIs übertragen.

OpenWeatherMap Bei OpenWeatherMap¹⁴ handelt es sich um eine freie und kostenlose Alternative, um Wetterdaten zu bestimmen. Dabei bietet OpenWeatherMap eine REST-API an, über die Anfragen getätigt werden können. Der Aufruf der API sieht folgendermaßen aus:

```
api.openweathermap.org/data/2.5/weather?lat=XX&lon=XX
```

Anschließend wird eine JSON bzw. XML Datei zurückgeliefert welches die angefragten Daten enthält. Dabei gibt es verschiedene Konfigurationen und Anfragemöglichkeiten.

Der einfachste Weg OpenWeatherMap in eine Android-Anwendung einzubauen, ist das Aufrufen der API über die Longitude- und Latitude-Werte, gemäß dem beispielhaften Aufruf. Die Longitude- und Latitude-Werte stammen aus der Antwort des Lagesensors (sh. Kapitel 3.2). Die Antwort der API muss anschließend traversiert und ausgewertet werden.

Wetterinformationen haben einen sehr geringen Einfluss auf die Aktivitätserkennung. Nur durch die Verknüpfung mit weiteren Informationen ist eine Abbildung einer Aktivität denkbar. Beispielsweise bei einer Bewegung die sowohl zu einem Fahrrad als auch einer S-Bahn innerhalb einer Stadt passen würde. Durch die Auswertung des Wetters kann somit zumindest wahrscheinlich ausgeschlossen werden, dass bei einen starkem Sturm der Nutzer das Fahrrad benutzt.

Dropbox Dropbox gehört zu den führenden Cloud-Speicher-Anbietern auf der Welt und wurde offiziell im Jahre 2010 veröffentlicht¹⁵. Um auf diesen Dienst

¹⁴ <http://openweathermap.org/>

¹⁵ <http://t3n.de/news/dropbox-erreicht-version-10-290884/>

zugreifen zu können, gibt es eine Vielzahl von APIs und SDKs. Dabei unterstützt Dropbox sowohl eine Vielzahl von Programmiersprachen, als auch die wichtigsten mobilen Betriebssysteme (u.a. Android).

Eine Android-Implementierung benötigt lediglich das Android Recht „android.permission.INTERNET“. Zudem muss ein App-Key, ähnlich der Application-ID von Facebook, in das Android-Manifest eingetragen werden. Nach Herunterladen und Eintragen des SDKs kann es schließlich benutzt werden¹⁶. Anschließend ist es möglich, Dateien hochzuladen, auszulesen oder herunterzuladen.

Dropbox wurde in die Implementierung, welche im nachfolgenden Kapitel vorgestellt wird, nicht mit einbezogen, da Cloud-Speicher dennoch eine geringe Aussagekraft über die Aktivität des Nutzers haben.

4 Umsetzung

Um die in Kapitel 3 aufgeführten Dienste abzugreifen und zu nutzen, wurde eine Beispielumsetzung in Form der App „Activitytracker“ erstellt. Der folgende Abschnitt handelt von der App selbst und deren Aussehen und den Implementierungsdetails der jeweiligen Zugriffe auf die verschiedenen APIs bzw. der Nutzung der zur Verfügung stehenden SDKs.

4.1 Main- und LocalActivity



Abb. 2. Haupt-Activity und Activity zum Zugriff auf lokale Daten und Sensoren

¹⁶ <https://www.dropbox.com/developers/core/sdks/android>

Der zentrale Einstiegspunkt der App ist die MainActivity (sh. Abbildung 2(a)). Hier können die drei Hauptbestandteile der App ausgewählt werden, der Zugriff auf die lokalen Daten und Sensoren, der Zugriff auf die Google-Dienste und der Zugriff auf Facebook-Daten. Dabei stellt die MainActivity lediglich drei Buttons dar, die wiederum die anderen Activities starten. Wird von hier aus nun der Button „Get Local Access“ aufgerufen, öffnet sich zunächst der Rechedialogue für die Benachrichtigungen (sh. Abbildung 2(b)). Dieser wird wie eine normale Activity über einen Intent gestartet. Der Aufruf sieht folgendermaßen aus:

```
Intent intent = new
    Intent("android.settings.ACTION_NOTIFICATION_LISTENER_SETTINGS");
startActivity(intent);
```

Über den Zurück-Button des Android Systems kann anschließend zu der Local-Activity zurückgekehrt werden (sh. Abbildung 2(c)). Auf dieser Activity existieren Buttons und Switches, welche gewisse lokale Daten zurückliefern oder Sensoren ansteuern. Dabei wird über Switches ein bestimmtes Verhalten ein- bzw. ausgeschaltet, über Buttons eine Ausgabe bewirkt. Da es sich um ein Proof of Concept handelt, werden die Ergebnisse der jeweiligen Anfragen über einfaches Logging auf die Console ausgegeben. Dieses Logging geschieht bei allen Ausgaben über:

```
Log.v(TAG, Message);
```

Das Tag stellt dabei immer den entsprechenden Klassennamen und die Message den jeweiligen auszugebenden Wert dar.

Zeit Die Zeit wird über eine normale Instanz von Calendar ausgegeben, dabei gibt Calendar.HOUR_OF_DAY die Zeit im 24h-Format zurück. Über diesen Calendar können jedoch auch viele andere Calendar-Daten abgerufen werden, wie das Datum in verschiedenen Formen.

```
Calendar calender = Calendar.getInstance();
Log.v(TAG, calender.get(Calendar.HOUR_OF_DAY) + ":" +
    calender.get(Calendar.MINUTE) + "." + calender.get(Calendar.SECOND));
```

Benachrichtigungen Um die Benachrichtigungen auslesen zu können, wird eine Klasse benötigt, die den NotificationListenerService auf folgende Art implementiert:

```
public class NotificationService extends NotificationListenerService {

@Override
public void onCreate() {
    super.onCreate();
```

```
notificationReceiver = new NotificationServiceReceiver();
registerReceiver(notificationReceiver, filter);
}}
```

Die beiden Methoden `onNotificationPosted` und `onNotificationRemoved` müssen implementiert werden, spielen aber beim Auslesen der `Notifications` keine Rolle. Die aktuellen `Notifications` müssen über den `BroadcastReceiver` ausgelesen werden, da sonst ein `NullPointerException` zurückgegeben wird. Beim `NotificationServiceReceiver` handelt es sich um eine interne Klasse, die von `BroadcastReceiver` erbt und dadurch die `onReceive`-Methode implementiert. In dieser können alle aktuellen Benachrichtigungen folgendermaßen abgegriffen werden:

```
@Override
public void onReceive(Context context, Intent intent) {
    StatusBarNotification[] notifications =
        NotificationService.this.getActiveNotifications();
}
```

Calendar Der Zugriff auf den Google Calendar wurde über die Content-Provider-Möglichkeit implementiert. Dazu muss ein `ContentResolver` erzeugt werden, welcher eine `URI`, die `String`-Abfrage und die `Projektion` übergeben bekommt. Dabei ist die `Abfrage` in eine `Hauptabfrage` und den dazugehörigen `Parametern` unterteilt. Das `String-Array` „`INSTANCE_PROJECTION`“ gibt an, welche `Spalten` zurückgeleifert werden sollen. Die `URI` gibt an, dass die `Anfrage` an den `Google Calendar` gestellt wird. Nach `Ausführen` der `Abfrage` wird ein `Cursor` zurückgegeben, über welchen anschließend iteriert werden kann.

```

public static final String[] INSTANCE_PROJECTION = new String[] {
    Instances.EVENT_ID, Instances.BEGIN,
    Instances.END, Instances.TITLE};

ContentResolver cr = getContentResolver();
String calUri = "content://com.android.calendar/instances/when";
Uri uriInst = Uri.parse(calUri).buildUpon().build();
String selectionInst = "(" + Instances.BEGIN + " > ?) AND (" +
    Instances.BEGIN + " < ?)";
String[] selectionInstArgs = new String[]
    {String.valueOf(System.currentTimeMillis()),
    String.valueOf(System.currentTimeMillis() + 86400000)};

Cursor curInst = cr.query(uriInst, INSTANCE_PROJECTION, selectionInst,
    selectionInstArgs, null);

```

Location Die Abfrage des Standorts wird über die `GooglePlayServices` implementiert. Dabei müssen die beiden Interfaces „`ConnectionCallbacks`“ und „`OnConnectionFailedListener`“ implementiert werden. Dadurch müssen vier verschiedene Methoden implementiert werden. Die wichtigste ist jedoch die Methode `onConnected()`. In dieser können vom `LocationClient` die jeweiligen Informationen über den Standort abgefragt und ausgewertet werden. Eine eventuelle Weiterverarbeitung ist dabei ebenfalls denkbar (z.B. für `OpenWeatherMap`).

```

public class LocationService extends IntentService implements
    GooglePlayServicesClient.ConnectionCallbacks,
    GooglePlayServicesClient.OnConnectionFailedListener {

    @Override
    public void onCreate() {
        super.onCreate();
        locationClient = new LocationClient(this, this, this);
        locationClient.connect();
    }

    @Override
    public void onConnected(Bundle arg0) {
        Location lastLocation = locationClient.getLastLocation();
    }
}

```

Proximity Sensor Zum Auslesen eines Sensors muss ein `SensorEventListener` implementiert werden. Durch diesen müssen die Methoden `onSensorChanged()` und `onAccuracyChanged()` implementiert werden. Dabei ist vor allem die erste Methode die wichtige, da hierrüber die Koordinaten des Sensors ausgelesen werden können. Zuvor müssen ein `SensorManager` und ein `Sensor` implementiert werden.

```
public class ProximityService implements SensorEventListener {

    @Override
    public void onCreate() {
        super.onCreate();
        this.sensorManager = (SensorManager)
            getSystemService(Context.SENSOR_SERVICE);
        this.sensor =
            this.sensorManager.getDefaultSensor(Sensor.TYPE_PROXIMITY);
        this.sensorManager.registerListener(this, sensor,
            SensorManager.SENSOR_DELAY_UI);
    }

    @Override
    public void onSensorChanged(SensorEvent event) {
        float x = event.values[0];
        float y = event.values[1];
        float z = event.values[2];
    }
}
```

Accelerometer Der Beschleunigungssensor wird genau wie der Näherungssensor ausgelesen. Lediglich eine Zeile muss folgendermaßen geändert werden:

```
this.sensor =
    this.sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
```

Anschließend kann mit den Daten genau wie beim Näherungssensor weitergearbeitet werden.

Touch Detection Das Auslesen der Touchscreen-Berührungen funktioniert seit Android 4.0 nur noch in den eigenen Activities. Dies funktioniert über die einfache Überschreibung der `dispatchTouchEvent`-Methode. In dieser werden alle `TouchEvent`s zurückgegeben, mit denen anschließend weitergearbeitet werden kann. In der Beispielanwendung wurde eine `AbstractActivity` implementiert welche lediglich diese Methode überschreibt. Anschließend erben alle weiteren Activities von dieser `AbstractActivity`.

```
@Override
public boolean dispatchTouchEvent(MotionEvent ev) {
    Log.v(TAG, ev.getAxisValue(MotionEvent.AXIS_X) + " " +
        ev.getAxisValue(MotionEvent.AXIS_Y));
    return super.dispatchTouchEvent(ev);
}
```

Screen Activity Um die Displayaktivität abzugreifen, wird ein BroadcastReceiver benötigt, welcher auf die Events „Intent.ACTION_SCREEN_OFF“ und „Intent.ACTION_SCREEN_ON“ hört. Um einen BroadcastReceiver zu implementieren, muss die onReceive-Methode implementiert werden. In dieser muss auf das jeweilige Event geprüft werden und kann anschließend verwendet werden.

```
@Override
public void onReceive(Context context, Intent intent) {
    if (intent.getAction().equals(Intent.ACTION_SCREEN_OFF)) {
        Log.v(TAG, "Screen turned off");
    } else if (intent.getAction().equals(Intent.ACTION_SCREEN_ON)) {
        Log.v(TAG, "Screen turned on");
    }
}
```

Damit der ScreenReceiver die Events bekommt, muss der Receiver erzeugt werden und anschließend registriert werden. Damit genau die beiden Events zurückgeliefert werden, muss zudem noch ein Filter übergeben werden, welcher angibt welche Events gefiltert werden. Nach Registrieren des Filters wird die onReceive-Methode jedes mal aufgerufen, wenn das jeweilige Event ausgelöst wird. Falls die Funktionalität nicht mehr benötigt wird, kann der ScreenReceiver mit der unregisterReceiver-Methode wieder entfernt werden.

```
IntentFilter filter = new IntentFilter(Intent.ACTION_SCREEN_ON);
filter.addAction(Intent.ACTION_SCREEN_OFF);
ScreenReceiver screenReceiver = new ScreenReceiver();
registerReceiver(screenReceiver, filter);
```

4.2 Google Activity

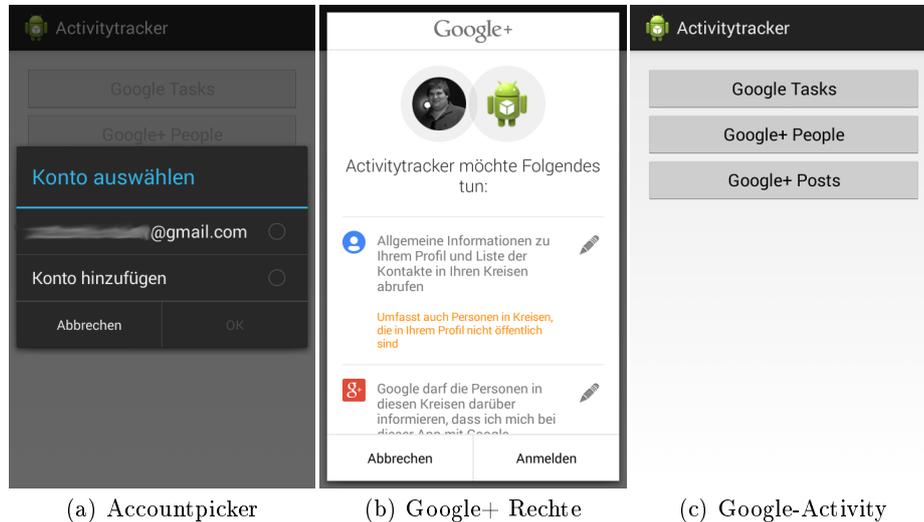


Abb. 3. Google-LogIn in der App

Die Google Activity (sh. Abbildung 3(c)) beinhaltet die gesamte Kommunikation mit den Google Services, dazu gehört das Authentifizieren mit einem Google-Account und das Autorisieren für den Zugriff auf die jeweiligen angeforderten Informationen. Nach einem Klick auf dem Google-Button (sh. Abbildung 2(a)) wird in der onCreate-Methode der Google Activity der Google Zugriff initiiert. Dabei werden die Permissions für die Kommunikation gesetzt, das credential erzeugt und gesetzt und die Clients für den Zugriff auf Informationen erstellt. In diesem Zustand wurde noch kein Account ausgewählt, dieser wird anschließend über das Starten des `credential.newChooseAccountIntent()` ausgewählt. Dabei wird der Accountpicker (sh. Abbildung 3(a)) gestartet, auf dem entweder ein konfigurierter Account des Smartphones ausgewählt werden kann oder die Wahl besteht, sich mit einem neuen Account anzumelden.

```
private void initializeGoogleAccess() {
    Collection<String> permissions = new ArrayList<String>();
    permissions.add(TasksScopes.TASKS);
    permissions.add(PlusScopes.PLUS_LOGIN);
    credential = GoogleAccountCredential.usingOAuth2(this, permissions);
    taskService = new Tasks.Builder(AndroidHttp.newCompatibleTransport(),
        GsonFactory.getDefaultInstance(), credential).build();
    plusService = new Plus.Builder(AndroidHttp.newCompatibleTransport(),
        GsonFactory.getDefaultInstance(), credential).build();
}
```

```

startActivityForResult(credential.newChooseAccountIntent(),
    REQUEST_ACCOUNT_PICKER);
}

```

Dieser Intent wird über die Methode `startActivityForResult` gestartet. Dabei wird der entsprechende Intent und ein Integer übergeben. Es wird diese Methode gewählt, da erwartet wird, dass nach dem Aufruf des Accountpickers zu der `GoogleActivity` zurückgekehrt wird. Mit dem übergebenen Integer kann in der `onActivityResult`-Methode identifiziert werden, welche Activity gestartet wurde. In dem speziellen Fall holen wir uns den ausgewählten Accountnamen, setzen diesen im `Credential` und blenden alle Buttons der Activity ein.

```

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent
    data) {
super.onActivityResult(requestCode, resultCode, data);
switch (requestCode) {
    case REQUEST_ACCOUNT_PICKER:
        if (resultCode == Activity.RESULT_OK && data != null &&
            data.getExtras() != null) {
            String accountName =
                data.getExtras().getString(AccountManager.KEY_ACCOUNT_NAME);
            if (accountName != null) {
                credential.setSelectedAccountName(accountName);
                enableButtons(true);
            }
        }
        break;
}
}

```

Die Buttons starten dann im Hintergrund asynchrone Tasks, sodass die Abfrage des `WebServices` asynchron zum eigentlichen UI-Thread geschieht. Die jeweiligen Requests werden dann ganz einfach über die erzeugten `taskService` und `plusService` abgesetzt. Anschließend werden, je nach Anfrage, eine Liste von `Google+ Personen`, `Google+ Posts` (nur öffentlich) oder `Google Tasks` zurückgegeben.

```

List<Person> people =
    plusService.people().list("me", "visible").execute().getItems();
List<Activity> activities =
    plusService.activities().list("me", "public").execute().getItems();
List<Task> tasks =
    taskService.tasks().list("@default").execute().getItems();

```

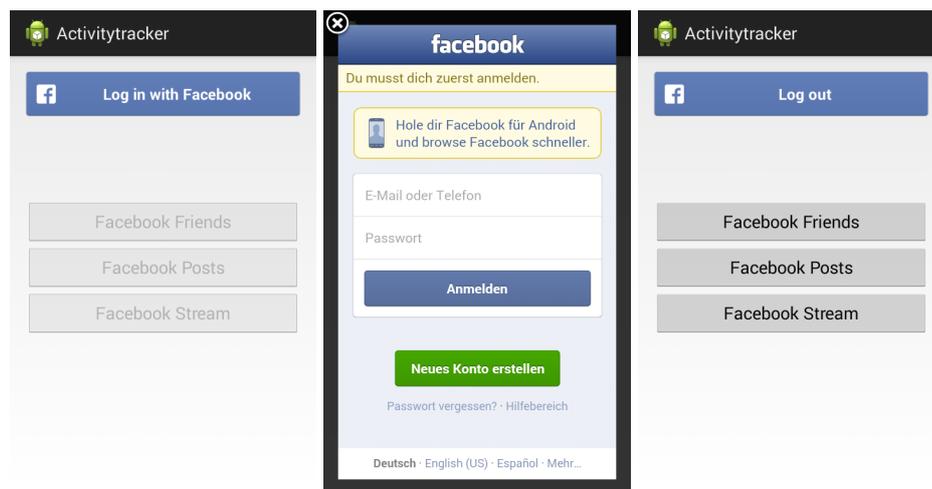
Beim ersten Ausführen der Requests wurden die Permissions noch nicht angefragt und der Nutzer konnte sie noch nicht bestätigen. Dadurch wird eine Exception geworfen, welche über ein Try-Catch abgefangen wird. Im Catch wird dann der Bestätigungsdialog für die Permissions aufgerufen (sh. Abbildung 3(b)).

```

catch (UserRecoverableAuthIOException userRecoverableException) {
    startActivityForResult(userRecoverableException.getIntent(),
        GoogleLoginActivity.REQUEST_AUTHORIZATION);
}

```

4.3 Facebook Activity



(a) Facebook-Activity (nicht angemeldet) (b) Anmeldungsdialog (c) Facebook-Activity (angemeldet)

Abb. 4. Facebook-LogIn in der App

In der Facebook-Activity ist die Anmeldung und die Abfrage von Daten von Facebook über das Facebook-SDK realisiert. Um sich bei Facebook anzumelden, wurde hier die Implementierung über den Facebook-Button gewählt. Dieser stellt sowohl einen speziellen Button in der Layout.xml dar, als auch einen speziellen LoginButton in der Programmierlogik. Wichtig ist hierbei das Setzen der Application-ID (sh. Kapitel 3.1). Beim UIHelper handelt es sich um ein Objekt, was die Verwaltung der Session vereinfacht und den Button jeweils passend zum LogIn-Status verändert.

```

uiHelper = new UiLifecycleHelper(this, null);
uiHelper.onCreate(savedInstanceState);
LoginButton loginButton = (LoginButton) findViewById(R.id.authButton);
loginButton.setApplicationId("1375505019377863");
loginButton.setUserInfoChangedCallback(new
    LoginButton.UserInfoChangedCallback() {

```

```

@Override
public void onUserInfoFetched(GraphUser user) {
    if (user != null) {
        Log.v(TAG, user.getFirstName() + " " + user.getLastName());
        updateButtons();
    }
}
}
}

```

Über den Druck auf den LogIn-Button wird der Anmeldedialog von Facebook aufgerufen (sh. Abbildung 4(b)). Ist die Anmeldung erfolgreich, werden die Buttons aktiviert und der Facebook-Button wechselt seine Erscheinung (sh. Abbildung 4(c)). Durch diese Initiale Anmeldung wird der Zugriff auf die öffentlichen Profildaten und die Freundesliste möglich. Falls die Anwendung noch unbekannt ist, wird um Bestätigung dieser Berechtigung gebeten (sh. Abbildung 5(a)). Anschließend kann ein speziell von dem Facebook-SDK bereitgestellter Request abgesetzt werden, welcher alle Freunde zurückgibt.

```

Request request =
    Request.newMyFriendsRequest(Session.getActiveSession(), new
        Request.GraphUserListCallback() {

            @Override
            public void onCompleted(List<GraphUser> users, Response response) {
                Log.v(TAG, response.toString());
                for (GraphUser graphUser : users) {
                    Log.v(TAG, "ID: " + graphUser.getId() + " Name: " +
                        graphUser.getName());
                }
            }
        });

request.executeAsync();

```

Um die anderen Daten abzurufen, wird ein normaler `newGraphPathRequest` benötigt, bei dem spezifiziert werden muss, was zurückkommen soll. Dieser Wert ist für die Posts „me/posts“ und für den Stream „me/home“. Anschließend wird bei einem normalen Request lediglich ein JSON-Response zurückgeliefert, was schließlich ausgewertet werden muss. Diese Daten können nicht ohne eine extra Berechtigung abgefragt werden. Dazu müssen auf der aktiven Session neue Leseberechtigungen angefragt werden. Dabei wurde die Berechtigung zum Lesen der Posts mit „user_status“ und die zum Lesen des Streams mit „read_stream“ festgelegt. Diese führen anschließend zu einem neuen Dialog, bei dem der Nutzer die Berechtigungen erneut bestätigen muss (sh. Abbildung 5(b) und 5(c)).

```

Session.getActiveSession().requestNewReadPermissions(new
    Session.NewPermissionsRequest(FacebookLoginActivity.this,
        "user_status"));
Request request =
    Request.newGraphPathRequest(Session.getActiveSession(), "me/posts"
        ,new Request.Callback() {

```

```

@Override
public void onCompleted(Response response) {
    Log.v(TAG, response.toString());
    });

request.executeAsync();

```



Abb. 5. Berechtigungsdialoge für Facebookzugriff

5 Fazit und Ausblick

Jedes Smartphone bietet die Schnittstelle zu zahlreichen Informationen. Egal ob über Sensoren oder Accounts und den zugehörigen Zugriffsmöglichkeiten über das Internet. Dabei ist der Zugriff auf die Daten mit den von Herstellern bereitgestellten APIs und SDKs relativ leicht. Google versucht zudem, mit der Platzierung der Google-Play-Services eine Schnittstelle unabhängig zur Android-Version anzubieten, die zusätzlich den Zugriff zu den wichtigsten Services bietet. Diese Strategie wird vermutlich in den nächsten Monaten noch weiter verfolgt, bis der Großteil der Neuerungen unabhängig von der eigentlich ausgebrachten Android-Version ist. Dies hat den Vorteil für die Anwender, dass sie die neuesten Änderungen auch mit einem älteren Smartphone bekommen und den Vorteil für die Entwickler, dass nicht mehr zu stark auf die Fragmentierung geachtet werden muss. Zudem werden in Zukunft weitere neue Sensoren und Weiterentwicklungen von Sensoren Einzug in Smartphones erhalten. Dabei ist das Galaxy S4 erst der erste Schritt in Richtung eines komplexen Geräts mit einer Fülle von

Sensoren. In Zukunft werden Smartphones entweder noch mehr Sensoren enthalten¹⁷ oder sich mit einer Fülle von tragbaren Geräten verbinden können (z.B. Sony SWR10/Core¹⁸). Außerdem werden weiterhin innovative Dienste für den Nutzer erscheinen, welche heutzutage häufig auf eine Cloud-Unterstützung setzen. Dazu gehören bspw. die Felder der Spracherkennung (z.B. Siri und Google Now) und der Gesichtserkennung¹⁹. Dadurch ist es relativ wahrscheinlich, dass für die führenden Smartphone-Betriebssysteme ebenfalls eine API oder ein SDK erscheint, über welche diese Dienste in andere Apps integriert werden können. Wie das Proof of Concept beweist, können viele relevante Daten ohne großen Aufwand von einem Smartphone heraus ausgelesen werden. Voraussetzung ist dabei jedoch prinzipiell die Einwilligung des Nutzers. Grundsätzlich existieren einige Informationen die eine starke Aussagekraft für die Aktivitätserkennung haben, die meisten sind jedoch erst durch eine Verknüpfung mit anderen Informationen auf Aktivitäten abbildbar. Durch diese Informationsspanne wird ein neues Feld von innovativen Anwendungen geschaffen werden, welches von personalisierten Assistenten, wie Google Now, bis hin zu neuartigen Sport- oder Musikapps reicht.

Literatur

1. Dewanto, L.: Autorisierungsdienste mit oauth. heise (2009), <http://www.heise.de/developer/artikel/Autorisierungsdienste-mit-0Auth-845382.html>
2. vor dem Esche, J., Hennig-Thurau, T.: German social media consumer report 2012/2013. Tech. rep., Westfälische Wilhelms-Universität Münster (2013), http://www.socialmediathinklab.com/wp-content/uploads/2013/02/WWU_Social-Media-Consumer-Report_0213_Ansicht.pdf
3. Facebook: Facebook developers, <https://developers.facebook.com/>
4. Fernando, N., Loke, S.W., Rahayu, W.: Mobile cloud computing: A survey. Tech. rep., Department of Computer Science and Computer Engineering, La Trobe University, Australia (2011), <http://www.sciencedirect.com/science/article/pii/S0167739X12001318/pdf?md5=456dbb2b20c584d0277bc06284107fe0&pid=1-s2.0-S0167739X12001318-main.pdf>
5. Fielding, R.T.: Architectural styles and the design of network-based software architectures. Tech. rep., University of California, Irvine (2000), http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm
6. Google: Android developers, <https://developer.android.com/>
7. Google: Google developers, <https://developers.facebook.com/>
8. Lane, N.D., Miluzzo, E., Lu, H., Peebles, D., Choudhury, T., Campbell, A.T., College, D.: A survey of mobile phone sensing. Tech. rep., Department of Computer Science and Computer Engineering, La Trobe University, Australia (2010), <http://nslab.kaist.ac.kr/courses/2011/cs710/paperlist/1-1.pdf>

¹⁷ <http://techcrunch.com/2013/12/24/apple-patents-integrated-heart-rate-monitor-for-smartphones-hover-touch-sensors/>

¹⁸ <http://www.androidnext.de/news/sony-core-und-smartband-swr10-activity-tracker-im-hands-on-video-ces-2014/>

¹⁹ <http://www.presse-text.com/news/20131220015>

9. Mell, P., Grance, T.: The nist definition of cloud computing. Tech. rep., National Institute of Standards and Technology (2011), <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>
10. Pessemier, T.D., Dooms, S., Martens, L.: Context-aware recommendations through context and activity recognition in a mobile environment. Tech. rep., Wica, iMinds-Ghent University, Ghent, Belgium (2013)
11. Pietsch, A.: Community online tracking. Tech. rep., Universität Rostock, Fakultät für Informatik und Elektrotechnik, Institut für Informatik (2009), http://wwwiuk.informatik.uni-rostock.de/fileadmin/wwwiuk/download/Publikationen/2009/alex_p_da.pdf

Overview over current Radio-based, Device-Free Context Recognition Research

Albu Alexandru*

Advisor: Markus Scholz[†]

Karlsruhe Institute of Technology (KIT)
Pervasive Computing Systems – TECO

*alexandru.albu@student.kit.edu

[†]scholz@teco.edu

Abstract. In this seminar paper I review the current research in the field of Device-Free Context Recognition. We categorized the reviewed papers based on their used features, algorithms and protocols. The final categorization could be used as a start point for answering old questions or raise new ones. The categories that I've used are: Context, Information Layer, Communication protocol, Number of Subjects, Training Requirement, Number of physical links.

Keywords: review, device-free, context, recognition

1 Introduction

"Ubiquitous computing has as its goal the enhancing computer use by making many computers available throughout the physical environment, but making them effectively invisible to the user." [8]. Following this definition Ubiquitous Computing require a lot of communicating devices that are deployed in our environment. With their increasing number it gets more complex and costly to control and use these devices. In the last years, trying to reduce this complexity, a number of algorithms were developed with the specific task of optimization. There were also added different sensors in order to increase the capabilities of this algorithms.

The reason behind Context Recognition is that the system can recognize the context of the subject so that the he could be more easy and intuitive to use. Context Recognition comprises all the approaches to comprehend the context in which a human find itself. Device-Free Context Recognition target all the approaches that don't require the person to carry any kind of device or tags being thus minim invasive.

In this paper we review and categorize some of the most recent researches in this area, that use radio signal distortions as sensor input either for indoor localization, subject count or presence detection.

2 Papers

2.1 Adaptive device-free passive localization coping with dynamic target speed

The purpose of this paper of Xiuyuan Zheng, Jie Yang, Yingying Chen, Yu Gan[1] is to present an optimization framework for classical active device-free location methods. The problem of real life speed changing targets is presented and one possible solution based on the developed framework is exemplified. The framework presented in this paper has three major components: speed change detection, determination of the time-window size and adaptive localization.

The paper brings into light the fact that most of the classical methods used in device-free location, assume that the target speed is constant and thus use a constant window size. This assumption fails in real life applications, where most of the time, the targets doesn't have constant speed. This fact is proved here using some experiments with targets that change their speed during the test. The results of this experiments prove that the problem raised is real and the need for optimization exist.

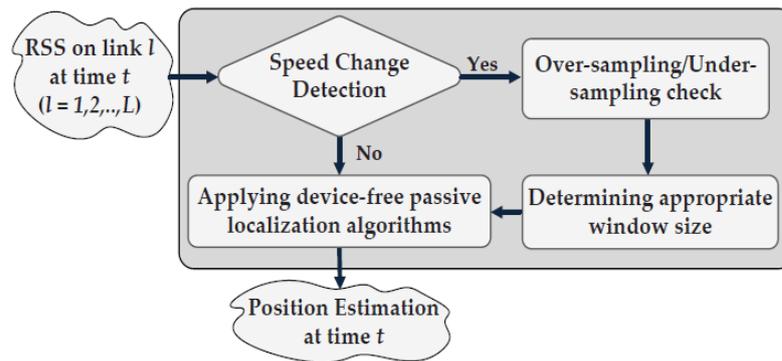


Fig. 1. Overview of the framework components[1]

The proposed solution of this paper is an adaptive speed change detection framework consisting of three components[Fig.1]: speed change detection, determination of the time-window size and adaptive localization. The framework is flexible and thus able to use any variance-based RSSI technique. This is possible because of the components delimitation. I will go through the components and explain the role that play and how they work together. The framework uses as input the RSSI values between links at different time-stamps. The first component of the framework, the speed change detection, decide if the target speed has changed. If the framework doesn't find any speed changing of the target, than the localization algorithm is applied using a fixed window size derived from empirical study. If the first component detects a speed change than the second

component is used to determine the appropriate window size which will be used in the localization algorithm.

The paper tests two approaches for the speed change detection: Average variance ratio (Detection AVR) and Variance distribution similarity (Detection VDS). The both scheme compare the current time interval with the previous one and based on the results, decide if the speed change event has occurred. The time intervals have constant size. The first scheme, Detection AVR, uses the averaged variation ratio, defined like this: $AVR = \frac{1}{L} \sum_{l=1}^L \frac{\sigma_{l,\Delta T}^2}{\sigma_{l,\Delta T'}^2}$. In this formula, L is the total number of links, $\sigma_{l,\Delta T}^2$ is the variance of link l for ΔT and $\sigma_{l,\Delta T'}^2$ is the variance of link l for $\Delta T'$. Upon the calculation of this value, it then compares it with two threshold constants, and decide if the event occurred. The second scheme, Detection-VDS, use the distribution similarity between the two intervals, and decide upon the similarity found if the event was present. This scheme utilize Kullack-leiber divergence metric. The both schemes were tested using experiments information and the results show that the Detection VDS has on average a 82% success rate, with 5% more than the first scheme. The tests also showed that the interval size is important, because its length influence the granularity of the variance. In conclusion, the Detection VDS was used, with a 1s time interval.

The second component of the framework is the window size determination. Once the speed-changed event occurred, the new window size is determined in two steps. The first one check to find out if the window size need to increase or decrease. This is accomplished by comparing the average variance of the current window with the average variance of the previous one. If the average of the current window is bigger, then the window size should be reduce, and if it is smaller, otherwise. The second step tries to find out what is the best new window size, according to the information obtained in the first step. In the paper is described a binary-search approach, that searches for the best solution. Finally, the last part is the localization algorithm, which according to the paper, can be any RSSI-variance based approach. In this research is tested a combination of Localization-RTI (radio tomographic imaging) and Localization ISM (Intersection Method). The first algorithm is used to develop a linear model using the RSSI values and the second one is used to calculate the coordinate of the moving target by finding the intersection of influential links.

The results of the framework evaluation showed an increase of location accuracy when speed-changing targets are monitored.

2.2 Time-of-Flight-Based Radio Tomography for Device Free Localization

In this paper, Jie Wang, Qinghua Gao, Hongyu Wang, Yan Yu and Minglu Jin[2] propose a novel Time of Flight based radio tomography (TRT). The idea behind these approach is that as the RSSI values, the distance measured by TOF is also influenced by shadowing the link between nodes. The proposed technique

use the TOF measurements as input data and utilize a modified particle filter algorithm with compression sensing technique.

It is stated that this novel TOF approach is more accurate than the traditional RSSI-based techniques because the RSSI measurements are less robust to noise, and thus can be affected by factors such as humidity, temperature. In this paper is taken into consideration that the location information is not only sparse, the proposed solution make use of a modified compression sensing particle filter algorithm with under sampled measurements. I will describe the structure of the system and then i will go into details about the used algorithms.

The solution proposed in this paper is a novel Time of Flight based localization technique. This technique is somehow similar to the RSSI-based one, the main difference being in the type of input data. The system structure is composed of nanoPAN nodes which measure sequentially the distance between them. For measuring, the system use the symmetrical double-side two-way ranging technique, that allows him to measure the distance without clock synchronization.

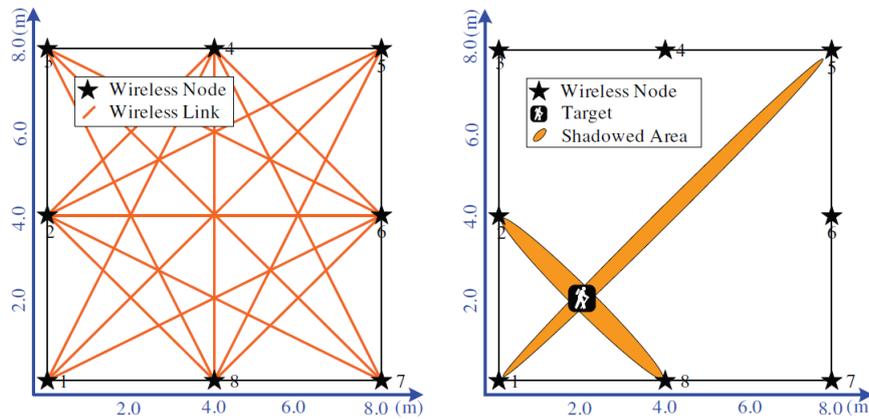


Fig. 2. a) wireless links b) shadowing effect[2]

In order to test if the TOF measurements are suitable for the device-free localization, some experiments using this system were made. The obtained results showed that the distance information is influenced by the target moving into the area. The tests also showed some limitation of the system. The shadowing effect is stronger and better biased when the target is close to one node and close to the line of sight. The shadowing effect states for the effect that the target has on a link. When a node transmits and the subject is on the line-of-sight, the rest of the link between the target and the receiver is influenced, thus shadow effect. It is also stated the length of the link has also a negative influence on the system performance. The reason for this result is that the target has a bigger influence on the measured distance(ToF) when it can block the line-of-sight[Fig.2]. Even

though this measured distance can be influenced even from a small object, the influence is small and hard to use as an accurate information. That's why when the link length increases the influence is dropping. The same happens with the distance between the target and the line-of-sight, the closer the subject is to the line-of-sight, the better can the system recognize it.

In order to test the performance of the proposed technique, the results were compared with the ones from RSSI based algorithms, under the same condition. In this experiment 8 active nodes and 1 that receiver were used on a surface 64 square meters and the localization was done every second. The scenario consisted of a target moving anticlockwise with a constant speed. The results of the experiment show an error less than 1.8 meter 80% of the time. Under the same condition and scenario were also tested a couple of RSSI-based methods. RSS-based Radio Tomography (RRT) using the same Compressive Sensing modified particle filter showed a median error of 1.27 meter compared with 1.02 for the proposed solution. The variance was also smaller for the TRT approach: 0.92 to 0.84. The multi-channel RSS-based RRT scheme using the same modified algorithm showed a better median error, 0.85 meter, but with a variance of 1.01.

The results showed in this paper state that the technique presented here obtained better results than the traditional RRT. It is also introduced a novel CS-modified PF algorithm for the location estimation that utilizes the full space-domain sparse. This algorithm is universal and it can be used in any sequential estimation problems. Even though this approach obtained good results it is noted that there are still problems to be solved like multi-target localization and computational time optimization.

2.3 MonoPHY: Mono-stream-based device-free WLAN localization via physical layer information

This paper the authors, Heba Abdel-Nasser, Reham Samir, Ibrahim Sabek and Moustafa Youssef[3], propose an approach for device-free localization using the information provided from the physical layer: Channel State Information and MIMO (Multiple-Input Multiple-Output). The technique used here takes advantage of the capabilities of a 802.11n network and it is implemented without using a special designed system, but using only an access point and a notebook, which are more common in households.

The problem stated in this paper is that even though device-free localization draws a lot of attention in the last years, the fact that the used techniques require special hardware or a large amount of nodes, it makes it difficult to deploy in real life scenarios. The proposed solution of this paper is a mono-stream 802.11n based system that uses the information provided by the physical layer. The advantage of the described system is that it doesn't require any special made hardware because it uses an access point and a notebook that allow the access to the physical layer information. In the following paragraph I will describe the system and the used technique.

In order to explain the algorithm used, I will first describe some used terms. First of all the IEEE 802.11n standard uses OFDM (Orthogonal Frequency Di-

vision Multiplexing) modulation that divides a wide channel into sub-carriers where each signal, transmitted on a sub-carrier, has a different signal strength and phase. The information about each sub-carrier is obtained from the Channel State Information on the physical layer. The other used information reside on MIMO technology, which provides multiple virtual links between each pair of transmitter-receiver antennas. The technique used for localization in this paper is based on the information provided by the two.

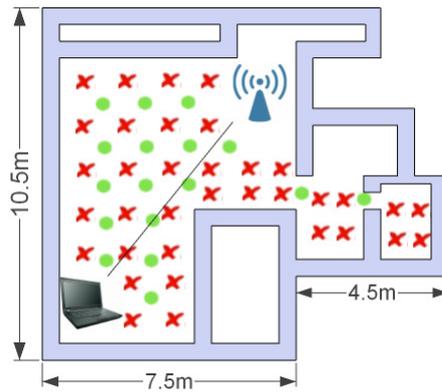


Fig. 3. Training phase[3]

Because is a single-link approach the designed system tries to compensate using the capabilities of the 802.11n network. As result using a single-link, the system manages to obtain information from $m \cdot n \cdot f$ virtual links, where m is the number of antennas of the AP, n is the number of antennas of the WiFi card and f is the number of subcarriers. Furthermore the signal strength for each virtual link will be mapped using a k -cluster approach. K-means clustering is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining. k -means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells[9].

The method described in this paper starts with an offline training phase[Fig.3]. In this phase, the target stands at different position in the defined location and the system computes a RF-based fingerprint for each location. This fingerprints will be used during the localization phase. The online localization algorithm captures the signal information and preprocess the obtain information, after which it filters the outline values. At this point the system runs a cluster builder using the k -means algorithm. The last component of the system, which computes the actual localization, tries to match the built cluster with the clusters computed in the offline training phase.

The system was tested on a normal apartment with an area of 100 square meters. The offline training phase computed 35 different locations distributed uniformly on the test area. The system used 2 receiver antennas, 1 transmitter antenna and number of 20 sub-carriers. The evaluation showed an error less than 1.36 median error which is 48.1 % less than in state-of-the-art device free localization with a single link.

During the test process a couple of important information were obtained. Firstly, the number of receiver antennas and their combinations influence the median error of the system. This shows that an increased number of receiver antennas doesn't lead to a better result. Secondly, an increase in window size increase the performance but slows the whole system down. An increase in the number of used sub-carriers results in a better accuracy up until 20 sub-carriers, at which point the improvement is not visible anymore.

Compared with the state-of-the-art single-link device free localization technique, the presented approach obtained an about 48.1% better accuracy. The paper presents a solution which has the advantage of using common hardware and not specially designed one, thus making the approach more easy to use.

2.4 Towards Omnidirectional Passive Human Detection

This paper Zimu Zhou, Zheng Yang, Chenshu Wu, Longfei Shangguan and Yunhao Liu[4] introduce the concept of Omnidirectional Passive Human Detection, a method used for realizing human detection. The key concept of this paper is the new approach to realize an algorithm that detect human presence within a disk-like area[Fig.4] of one node. The experimental results showed an average false positive of 8% and an average false negative of 7%.

This research points out the need for device-free human detection, a technique that can be used in different areas such as intruder detection, indoor human localization and can also replace existing expensive techniques. In this paper it is also pointed out the need for a omnidirectional human detection technique, which, even though the antennas used in common device-free localization hardware are omnidirectional, wasn't experimented before. Now a days such a disk-like proximity detection can be achieved only using expensive radars, which makes the proposed technique even more attractive.

There are two types of detection presented in this paper namely Equalized Decision and Azimuth Distinction. The first one, which is the centerpiece of the research determines if a person is present within a near-disk region or not with the same confidence on all positions. The second one is not the primary subject of this paper, but results of this distinction are also presented here. It discriminates the particular azimuth of the human.

In order to explain the technique I will first define a couple of features used in this scheme. Even though RSSI is used in the most device-free detection schemes, the fact that it yields limited resistance to environmental noise makes this feature not the most desired one. Hence, two other features derived from the Channel State Information of the physical layer are used. First, the Channel Impulse Response(CIR) which is used to distinguish different paths from the time domain.

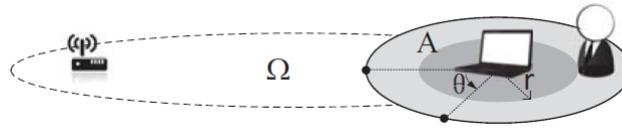


Fig. 4. The omnidirectional passive Human Detection Problem[4]

The second one is the Channel Frequency Response(CFR) which is provided in modern multi-carrier radio such as OFDM. Both features are correlated[Fig.5] and there is possible to convert the information from CFR into CIR information using Fourier transformation. In this paper the CFR features are used because the experimenter showed two limitations of the CIR: a lack of TX-RX synchronization and on the 20MHz bandwidth not all the CIR components are relevant in comparison with the CFR.

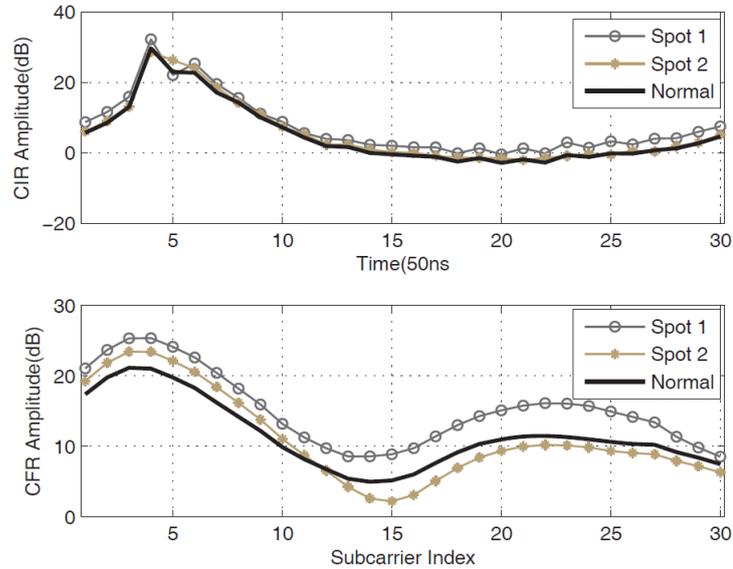


Fig. 5. Amplitude of CIR and CFR[4]

In the presented solution, the authors use a fingerprinting approach using a K-dimensional vector of amplitude histograms of CFRs as signatures and Earth Mover’s Distance(EMD) for signature recognition. In the paper there are presented a couple of findings that were revealed during the experiments. The tests showed that using amplitudes of CFRs we are able to discriminate the environmental noise and movement and focus on the human presence. The reason for

using histograms in this scheme is that the circle-like distribution of the CFR signatures makes it difficult to model using clusters. Following the selection of histograms, the use of EMD for signature comparison seems like the right one. The EMD is also calculated separately for each sub-carrier because otherwise the spectral structure will be compromise.

The fingerprinting approach is built in two phases. The training phase and the monitoring phase. The information obtained from the CSI are converted into amplitudes of CFRs and then the resulting histograms are used to build signatures with a fixed window size. In the first phase, the person is placed in different location in order to built a signature database. In the second phase the signature computed in each window is compared with the ones stored in the database and if a number of N consecutive windows are found in the database then the system yields a detect event. For the Azimuth Distinction the same algorithm is used but for each signature in the monitoring phase, the most probable match in the database will be used to determine the direction of the person.

In the experiment a TP-LINK TL-WR741N wireless router and a laptop with Intel 5300 NIC receiver were used. The driver for the Intel receiver was changed in order to access the CSI data. There were used a number of 30 sub-carriers and a transmission rate of 20 packets per second. There were two location used: a conference hall that was almost empty and a small computer lab with a lot of background infrastructure. There were built a number of test cases with in different position in each location. The signatures were built in each test case with a person standing in 4 directions and at 0.5 or 1 meter away from the link. The experiment lead to some findings such as: a small rise of average false positive between 0.5m and 1m cases, the performance gap between the two locations was relative small and the best performance was obtain when the person was on the direct link.

2.5 Device-Free Indoor Localization for AAL Applications

This paper of Paolo Barsocchi, Francesco Potort and Paolo Nepa[5] presents a method for device-free indoor localization designed for Ambient Assisted Living Applications(AAL). In this applications the position accuracy is less important as in other applications but it tries to accomplish good results according to the EvAAL competition. Other metrics include: user acceptance, availability, interoperability and installation complexity.

The problem which is raised in this paper is that the existing localization techniques are not built to fulfill the requirements of AAL applications. Such applications are evaluated on other metrics as the normal localization algorithms. For example, a localization system should be able to change its accuracy according to the application needs. In the paper this problem is tried to be solved with the pre knowledge of the application requirements.

The proposed solution is leveraging the RSSI variances of the lines of sight[Fig.6] built between the nodes of the system. The nodes are build into a mesh, thus their lines of sight divide the location into different perimeters The key concept of this algorithm is that, when a person crosses one line of sight the system

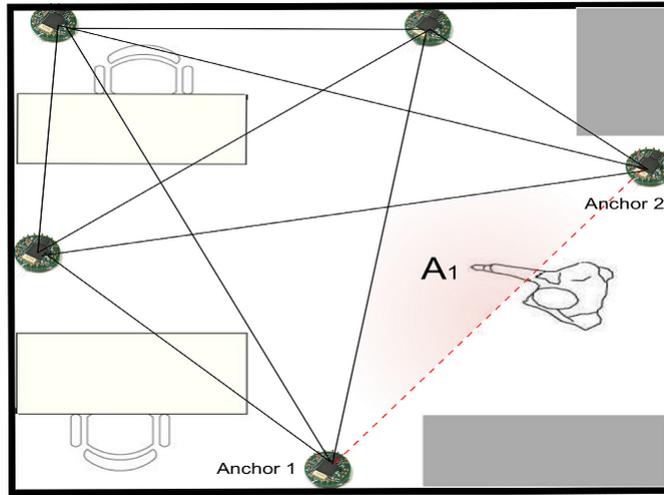


Fig. 6. The proposed AAL compliant system[5]

recognize it and it further know in which perimeter is the person now. At each step, according to the area in which the system thinks the person is, it assume its location is the centroid of that area. With this algorithm, one is able to arrange the nodes in the location in such a manner that the most important ones will obtain a better accuracy. The system is also capable to make an error estimation of the algorithm by taking the maximum distance between a centroid and its farthest away point.

This algorithm was tested using five IRIS wireless sensors nodes in a indoor location of 23 m^2 . The nodes use the 2.4 GHz band using the 802.15.4 standard. The nodes send using a token-based algorithm a broadcast packet that all the other ones receive it and thus obtain the RSSI value between the nodes. The duration of the experiment was 668s and it collected more than 40000 RSSI measurements. The five nodes were positioned among the walls of a rectangular area and the person walked exactly between them from one side of the room to the other.

The detecting algorithm was based on the fact that the variance of the RSSI measurements increases when a person cross a line-of-sight between the nodes. In order to evaluate the performances of the system a ROC curve was used, which is commonly used in detection algorithms. The paper notes that even the results of ROC were poor and highly variable, the actual localization performance was good. The system achieved a median error of 1 meter with a RMS of 1.7 meter, and a median error of 0.4 with a RMS of 0.8.

The paper introduces an algorithm that targets device-free human localization in AAL applications. The requirements of this applications are different and don't focus just on the accuracy of the localization.

2.6 SCPL: Indoor Device-Free Multi-Subject Counting and Localization Using Radio Signal Strength

In this research, Chenren Xu, Bernhard Firner, Robert S. Moore, Yanyong Zhang, Wade Trappe, Richard Howard, Feixiong Zhang and Ning An[6] extend the device-free localization technique by introducing algorithms that are able to count the number of persons and to localize them. The name, SCPL, stands for sequential counting and parallel localization in order to describe the manner in which the algorithms work. This technique uses a fingerprinting approach and it uses RSSI values between the links.

The key problem stated in this paper is that the impact of multiple subjects on the links RSSI is non-linear and thus hard to be decomposed using standard approaches. There are also stated a couple of problems of the currently used techniques. The location-based technique, which is based on a RSSI fingerprinting of the area with the subject on different locations, lacks the scalability property. The other option is a link-based approach, which captures when a user passes a line-of-sight between nodes, has the downside of assuming that the number of subjects is known. There is also called the need for techniques that are able to provide localization capabilities for more than one subject and the technique presented here tries to accomplish that.

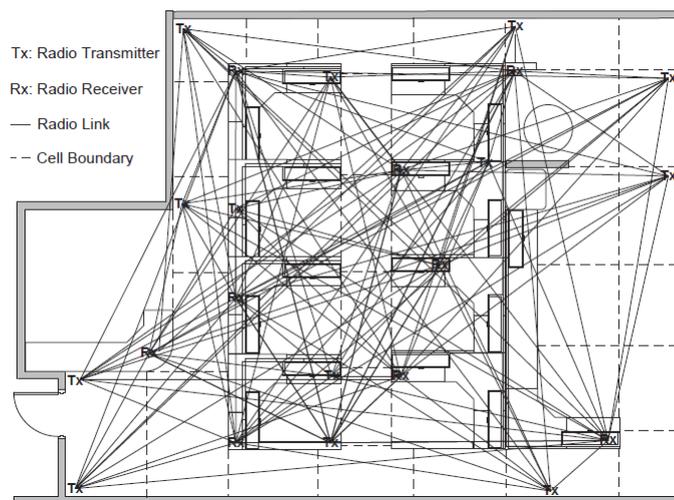


Fig. 7. Radio Link Distribution[6]

The proposed solution consists of two phases. In the first phase the algorithm counts sequentially how many persons are present in the area using successive cancellation. The cancellation raises other problems because the subjects are influencing the RSSI values in a non-linear way. The proposed technique uses coefficients in order to compensate for this effect. The second phase realizes the

localization of the subjects once their number is known. This process is done in a parallel fashion and is modeled as a conditional random field. In the following I will go into more details about each phase.

Before the first phase, there were some tests made in order to obtain a better understanding of the impact of more persons on the links. Two main conclusion were drawn from this experiments: the absolute RSSI mean difference is a suitable metric when one tries to compute how the links RSS value changed and the second result was that computed difference is not suitable for subject detection because the value is not linearly correlated with the number of subjects. Having this results, the counting algorithms does a successive cancellation after each new added subjects, but the algorithm uses some coefficients computed in the training phase, in order to avoid the problem of non-linearity. The algorithm has three steps what run as long as there are still subjects uncounted: Presence Detection, Cell Identification, Contribution Subtraction.

In the next phase, using the known number of subjects in the area, starts the localization algorithm. The algorithms uses a different technique as the one presented for counting because this new one can provide better results. The main problem found in the localization process is that the multi-path effect of the signal which can cause prediction errors.

The proposed solution was tested in two different environments: an office space of $105m^2$ consisting of cubicles and an exposition hall of $400m^2$. The used hardware consisted of Chipcon CC1100 radio transceivers with 16-bit Silicon Laboratories C8051F321 microprocessors and USB dongles using the same CC1100 as receivers. The used frequency was 909.1 MHz using MSK modulation and a output power of 0 dBm. For the office area were used 13 transmitters and 9 receivers which weren't placed for this experiment, so their position wasn't optimal computed. For the hall area were used 12 transmitters and 8 receivers that were placed in an optimal manner. Optimal was defined here when the link density per cell was uniformly distributed[Fig.7].

The above presented technique obtained good results as is follows: in the office scenario, for the counting of subjects, obtained 84% across all cases and for the localization the error distance was 1.25m with 0-order trajectory ring and 1.08m with 1-order trajectory ring. In the second area, for the counting phase, for the counting phase, the average was 86% and for localization the error was 1.49m using a 2-order trajectory ring. These results show that the method presented here is suitable and the paper also states a couple of performance improvements that could be made.

2.7 System Design for Passive Human Detection using Principal Components of the Signal Strength Space

The system presented in this paper from Bojan Mrazovac, Milan Z. Bjelica, Dragan Kukulj, Saša Vukosavljev and Branislav M. Todorovic[7] was designed for the detection of human presence with the purpose of being used in smart homes. In this paper it is stated that this method is the first the uses Principal

component analysis(PCA) for the RSSI data set. PCA is a linear transformation that defines a new coordinate system for the data set.

The problem raised in this research is the need of human detection in order to implement smart houses. The main purpose is to save energy based upon the presence human in a room. Existing human detection system such as infrared, 3D camera, ultrasonic radars are too complex to use because they require infrastructure that need to be wired and also require large computational time. This research introduces a system that doeskin have this requirements.

The proposed solution is based on the fact that human presence violates the received signal strength between communicating wireless nodes. One of the problems that the system tries to avoid is caused by the fact that radio wave characteristics vary significantly in indoor spaces mainly because the multipath fading. There are two main causes for these variations: device properties and the propagation medium. The human presence lies into the second category because the human body can scatter, reflect or absorb the radio signal. Because of this property it is thus possible to draw conclusion about human presence by monitoring the signal strength.

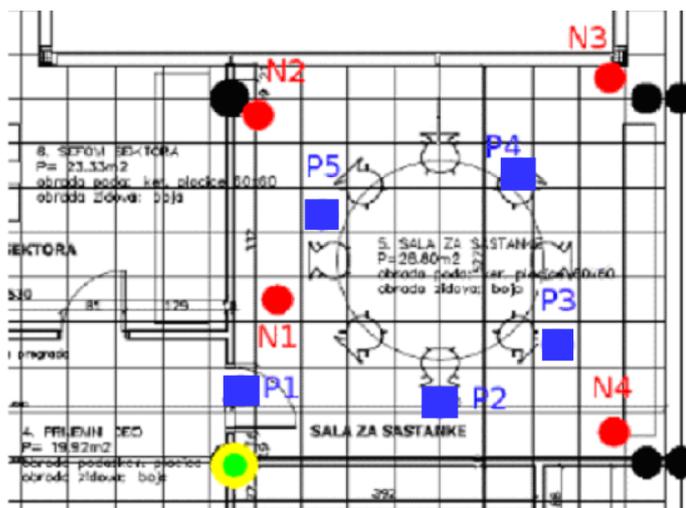


Fig. 8. Testing room layout[7]

The system presented in this paper claims to be able to not only detect human presence when the person is standing but also detect it when the person lies or sits. The starting point of the algorithm is based on the mean and variance of the RSSI values. When the variance exceeds some bounds the detection is reported. In order to optimize this approach the authors used principal components analysis which is often used in pattern recognition. PCA is a variable reduction procedure. It is useful when samples are obtained on a large number

of variables that are mutually correlated. It then shows the similarities and differences. PCA can also be used to compress the data by reducing the number of dimensions, without much loss of information.

The system was designed on a master-slave base. The master was a home controller which polls each node every 100ms and saves the received data. After it saves this information for 3 seconds the PCA algorithm is used to detect the human presence. When the node receive the pooling command it broadcast the RSSI values stored from the packets received from the other nodes.

The home controller was installed on a PC and four nodes that worked on the ZigBee protocol enabled by CC2531 USB dongle. The experiment was made in room of 536cm x 530cm[Fig.8]. The results of the experiment showed that the used PCA algorithm has an accuracy of 96% with a window size of 32 samples and of 93% with a window size of 12 samples. It is stated that an improvement could be achieved by using additional filter function that would isolate all the values below the calculated threshold.

The proposed solution successfully uses principal components analysis on the RSSI data set in order to obtain accurate indication of human presence.

3 Categorization

One of the purposes of this paper was to find key attributes of the papers in order to build a meaningful categorization. The reason behind this idea was to sort and to catalog existing research and make it better accessible and easier to understand existing and open research questions. Knowing how similar are two researches will enable new researches to compare the right papers or to see which combinations of this technologies and algorithms was never used.

I propose the following categories:

- **Layer:** The communication layer that provides the data used in the algorithms.
- **Context:** Which contextual information is recognized or should be recognized.
 - Localization
 - Counting
 - Detection
- **Communication Protocol** The protocol that the wireless sensors use to communicate with each other.
- **Number of subjects:** Whether or not the system works with multiple subjects
- **Number of physical links**
 - single: there are only two nodes that communicate
 - multiple: sensors are build into a mesh topology
- **Requires training** Whether or not the system requires a training phase. (*for example to build RF signatures*)

Paper	Layer		Context			Communication Protocol	Subject Number		Physical Links		Training
	Physical Layer	MAC	Localization	Counting	Detection		sing.	multi.	sing.	multi.	
[1]		RSSI	X			RFID	X			X	
[2]		ToF	X			802.15.4a	X			X	
[3]	CSI MIMO		X			802.11n	X		X		X
[4]	CRF				X	802.11n	X		X		X
[5]		RSSI	X			802.15.4a	X			X	
[6]		RSSI	X	X				X		X	X
[7]		RSSI			X	802.15.4a	X			X	

Table 1. Categorization results

4 Conclusion

In this seminar paper I reviewed some current researches in the field device-free context recognition. As our categorization showed this researches use different approaches and have different goals. The categorization has as main purpose to answer old questions and to find new ones. It can also be considered a guide for the new researches and also a starting point for new researchers.

In the last years the interest for device-free context recognition has grown fast. In this paper we tried to show that the increasingly number of researches should be categorized in order to be easily identified and also compared. We've proposed a possible categorization of the presented paper, which gives us the possibility to compare correctly the results of this researches. We can also see from such a categorization what combination of sensors, topology and algorithms was never tested. We think that this process of categorization should be a perpetual one and thus we aim to categorize more papers in this area.

References

1. Xiuyuan Zheng, Jie Yang, Yingying Chen, Yu Gan: Adaptive Device-Free Passive Localization Coping with Dynamic Target Speed. In: 2013 IEEE INFOCOM
2. Jie Wang, Member, IEEE, Qinghua Gao, Hongyu Wang, Member, IEEE, Yan Yu, and Minglu Jin, Member, IEEE: Time-of-Flight-Based Radio Tomography for Device Free Localization. IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, VOL. 12, NO. 5, MAY 2013
3. Heba Abdel-Nasser, Reham Samir, Ibrahim Sabek, Moustafa Youssef: Monophy: Mono-Stream-based Device-free WLAN Localization via Physical Layer Information. In: 2013 IEEE Wireless Communications and Networking Conference (WCNC): SERVICES & APPLICATIONS
4. Zimu Zhou, Zheng Yang, Chenshu Wu, Longfei Shangguan, and Yunhao Liu: Towards Omnidirectional Passive Human Detection. In: 2013 IEEE INFOCOM
5. Paolo Barsocchi, Francesco Potort and Paolo Nepa: Device-Free Indoor Localization for AAL Applications
6. Chenren Xu, Bernhard Firner, Robert S. Moore, Yanyong Zhang Wade Trappe, Richard Howard, Feixiong Zhang, Ning An: SCPL: Indoor Device-Free Multi-Subject Counting and Localization Using Radio Signal Strength
7. Bojan Mrazovac, Milan Z. Bjelica, Dragan Kukolj, Saša Vukosavljev, Branislav M. Todorovic: System Design for Passive Human Detection using Principal Components of the Signal Strength Space. In: 2012 19th IEEE International Conference and Workshops on Engineering of Computer-Based Systems
8. Ubiquitous Computing, Mark Weiser, August 16, 1993 <http://www.ubiq.com/hypertext/weiser/UbiCompHotTopics.html>
9. Wikipedia, the free encyclopedia http://en.wikipedia.org/wiki/K-means_clustering

Einfluss von Fenstergröße, Überlappung der Fenster und Abtastrate bei funkbasierter gerätefreier Aktivitätserkennung

Nikolai Hartmann*

Advisor: Markus Scholz†

Karlsruhe Institute of Technology (KIT)

Pervasive Computing Systems – TECO

*hartmann@teco.edu

†scholz@teco.edu

Abstract. Dieser Artikel untersucht, welchen Einfluss verschiedene Parameter auf die Erkennungsrate bei funkbasierter gerätefreier Aktivitätserkennung haben. Die Auswertung basiert auf denselben Rohdaten, wie die Untersuchungen in "Device-Free and Device-Bound-Activity Recognition using Radio Signal Strength" [6]. Im Speziellen geht es hier um die Auswirkungen von Fenstergröße, Überlappung der Fenster und Abtastrate auf die Genauigkeit der Aktivitätserkennung. Auch hier wurden *Bayes*, *kNN* und *C4.5* zur Klassifizierung verwendet. Für alle Fenstergrößen außer 5 sind Erkennungsraten von über 50 % möglich. Die beste Erkennung (53,7 %) gelingt mit *C4.5* bei Fenstergröße 40, 75 % Überlappung und 20 Hertz Abtastrate. Bei Fenstergröße 40 sind 20 Hertz zur Erkennung der getesteten Aktivitäten ausreichend. Ist die Fenstergröße doppelt oder genau so groß wie die Abtastrate sind die Erkennungsraten im Schnitt am besten.

1 Einleitung

Mark Weiser beschreibt in [7] seine Vision vom Ubiquitous Computing. Die Idee einer Welt, in der vernetzte Computer allgegenwärtig sind und intelligente Umgebungen schaffen. Der erste Teil ist schon weit fortgeschritten, immer mehr Technik, die in der Lage ist, immer mehr zu steuern, ist im Alltag allgegenwärtig. Diese Geräte sind zumindest teilweise in der Lage, sich zu vernetzen. Um die Vision zu vervollständigen, fehlt noch die Schaffung der intelligenten Umgebung. Dafür entscheidend ist, dass die einzelnen Geräte erkennen können, was in ihrer Umgebung passiert. Dies erfordert geeignete Sensortechnik und Möglichkeiten auf die Informationen aus den Sensordaten intelligent zu reagieren.

Aktivitätserkennung ist ein zentrales Thema im Pervasive Computing. In [6] wurde gezeigt, dass funkbasierte Aktivitätserkennung gerätefrei möglich ist.

Zur Erkennung wurden 8 Netzknoten im Raum angebracht. Die Knoten wurden so programmiert, dass sie reihum nacheinander Datenpakete losschickten, die von den jeweils nicht-sendenden Knoten empfangen wurden. Jedes Paket beinhaltet zum einen die Kennung des Knotens, der es verschickt hat. Zum anderen beinhaltet es für jeden anderen Knoten, wie gut die Empfangsqualität des letzten von dort stammenden und am absendenden Knoten ankommenden Paketes war. Somit weiß jeder Knoten zu jedem Zeitpunkt, wie das Netzwerk für jeden Knoten zum vorherigen Zeitpunkt aussah.

Die anfallenden Daten wurden serialisiert. Zur Auswertung wurden nicht die Rohdaten selbst verwendet, sondern immer Pakete mit Durchschnittswerten über einen Zeitraum von einer Sekunde, in der sich die Aktivität nicht geändert hat. Ausgewertet wurden die Daten dann mit Orange¹, einem Data-Mining-Framework, zunächst wurde mit 10-facher Kreuzvalidierung evaluiert, in einer zweiten Auswertung dann mit Training-Test-Splits, um die Einflüsse der Tür oder der jeweiligen Testperson auf das Gesamtergebnis beurteilen zu können.

Je nach verwendetem Klassifikator wurden für die gerätefreie funkbasierte Erkennung konnte in Train-Test-Splits nach Person zwischen 46,5% und 49,1% Genauigkeit erreicht. Im Folgenden werden nun andere Konfigurationen untersucht werden, um zu sehen, ob es möglich ist, bessere Erkennungen zu erreichen oder mit weniger Daten ähnlich gute Ergebnisse zu erhalten.

2 Bestehende Untersuchungsergebnisse und Datensatz

2.1 Algorithmen

kNN: kNN ist die Abkürzung für k-Nearest-Neighbors oder k-Nächste-Nachbarn. Anschaulich werden alle zu klassifizierenden Daten als Punkte im mehrdimensionalen Raum aufgefasst. Werte werden Klassen zugeordnet, indem die räumlich am nächsten liegenden Punkte betrachtet werden. Die Klasse, der die meisten Nachbarn angehören, wird auch dem zu klassifizierenden Punkt zugewiesen. k ist eine Konstante, die angibt, wie viele Nachbarn betrachtet wurden[1]. Sowohl in den Analysen in [6], als auch in den in dieser Arbeit durchgeführten Auswertungen, wurde $k = 10$ gewählt.

Bayes: Klassifizieren mit Bayes basiert auf bedingten Wahrscheinlichkeiten. Die Grundannahme hier ist, dass alle Werte nur von der Klasse abhängen. Dementsprechend ist die Erkennung mit einem Bayes-Klassifikator schlecht, wenn die Werte sehr stark korreliert sind. Bayes-Implementierungen haben eine sehr geringe Laufzeit und liefern in der Praxis meist gute Ergebnisse.[4]

C4.5: Ein von Ross Quinlan entwickelter Algorithmus, um Entscheidungsbäume aufzubauen. Diese Bäume können zur Klassifizierung verwendet werden.

¹ <http://orange.biolab.si/>

Beim Aufbau des Baumes werden die einzelnen Werte auf Entropie untersucht. Je größer der Informationsgehalt pro Wert erscheint, desto weiter oben wird die entsprechende Entscheidung im Baum verzweigt.[5]

Train-Test-Split: Bei einem Training-Test-Split wird der Testdatensatz abhängig von einem Parameter aufgeteilt. Bei Aktivitätserkennung bietet sich beispielsweise ein Split nach Testperson an, wie er in [6] und auch in diesen Untersuchungen durchgeführt wurde. Dann werden die Klassifizierer zunächst mit den Daten, die die erste Person erzeugt hat, trainiert und die Erkennung mit den Daten der zweiten Person getestet. Im zweiten Schritt wird dann mit den Daten der zweiten Person trainiert und mit der ersten Person die Erkennung getestet.

Durch Train-Test-Splits lässt sich der Einfluss des gewählten Parameters auf die Erkennung bestimmen.

Kreuzvalidierung: Wenn mit Kreuzvalidierung ausgewertet wird, dann wird der Datensatz in gleiche Teile aufgeteilt. Im Versuch waren es zehn. Nun werden die Erkenner mit jedem der erzeugten Datensätze getestet, nachdem sie zuvor mit den jeweils anderen Datensätzen trainiert wurden. Die Gesamtgenauigkeit ist dann der Durchschnitt aus den Genauigkeiten pro erzeugtem Teil. Sehr wichtig ist hierbei, dass es keine Überlappung zwischen den einzelnen Fenstern gibt, da sonst die Daten, mit denen die Erkenner trainiert wurden, auch im Testdatensatz vorkommen und diese Daten so das Ergebnis künstlich verbessern werden.

Prävalenz: Von einigen Aktivitäten gibt es mehr Datenpakete und damit auch mehr Fenster. Um sicherzustellen, dass die Klassifizierer für alle Aktivitäten gleich gut trainiert werden, wird die Anzahl der Fenster für alle Aktivitäten angeglichen. In der originalen Auswertung wurden zufällige Fenster entfernt, in den hier folgenden Untersuchungen wurde die zeitliche Abfolge beibehalten, und spätere Fenster verworfen.

Ausgewählt wurde in jedem Fall so, dass es pro Aktivität, Testperson und Türzustand gleich viele Instanzen erhalten blieben.

2.2 AH 2013 - Experimente und Datensatz

Dieser und der nächste Abschnitt beziehen sich direkt auf [6]. In einem Büroraum für zwei Personen wurden acht Sende-Empfangsknoten in bekannter Topologie angebracht. Zwei Testpersonen führten dann verschiedene Aktivitäten im Raum durch, einige davon allein, andere zu zweit. Zusätzlich zu den Funkknoten trägt die im Raum aktive Person auch noch Beschleunigungssensoren.

Die getesteten Aktivitäten waren Stehen, Laufen, Sitzen, Sitzen und Tippen, liegen, liegen und winken, sowie nicht im Raum.

Die Daten, die per Funk anfallen, werden zusammen mit den Beschleunigungsdaten annotiert und serialisiert. Annotiert wird nicht nur die aktuelle Tätigkeit, sondern auch welche Person diese durchführt und ob und wie weit die Tür geöffnet ist.

2.3 AH 2103 - Auswertung und Ergebnisse

Die Auswertung der Daten erfolgte in zwei Schritten. Zunächst wurde der gesamte Datensatz wieder deserialisiert und vorverarbeitet, so dass er in einem zweiten Schritt mit Orange eingelesen und ausgewertet werden konnte.

In der **Vorverarbeitung** wurde, um die Werte über den drahtlosen Empfang zu stabilisieren, immer der Durchschnitt eines Knotenpaares verwendet. Im nächsten Schritt wurden die Daten der Beschleunigungssensoren mit den stabilisierten Funkdaten verknüpft.

Nun wurden immer 40 aufeinanderfolgende Pakete betrachtet, wenn sich laut Annotation die Aktivität innerhalb eines solchen Fensters nicht änderte, wurden Varianz und Durchschnitt der einzelnen Werte innerhalb der Pakete zur Analyse vorgemerkt. Änderte sich die Aktivität bevor 40 Pakete gefunden wurden, wurden diese Pakete nicht betrachtet. Nach einem vollständigem Fenster wurde mit dem nächsten darauf folgenden Paket beginnend weitergesucht, so dass es keine Überlappungen zwischen den einzelnen Fenstern gab. Zusätzlich wurde jeweils am Ende einer Aktivität noch eine Sekunde an Daten ignoriert, um mögliche Ungenauigkeiten in der Annotation auszugleichen.

Im letzten Schritt der Vorverarbeitung wurde die Prävalenz angeglichen, d.h., es wurde gezählt, wie viele Fenster pro Aktivität erkannt wurden, um dann von den häufigeren Aktivitäten zufällige Fenster zu löschen, bis für alle Aktivitäten, Personen und Türzustände je gleich viele Fenster im Datensatz waren. Diese Anpassung wurde durchgeführt, damit die verschiedenen Erkenner für alle Aktivitäten gleich gut trainiert werden.

In der **Auswertung** wurden dann über Training-Test-Splits und Kreuzvalidierung überprüft, ob und wie gut sich die gewählten Aktivitäten erkennen lassen. Als Erkenner wurden *kNN*, *Bayes* und *C4.5* verwendet. Ausgewertet wurden zunächst nur die Beschleunigungsdaten, dann der gesamte Datensatz und zuletzt die Funkdaten ohne Beschleunigungsdaten. Dass die Erkennung mit den Beschleunigungsdaten möglich ist, zeigte unter anderem [2]. Die Auswertung der Beschleunigungsdaten diente hier vor allem der Referenz.

Ausgewertet wurde nun zunächst mit Kreuzvalidierung, deshalb wurden für die Untersuchung Fenster ohne Überlappung gewählt. Spätere Auswertungen erfolgten dann noch mit Training-Test-Splits, um zu überprüfen, welchen Einfluss die Testperson oder der Türzustand auf die Erkennung haben.

Ergebnisse: Eine rein funkbasierte Erkennung ist möglich. Es wurde mit Kreuzvalidierung und *kNN*-Erkener eine Genauigkeit von 89,4 % erreicht. Die Auswertung der Beschleunigungsdaten ergab eine Genauigkeit von 85 %. Wurde der Datensatz nach Türzustand aufgeteilt, sank die Erkennungsrate in diesem Train-Test-Split auf 59 % ab.

2.4 Eigene Untersuchungen

Nachdem im ursprünglichen Paper gezeigt wurde, dass Aktivitätserkennung gerätefrei möglich ist, soll hier untersucht werden, ob es bestimmte Konfigurationen gibt, die eine bessere Erkennung ermöglichen. Der gesamte Rohdatensatz wurde ursprünglich bei fester Fenstergröße, ohne Überlappungen und mit einer Abtastrate von 40 Hertz analysiert. Im Folgenden werden nun verschiedene Fenstergrößen, Überlappungen und Abtastraten auf ihre Erkennung hin getestet.

Dafür gab es keine neuen Experimente, die folgende Untersuchung beruht auf dem selben Datensatz wie die originale Untersuchung. In der Vorverarbeitung werden jedoch die Fenstergröße, Überlappung und auch die Abtastrate angepasst.

Die Erkennungsgenauigkeit wird hier nur mit Training-Test-Splits ermittelt. Es wird keine Kreuzvalidierung verwendet, da in den meisten getesteten Konfigurationen tatsächlich überlappende Fenster vorkommen. Würde man hier dennoch mit Kreuzvalidierung testen, kämen einzelne Daten sowohl im Trainings- als auch im Testdatensatz vor und würden so das Ergebnis künstlich verbessern.

Als weiterer Unterschied zur Originaluntersuchung ist noch zu nennen, dass hier bei der Prävelenzanpassung keine zufälligen Fenster gezogen werden, sondern die ursprüngliche Reihenfolge beibehalten wird.

3 Methodik

3.1 Parameter

Fenstergröße: Es werden nicht einfach alle gesammelten Datenpakete verwendet, um die Erkennung zu trainieren oder zu testen, sondern immer eine feste Anzahl Pakete gruppiert. Hierbei wird überprüft, dass alle Pakete innerhalb eines Fensters laut Annotation dieselbe Aktivität darstellen. In der ursprünglichen Untersuchung wurden Fenster mit 40 Samples verwendet.

Ändert sich die Aktivität laut Annotation bevor ein Fenster voll ist, werden die bisher für das Fenster vorgesehenen Daten verworfen, und es wird erneut versucht, die benötigte Anzahl an Datenpaketen mit der gleichen Aktivität zu finden.

Für ein Fenster werden dann jeweils Varianz und Durchschnitt über die einzelnen Werte innerhalb der betrachteten Pakete gebildet. Also beispielsweise der Durchschnitt und die Varianz über die Werte, die der Beschleunigungssensor in X-Richtung bei allen Samples innerhalb des Fensters aufgezeichnet hat. Betrachtet wurden Fenstergrößen von 5 Samples bis 120 Samples.

Verschiebung: Wurde, wie oben beschrieben, ein Fenster gefunden und die Features darin berechnet, wird bei einer Verschiebung von 100 %, d.h. bei einer Überlappung von 0 %, mit dem nächsten Datenpaket beginnend weiter gesucht.

Ist ein kleinerer Wert für diesen Shift angegeben, so werden einige Pakete, die bereits im letzten Fenster waren, im aktuellen wieder verwendet werden.

Analysiert wurden Verschiebungen relativ zur Fenstergröße, jeweils um ein Viertel, die Hälfte und drei Viertel der Fenstergröße. Zusätzlich wurden auch die Verschiebung um jeweils ein Sample und auch die Verschiebung um die Fenstergröße selbst, also größt- und kleinstmögliche Überlappung betrachtet.

Abtastrate: Zusammen mit der Fenstergröße bestimmt die Abtastrate die Dauer eines Fensters. Die Samplingrate beschreibt, wie viele Pakete pro Sekunde aufgezeichnet wurden. Mehr als 40 Hertz sind hier nicht notwendig, in [3] wurde gezeigt, dass menschliche Aktivitäten immer unter 20 Hertz sind.

Instanzen: Die Anzahl der Instanzen ist genau die Anzahl von Fenstern, die nach der Prävalenzangleichung noch im Datensatz sind. Bei einigen Konfigurationen verblieben nach der Vorverarbeitung weniger als 100 Instanzen. Diese Konfigurationen wurden nicht näher betrachtet; nicht sicher ist, ob diese Werte repräsentativ sind.

3.2 Anpassungen am originalen Datensatz

Um die beschriebenen Untersuchungen mit den geänderten Parametern durchführen zu können, wurde die Vorverarbeitung der Daten erweitert. Die Erweiterungen, um mit zusätzlichen Fenstergrößen und Verschiebungen zu arbeiten, konnten einfach umgesetzt werden, da lediglich einzelne Grenzen in der Berechnung der Feature-Windows angepasst werden mussten.

Eine Änderung der Abtastrate war ursprünglich nicht vorgesehen. Die hier gewählte Vorgehensweise ermöglicht es nun zumindest Datensätze zu erzeugen, deren Frequenz ganzzahlige Teiler der ursprünglichen Abtastrate von 40 Hertz sind, zusätzlich möglich sind also Abtastraten von 1, 2, 4, 5, 8, 10 und 20 Hertz. Dabei werden einzelne Pakete ausgelassen, um beispielsweise eine Abtastung mit fünf Hertz zu simulieren, wird nur jedes achte Datenpaket tatsächlich verwendet, für 20 Hertz würde jedes zweite Datenpaket verworfen.

3.3 Zusammenfassung

Im Folgenden wird nun betrachtet, wie sich die Fenstergröße, Fensterverschiebung und Abtastrate auf die Aktivitätserkennung auswirken. Als Klassifizierer werden, wie auch im ursprünglichen Paper *kNN*, *Bayes* und *C4.5* verwendet, um allgemeinere und weniger vom Klassifikator abhängige Ergebnisse zu erhalten. Evaluiert wird hier nur mit Training-Test-Splits auf die Testperson, nicht mit Kreuzvalidierung, da es Überschneidungen zwischen den einzelnen Feature Windows gibt.

Die Rohdaten wurden vorverarbeitet, um alle Kombinationen aus diesen Parameter testen zu können:

- Fenstergröße: 5, 10, 20, 40, 60, 80 und 120
- relative Verschiebung: 25 %, 50 %, 75 %, 100 % und absolut 1
- Abtastrate: 1, 5, 10, 20 und 40 Hertz

4 Ergebnisse

Für jede der 7 gewählten Fenstergrößen 5, 10, 20, 40, 60, 80 und 120, ergeben sich jeweils 25 Werte - 5 verschiedene Abtastraten und für jede davon je 5 verschiedene Verschiebungen. Insgesamt also theoretisch 175 Erkennungsraten. Hierbei fallen jedoch ab einer Fenstergröße von 60 jeweils die 5 Werte für Abtastung mit einem Wert pro Sekunde weg, was daran liegt, dass es keine Aktivitäten gab, die länger als eine Minute am Stück durchgeführt wurden. Bei einigen Konfigurationen mit großer Fenstergröße und kleinen Abtastraten fielen zudem nur einzelne Aktivitäten heraus.

Zusätzlich wurden die Erkennungsraten nur näher betrachtet, wenn es bei den jeweiligen Parametern mehr als 100 Instanzen gab.

Die Tabellen lesen sich wie folgt: In der Kopfzeile ist die Überlappung für die jeweilige Spalte angegeben. In der ersten Spalte sind die jeweiligen Abtastraten angegeben. In den übrigen Feldern sind in grau unterlegt die Erkennungsraten, zuerst *kNN*, dann *C4.5*, zuletzt *Bayes*. Darüber jeweils die Anzahl der Instanzen, die in der jeweiligen Konfiguration vorkamen.

4.1 Fenstergröße 5

Table 1. Fenstergröße 5

Samplingrate	0	25 %	50 %	75 %	1
Instanzen:	84	84	84	84	84
1	47.61 27.38 48.8	39.28 29.76 36.9	48.8 39.28 50.0	54.76 27.38 39.28	54.76 27.38 39.28
Instanzen:	1764	2898	4326	8610	8610
5	45.01 44.21 45.69	47.92 43.3 44.96	47.89 43.29 44.96	46.65 45.38 45.05	46.65 45.38 45.05
Instanzen:	3570	5922	8862	17682	17682
10	44.5 46.21 42.38	45.87 43.7 42.58	45.15 42.77 43.1	44.25 42.28 43.58	44.25 42.28 43.58
Instanzen:	7140	11928	17850	35658	35658
20	43.65 42.98 43.02	44.58 43.72 43.3	43.88 42.1 42.66	44.4 44.11 43.14	44.4 44.11 43.14
Instanzen:	14364	23940	35910	71736	71736
40	40.81 43.94 40.97	40.41 40.81 41.14	40.02 43.85 40.75	39.29 42.63 40.87	39.29 42.63 40.87

In Tabelle 1 sieht man eine Besonderheit durch die kleine Fenstergröße: Ein Shift von 25 % entspricht dem festen Shift von 1. Es gibt hier also für jede der 4 Frequenzen 4 Instanzzahlen und je Erkennen auch nur 4 Genauigkeitsangaben je Frequenz.

Verwertbare Daten gibt es hier ab 5 Hertz Abtastung, bei geringerer Abtastrate gibt es jeweils nur 84 Instanzen.

Einfluss von Fenstergröße, Überlappung der Fenster und Abtastrate bei funkbasierter gerätefreier Aktivitätserkennung 209

Mit *Bayes* ist eine konstante Erkennung möglich, bei 5 Hertz am besten, circa 45 %, bei 10 Hertz und 20 Hertz etwa 42% und bei 40 Hertz etwa 40%.

Die Erkennungsrate von *kNN* sinkt mit steigender Samplerate. Bei 5 Hertz ist die Erkennung bei kleinem und mittlerem Overlap mit **knapp 48%** fast 2 Prozentpunkte besser als ohne Overlap oder mit einer Verschiebung von 1 Sample pro Fenster. Die Erkennung bei 10 und 20 Hertz Abtastung liegt unabhängig von der Fensterverschiebung bei rund 44%. Bei 40 Hertz liegt ist die Erkennung noch mal etwa 4% schlechter.

C4.5 liefert hier Erkennungsraten von rund 43%. Nur 2 Werte sind hier deutlich entfernt, die Erkennungsrate bei 10 Hertz ohne Overlap ist 46%, das Minimum für Erkennung mit *C4.5* bei einer Fenstergröße von 5 beträgt 40% und ist bei 40 Hertz und einem Overlap von 25%.

Die beste Erkennung, die auf dem Testdatensatz hier mit *kNN* erreicht wurde, ist bei Overlaps von 25% und 50% und 5 Hertz Abtastrate **knapp 48%**. Der größte Wert, der mit *C4.5* hier erreicht wurde, ist 46%, ausgewertet ohne Überlappung bei 10 Hertz. Der Spitzenwert für den *Bayes*-Klassifikator ist 45,6% bei Abtastung mit 5 Hertz ohne Überlappung.

4.2 Fenstergröße 10

Table 2. Fenstergröße 10

Samplingrate	0	25 %			50 %			75 %			1				
Instanzen:	108	180			180			324			576				
1	49.07	28.7	41.66	47.77	33.88	40.0	47.77	38.33	44.44	52.16	36.11	43.51	49.13	33.5	43.57
Instanzen:	882	1050			1260			2646			4914				
5	49.43	49.31	47.95	48.76	51.52	47.61	46.98	47.77	47.46	46.52	52.22	46.78	46.64	48.53	46.52
Instanzen:	1764	2520			3486			8652			17220				
10	47.27	43.87	48.07	48.09	46.19	47.18	47.61	46.7	44.46	48.49	46.09	45.33	48.87	45.66	46.77
Instanzen:	3570	5082			7056			17640			35238				
20	44.87	49.97	45.12	45.23	48.07	45.12	46.62	42.6	43.62	45.1	48.07	44.32	43.17	43.07	45.26
Instanzen:	7140	10206			14280			35658			71316				
40	42.52	45.86	44.69	42.71	43.81	44.59	42.19	46.08	44.73	41.24	44.1	44.83	40.66	43.48	44.68

Mit steigendem Overlap steigt bei einer Abtastrate von 1 Hertz die Erkennungsrate von *C4.5* zunächst von 28,7% über 33,8% bis auf 38,3% bei einem Overlap von 50% an. Danach fällt der Wert wieder ab, zunächst auf 36% bei 25% Verschiebung, dann auf 33% bei fester Verschiebung von 1 Sample pro Fenster.

Bei Abtastung mit 5 Hertz ist der beste Wert für Fenstergröße 10, bei einem Overlap von 75% erreicht *C4.5* eine Erkennungsrate von **52%**, ähnlich gut ist die Erkennung bei einem Shift von 25% mit 51%. Die übrigen drei Werte sind gerundet bei 48%.

Für 10 Hertz ist bei Overlaps von 25, 50 und 75% die Erkennung bei 47 bis 48%.

Bei Verschiebung von konstant 1 liegt diese bei 45%, ohne Overlap bei lediglich 43%.

Bei 20 Hertz ist die Tendenz für die Erkennungsrate bei *C4.5* mit steigendem Overlap fallend, mit einer Ausnahme bei 50%, dort gibt es einen leichten Anstieg. Die Erkennung beginnt mit knapp 50% ohne Overlap, und fällt dann Stück für Stück, mit der erwähnten Ausnahme bei 50% Overlap, auf 43% ab.

Der beste Wert bei Abtastung mit 40 Hertz für *C4.5* ist bei 50% Overlap und beträgt 46%, ohne Overlap ist die Erkennung etwa 1% schlechter. Die übrigen drei Werte liegen zwischen 43,5 und 44,1%.

Bayes zeigt bei 1 Hertz die beste Erkennung für 50% Shift, mit etwa 44%. Die Erkennung für größere Overlaps ist jeweils bei jeweils 43,5%, bei den beiden kleineren im Schnitt bei knapp 41%.

Die Erkennungsrate mit *Bayes* bei einer Abtastung mit 5 Hertz steigt mit steigendem Shift leicht an. So ist die Erkennung bei minimalem Shift 46,5% und bei maximalem Shift bei 48%.

Für 50% und 75% Overlap sind die Werte mit 10 Hertz Abtastung bei rund 45% die schlechteren Erkennungen für *Bayes*. Die anderen drei Werte liegen bei rund 47%.

Die Erkennungsraten bei 20 Hertz sind ähnlich verteilt wie bei 10 Hertz. Für keinen Overlap, 25% und Shift von 1 ist die Erkennung mit rund 45% etwas besser als bei den anderen beiden Shift-Werten, wo die Erkennung bei rund 44% liegt.

Bei Abtastung mit 40 Hertz ist die Erkennung mit *Bayes* unabhängig von der Verschiebung rund 44,7 %.

kNN erkennt bei 10 Samples pro Fenster und einem Sample pro Sekunde ohne Overlap 49%, und jeweils 47% für einen Overlap von 25% und 50%. Für einen Overlap von 75% ist *kNN* bei einem Hertz mit **52%** der beste betrachtete Erkennner für eine Fenstergröße von 10, *C4.5* erreichte diesen Wert bei einer Abtastung mit 5 Hertz, bei gleichem Overlap. Bei minimalem Shift lag die Erkennung bei 49%.

Bei 5 Hertz Abtastrate sind die Werte für *kNN* knapp schlechter als bei 1 Hertz. Für keinen Overlap und 25% Overlap sind sie gleich, bei einem Overlap von 50% etwa ein Prozent schlechter. Für die beiden größten Overlapwerte lag die Erkennungsrate bei 46,5%.

Für 10 Samples pro Fenster kann eine Erkennungsrate von **52%** erreicht werden, entweder mit *kNN* 75% Overlap und 1 Hertz Abtastung, oder mit *C4.5*, dann bei 10 Hertz und 75% Overlap. *Bayes* erreicht bei dieser Fenstergröße 48% bei 10 Hertz ohne Overlap.

Table 3. Fenstergröße 20

Samplingrate	0	25 %	50 %	75 %	1										
Instanzen:	33	33	33	33	165										
1	37.22	27.77	34.44	37.22	27.77	34.44	37.22	27.77	34.44	49.11	34.22	42.33			
Instanzen:	210	210	210	210	840										
5	48.09	42.85	40.47	48.57	38.57	40.0	49.52	39.52	41.42	48.41	38.88	40.07	46.3	40.47	40.35
Instanzen:	840	966	1218	2058	8946										
10	49.88	49.52	48.33	49.58	49.48	48.75	49.5	48.11	47.7	48.73	46.4	47.13	50.05	49.28	45.96
Instanzen:	1764	2310	3486	6888	33642										
20	47.56	42.17	48.01	48.35	49.61	47.83	49.28	46.81	47.24	49.31	48.92	47.38	47.93	49.63	47.53
Instanzen:	3570	4746	7056	14112	70476										
40	45.96	51.84	47.36	46.01	47.8	46.86	45.3	47.68	47.32	45.15	46.45	47.1	43.44	46.0	47.27

4.3 Fenstergröße 20

Für Abtastung mit einem Hertz kamen nur bei Shift = 1 über 100 Instanzen zusammen. Andere Überlappungskonfigurationen werden daher nicht betrachtet.

Bei einem Hertz und Verschiebung von einem Sample pro Fenster zeigte *kNN* eine Erkennungsrate von 49%.

Bei 5 Hertz Abtastrate ist *kNN* bei 50% Overlap mit einer Erkennung von 49,5% am besten. Für 75 und 25% Overlap ist die Genauigkeit bei 48,5%. Ohne Overlap beträgt sie 48%, bei maximalem Overlap 46%.

Außer bei 75% Overlap, ist die Erkennung bei 10 Hertz mit rund 50% bei allen Shifts recht hoch, am besten bei genau 50% und maximalem Overlap.

Bei 20 Hertz und 50 oder 75% Overlap ist die Erkennungsrate bei 49,3%, für die übrigen Shifts etwa einen bis eineinhalb Prozentpunkte schlechter.

Bei 40 Hertz Abtastrate fällt mit größerer Überlappung ab. Ohne Überlappung und mit 25% Überlappung wird eine Genauigkeit von 46 % erreicht. Bei größerer Überlappung fällt der Erkennungswert auf 45 , beim größten betrachteten Overlap auf 43 %.

Die Erkennungsraten mit *Bayes* sind bei Fenstergröße 20 je Frequenz vergleichsweise stabil. Alle Werte bei 5 Hertz bewegen sich zwischen 40,5% und 41,5%. Wobei der größte Werte bei 50% Overlap auftaucht.

Für 10 Hertz zeigt sich ein ähnliches Bild, die Werte schwanken zwischen 46 und und 48,7%, beste Erkennung bei einem Overlap von 25%. Bei 20 Hertz bleibt die Erkennung stabil bei über 47%, ebenso bei 40 Hertz.

C4.5 liefert bei 5 Hertz für 25 bis 75% Overlap Werte um die 39%, für maximalen Overlap 40% und bei keiner Überlappung 42,8%.

Bei einer Abtastrate von 10 Hertz ist die Erkennungsrate bei Overlaps von 50% und 75% mit rund 47% am niedrigsten, bei den anderen drei Werten beträgt die Erkennung etwa 49,3%.

Bei einer Abtastung von 20 Hertz und Fenstern ohne Überlappung ist die Erkennung für *C4.5* mit 42% am schlechtesten. Bei Overlap 50% ist die Erkennung mit 46,8% besser, für die übrigen drei Werte im Schnitt wieder bei 49,3%.

Die beste Erkennung für Fenstergröße 20 bei allen betrachteten Konfiguratio-

nen zeigt *C4.5* bei 40 Hertz und überlappungsfreien Fenstern mit **51,8%**. Mit steigender Überlappung fällt der Wert dann stetig von 47,8% bis 46% ab.

Mit einem *Bayes*-Klassifikator sind bei einer Fenstergröße von 20, Fensterverschiebung von 25% und Abtastrate von 10 Hertz 48,7% Erkennungsrate möglich. *kNN* erreicht bis zu 50% Erkennung, berechnet für 10 Hertz und eine Verschiebung von einem Sample je Fenster. Mit dem *C4.5*-Klassifizierer gelang die beste Erkennung für diese Fenstergröße, **51,8%**, bei einer Verschiebung von 20 Samples pro Fenster abgetastet mit 40 Hertz.

4.4 Fenstergröße 40

Table 4. Fenstergröße 40

Samplingrate	0	25 %			50 %			75 %			1				
Instanzen:	11	11	11	11	11	11	11	11	11	33					
1	80.35	80.35	85.71	80.35	80.35	85.71	80.35	80.35	85.71	80.35	85.71	73.8	57.73	84.52	
Instanzen:	37	37	37	37	37	37	37	37	37	74					
5	32.3	30.11	21.63	32.3	30.11	21.63	32.3	30.11	21.63	32.3	30.11	21.63	33.84	21.63	31.28
Instanzen:	210	210	210	210	210	210	210	210	210	252	1386				
10	49.52	36.19	42.38	49.52	40.47	42.85	47.61	38.57	44.28	47.61	46.03	37.69	45.52	37.08	39.32
Instanzen:	672	966	1176	2016	17346										
20	48.36	45.08	43.6	49.89	48.65	48.03	49.74	52.89	47.87	49.55	53.72	45.98	49.16	49.06	45.31
Instanzen:	1764	2310	3486	6888	66402										
40	48.07	41.43	48.46	49.48	44.93	47.83	50.02	44.49	48.22	50.08	46.15	48.27	48.57	48.73	48.38

Hier gibt es wegen zu geringer Instanzzahlen erst ab 10 Hertz Abtastrate verwertbare Ergebnisse.

kNN fällt bei 10 Hertz mit größer werdendem Overlap ab. Ohne Überlappung und bei 25% Überlappung wird eine Erkennung von 49,5% erreicht, für eine Überlappung von 50 und 75% 47,6%, bei maximalem Overlap 45%. Bei 20 Hertz Abtastrate sind die Werte für *kNN* alle zwischen 48,3 und 49,8%, der größte Wert bei 25% Overlap, der kleinste ohne Overlap. Wird mit 40 Hertz abgetastet sind alle Werte über 48%. Der beste Wert ist 50% Erkennung jeweils bei 50 und 75% Overlap.

Der *Bayes*-Klassifikator kommt bei 10 Hertz ohne Überlappung und mit 25% Overlap auf eine Erkennungsrate von 42,5%, verbessert sich dann bei 50% Overlap auf 44%. Die Erkennungsraten für die größten Verschiebungswerte sind unter 40%.

Für 20 Hertz ist die Erkennung ohne Overlap bei 42% am schlechtesten. Bei 25 und 50% Overlap erkennt *Bayes* 48%. Die übrigen beiden Werte sind 45%. Bei der größten Abtastrate ist die Erkennungsrate zwischen 48,5% bei keiner Überlappung, und 47,8% bei 25% Overlap. Die übrigen Werte sind alle über 48%.

C4.5 kommt bei 10 Hertz und 75% Überlappung auf 46% Genauigkeit, mit Abstand der beste Wert für diese Abtastrate, die übrigen Werte sind 40% und weniger.

Bei einer Abtastrate von 20 Hertz erreicht *C4.5* die größte Genauigkeit Erkennung über Fenster mit 40 Samples, **53,7%**, bei einem Overlap von 75%, ähnlich gut ist auch der Wert für 50% Überlappung, 52,9%. Die übrigen Werte sind unter 50%, der kleinste mit 45 Prozentpunkten wird bei Auswertung ohne überlappende Fenster erreicht.

Bei 40 Hertz steigt die Erkennungsrate von *C4.5* mit größerem Overlap an. Die Erkennungsrate ohne Overlap ist 41,4%, diese steigt dann bis 48,7% bei maximaler Überlappung.

Bei Fenstergröße 40 finden sich bei allen Klassifikatoren vergleichsweise gute Erkennungsraten, wenn mit mindestens 20 Hertz abgetastet wird und die Überlappungen größer als 50 % sind. Die größten Genauigkeiten für *kNN* und *C4.5* sind auch in diesem Block. Die beste Erkennung mit Bayes gelingt zwar bei 40 Hertz, allerdings ohne Overlap (48,5%). Die genaueste Erkennung für *kNN* ist bei 40 Hertz und 75% Overlap und beträgt 50%. Die global beste Erkennung für Fenstergröße 40 erreicht *C4.5*, bei 20 Hertz, aber ebenso wie *kNN* bei einer Überlappung 75%, **53,7%**.

4.5 Fenstergröße 60

Table 5. Fenstergröße 60

Samplingrate	0	25 %	50 %	75 %	1
Instanzen:	72	72	72	108	1044
5	36.11 31.94 40.27	36.11 31.94 40.27	36.11 31.94 40.27	47.22 37.96 42.59	47.7 40.03 42.33
Instanzen:	78	78	78	117	741
10	22.81 14.08 23.21	22.81 14.08 23.21	22.81 14.08 23.21	37.83 21.29 35.44	34.69 28.71 32.37
Instanzen:	294	294	420	714	8190
20	45.91 43.19 41.83	42.51 45.23 40.81	46.9 49.04 44.76	49.29 49.71 42.29	46.25 46.27 41.75
Instanzen:	1176	1512	2058	3654	49098
40	49.91 43.11 47.53	50.66 40.8 48.01	49.17 47.76 47.52	50.19 46.6 47.56	49.5 47.21 47.8

Bei Fenstergröße 60 können bei 5 und 10 Hertz jeweils nur die Werte betrachtet werden, die für Überlappungen von 75% und für den Shift von 1 errechnet wurden, bei den anderen Verschiebungen gab es zu wenig Instanzen, um eine sinnvolle Betrachtung zu ermöglichen.

Für eine Abtastung mit 1 Hertz gibt es ebenso keine Werte, da keine Aktivitäten länger als 60 Sekunden am Stück aufgezeichnet wurden.

Für die beiden gültigen Werte bei 5 Hertz ist die Erkennungsrate mit *kNN* bei 47 %.

Bei Abtastung mit 10 Hertz beträgt die Erkennung bei 75 % Überlappung 37 %, für eine Verschiebung von 1 34 %.

Mit einer Abtastrate von 20 Hertz gibt es jetzt für alle getesteten Überlappungen wieder ausreichend viele Instanzen. Ohne Überlappung, bei 50% Overlap und bei maximalem Overlap ist die Erkennungsrate bei rund 46 %. Für den Overlap von 25 % fällt die Genauigkeit auf 42,5 % ab. Bei einem Overlap von 75 % war die Genauigkeit für *kNN* bei einer Fenstergröße von 60 Samples und einer Abtastung von 20 Hertz 49 %.

Die Erkennungswerte mit *kNN* bei 40 Hertz sind alle über 49 %, der beste bei einem Overlap von 25 % mit **50,7 %**.

Die Erkennungsraten für die beiden gültigen Verschiebungswerte mit 5 Hertz Abtastrate bei *Bayes* sind 42 %.

Bei 10 Hertz Abtastung fällt die Erkennungsrate auf 35 % für einen Overlap von 75 % und 32 % bei maximaler Überlappung.

Ab 20 Hertz gibt es wieder für alle Verschiebungswerte verwertbare Genauigkeiten. Bei 50 % Overlap ist die Erkennung mit *Bayes* für 20 Hertz mit 44 % am besten, bei den anderen Verschiebungswerten erreicht Bayes nur eine Genauigkeit von 41 bis 42 %.

Bei Abtastung mit 40 Hertz ist die Erkennungsrate mit *Bayes* bei allen Overlaps über 47,5 %, der beste Wert ist 48 % bei einem Overlap von 25 %.

C4.5 erreicht bei einer Fenstergröße von 60 Samples und einer Abtastrate von 5 Hertz für einen Overlap von 75 % eine Genauigkeit von 38 %. Bei maximaler Überlappung ist die Erkennungsrate 40 %.

Die Werte für 10 Hertz sind sehr niedrig, mit steigender Überlappung 21,3% und 28,7%.

Der kleinste Wert für *C4.5* und 20 Hertz ist hier 43,2% ohne Überlappung. Bei Überlappungen von 50 und 75% wird eine Genauigkeit von über 49% erreicht. Für maximale Überlappung beträgt die Genauigkeit hier 46%, bei Überlappung von 25% werden 45,2% Erkennungsrate erreicht.

Mit 40 Hertz Abtastrate ist die Erkennungsrate von 40,8% für *C4.5* bei einem Overlap von 25% die schlechteste. Gibt es keinen Overlap, so steigt die Erkennungsrate hier auf 43% an. Für die übrigen Überlappungswerte ist die Erkennung bei rund 46%.

Die beste Erkennung für Fenstergröße 60 insgesamt gelang mit *kNN*. Bei einer Abtastrate von 40 Hertz und einer Überlappung von 25% wurden **50,7%** Genauigkeit erreicht. Mit diesen Parametern ist auch die Erkennung für den *Bayes*-Klassifikator optimal, dieser zeigt eine Erkennung von 48%. Der beste Wert für den *C4.5*-Erkennung ist 49,1%, erreicht bei einer Abtastrate von 20 Hertz und 75% Überlappung. Bei 40 Hertz und vorhandenem, aber nicht maximalem Overlap sind die Genauigkeiten unabhängig vom Klassifikator hier recht gut.

4.6 Fenstergröße 80

Analog zur Fenstergröße 60 gibt es auch hier keine Werte für Abtastung mit einem Sample pro Sekunde. Für 20 und 40 Hertz gibt es verwertbare Genauigkeiten

Table 6. Fenstergröße 80

Samplingrate	0	25 %	50 %	75 %	1
Instanzen:	35	35	35	35	140
5	20.42 22.71 34.47	20.42 22.71 34.47	20.42 22.71 34.47	20.42 22.71 34.47	40.07 32.92 36.51
Instanzen:	37	37	37	37	111
10	21.63 30.11 21.63	21.63 30.11 21.63	21.63 30.11 21.63	21.63 30.11 21.63	38.88 20.41 38.98
Instanzen:	210	210	210	210	2562
20	47.14 41.42 41.42	47.14 40.47 43.33	45.71 43.8 40.95	42.85 43.8 37.61	43.63 43.4 39.18
Instanzen:	840	966	1176	2016	34146
40	49.4 42.85 47.73	50.0 49.48 48.86	50.08 49.74 47.61	50.04 50.04 46.37	50.1 49.11 46.15

in allen Überlappungseinstellungen, bei 5 und 10 Hertz nur für einen Shift von 1.

kNN leistet eine Erkennungsrate von 40% bei einer Fensterverschiebung von 1 und 5 Hertz Abtastrate. Bei doppelter Abtastrate und gleicher Verschiebung sind es 38%.

Die Werte für keine und 25% Überlappung bei 20 Hertz sind gleich und betragen 47,1%. Bei einer Überlappung von 50% sinkt die Erkennungsrate auf 45,7%, bei der nächstgrößeren Überlappung fällt die Erkennung nochmals ab, auf 42,9%. Der Wert für die maximale Überlappung ist wieder etwas größer und beträgt 43,6%.

Bei 40 Hertz sind alle Erkennungsraten, außer der für keine Überlappung, über 50%. Der beste Wert hier ist **50,1%**, sowohl bei 50% Overlap als auch bei maximalem Overlap.

Bei Abtastung mit 5 Hertz erreicht $C4.5$ eine Erkennungsrate von 32,9% für den einzigen Überlappungswert mit ausreichend Instanzen. Bei 10 Hertz sinkt die Genauigkeit auf 20,4%.

Für keine und 25% Überlappung erreicht $C4.5$ eine Genauigkeit von rund 41%. Die drei übrigen Werte sind etwa bei 43,6%.

Der Erkennungswert ohne Überlappung ist mit 42,9% vergleichsweise schlecht, die übrigen Werte sind alle größer als 49%, die beste Erkennungsgenauigkeit ist 50% bei einer Überlappung von 75%.

Der einzige für 5 Hertz verwertbare Wert erreichte mit $Bayes$ zu 36,5% Genauigkeit. Der entsprechende Wert bei 10 Hertz ist 39%.

Der für 20 Hertz beste Wert mit $Bayes$ ist 43,3% bei einem Overlap von 25%. Die Genauigkeiten für 50% Overlap und keine Überschneidung sind ebenfalls über 40. Bei 75% Overlap ist die Genauigkeit bei 37,6%, beim maximalen Shift ist die Erkennung etwa 1,5 Prozentpunkte besser.

Bei Abtastung mit 40 Hertz ist die Genauigkeit mit $Bayes$ immer über 46%, der beste Wert beträgt 48,9% bei 25% Overlap.

Auch hier sind die Erkennungsgenauigkeiten bei 40 Hertz, wenn es Overlaps gibt, wieder unabhängig vom Klassifikator vergleichsweise gut. Der größte erreichte Erkennungswert ist hier **50,1%**, erreicht von kNN bei 40 Hertz, sowohl bei 50, als auch bei 75% Overlap. Der beste Wert für $C4.5$ ist 50% erreicht bei 40

Hertz und 75% Overlap. Auch bei einer Abtastrate von 40 Hertz, aber bei einem Overlap von 25% ist das Maximum für *Bayes* bei 48,9%.

4.7 Fenstergröße 120

Table 7. Fenstergröße 120

Samplingrate	0	25 %	50 %	75 %	1
Instanzen:	30	30	30	30	30
5	30.55 34.72 25.0	30.55 34.72 25.0	30.55 34.72 25.0	30.55 34.72 25.0	30.55 34.72 25.0
Instanzen:	72	72	72	108	2016
10	34.72 25.0 36.11	34.72 25.0 36.11	34.72 25.0 36.11	51.85 37.03 44.44	47.91 38.49 44.04
Instanzen:	78	78	78	117	1326
20	25.39 19.44 21.82	25.39 19.44 21.82	25.39 19.44 21.82	33.59 27.77 36.37	33.52 33.63 31.86
Instanzen:	294	294	420	714	16212
40	43.53 42.51 41.49	42.85 45.91 41.49	45.71 41.66 45.47	49.01 45.79 42.71	47.71 46.65 42.58

Bei einer Fenstergröße von 120 gibt es nur mit 40 Hertz für alle Overlap-Werte verwertbare Daten. Bei 10 und 20 Hertz sind zusätzlich die Shifts von 25% der Fenstergröße und 1 verwertbar.

Mit *C4.5* ist bei 10 Hertz und einem Overlap von 75 % eine Genauigkeit 37% erreicht, beim größten Overlap sind es 38,5 %.

Bei 20 Hertz und 75 % Overlap ist die Erkennung etwa 10 Prozentpunkte schlechter als bei 10 Hertz und gleicher Überlappung. Die Erkennungsrate steigt auf 33,6 %, wenn der Shift minimal wird.

Wird mit 40 Hertz abgetastet, gibt es wieder Werte für alle Verschiebungen. Die kleinsten Werte sind bei 50% Overlap und bei keiner Überlappung, etwa 42 %. Bei 25 und 75 % Overlap steigt die Erkennungsrate bis knapp unter 46 %, für maximalen Overlap sind es dann 46,7 %.

Bayes zeigt für die beiden Werte, die mit 10 Hertz abgetastet wurden, jeweils über 44%.

Bei 20 Hertz ist die Erkennungsrate bei *Bayes* deutlich unter 40%, für maximalen Overlap tatsächlich nur 31%.

Für keinen und 25% Overlap ist die Erkennungsrate bei 40 Hertz 41,5%. Bei einer Überlappung von 50% steigt die Erkennung auf den für *Bayes* besten Wert bei einer Fenstergröße von 120 Samples, 45,5%. Die beiden übrigen Werte sind etwa einen Prozentpunkt besser, als der Wert für keinen Overlap.

Die beste Erkennungsrate für eine Fenstergröße von 120 gelingt mit *kNN* bei 10 Hertz und 75% Overlap, **51,9%**, für den größeren Overlap bei gleicher Abtastung sinkt die Erkennungsrate auf 48% ab.

Die beiden Werte für 20 Hertz sind um die 33,5%.

Bei 40 Hertz Abtastung und 75% Overlap beträgt die Genauigkeit 49%, beim größten Overlap sind es 47,7%. Bei einer Überlappung von 50% beträgt die

Genauigkeit 45,7%, die beiden kleineren Overlaps ermöglichen Erkennungsraten von etwa 43%.

Die beste Erkennung unabhängig vom Klassifikator war hier **51,9%**, bei 10 Hertz und 75% Overlap mit einem *kNN*-Erkennung. Die beiden anderen Erkennung waren jeweils bei 40 Hertz, *Bayes* bei einem Overlap von 50% und einer Erkennungsrate von 45,5%, *C4.5* erkannte 46,7% beim maximalen Overlap.

5 Diskussion der Ergebnisse

Bei Fenstergrößen zwischen einschließlich 40 und 80 wurden unabhängig vom Erkennung die besten Werte, sehr Nahe an 50%, erreicht.

Mit größeren Fenstern werden bei kleinen Abtastraten teilweise nur noch sehr wenige Instanzen gefunden. Um stabile Ergebnisse zu betrachten, wurden nur jene Werte ausgewertet, für die es mehr als 100 Instanzen gab. Für große Fenstergrößen, insbesondere bei 160 fallen teilweise Aktivitäten komplett weg, deswegen wurden diese nicht näher betrachtet. Unter anderem fehlt auch eine Aktivität bei Fenstergröße 120, Abtastung mit 10 Hertz und Überlappung von 75 %. Deshalb ist die dort mit *kNN* erreichte Genauigkeit von fast 52 % nur eingeschränkt als wirklich guten Wert zu betrachten.

Die beste Erkennungsrate, die mit den getesteten Einstellungen erreicht wurde, ist 53,7 %, die Fenstergröße war 40, abgetastet wurde mit 20 Hertz und 75 % Überlappung. Der Erkennung war *C4.5*.

Die schlechteste Erkennungsrate ist 20,4 %, bei einer Fenstergröße von 80, maximaler Überlappung und 10 Hertz Abtastrate, ebenfalls klassifiziert mit *C4.5*.

Die beste Erkennungsrate bei **Fenstergröße 5** ist mit 48 % vergleichsweise gering (*kNN*, Overlap 25 %, 5 Hertz). Bei dieser Fenstergröße gibt es nur einen Genauigkeitswert unter 40 %.

Bei **Fenstergröße 10** ist die größte erreichte Erkennungsgenauigkeit 52,2 %. Dieser Wert taucht zweimal in bei dieser Fenstergröße auf: Einmal mit *kNN*-Erkennung bei 1 Hertz und 75 % Überlappung und bei 10 Hertz und ebenfalls 75 % Overlap, erkannt mit *C4.5*. Auffällig schlecht waren hier die Genauigkeiten von *C4.5* bei Abtastung mit 1 Hertz, unabhängig von der Fensterverschiebung immer mindestens 7 Prozentpunkte schlechter als *Bayes*, ohne Überlappung 20 % schlechter als *kNN*.

Die beste erreichte Erkennung bei **Fenstergröße 20** war 51,8 %, gemessen ohne Überlappung, mit 40 Hertz Abtastrate und *C4.5*-Erkennung. Wird mit mindestens 10 Hertz abgetastet ist Erkennung hier immer größer als 42 %.

Mit **Fenstergröße 40** wird die beste Erkennung unter den getesteten Konfigurationen erreicht (*C4.5* 53,7 %, 20 Hertz, 75 % Überlappung). Auffällig ist die vergleichsweise schlechte Erkennung mit *C4.5* bei 10 Hertz Abtastrate, mit Ausnahme von 46 % Erkennung bei 75 % Overlap sind die anderen Werte für diesen Erkennen hier alle unterhalb von 40,5 %. Wird mit 20 oder 40 Hertz abgetastet so erreicht *kNN* immer eine Genauigkeit über 48 %. Bei 40 Hertz ist *C4.5* außer bei einem Shift von 1 der schlechteste Erkennen, mit lediglich einem Wert über 45 %. Die Erkennung war bei Abtastung mit 20 Hertz nicht nennenswert schlechter als bei 40 Hertz.

Bei **Fenstergröße 60** sind alle Genauigkeiten über 40 %, wenn mit mindestens 20 Hertz abgetastet wird. Bei 40 Hertz ist *kNN* sehr gut, alle Erkennungsraten sind über 49 %. Auch die beste Erkennung für diese Fenstergröße ist in diesem Bereich, 51,9 % bei 25 % Überlappung. Hier ist die Erkennung mit 20 Hertz im Schnitt etwa 3 % schlechter als bei 40 Hertz.

Fenstergröße 80: Wird mit 40 Hertz abgetastet, sind die Erkennungsraten unabhängig vom Erkennen über 46 %, nur *C4.5* ohne Überlappung hat mit 42,9 % eine geringere Genauigkeit. Bei maximaler Überlappung ist hier auch die beste Genauigkeit für diese Fenstergröße, 50,1 %. Ab dieser Fenstergröße gibt es wieder Erkennungsgenauigkeiten deutlich unterhalb von 40 %.

Der einzige Erkennungswert über 50 %, der bei **Fenstergröße 120** erreicht wurde, ist bei 75 % Überlappung und 10 Hertz Abtastfrequenz mit *kNN* erkannt worden. Problematisch hierbei ist, dass es in dieser Konfiguration nicht mehr für alle durchgeführten Tätigkeiten Samples gibt. Die Vergleichbarkeit dieses Wertes ist also fraglich. Die Erkennungsraten sind im Schnitt hier deutlich schlechter als bei kleineren Fenstern.

Der Erkennen mit den besten Genauigkeiten je Fenstergröße war *kNN*, hier ist der Wert von für die Fenstergröße 120 noch mitgezählt. Wird dieser Wert nicht mitgezählt, so ist *C4.5* ebenso bei drei Fenstergrößen der beste Erkennen. Allerdings ist *C4.5* auch bei vielen Konfigurationen der schlechteste Erkennen.

Bayes wurde hier bisher kaum als bester, guter oder besonderer Erkennen erwähnt. *Bayes* ist meist ein wenig schlechter als *kNN* oder *C4.5* - insgesamt aber sehr stabil.

Für den ausgewerteten Datensatz waren besonders gute Erkennungsraten bei großen Abtastraten und Fenstergrößen zwischen 20 und 80, jeweils einschließlich. Bei 40 Hertz war die Erkennung mit *Bayes* bei diesen Fenstergrößen meist recht stabil und schwankte über die verschiedenen Überlappungen jeweils um höchstens eineinhalb Prozentpunkte. *kNN* hatte meistens ohne Überlappung und bei der kleinsten Verschiebung die am schlechtesten Werte bei diesen Konfigurationen. Die Erkennungsrate mit *C4.5* folgte bei diesen Fenstergrößen und 40 Hertz

Abtastung keinem festen Schema.

Auch fällt auf, dass die Erkennungsraten unabhängig vom Shift relativ gut sind, wenn ein Fenster ein oder zwei Sekunden lang ist, also die Fenstergröße doppelt oder genau so groß wie die Abtastrate ist.

Für die Untersuchungen gab es nur Daten zu den Aktivitäten Laufen, Stehen, Sitzen, Sitzen und Tippen, Liegen, Liegen und Winken, sowie Aufzeichnungen von keiner Aktivität im Raum. Alle Ergebnisse sind nur für diese Aktivitäten und den in [6] genau beschriebenen Versuchsaufbau gültig. Für die Zukunft wäre es wünschenswert, allgemeinere Testdaten über mehr Aktivitäten zur Auwertung zur Verfügung zu haben.

Viele der Aktivitäten im ausgewerteten Datensatz waren für Konfigurationen mit großen Fenstern oder geringen Abtastraten zu kurz, so dass bei solchen Einstellungen nur sehr wenige oder gar keine Instanzen dieser Aktivitäten ausgewertet werden konnten. Ein größerer Testdatensatz, in dem die verschiedenen Tätigkeiten länger am Stück durchgeführt werden, wäre sicher eine gute Grundlage die Erkenntnisse aus diesem Paper zu vervollständigen.

6 Schlussfolgerung

In dieser Arbeit wurde der Datensatz aus [6] mit verschiedenen Parametern für Fenstergröße, Abtastung und Überlappung ausgewertet, mit dem Ziel Konfigurationen zu finden, die besonders gute Erkennungsraten zeigen.

Der Datensatz enthält Aufzeichnungen aus Experimenten, die zeigten, dass eine gerätefreie, funkbasierte Aktivitätserkennung auf Basis der Empfangsstärken möglich ist. In der originalen Untersuchung wurden diese Daten mit 40 Hertz abgetastet und annotiert, so dass jedem Sample eine Aktivität zugeordnet werden konnte. Bevor mit diesen Daten dann *kNN*, *C4.5* und *Bayes* als Erkennen trainiert und getestet wurden, gab es noch eine 2-schrittige Vorverarbeitung. Im ersten Schritt wurden immer 40 Samples, die zur gleichen Aktivität gehören, zu einem Fenster zusammengefasst, und für die einzelnen Werte darin jeweils Durchschnitt und Varianz berechnet. Samples, die in einem Fenster vorkamen, wurden für kein weiteres Fenster verwendet. Anschließend wurde die Prävalenz angeglichen, das heißt, es wurde die Tätigkeit gesucht, zu der am wenigsten Fenster gefunden wurden, und von allen anderen Tätigkeiten so viele Fenster verworfen, bis alle Aktivitäten gleich viele Instanzen hatten. Mit Training-Test-Splits auf die Testperson wurde mit *kNN* als Klassifikator 49 % Genauigkeit erreicht, mit *C4.5* 46,3 % und mit *Bayes* 48,2 %.

Hier wurde die Vorverarbeitung angepasst, um die Erkennen mit verschiedenen Fenstergrößen, Überlappungen und Abtastraten zu trainieren. Getestet wurden Fenstergrößen von 5, 10, 20, 40, 60, 80 und 120, jeweils ohne Überlappung, mit einer relativen Überlappung von 25 %, 50 % und 75 %, sowie einer Ver-

schiebung von je einem Sample zwischen den Fenstern. Um die Abtastung anzupassen wurden Samples verworfen, um eine geringere Abtastung zu simulieren. Sollte mit 20 Hertz getestet werden, so wurde jedes zweite Sample verworfen, bei 1 Hertz wurde nur jedes vierzigste Sample verwendet. Getestet wurde mit 1, 5, 10, 20 und 40 Hertz.

Für alle getesteten Überlappungen und Frequenzen, sowie für für alle Fenstergrößen außer 5, konnten mit mindestens einem Erkennen Genauigkeiten von über 50 % erreicht werden.

Bei Fenstergröße 40 war mit 20 Hertz Abtastung die Genauigkeit ähnlich gut wie bei 40 Hertz. Auch bei Fenstergröße 20 zeigt sich von 20 zu 40 Hertz eine ähnliche Tendenz. Bei diesen Fenstergrößen sind zur Erkennung 20 Hertz ausreichend.

Für alle Erkennen gibt es Konfigurationen, die besser sind als die Werte bei [6]. Mit den getesteten Daten sind für *kNN* 52 % Erkennung möglich, 3 Prozentpunkte mehr als in der ursprünglichen Konfiguration, für *C4.5* ist eine Erkennung von 53,7 % möglich, das entspricht einem plus von 7 %. Die beste erreichte Erkennungsrate mit *Bayes* war mit 48,9 % über ein halbes Prozent besser als das ursprüngliche Ergebnis.

References

1. Altman, N.: An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician* 46(3), 175–185 (1992)
2. Bao, L., Intille, S.: Activity recognition from user-annotated acceleration data. In: Ferscha, A., Mattern, F. (eds.) *Pervasive Computing, Lecture Notes in Computer Science*, vol. 3001, pp. 1–17. Springer Berlin Heidelberg (2004), http://dx.doi.org/10.1007/978-3-540-24646-6_1
3. Bouten, C., Koekkoek, K., Verduin, M., Kodde, R., Janssen, J.: A triaxial accelerometer and portable data processing unit for the assessment of daily physical activity. *Biomedical Engineering, IEEE Transactions on* 44(3), 136–147 (1997)
4. Caruana, R., Niculescu-Mizil, A.: An empirical comparison of supervised learning algorithms. In: *Proceedings of the 23rd International Conference on Machine Learning*. pp. 161–168. ICML '06, ACM, New York, NY, USA (2006), <http://doi.acm.org/10.1145/1143844.1143865>
5. Quinlan, J.R.: *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1993)
6. Scholz, M., Riedel, T., Hock, M., Beigl, M.: Device-free and device-bound activity recognition using radio signal strength. In: *Proceedings of the 4th Augmented Human International Conference*. pp. 100–107. AH '13, ACM, New York, NY, USA (2013), <http://doi.acm.org/10.1145/2459236.2459254>
7. Weiser, M.: The computer for the 21st century. *SIGMOBILE Mob. Comput. Commun. Rev.* 3(3), 3–11 (Jul 1999), <http://doi.acm.org/10.1145/329124.329126>

Gamification for Environmental Sensing

Frederik Schöll*

Advisor: Matthias Budde†

Karlsruhe Institute of Technology (KIT)
Pervasive Computing Systems – TECO

*uxcpx@student.kit.edu

†budde@teco.edu

Abstract. Gamification describes the basic approach of using game elements in non-gaming systems. This paper discusses how gamification can be used in participatory sensing projects - especially in matters of environmental sensing - and discusses the use of gamification in existing projects as well as the potential of gamification in regard to those projects.

Keywords: gamification, environmental sensing, participatory sensing, game techniques

1 Introduction

Over the past 20 years there's been a lot of progress in the game industry, leading to hundreds of millions players all over the world. This huge success is not only interesting in a financial aspect but is also bonding players to their games for plenty of hours, raising one question: Is it possible to use those powerful motivators to other purposes than just gaming?

In the 2000s *serious games* began to arise using those incentive systems for educational purposes. At the same time researches in human-computer interfaces began and with that the exploring of user experiences, design and incentive systems. [3] Now, a new design approach gained traction: gamification. Instead of developing full-fledged games single game elements are taken from the gaming context and used in non-gaming systems or products. This gamified experience helps to motivate and improve user experiences and increases the user activity as well as building his loyalty to such a non-gaming system or product. Therefore different game mechanics like achievements, leaderboards, leveling or badges are implemented to satisfy basic human needs like competition or self-expression. With Foursquare and Nike+ as commercial pioneer products, gamification reached it's break through and is now found in numerous applications and marketing campaigns. [3] [5] [4] [9]

In a nutshell, gamification doesn't mean to turn an application into a game but rather uses well-known game mechanics to make an application feel like a game. Gabe Zichermann, a familiar head in the gamification scene, defines it as follows:

The process of game-thinking and game mechanics to engage users and solve problems. [15]

This distinguishes gamification from serious as well as pervasive games. Former are full-fledged games which are merely designed for educational, training or simulation purposes rather than entertainment. Common examples for serious games are edutaining games which combine education with entertainment (for example quiz games), game learning or simulation games (for example the Microsoft Flight Simulator). [13] Pervasive games, however, adopt a complete different methodology of gamification elements. While serious games normally are build as traditional video games, pervasive games extend the game into the real world which means game boundaries are extended in temporal, spatial or social dimensions. Such games can e.g. be location-based where players have to reach different locations to progress in the game or use augmented reality to extend the perception of reality with in-game or gameplay information. [14]

2 Motivation

Participating in communities is one of the basic human behaviors which yet existed in the stone age. Since then humans participated in all kind of communities, starting from the importance of surviving to social and political engagement and even organizing free-time activities. With the growing popularity of the internet in the early 2000s and the arising ease of connecting to other people a whole new form of communities was born. Online communities in which still millions of people participate to organize and reach their individual goals.

But that's not all. When the smartphone market reached it's turning point towards mass production, smartphones and mobile internet turned into an ubiquitous part of the human life. That's not only interesting because each of those smartphones can act as a node in a location-aware, information collecting network, but also has the power to use internal and external sensors to collect this data and information autonomous. With this potential of huge distributed data collection networks, phenomenas can be observed, examined and evaluated that had been unobservable before. [2]

While the technical requirements pretty much meet up the conditions of participatory sensing there's another side which has to be considered. Communities may be the base element of our society but that doesn't mean they just work out of the box. In fact, most participants won't take active part in such a community if they can't see an individual benefit from participating. The solution to this could be gamification.

3 Incentive Systems

There are a lot of psychological theories and models on how to motivate people to do or repeat specific actions. Gamification mostly uses the reward system, meaning every time a user finishes a given task they get rewarded. This a much

practiced model and is often used in money rewarding systems, as money is one of the most powerful motivators. Due to the fact that this is a very expensive approach, particularly for non-commercial systems, gamification uses social and psychological processes to provide rewards. Those processes could be to perceive self-efficacy, an identification with a community or social approval inside a community. [3] In matters of gamification, Zichermann and Cunningham recommend in their book ([15]) a concrete scheme of rewards called SAPS.

3.1 SAPS

SAPS stands for Status, Access, Power and Stuff and orders the rewards from the most important and cheapest to the most expensive and least important. *Status* as the most important reward describes the position or rank of an individual within a social group. Users are given the ability to improve this rank by earning status rewards. Those rewards could be badges or leaderboards (which both will be explained in the next section). The second most important reward is *Access*. That means that users have to complete certain tasks to either get early access to or unlock new functions, locations or products. A typical implementation for access would be levels, i.d. users have to reach a certain level to unlock a new feature. Another good motivational reward would be the *Power* over other users. Valuable users inside the community could for example be given a job as moderator on the official forum. As this can only be applied to a certain amount of users and it requires a control instance to ensure that this power isn't misused, it's much less important than *Status* and *Access*. Finally *Stuff* in terms of giving away free items or virtual goods can be a strong incentive as well, however, as soon as the free item has been given away this incentive is over. [15]

3.2 The Bartle Test of Gamer Psychology

While the reward system as psychological theory as well as the SAPS scheme are both a general approach to motivate the users, Richard Bartle goes a step further with his *Bartle Test of Gamer Psychology*¹. In this publication he developed a series of questions to classify gamers based on their gaming preferences into four archetypes: [1]

- **Achievers** reach for rewarding goals like earning points, increasing their in-game level, earn badges or other achievements. They are *acting* on the *world*.
- **Explorers** like to explore the possibilities inside a game, discover new things or let themselves be surprised by the game. Knowledge of the game is more important than achievements or rankings. They are *interacting* with the *world*.

¹ A recent implementation of this test can be found here:
<https://www.gamerdna.com/quizzes/bartle-test-of-gamer-psychology>

- **Socializers** are more interested in connecting to other players and their influence to them than in the actual game. They are *interacting* with other *players*.
- **Killers** are similar to achievers, they also go for rankings and leaderboards to show themselves off. But instead of achievers, they often play a win-lose game: winning is just one part of the game, their superiority over other players and beating them is as important to them as their reputation. They are *acting* on other *players*.

Since each of those personality types goes for different objectives it's important to adapt a gamification concept and the reward system to those archetypes which should be addressed most.

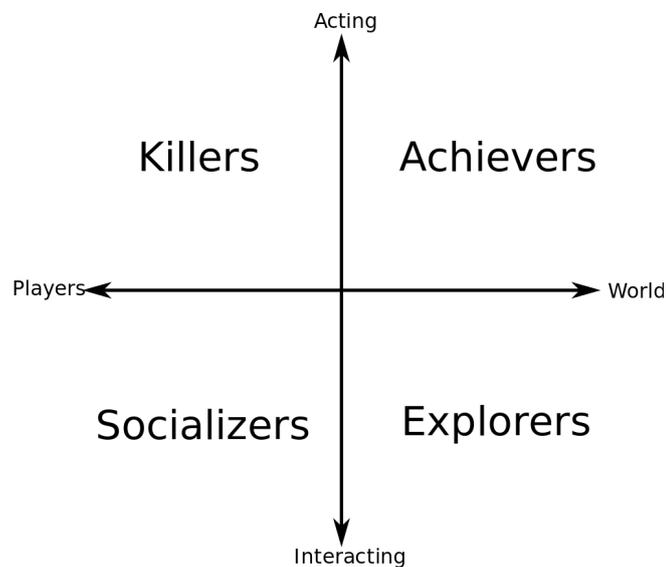


Fig. 1. Player archetypes according to the Bartle Test. Source: wikipedia.org

4 Game Techniques

Before implementing gamification elements it's useful to know the functionality of games. One approach to define a general underlying concept which applies to all games is the MDA (Mechanics-Dynamics-Aesthetics) Framework, which breaks the games into their core components and tries to formalize them.

First of all games exist from *rules*. Those rules define the gameplay and set the borders within the players can act. Every player can create his own *system* how he interacts within those borders. The result of is *fun*. In formalization, the MDA framework describes the design counterparts of rules, system and fun as *Game Mechanics* leading to *Game Dynamics* which then are leading to *Game Aesthetics*. [7]

Although the MDA framework has been designed for actual games, it can be assigned to gamification as well. In the context of gamification the game mechanics describe the rules and techniques which are employed in gamified systems. Those game mechanics could for example be points, badges or levels. Following different game mechanics then lead to game dynamics where user are being rewarded or creating competition among each other. The result then are different game aesthetics like sensation, discovery or expression. [7]

4.1 Game Mechanics

Game mechanics can be powerful incentives for all player types. Since *Killers* and *Achievers* are acting on the world respectively on other players, those game mechanics are most attractive for them. *Socializers* and *Explorers* on the other hand, are more focused on the interaction. This means the incentives for them depend more on the actual implementation and the possibilities to interact within the game mechanics than on the game mechanics itself.

Points Points are probably the most widely used game mechanic in gamified systems and sometimes even seen as mandatory for all gamified systems². They can be used to reward users across multiple dimensions or categories. Common types are Experience Points (XP) used for ranking players, Redeemable Points (RP) used for virtual economies (often named coins or cash), Skill Points used as score for special activities alongside the core (e.g. different points for commenting and participating) and Karma Points which can be exchanged between players to improve socializing and including altruism as game dynamic. As points are a very common game mechanic they work as incentive for every player type. While the typical experience points attract mostly *Killers* and *Achievers*, Karma Points for example are working good with *Socializers*. [8]

Levels Levels are used to indicate progress and are one of the most powerful motivation for gamers. In most cases they are reached by accumulating experience points. Unlocking features with a new level is another important factor in matters of motivation. As well as points they work as incentive for every player type, but especially for *Killers* and *Achievers*. [8]

Badges Badges actually exist since the middle age when they were worn by important persons. Since then they established in many organizations, especially in the military. They give users the ability to accomplish goals and signaling their status in a community. Challenges, Trophies and Achievements are very similar to Badges. They mostly attract *Achievers*, but can work as incentive for the other player types as well. [8]

² Zichermann & Cunningham in *Gamification by Design* [15]



Fig. 2. Some of the famous foursquare badges. Source: adibbehjat.com

Leaderboards & Competitions Leaderboards are another very wide spread game mechanic which are using competition to motivate players and allow them to compare themselves to others. Competitions and leaderboards are much alike, while competitions allow players to challenge and compare to other players directly (for example their friends), leaderboard scopes usually are more extensive. Again, both attract Achievers and Killers the most, but competitions can be a good incentive for Socializers too. [8]

Virtual Goods Virtual goods are non-physical objects which can be used in online communities, like avatars, decorations or clothing. As these are intangible they often have to be purchased and help the players to create an virtual identity within the community. Rewarding users with virtual goods can be a strong incentive, especially for *Achievers* and *Socializers*. However, while other game mechanics can create constant incentives, the incentive of a virtual good as reward is over as soon as this item is given out. [8]

4.2 Game Dynamics

How the game mechanics act on the players at run-time and how they influence each others' outputs over the time is described by the game dynamics. The key game dynamics are the following:

Reward Reward is referred to any kind of value given for usually fixed actions where users get a reward for completing a certain task. That gives users a motivation to work towards those goals because they know what to do for a certain reward as well as what they get for it. Despite rewarding fixed actions, there are sudden (given out unexpectedly) and random rewards (type of reward is random and creates a suspense for the user). [8]

Status Status within a social group is achieved by establishing a level of recognition, fame, reputation and appreciation. Those dynamics are driven by all of the game mechanics with levels being the most important. [8]

Self-expression Another important factor besides the Status is to let the users create their own and unique identity within a social group and show off their style and personality. A common way for players to achieve this are virtual goods. [8]

Competition vs. Altruism Competition is another pervasive behavior in human societies, found in sports, politics, economics, education. . . As humans tend to compare themselves to each other a lot and thereby gain satisfaction, competition is a very powerful motivator. Typical game mechanics to create competition amongst the players would be leaderboards and challenges.

On the other hand altruism can work as powerful motivator as well. Especially for Socializers gift-giving can be very interesting as they often use gamified systems to foster relationships.

4.3 Game Aesthetics

Finally the game aesthetics describe the emotional responses which are invoked in the player when the game dynamics are applied. The most important aesthetics are Sensation, Fantasy, Narrative, Challenge, Fellowship, Discovery, Expression and Submission. As those aesthetics are not that important for gamification (unlike for the creation of games), this paper will not discuss those in detail. [7]

5 Gamified Systems for Environmental Sensing

This section will give an overview over existing gamified systems for environmental sensing. Since there are nearly no such systems at all at the time this paper has been written, I will start with an example for a gamified system as well as an example for an environmental sensing system before I'm going to introduce a gamified system for environmental sensing.

5.1 Example for a Gamified System: Stackoverflow.com

Stackoverflow.com is a website where users can create and answer questions related to computer programming. Although stackoverflow.com doesn't look like a game at all, it uses a lot of gamification and implements the SAPS reward scheme:

Status: Status is achieved by increasing the so called reputation which users gain if other users vote up their questions or answers, or loose if they get voted down. This reputation is also be used to create bounties where users can set a certain amount of his own reputation on a question of him. This works as an additional motivator as the user who answers the question will get this reputation. Besides of the reputation the users can earn badges if they fulfill certain tasks (for example if a user writes an answer which gets voted up at least 100 times). To display the status of the users there are different leaderboards ranking the users by reputation and badges.

- Access:** With increasing reputation the user unlocks new features. Answering questions or voting for example is not possible until the user reaches a certain level of reputation. The higher the reputation gets the more interesting are the unlocks. Reaching a reputation of 200 for example disables some ads on the website.
- Power:** At a certain amount of reputation the unlocks give the users power of other users. Users with a very high reputation are able to edit or close questions and answers from other users, flag them, manage tags or even (un-) delete questions and answers.
- Stuff:** Rewarding users with stuff is currently³ not implemented. Since stuff is the most expensive and least important reward this should not have a high impact on the gamification concept.

In terms of game mechanics, the reputation would be a mixture of experience and karma points, since they are not only used to unlock features but also can be exchanged between players with bounties. The resulting game dynamics are status, competition and in a way self-expression as the users can upload own avatars and describe themselves very detailed on their profile.

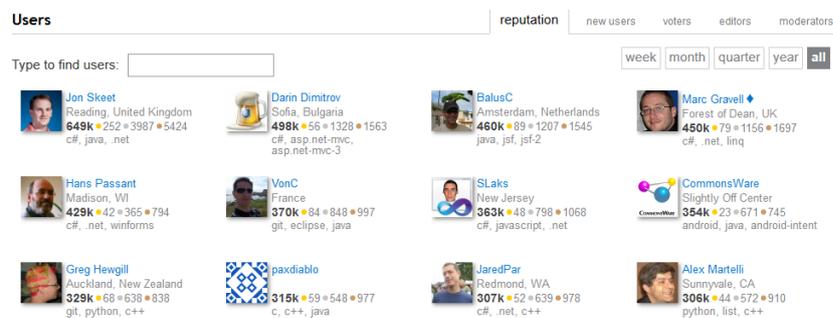


Fig. 3. The Stackoverflow.com leaderboard as of February 16th, 2014. Source: stackoverflow.com

5.2 Example for a Environmental Sensing System: PEIR

PEIR is a mobile application which uses participatory sensing to estimates the personal impact on the environment in relation to carbon dioxide and particular matter in general as well as near sensitive sites (like hospitals, schools, etc.). Additional PEIR also calculates estimated, personal exposures to smog and fast-food eating establishments.

To calculate the personal, environmental impact PEIR uses the time and the location from the user's smartphone and uploads it to a private repository. First of all this data is processed for the further analyses and classified whether the user is staying at one position, walking or driving (detecting biking and the use of mass transits are being developed). In the next step the previously obtained

³ As of March 2014.

data is combined with static and dynamic data sources, like road maps, specific locations, weather, traffic conditions and vehicle emissions. Based on this data collection the estimated impact on carbon dioxide and particulate matter (PM 2.5) is calculated, especially for sensitive sites like hospitals and schools.

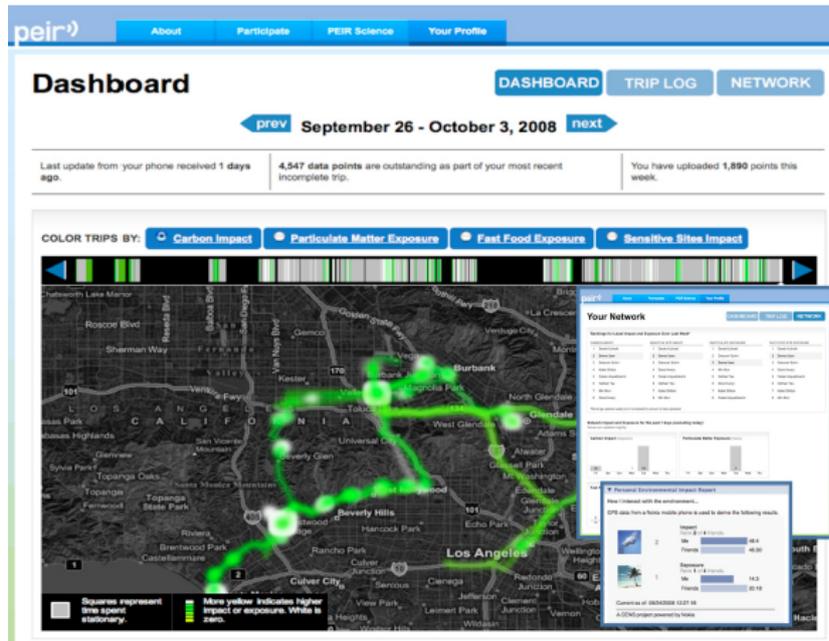


Fig. 4. Screenshot of PEIR. Source: [12]

5.3 Mobile Application for Noise Pollution Monitoring through Gamification Techniques

Noise pollution nowadays can be a serious problem, especially in cities. Medical studies state that besides of stress and annoyance noise pollution can affect the human health seriously. This not only implies obvious problems like sleep issues or hearing impairment but also much more severe effects on the cardiovascular as well as the immune system. Since the daily noise is inevitable it's not possible to avoid noise pollution at all, but the approach of using a mobile application can identify potential and actual problematic areas in real-time. While this could also be monitored by sensor networks, a mobile application using participatory sensing can provide a much more accurate and cheaper solution. Latter is confirmed by the fact that the authors of this study use the microphones of personal smartphones and therefore avoid expenses for additional hardware. Since the data collection depends on voluntary participation of citizens gamification is an important point in keeping participants motivated to collect the data. [10]

The gamification elements are based on the SAPS model and differ in two version of the prototype. The first one is called *Noise Battle* and is targeting

Achievers while the second one, *Noise Quest*, tries to reach out for the *Explorers*. Following, the implementation of the SAPS reward elements:[10] [6]

- Status: Status is achieved as users gain points for collecting noise pollution information. These points increase the users general score and therefore their level. As status is a general and very important gamification element it's implemented in both versions of the mobile application.
- Access: Access differs in the two versions. The *Noise Battle* prototype encourages users to battle over city areas while the *Noise Quest* prototype demands that users unlock quests.
- Power: Power is included only in the *Noise Battle* prototype and allows good players to send sounds to other players.
- Stuff: In this case stuff specifies a set of avatars and items as reward. It's included in both prototypes.

Similar mobile applications for gathering noise pollution data are for instance *Lärmometer*⁴, *Noise Tube*⁵ or *NoiseSpy*⁶. In contrast to the *Noise Battle / Quest* application they use less or no gamification elements at all and in particular they have no noticeable gamification concept.

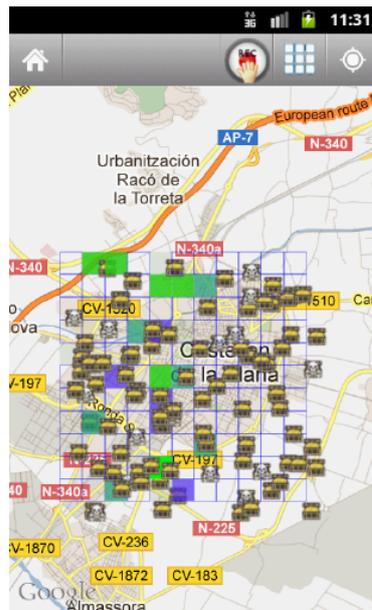


Fig. 5. Screenshot of Noise Battle. Source: [6]

⁴ Lärmometer – a Mobile Noise Mapping Application

⁵ <http://www.noisetube.net>

⁶ NoiseSPY: A Real-Time Mobile Phone Platform for Urban Noise Monitoring and Mapping

5.4 Nissan Carwings

Nissan Carwings⁷ is just one of the many gamified systems by the automobile industry. Besides Nissan there are similar systems from Toyota⁸, Chevrolet⁹ or Honda¹⁰ just to name a few. Those systems measure the environmental impact of the driver by evaluating the use of the accelerator and brake pedal, the covered distances and the usage of heater and air conditioner. On the basis of this eco rating Carwings creates a competition amongst its users by offering leaderboards and in addition are ranking the players to distribute Bronze, Silver, Gold and Platinum trophies. Although this is currently just a promotion for green driving and most likely used to influence the drivers to improve the battery duration in electric cars, using the gained data in conjunction with additional data sources could easily provide a system for environmental sensing. If, for example, this system would be deployed to standard cars with combustion engines the environmental impacts of numerous drivers could be measured in real-time (similar to PEIR).



Fig. 6. Screenshot of Nissan Carwings. Source: green.autoblog.com

⁷ <http://www.nissan-global.com/EN/ENVIRONMENT/SOCIAL/CARWINGS/>

⁸ Toyota Entune Eco Dashboard

⁹ Onstar Volt Driver Challenge App (<http://www.chevrolet.com/voltdcapp.html>)

¹⁰ Honda Eco-Assist

5.5 Other systems for environmental sensing

There are a lot other systems used for environmental sensing which use gamification elements sporadically or renounce them completely. The following list shows a selection of such systems:

- **Sense the Beach**¹¹ calls on their users to report polluted beaches. Therefore an android application has been developed which allows the user to commit geo-tagged pictures of the pollution. Besides of a leaderboard ranking the users with the most commits no gamification elements are used.
- **FracTracker**¹² is a mobile application which allows their users to submit observations of potential industrial impacts and suspected well water impacts.
- **Naturgucker.de**¹³ **and Artenfinder.de**¹⁴ are German websites for wildlife conservations. Therefore a website (naturgucker.de) or mobile applications (artenfinder.de) are provided so users can report occurrences of endangered species.
- **P-Sense[11]** is a 'participatory sensing system for air pollution monitoring and control'. Sensors are used to measure carbon dioxide, humidity, temperature, combustible gases and carbon monoxide. The author describes the incentives as work in progress.

6 Discussion and Conclusion

6.1 Discussion about the highlighted systems

PEIR, the Personal Environmental Impact Report, as a Platform for Participatory Sensing Systems Research is a very powerful tool in matters of environmental sensing. PEIR only needs to trace the location in connection with the time of their participants. This means there are no additional expenses for hardware as PEIR uses personal smartphones and there's less need for involvement of the users as the data tracing could work autonomous. Unfortunately PEIR has no gamification concept at all. It uses leaderboards to let the users compare themselves to either all participants or their Facebook friends, but the reason for implementing those leaderboards primarily focuses on the comparison. Users need to know comparative values to be able to rate their own carbon dioxide and particular matter impact (resp. exposure).

Mobile Application for Noise Pollution Monitoring through Gamification Techniques currently has the best gamification approach of the systems developed for environmental sensing. They did not only use a well-known reward

¹¹ <http://sensethebeach.appspot.com/>

¹² <http://www.fractracker.org/>

¹³ <http://naturgucker.de/>

¹⁴ <http://artenfinder.de>

scheme (SAPS) but also looked into the *Bartle Test* and stuck to two players archetypes they mainly reach out for. Namely this is the *Achiever* which should be addressed by the *Noise Battle* prototype and the *Explorer* which should be addressed by the *Noise Quest* prototype. As this application still is in the state of prototyping it's difficult to evaluate the efficiency and effectivity of the deployed game mechanics and the resulting game dynamics. In addition the description of their implementation of the SAPS model is kept fairly general, making it difficult to clearly distinguish the purpose of some gamification elements.

Summing up the *Noise Battle / Quest* application has a sophisticated approach for their gamification concept. In their final version they shouldn't use one application per player archetype but instead try to reach all four of Bartle's player archetypes with the same application. Especially the *Socializers* ought to be caught rather easy with the present availability of social networks.

Nissan Carwings and alternative gamified systems in cars are an interesting approach in matters of environmental sensing as well as gamification. Currently both are implemented fairly rudimentary. For environmental sensing data is available but not analyzed and gamification is implemented rather sporadic instead of following a concept. Depending on their environmental impact the drivers are rated using a Bronze, Silver, Gold or Platinum *Achievement*. Additionally the savings in the carbon dioxide emissions are counted as saved trees which equates to the *Reward* game dynamic. Since the amount of achievements is insufficient for long term motivation and the reward is more a statistic than an actual reward, there's a lot of unused potential. Furthermore the addressed archetypes according to Bartle are *Achievers* only. Reaching out for the other three archetypes wouldn't be a mistake.

6.2 Conclusion

Gamification has proven itself to be a powerful motivator when Nike+ and Foursquare included it into their applications and now are having 18 million¹⁵ (Nike+) respectively 45 million¹⁶ users (Foursquare) to show and still hold on to their gamification concepts. Since environmental sensing systems especially rely on the voluntary input of their users gamification could function as a huge motivator to keep the users participating. Unfortunately there is no productive system for environmental sensing that uses a gamification concept¹⁷. Some of those systems are in fact using some gamification elements, but that seems to be implemented rather 'accidentally' or 'random'. An example for the former would be PEIR, which uses a leaderboard that is not due to gamification decisions but rather comparability. 'Randomly' implemented gamification would be for example the Nissan Carwings which actually uses gamification elements

¹⁵ <http://nikeinc.com/news/nike-evolves-just-do-it-with-new-campaign>

¹⁶ <http://blog.foursquare.com/post/70494343901/ending-the-year-on-a-great-note-and-with-a-huge-thanks>

¹⁷ As of March 2014.

like achievements but it doesn't seem as there would be a concrete gamification concept behind it, at least not for a long-term motivation. In a nutshell gamification has a great potential to act as motivator but it's big potential is currently unused in environmental sensing systems.

References

1. Bartle, R.A.: Hearts, clubs, diamonds, spades: Players who suit muds, <http://www.mud.co.uk/richard/hcds.htm>, published on Website
2. Burke, J.A., Estrin, D., Hansen, M., Parker, Andrew and Ramanathan, N., Reddy, S., et al.: Participatory sensing. Paper, Center for Embedded Network Sensing, UCLA (2006), <http://escholarship.org/uc/item/19h777qd>
3. Deterding, S.: Gamification: Designing for motivation. *interactions* 19(4), 14–17 (Jul 2012), <http://doi.acm.org/10.1145/2212877.2212883>
4. Deterding, S., Dixon, D., Khaled, R., Nacke, L.: From game design elements to gamefulness: Defining "gamification". In: *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*. pp. 9–15. MindTrek '11, ACM, New York, NY, USA (2011), <http://doi.acm.org/10.1145/2181037.2181040>
5. Deterding, S., Sicart, M., Nacke, L., O'Hara, K., Dixon, D.: Gamification. using game-design elements in non-gaming contexts. In: *CHI '11 Extended Abstracts on Human Factors in Computing Systems*. pp. 2425–2428. CHI EA '11, ACM, New York, NY, USA (2011), <http://doi.acm.org/10.1145/1979742.1979575>
6. García-Martí, I., Rodríguez-Pupo, L.E., Díaz, L., Huerta, J.: Noise battle: A gamified application for environmental noise monitoring in urban areas (2013), http://www.agile-online.org/Conference_Paper/CDs/agile_2013/Short_Papers/SP_S1.3_Garcia_Marti.pdf
7. Hunnicke, R., Leblanc, M., Zubek, R.: Mda: A formal approach to game design and game research. In: *In Proceedings of the Challenges in Games AI Workshop, Nineteenth National Conference of Artificial Intelligence*. pp. 1–5. Press (2004)
8. Inc., B.: Gamification 101: An introduction to the use of game dynamics to influence behavior (2010), white Paper
9. Liu, Y., Alexandrova, T., Nakajima, T.: Gamifying intelligent environments. In: *Proceedings of the 2011 International ACM Workshop on Ubiquitous Meta User Interfaces*. pp. 7–12. Ubi-MUI '11, ACM, New York, NY, USA (2011), <http://doi.acm.org/10.1145/2072652.2072655>
10. Martí, I., Rodríguez, L., Benedito, M., Trilles, S., Beltrán, A., Díaz, L., Huerta, J.: Mobile application for noise pollution monitoring through gamification techniques. In: Herrlich, M., Malaka, R., Masuch, M. (eds.) *Entertainment Computing - ICEC 2012, Lecture Notes in Computer Science*, vol. 7522, pp. 562–571. Springer Berlin Heidelberg (2012), http://dx.doi.org/10.1007/978-3-642-33542-6_74
11. Mendez, D., Perez, A., Labrador, M., Marron, J.: P-sense: A participatory sensing system for air pollution monitoring and control. In: *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on*. pp. 344–347 (March 2011)
12. Mun, M., Reddy, S., Shilton, K., Yau, N., Burke, J., Estrin, D., Hansen, M., Howard, E., West, R., Boda, P.: Peir, the personal environmental impact report, as a platform for participatory sensing systems research. In: *Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services*. pp.

- 55–68. MobiSys '09, ACM, New York, NY, USA (2009), <http://doi.acm.org/10.1145/1555816.1555823>
13. Susi, T., Johannesson, M., Backlund, P.: Serious games : An overview. Tech. Rep. HS- IKI -TR-07-001, University of Skövde, School of Humanities and Informatics (2007), <http://urn.kb.se/resolve?urn=urn:nbn:se:his:diva-1279>
 14. Walther, B.K.: Atomic actions – molecular experience: Theory of pervasive gaming. *Comput. Entertain.* 3(3), 4–4 (Jul 2005), <http://doi.acm.org/10.1145/1077246.1077258>
 15. Zichermann, G., Cunningham, C.: *Gamification by Design: Implementing game mechanics in web and mobile apps*. O'Reilly Media, Inc. (2011)