Karlsruhe Reports in Informatics 2015,8
Edited by Karlsruhe Institute of Technology,
Faculty of Informatics
ISSN 2190-4782

Ubiquitäre Systeme (Seminar) und Mobile Computing (Proseminar) SS 2015

Mobile und Verteilte Systeme
Ubiquitous Computing

Teil XIII

Herausgeber:
Martin Alexander Neumann, Anja Bachmann,
Matthias Berning

2015

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

Karlsruhe Institute of Technology

Ubiquitäre Systeme (Seminar) und Mobile Computing (Proseminar) SS 2015

Mobile und Verteilte Systeme Ubiquitous Computing

Teil XIII

Herausgeber

Martin Alexander Neumann
Anja Bachmann
Matthias Berning

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

Interner Bericht 2015-08
ISSN 2190-4782

Vorwort

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

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

Dieser Seminarband fasst die Arbeiten der Seminare des Sommersemesters 2015 zusammen. Die Themenvielfalt der hier zusammengetragenen Aufsätze umfasst Ambulantes Assessment, Social Sensing, Augmented Reality und Dynamische Softare Updates für Java. Wir danken den Studierenden für ihren besonderen Einsatz, sowohl während des Seminars als auch bei der Fertigstellung dieses Bandes.

Karlsruhe, den 31. September 2015

Martin Alexander Neumann
Anja Bachmann
Matthias Berning

Inhaltsverzeichnis

Smartphonebasiertes Ambulantes Assessment

Christian Biegemeier*
Betreuer: Anja Bachmann†

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

Zusammenfassung. Diese Ausarbeitung befasst sich mit der Unterstützung der Datenerfassung in Experimenten im natürlichen Umfeld eines Probanden. Im Verlauf eines Experimentes werden häufig Fragebögen erhoben, wobei sich die Problematik des richtigen Zeitpunktes der Abfrage sowie die sinkende Teilnahmebereitschaft bei zu vielen Fragen ergibt. Um diesen Problemen entgegen zu wirken, wird das ambulante Assessment verwendet, was den Einsatz computergestützter Datenerfassung im alltäglichen Leben beschreibt. Ein Problem, das sich allerdings aus dessen Einsatz ergibt, ist der ungewohnte Umgang mit speziellen sensorbehafteten Geräten, was zu deren häufigen Vergessen führen kann. In dieser Arbeit wird daher eine Möglichkeit der Datenerfassung thematisiert, die den meisten Probanden sowohl zugänglich als auch vertraut ist - die Datenerfassung über Smartphones. Smartphones bieten weiterhin die Möglichkeit, durch die Vielzahl an integrierten Sensoren, verschiedenste Kontexte des Nutzers zu erfassen. Ihr Einsatz in diesem Kontext wurde bereits in anderen Arbeiten praktiziert, jedoch lösten diese dabei nicht das allgemeine Problem des großen Umfangs eines Fragebogens und der damit einhergehenden Demotivation. In dieser Ausarbeitung wird daher der Einsatz der Smartphonesensorik zur Erleichterung der Eingabe des Nutzers für elektronische Fragebögen thematisiert.

1 Einleitung

Die Erfassung von Daten spielt in Zeiten von „Big Data“ eine große Rolle. Doch nicht nur für Unternehmen ist die Gewinnung von Informationen sinnvoll, auch Einzelne können von der Datenerfassung profitieren. Diese kann sich dabei beispielsweise auf die Überwachung der Vitalwerte einer Person beziehen. Allerdings kommt es hierbei stark auf die richtige Methode an. Schrader, Schoel und Scheler zeigten schon 1990 dass sich stationäre Messungen auf deren Ergebnisse auswirken [26]. Sie zeigten in ihrem Versuch, dass mehr als zwanzig Prozent der Untersuchten zwar beim Arzt einen erhöhten Blutdruck aufwiesen, was sich allerdings bei Alltagsmessungen nicht bestätigte. Durch diese so genannte Praxishypertonie sind laut Schrader et al. wahrscheinlich hunderttausende Personen falsch diagnostiziert und behandelt worden. Gleichzeitig zeigt es aber auch, wie zweifelhaft die Generalisierbarbeit von Labormessungen sein kann. Um solche

Fehlmessungen zu eliminieren, sollten personenbezogene Messdaten möglichst unter alltagsnahen Bedingungen gemessen werden. Doch nicht nur im Bereich der Medizin ergeben sich Probleme mit stationären Messungen. Viele psychologische Überwachungsaufgaben lassen sich ebenfalls schlecht unter Laborbedingungen rekonstruieren. Beispielweise ergibt sich die Problematik an riskanten Arbeitsplätzen oder bei bestimmten Verhaltensstörungen [15]. Die Problematik an riskanten Arbeitsplätzen liegt darin begründet, dass sich das Verhalten sowie auch die messbaren Werte in Gefahr verändert und dies schlecht im Labor rekonstruiert werden kann. Ähnliches gilt bei Verhaltensstörungen, diese können teilweise nur in bestimmten Situationen aber auch sporadisch auftreten.

Eine Lösung liefert das in dieser Ausarbeitung behandelte ambulante Assessment. Diese Messstrategie beschäftigt sich mit der Erfassung von Daten durch computergestützte Methoden, während die zu untersuchende Person ihren alltäglichen Tagesablauf nachgehen kann. Hierbei sollten Datenerfassungssysteme zum Einsatz kommen, welche gut in den Alltag etabliert werden können und nicht beeinträchtigend auf diesen wirken. Aus heutigem Stand ist es naheliegend, dafür auf die Erfassung durch Smartphones zurückzugreifen, da diese nicht nur eine Vielzahl an integrierten Sensoren mit sich bringen, sondern auch weit verbreitet sind [1] und häufig unmittelbar am Körper getragen werden. Durch integrierte Sensoren wie den GPS-Sensor oder den Beschleunigungssensoren können beispielsweise die Lokation des Nutzers bestimmt oder der aktuelle Bewegungszustand (stehen, laufen, sitzen oder liegen) geschätzt werden.

Zum besseren Verständnis der Ausarbeitung wird im folgenden Kapitel auf die wichtigsten Begriffe und Kennzeichen des ambulantem Assessment eingegangen. Im Anschluss werden die Vorzüge des Einsatzes des ambulanten Assessments, einige Einsatzmöglichkeiten, die grundlegenden Strategien sowie die Problematik der Akzeptanz und Compliance bei Probanden erläutert. Da ein wesentlicher Aspekt zur Erleichterung der Datenerfassung die Einbeziehung des Kontextes des Probanden darstellt, wird in Kapitel „Kontexterfassung durch Smartphones“ genauer auf diesen eingegangen und mithilfe von Beispielen illustriert. Zuletzt wird dem Leser ein Fazit des smartphonebasierten ambulanten Assessment gegeben sowie auf mögliche Erweiterungen und nicht betrachtete Einsatzgebiete für weitere Ausarbeitungen eingegangen.

2 Grundlagen

In diesem Kapitel werden Begriffe, deren Bedeutung und Unterschiede zueinander für das Verständnis dieser Ausarbeitung hilfreich sind, erläutert.

2.1 Kontext und Kontexterfassung

Im Allgemeinen wird unter dem Begriff des Kontextes die Umgebung einer Sache verstanden, z.B. wie etwas im Kontext gesehen wird. Da dies allerdings nur unpräzise wiedergibt, was unter dem Begriff zu verstehen ist, definiert Augsburg

diesen folgendermaßen: „Kontext ist jede Information, welche dazu genutzt werden kann, die Situation einer Entität zu beschreiben, wobei eine Entität eine Person, ein Platz oder ein Objekt sein kann“ [6]. Zusätzlich wird dabei unter *Gerätekontext*, *Nutzerkontext* und *Umweltkontext* unterschieden. Demzufolge ist also unter einer Kontexterfassung die möglichst genaue Wiedergabe des aktuellen Kontextes zu verstehen.

Wichtig in der Betrachtung der Umwelt durch Sensorik ist es zu beachten, dass immer nur ein Bruchteil des kompletten Kontextes erfasst und somit in die Klassifikation mit einfließen kann. Ein typisches Beispiel im Bereich der Smartphone Kontexterfassung ist das Kategorisieren des aktuellen Bewegungszustandes in die Bereiche: stehen, laufen, rennen und liegen. Nimmt man an, dass sich das Handy des Nutzers in dessen Tasche befindet, wird bspw. dessen Handbewegung nicht zur besseren Klassifikation mit einfließen können, obwohl dies ein Teil des Kontextes ist.

2.2 Monitoring

Die Begriffe ambulantes Assessment und ambulantes Monitoring werden häufig synonym gebraucht. Allerdings wird unter dem Begriff Monitoring eher eine kontinuierliche Messung von beispielsweise physiologischen Daten verstanden, während sich das Assessment vornehmlich auf die Eingabe von Daten in Tagebüchern oder digitalen Fragebögen durch die Testperson bezieht [25]. Fahrenberg et al. [15] empfehlen im deutschsprachigen Raum den Begriff ambulantes Assessment. Da dieser jegliche Form der Datenerhebungen mit einbezieht, ist damit auch das Monitoring eingeschlossen.

2.3 Feld und Labor

Im Kontext von Feld- und Laborexperimenten ist der Begriff der *Natürlichkeit* ein wichtiger Aspekt, weshalb nun in Kürze auf diesen eingegangen wird. Die Natürlichkeit eines Experimentes wird unter den Gesichtspunkten des *Verhaltens*, des *Kontextes* sowie der *Bedingungsänderung* gesehen. Dies bedeutet laut Patry, dass das Verhalten eines Menschen als natürlich zählt, wenn sich dieser so verhält wie in seiner „gewohnten sozialen und materiellen Umwelt, auch wenn er nicht Gegenstand einer Untersuchung ist, was er tut, wenn kein Versuchsleiter in direkt oder indirekt beeinflusst oder veranlasst etwas zu tun“ [23]. Der Begriff der Natürlichkeit ist deshalb so wichtig zu verstehen, da ein Experiment das einen gewissen Sachverhalt erläutert oder begründet, nur dann relevant ist, wenn sich das Verhalten der Probanden im Experiment auch in der Realität widerspiegelt - also natürlich ist.

In der Methodologie werden typischerweise Laborexperimente und Feldexperimente unterschieden. Im Folgenden werden die Unterschiede der Experimenttypen genauer erläutert, siehe auch Tabelle 1.

Klare Vorzüge eines **Laborexperimentes** sind deren Kontrollierbarkeit, da diese durch zweckmäßige Isolierung von Bedingungen des Versuches und deren Kontrolle eine präzise Prüfung der Hypothese ermöglichen (siehe Tabelle 1, a).

Gleichzeitig ist die Kontrollierbarkeit und Einschränkung auch dessen größter Nachteil. Aufgrund der strikten Kontrolle entsteht eine künstliche Umweltbedingung der Untersuchung (siehe Tabelle 1, c). Da die daraus resultierenden Artefakte (durch 'unechtes' Verhalten) nicht klar erkennbar sind, wirkt sich dies oft negativ auf die externe Validität aus und führt im Allgemeinen zu einer geringeren Relevanz der Ergebnisse. Weiterhin können in Laborexperimenten Effekte wie Gruppenzwang oder die Hilfsbereitschaft schlecht rekonstruiert werden (siehe Tabelle 1, d).

Dem gegenüber stehen die realitätsnäheren Untersuchungsbedingungen von **Feldexperimenten** (siehe Tabelle 1, b). Hieraus entsteht im Gegensatz zu Laborexperimenten eine höhere externe Validität, woraus eine größere Relevanz der Ergebnisse zu folgern ist. Bei Feldexperimenten können Situationen erfasst werden, die aus den Einschränkungen im Labor nicht zu erreichen sind. Als Grund für Feldexperimente werden daher des Öfteren das natürliche Umfeld des Probanden und der natürliche Umgang mit seiner Umwelt im sozialen sowie im materiellen Sinne genannt (siehe Tabelle 1, a). Allerdings treten hier Nachteile wie eine erschwerte Schlussfolgerung durch eine niedrigere Zuverlässigkeit der Datenerhebung sowie einer schlechteren Kontrollierbarkeit auf.

Zusammengefasst stehen sich bei den Experimenttypen die Ziele der Aussagenpräzision und der Aussagengeneralisierbarkeit gegenüber [14]. Daraus resultiert eine für jedes Experiment einzeln abzuwägende Entscheidung, je nach Ziel und Fokus des Experimentes.

2.4 Reportquellen

Unabhängig davon, ob eine Experiment im Feld oder Labor stattfindet, werden typischerweise Daten erhoben, um eine These zu prüfen oder aufzustellen. Hierbei lassen sich drei Typen von Reportquellen unterscheiden [25]:

- *Selbstbericht*: Bei dieser Art bewertet sich der Proband selbst. Z.B. das Führen eines elektronischen Tagebuches. (s.u.)
- *Verhaltensbeobachtung*: Andere Personen übernehmen die Messung und Dokumentation der Daten. (s.u.)
- *Automatischer Bericht*: Hierbei erfolgt die Datenaufnahme automatisiert. Es wird keine Nutzereingabe zur Datenaufnahme benötigt.

Da der Fokus dieser Arbeit auf Methoden liegt, bei welcher der Nutzer eine Interaktion tätigt, werden im Folgenden nur Selbstberichte sowie Verhaltensbeobachtungen als Reportquellen betrachtet.

Selbstbericht

In Selbstberichten muss der Teilnehmer eines Experimentes meist einen Fragebogen oder Protokollblätter nach seiner eigenen Erinnerungen und Gefühlen ausfüllen. Selbstberichte beziehen sich je nach Fragestellung der Untersuchung auf einzelne Aspekte von Tätigkeiten, aktuellem Befinden, Ausprägung von Symptomen usw [14]. Um Selbstberichte über den Tagesverlauf und Befindlichkeiten

Tabelle 1. Vorzüge und Nachteile von Laborexperiment und Feldstudie [14]

	Laborexperiment	**Feldstudie**
a.	Präzision der Hypothesenprüfung: Zuverlässigkeit der Schlussfolgerung aufgrund von Isolation und aktiver Manipulation der UV[1], Kontrolle der Fehlervarianz	Ökologische Validität[2] aufgrund naturalistischer Beobachtungen und repräsentativer Versuchsplanung
b.	relativ höhere interne Validität[3]	relativ höhere externe Validität[4]
c.	Künstlichkeit der Untersuchungsbedingungen, u.U. geringe Relevanz für Teilnehmer	multiple Effekte, deren Komplexität schwer kontrollierbar ist
d.	Effektstärken aus ethischen und praktischen Gründen beschränkt	höhere, realistische Effektstärken möglich

[1] UV (Unabhängige Variable): Variable in einem Experiment, mit deren Hilfe eine abhängige Variable (z.B. Lernerfolg) vorhergesagt werden soll
[2] Ökologische Validität: Empirische Gültigkeit des Befundes einer psychologischen Untersuchung im Alltagsgeschehen
[3] Interne Validität: Trifft Aussage darüber inwieweit das gemessen wird, was gemessen werden soll
[4] Externe Validität: Generalisierbarbeit und Repräsentativität des Experimentes

zu gewinnen, werden seit langem psychologische Fragebogen, Skalen und Protokollblätter verwendet. Hintergrund solcher Berichte ist meist das Ergründen (Diagnostik) von psychologischen und medizinische Beschwerden wie Schmerzen oder andere Symptome. Darüber hinaus können regelmäßige Tagesprotokolle zur selbstständigen Überwachung (Selbstmonitoring) von Gesundheitsstörungen oder chronischen Krankheiten dienen [14].

Fahrenberg warnt allerdings davor, dass Selbstbeurteilung nicht als Verhaltensdaten ausgelegt werden dürfen, da die Erfassung von tatsächlichem Erleben und Verhalten „auf einem unberechtigten Optimismus bezüglich der Leistungsfähigkeit des menschlichen Gedächtnisses und des subjektiven statistischen Inferenzvermögens“ [15] des Befragten beruht. Demzufolge führen Selbstauskünfte oft zu schwerwiegenden Fehleinschätzungen, wodurch große Diskrepanzen zwischen den Selbstberichten und den Verhaltensweisen, Beurteilungen durch Bezugspersonen oder physiologischen Messwerten entstehen können (so genannter retrospektiver Bias). Als Beispiel hierfür nennen Foerster, Leonhart und Fahrenberg 2003 [14]:

- „Diskrepanzen zwischen chronischen körperlichen Beschwerden und objektivierbaren medizinischen Befunden“ sowie
- „Diskrepanzen zwischen Selbsteinstufungen von emotionaler Aktivierung/ Beanspruchung bzw. Stressreaktionen und den Messungen der körperlichen Aktivierungsprozesse“

Verhaltensbeobachtung

Die Psychologie beschäftigt sich zu einem großen Teil mit Verhaltenswissenschaft. Hierbei geht es um das spontane Verhalten im Alltag, in sozialen Gruppen, am Arbeitsplatz, während der Freizeit etc. „Ein großer Teil der psychologischen Forschung wurde als Verhaltensexperiment mit standardisierter Verhaltensbeobachtung im Labor durchgeführt“ [14]. In der klinischen Psychologie wurden allerdings Verfahren der Verhaltensdiagnostik entwickelt, um unter alltäglichen oder lebensnah konstruierten Bedingungen Verhaltensstörungen und Symptome zu erfassen [14]. Verhaltensweisen wie z.B. Tätigkeiten, soziale Interaktionen, das Essverhalten, Rauchen und Trinken oder auch bestimmte Krankheitssymptome können als Selbstbeobachtungen protokolliert werden. Sie könnten aber auch durch unabhängige (Verhaltens-)Beobachter registriert werden, in so genannten Verhaltensbeobachtungen. Allerdings gibt es in der Methodenlehre der Psychologie bereits seit einiger Zeit eine Auseinandersetzung über die Validität solcher Verhaltensbeobachtungen. Einerseits kann durch mehrere Beobachter festgestellt werden, wie gut deren Aussagen übereinstimmen, sodass ein hohe Validität erreicht werden kann. Andererseits wird die Beschreibung des Verhaltens von der Beurteilung des Kontextes und von den Konzepten des Beobachters beeinflusst, sodass bei mehreren Beobachtern mehrere Wahrnehmungen aufeinander treffen können [14]. Das bedeutet dass auch bei Verhaltensbeobachtungen ähnlich wie bei Selbstberichten, keine voraussetzungsfreie Wahrnehmung des Sachverhaltes erfolgen kann und damit die externe Validität der Untersuchung sinkt.

2.5 Daten eines computergestützten Protokolls

Sowohl Selbstbericht wie auch Verhaltensbeobachtungen werden im Bezug dieser Ausarbeitung mithilfe von computergestützten Protokollen, wie bswp. elektronische Fragebögen, erhoben. Die Daten die in einem solchen computergestützten Protokoll erfasst werden, sind meist objektive Settingmerkmale und Umgebungsmerkmale sowie Erlebnisberichte, Verhaltensberichte und Testwerte/ Messwerte [14]. Der Begriff Setting bezieht sich dabei auf die Umgebung und den Ort der Handlung. Das Merkmal eines Setting (Settingmerkmal) beschreibt, ob sich der Proband in einem Geschäft, in einer Freizeitaktivität, auf Märkten oder in einem Verkehrsmittel befindet. Weiterhin sind aber auch die umgebenden Objekte selbst, wie z.B. die umgebenden Personen gemeint. Unter Umgebungsmerkmalen meint man dagegen die physikalischen Eigenschaften der Umgebung, wie die Temperatur, Schallpegel etc. In den Selbstberichten trägt der Proband Daten wie sein Stimmung- oder Emotionsbild, Gedanken, Schmerzen usw. ein. Im Vergleich zu Selbstberichten werden bei Verhaltensberichten die vollführte körperliche Aktivität, Bewegung oder Tätigkeit festgehalten. Unter Messwerte wird meist das Protokollieren von Messwerten wie Blutzucker, Blutdruck aber auch psychologische Testwerte wie Reaktionszeiten verstanden. Folgende Aufzählung

soll mögliche Inhalte eines computergestützten Protokolls darstellen, wie es Fahrenberg et al. in dem Buch „Alltagsnahe Psychologie“ beschreiben [14]:

- *Objektive Settingmerkmale*: Ort und Zeit, Personen und Objekte, Aufgaben und Reizbedingungen, ggf. besondere Ereignisse
- *Umgebungsmerkmale (ambiente Parameter)*: Schallpegel, Helligkeit, Temperatur, Wetterbedingungen
- *Selbstbericht*: Subjektiver Zustand (Stimmung, Emotion, Beanspruchung), Schmerzen und andere Symptome, Motive und Gedanken über Situation, Kommentare zu Aufgaben und speziellen Ereignissen
- *Verhaltensbericht*: Tätigkeiten, körperliche Aktivität, Bewegungen
- *Testwerte, Messwerte*: Verhaltensmessungen, psychologische Testwerte (z.B. Reaktionszeiten), Selbstmessungen mit Instrumenten (z.B. Blutdruck, Blutzucker)
- *Weitere Aspekte und Daten*: Kommentare zur Akzeptanz der Methodik, Artefakte, Alarm bis zum Beginn der Eingabe, Compliance und fehlende Daten

Ein Anwendungsgebiet für ein solches computergestütztes Protokoll bietet das ambulante Assessment. Dieses nutzt Sensorik zur automatischen Erfassung von Informationen, um so dem Probanden die Eingaben in das Protokoll zu erleichtern, aber auch zum eventbasierten Triggern von Abfragen von Messwerten, Selbstberichten, etc.

3 Ambulantes Assessment

In diesem Kapitel wird näher auf eine Erfassungsart für Feldexperimente eingegangen, das ambulante Assessment. Dieses nutzt Reportquellen wie Selbstberichte und automatische Berichte, um mithilfe computergestützter Technologie Daten im Alltag von Probanden zu erfassen. Fahrenberg et al. definieren ambulantes Assessment folgendermaßen: „Ambulantes Assessment ist die systematische Erfassung psychologischer und/oder physiologischer Daten unter alltäglichen Bedingungen (am Arbeitsplatz, in der Schule, in der Freizeit) für einen bestimmten Zweck, welcher in der strategischen Beziehung zwischen Daten, theoretischen Konstrukten und Kriterien expliziert wird“ [14].

In ambulanten Assessment-Studien geht es um verschiedene Fragestellungen und Anwendungen auf vielen Gebieten der Psychologie und der Psychophysiologie [14]. Um einen Eindruck zu bekommen, wie vielfältig die Einsatzbereiche des ambulanten Assessments sind, werden im Folgenden typische Fragestellungen aufgezählt, die mit dieser Methode untersucht werden können [15]:

- Verlaufsmuster im Erleben und Verhalten über kürzere und längere Zeiträume
- Vorhersage von Verhalten unter Alltagsbedingungen
- Intraindividuelle Variabilität
- Analyse der zeitliche Sequenz von Situationsbedingungen von Verhalten und Erlebten im Feld

- Akkuratheit der Wahrnehmung der Gefühle, Stimmungen und Emotionen des Partners
- Hypothesen über situationale und interpersonelle Effekte

Allen gemeinsam ist der Hintergrund, unter Alltagsbedingungen mit möglichst geringer *methodenbedingter Reaktivität* Daten zu erfassen um praktische Ziele wie „Screening, Klassifikation, Selektion, Diagnose, Evaluation von Maßnahmen, sowie individuelle durch Unterstützung von Selbstmonitoring, Selbstmedikation und Selbstmanagement, eine hohe empirische Gültigkeit und Nützlichkeit“ [14] zu erreichen. Die methodenbedingte Reaktivität beschreibt dabei die Verzerrung der Messung durch die Eigenheiten der Assessmentstrategie auf das Verhalten, das Befinden und den körperlichen Zustand der zu untersuchenden Person [14,25].

Um die Bedeutung des ambulanten Assessment zu verdeutlichen, wird im Folgenden auf dessen Vorzüge genauer eingegangen.

3.1 Anwendungsbegründung

Im Bereich der Psychologie „wird das Ambulante Assessment künftig die Methode der Wahl sein, statt der gegenwärtig weithin dominierenden Fragebögen“ [13]. Im Folgenden werden Gründe genannt, warum Fahrenberg et al. diese These stellen und allgemeine Vorzüge präsentiert, um die Relevanz dieser Methode zu verdeutlichen.

Ambulantes Assessment bietet viele Vorzüge gegenüber den konventionellen Papier-und-Bleistift-Methoden (z.B. Fragebogen), gerade was die flexible Gestaltung des Layouts der Fragen, maßgeschneiderte Antwortmöglichkeiten sowie verschiedene Antwortmodi (Listen, Skalen, Texteingabe, Verzweigung usw.) in sequentiellen oder hierarchischem Aufbau angeht [14]. Eine von diesen ist sicherlich die technische Zuverlässigkeit von computergestützten Datenerhebungen sowie die einfache und genauere zeitliche Protokollierung der Einträge - statt nur darauf hoffen zu können, dass sich die zu Untersuchenden auch an den vereinbarten Terminplan halten. Ob absichtlich oder unabsichtlich, es können Signale überhört werden und es kommt zu Datenausfällen (missing data). Aus den elektronischen Daten können allerdings Informationen wie die Latenzzeiten zwischen Signalen und Beginn der Eingabe oder Variationen in der Bearbeitungszeit extrahiert werden um so auf eventuelle situative Störungen oder Motivationsprobleme hinweisen zu können [14].

Weiterhin bieten die elektronischen Daten eine leichtere und bequemere Analyse und ermöglichen so auch eine schnellere und individuellere Anpassung an die Person [13]. Auch bezüglich der langfristigen Aufzeichnung, Datensicherung und Aufbewahrung sind elektronische Daten eine große Erleichterung, dem Probanden sowie dem Untersuchenden gegenüber. Nicht außer Acht gelassen werden sollte der dadurch mögliche direkte Datentransfer für die Weiterverarbeitung im ökonomischen Sinne sowie zur Vermeidung von Fehlerquellen durch manuelle Übertragung der Daten. Fehlerquellen bei manueller Eingabe könnten dabei

z.B. das falsche Kopieren des Datensatzes oder die falsche Eingabe bei der Übernahme der Daten von einem Protokoll in ein Anderes sein. Diese Übertragung geschieht heutzutage meist über kabellose Verfahren wie WLAN, Bluetooth usw.

Wie bereits angedeutet, bieten computergestützte Verfahren eine leichte Anpassung und innovative Möglichkeiten [14]:

- individuelle Systemanfragen nach speziellen Informationen (Expertise) und Zugriff auf graphische Darstellungen oder statistische Analysen der bisher eingegebenen Daten sowie Vorhersagen des weiteren Verlaufs
- systematisches Angebot von Informationen, z. B. möglichen Maßnahmen und Behandlungshinweisen
- interaktives Monitoring mit Rückfragen an den Untersuchten aufgrund zusätzlicher Informationen (z.B. aufgrund von online Analyse von Verhaltensmessungen oder physiologischen Daten)

Ein großer Vorteil der durch den Einsatz des Smartphones geboten wird, liefert die häufige Mitführung des Datenaufzeichnungsgerätes, ohne dass die zu Untersuchenden dies als Last empfinden. Ein Beispiel für eine solche Smartphone-Anwendung ist in Abbildung 1 zu sehen.

In diesem können Nutzer ihr aktuelles Stimmungsbild eingeben und direkt an den Forscher schicken. Hierbei ist zu erwähnen, das Experten im Umgang mit der Anwendung 'movisenseXS' sich die Applikation selbst erstellen können, woraus individuelle User Interfaces entstehen. Unbestritten liefert eine solche Anwendung, unabhängig von der genauen Zusammensetzung, einen enormen Vorteil gegenüber der nicht elektronischen Methode, allerdings birgt diese Art der Erhebung immer noch Nachteile derart, dass die Nutzer der Applikation teilweise gewisse Zeit benötigen, um den elektronischen Fragebogen auszufüllen und so eine Demotivierung stattfindet. Um dieses Problem zu lösen, können die bereits integrierten Sensoren verwendet werden, die den aktuellen Kontext des Nutzer ermitteln und so bereits einige Fragen an den Nutzer entfallen. Die Daten die dafür benötigt werden, können dabei auf unterschiedliche Weise (Assessmentstrategien) gewonnen werden. Im Folgenden wird auf diese genauer eingegangen.

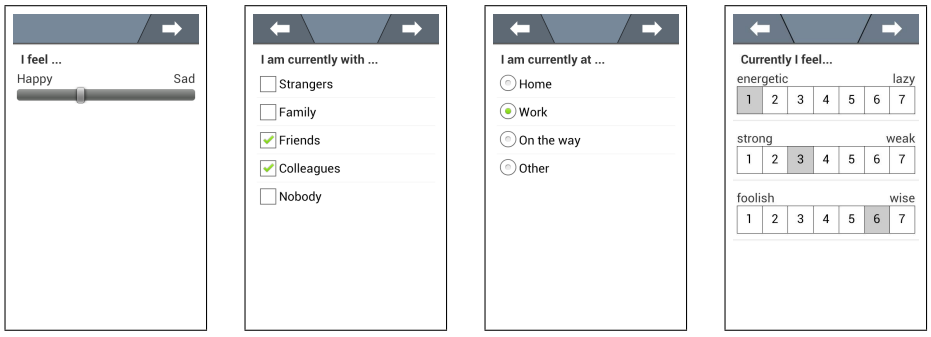


Abb. 1. Beispiel UI - 'movisenseXS' [2]

3.2 Assessmentstrategien

Für die alltagsnahe Forschung wurden mittlerweile spezielle Assessmentstrategien (Untersuchungspläne) (siehe Tabelle 2) zur Datenerfassung entwickelt. Diese unterscheiden sich hinsichtlich ihrer Blickwinkel zwischen Selbstberichten, Verhaltensmessungen, physiologischen Messungen und deren Besonderheiten. Selbstberichte können dabei abhängig von der Fragestellung unter anderem zu zeitlich fixen Terminen, zu zufälligen Terminen mit bestimmter Anzahl pro Tag sowie abhängig von der körperlichen Aktivität oder des körperlichen Zustandes (wie Herzrate, Blutdruck, etc.) erfasst werden. Die Erfassung der Daten bezieht sich dabei auf die Aufnahme sowie die Eintragung von Daten in den elektronischen Fragebogen durch den zu Untersuchenden. Hierfür wird die (Bewegungs-)Aktivität entweder kontinuierlich aufgenommen oder in festgelegten Intervallen gemessen. „Zunehmendes Interesse finden das computergestützte *Interaktive Monitoring* und *Feld-Experimente*“ [13].

Tabelle 2. Assessmentstrategien [13]

Strategie	Erläuterung
Kontinuierliches Monitoring	Fortlaufende Aufzeichnung als „Datenstrom“, ohne sonstige Einflussnahme. Z.B. Registrierung der Körperposition und Bewegungsmuster
Zeit- oder ereignisabhängiges Monitoring	Erfassung von Daten zu verschiedenen Tageszeiten oder bei relevanten Situationsbedingungen. Z.B. automatische GPS Lokation in Abständen von 20 Minuten
Kontrolliertes Monitoring	Auswahl natürlicher Settings um inter-und intraindividuelle Referenzwerte zu gewinnen. Z.B. Arbeitsplatz oder Familie
Feld-Diagnostik	Nach einem festgelegten Zeitplan, standardisierte Tests unter Alltagsbedingungen durchführen
Feld-Experiment	Prüfung der Hypothese im Alltag aufgrund von Zuweisungen zu Bedingungen (wie Arbeitsanforderungen, Interventionen oder standardisierter Settings) durch computergestützte Datenerhebung
Interaktives Monitoring	Protokollierung des Settings und des Befindens, bei dem Auftreten einer Veränderung eines physiologischen Parameters. Zufällig ausgelöste Protokolle können dabei als Referenzwert dienen
Symptom-Monitoring und Selbstmanagement	Überwachung von chronischen Krankheiten oder Verhaltensproblemen, Training neuropsychologischer Funktionen im Alltag

3.3 Nachbefragung

Unabhängig von der Assessmentstrategie beziehen sich die erfassten Daten meist nur auf den zu untersuchenden Kontext. Um Allerdings spezielle Probleme mit den angewandten Strategien zu erfassen, ist es meist unumgänglich ein Postmonitoring Interview durchzuführen oder zumindest dem Proband ein Fragebogen vorzulegen oder mitzugeben. Der Untersuchende kann während des Postmonitoring Interviews die aktuellen Eindrücke des Probanden festhalten. Zur Darstellung der Güte des Rekords ist laut Fahrenberg eine siebenstufige Skala empfehlenswert, wie in Abbildung 2 veranschaulicht (wobei Skalenstufe 1 = gar nicht; Skalenstufe 7 = völlig).

Fragebögen, die an die Probanden des Experiments verteilt werden, um die Methodik des Monitorings zu evaluieren, können ebenfalls mit einer siebenstufigen Skala beantwortet werden. Folgende Fragen könnten für Postmonitoring Fragebogen verwendet werden [14]:

- „Kam es durch die Untersuchungsmethodik zu Änderungen Ihres Verhaltens?“
- „Haben Sie vermehrt auf Ihren psychischen Zustand geachtet?“
- „Reagiert Ihre Umwelt negativ auf ihre Ausrüstung?“
- „Waren Ihnen die Reaktionen Ihrer Umwelt unangenehm?“
- „Wie störend war im allgemeinen die Programmverarbeitung?“
- „Wurde die Bearbeitung zu häufig verlangt?“
- „Wie lästig waren Ihnen die Selbsteinstufungen?“
- „Waren es zu viele Einstufungen pro Eingabe?“
- „Wie lästig war Ihnen der Konzentrationstest?“
- „Wie lästig war Ihnen der Gedächtnistest?“

Zweck der Nachbefragungen ist meist die Ergründung von Aspekten wie die *Akzeptanz* der Methodik, die *Compliance* oder die *methodenbedingten Reaktivität* des Probanden.

Bewertungsskala	1	2	3	4	5	6	7
Akzeptanz der Methodik							
Compliance							
Methodenbedingte Reaktivität							

Abb. 2. Nachbefragung - Postmonitoring-Interview [14]: Skalenstufe 1 = gar nicht; Skalenstufe 7 = völlig

3.4 Akzeptanz und Compliance

Die Notwendigkeit im Laufe einer Studie einen Probanden meist mehrfach einen Fragebogen ausfüllen lassen zu müssen, kann die Akzeptanz und die Compliance bedrohen, welche allerdings für den Erfolg der Testreihe maßgeblich sind. Die *Akzeptanz* bezieht sich dabei auf die Einstellung der Testperson bei dem ambulanten Versuch mitzuarbeiten, was die Qualität der Datenerhebung beträchtlich beeinflusst. Unter der *Compliance*, wird die Bereitschaft sich an die Protokollierung von Selbstberichten wie verabredet zu halten verstanden [14]. Diese kann unter anderem durch eine erhöhte Motivation, eine Honorierung für die Teilnahme oder einem eigenen Nutzen - wie die aus den Experimenten resultierende Erkenntnisse über die eigene Erkrankung - gesteigert werden [25].

Für die Akzeptanz ist es besonders wichtig, dass die Messvorrichtung als nicht störend empfunden wird. Werden zum Beispiel schlecht klebende Elektroden oder bewegungseinschränkende Verfahren für den Versuch gewählt, reduziert dies entscheidend die Akzeptanz. Wie gut diese für den jeweiligen Versuch gewesen ist, lässt sich in der Nachbefragung (Kapitel 3.3) erfassen. Laut Reuschenbach ist die Akzeptanz der computerisierten Eingabeform bezüglich des ambulanten Assessment i.d.R. höher als bei Papier-Bleistift Verfahren [25].

Allerdings kommt es laut klinischen Stichproben häufiger zu unvollständigen Angaben und möglicherweise auch technischen Ausfällen. Der Einsatz von Technik, besonders bei technikaffinen Personen, sichert wiederum die Compliance. Diese lässt sich durch die aus den Daten zu erhaltende Antwortrate („Response rate“) bestimmen. Diese spiegelt wieder, auf wie viele Anfragen eine Eingabe in das Eingabegerät durch die Testperson erfolgten [25]. Allerdings wird bei der Response rate nicht die Vollständigkeit der Eingabedaten betrachtet. Ein Ausbleiben oder nur teilweise Eingabe der Dateneingabe gefährdet die Repräsentativität der Daten und damit letztendlich die Reliablitität und Validität der gesamten Messung. Selbst bei schwer erkrankten Patientengruppen, beispielsweise Patienten mit Stammzellentransplantation, ist die Messung mittels computergestützten Verfahren durch eine hohe Akzeptanz und hohe Response Rate gekennzeichnet [25].

Die Akzeptanz und Compliance werden weiterhin durch den Aufwand der Dateneingabe beeinflusst. Das Problem liegt häufig in der Summe der vielen Erhebungszeitpunkte und/oder der vielen benötigten Dateneingaben (Fragen). Wird die Person mit einer Vielzahl von Fragen konfrontiert, sinkt schnell die Teilnehmermotivation und auch die Häufigkeit der Missing Data steigt an. Die Missing-Data-Häufigkeit zeigt laut Reuschenbach meist einen U-förmigen Verlauf: Fehleingaben am Anfang können durch Unsicherheit bei der Eingabe begründet sein, was durch ein bessere Unterweisung in der Handhabung der Applikation des Smartphones größtenteils behebbar sind. Spätere Missings sind durch eine abnehmende Motivation der Testperson zu erklären, was durch eine kontextbasierte Reduzierung der Anzahl sowie der Frequenz der Fragen reduziert werden kann. Neben dieser so genannten *Signal-Compliance* existiert auch eine *Settings-Compliance*. Bei dieser geht es um die Bereitschaft in bestimmten Situationen Angaben zu machen. Eine Eingabeaufforderung beim Kinobesuch oder

während des Sports bleibt eher unbeantwortet als in der häuslichen Umgebung. Auch diese Art der Compliance kann in bestimmtem Maßstab durch Kontexterkennung behoben werden. Beispielweise könnte die Eingabeaufforderung auf einen späteren Zeitpunkt verschoben werden, falls der Nutzer sich in einem ermittelten nicht passenden Kontext befindet. Erfolgen häufige Erinnerungen an die Dateneingabe und besteht keine Möglichkeit diese abzubrechen, kann es zu Eingaben mit falschen Daten kommen, so genannten *Fake-Compliances*. Daher ist es in den meisten Fällen ratsam, in die Applikation eine gut sichtbare Abbruchmöglichkeit einzubetten. Folgende Maßnahmen führen im Allgemeinen zu einer Erhöhung der Compliance [25]:

- Vorheriges Training mit der Erhebungstechnologie
- Möglichkeit anbieten, Beantwortung der Items zeitlich zu verschieben oder einen begrenzten Umfang der Angaben zu verweigern („Kontrolle über die Kontrolle“). Allerdings sollten im Postmonitoring Interview (siehe Kapitel 3.3: „Nachbefragung“) die Gründe dafür dann erfragt werden
- Signalgeber (automatisierte Erinnerungen) an nicht erfolgte Einträge
- Bedienfreundlichkeit der Applikation beachten
- Adaptive Abfolge und Auswahl der Fragen durch den Probanden
- An soziale Verantwortlichkeit appellieren, indem Wichtigkeit einer exakten Dateneingabe betont wird und dass die Compliance Teil der Erhebung ist
- Direkte Rückmeldung der Ergebnisse und der Auswertung sowie Feedback über Eingabequote

Eine Methode, die Akzeptanz sowie die Compliance eines Nutzers während eines Experimentes aufrecht zu halten, könnte die Reduzierung der Eingabedaten für diesen sein. Dies kann erreicht werden, indem bestimmte Eingabeparameter bereits durch die Erfassung des Kontextes angepasst werden. Des Weiteren ist der Einsatz einer kontextbasierte Abfragen möglich.

4 Kontexterfassung durch Smartphones

Die Erfassung des Kontextes, durch die Sensoren des Smartphones, kann für das ambulante Assessment unter anderem eine wesentliche Erleichterung der Dateneingabe für den Probanden bieten, womit eine höhere Akzeptanz erreicht werden kann. Es ist durch heutige Techniken und Methoden möglich, Kontextinformationen über den Probanden zu erfassen und so diesem eine reduzierte Anzahl oder zumindest zielgerichtetere Fragen zu präsentieren. Zudem ermöglicht es eine zuverlässigere Information über den Kontext. Beispielhaft kann hier genannt werden, dass der Proband bei jeder Befragung, die er an seinem Smartphone ausführt, seinen Standort mit angeben soll. Leichter und meist auch genauer könnte dies aber über den Sensor GPS erfasst werden. Neben diesen einfach aus den Sensoren ersichtlichen Kontexten können durch das Zusammenspiel von Sensoren auch schwerere Kontexte wie die Emotionserkennung extrahiert werden.

4.1 Relevanz des Smartphones im ambulanten Assessment

Smartphones erhalten einen immer größeren Stellenwert in unserer Gesellschaft, wie Umfragen über die Verbreitung von Smartphones zeigen [1]. Schränkt man zusätzlich die Altersgruppe von Personen auf über 13 Jahre ein, erhöht sich die Anzahl der Smartphonebesitzer auf über die Hälfte aller deutschen Bundesbürger [5]. Diese Einschränkung ergibt im Bereich des ambulanten Assessments durchaus Sinn, da diese sich meist mit Jugendlichen und Erwachsenen beschäftigt.

Durch die große Verbreitung lassen sich wesentliche Vereinfachungen in der Gestaltung von Studien treffen. Beispielweise muss nur noch den wenigsten Teilnehmern ein Aufzeichnungsgerät gestellt werden, was eine erhebliche Kostenreduktion sowie eine Senkung des organisatorischem Aufwands bedeutet. Weiterhin bieten Smartphones den Vorteil eines bereits in den Alltag eingebetteten Systems, sodass Nutzer das Gerät nicht als zusätzliche Last empfinden. Ein technischer Vorteil, der sich im Laufe der Entwicklung des Smartphones in den letzten Jahren offenbarte, sind die steigende Anzahl an integrierten Sensoren. Standardmäßig gehören dazu die Sensortypen: Beschleunigungssensor, der Lichtsensor, der Berührungssensor (Touch), der Lagesensor, das GPS, der Rotationssensor sowie der Kompass. Ein weiterer technischer Aspekt, der für den Einsatz von Smartphones spricht, sind die angebotenen drahtlosen Schnittstellen wie WLAN oder Bluetooth (*Telemetrie*). Diese ermöglichen es, die erhobenen Daten sofort zum Forschenden zu schicken, ohne dass dieser sich direkt mit den Probanden treffen muss. Gerade bei weltweit angelegten Studien sorgt dies für eine große Reduzierung des Aufwands, sowohl im Kosten- wie auch im organisatorischen Bereich.

Der Einsatz des Smartphones und dessen Sensorik ermöglicht viele Arten den Kontext des Nutzers zu erfassen bzw. zu analysieren. Eine Variante der Kontexterfassung ist das Affective Computing.

4.2 Affective Computing

Sensoren können dafür genutzt werden den Kontext des Probanden zu erkennen und diesem bspw. Hilfe bei der Bearbeitung seines computergestützten Fragebogens anzubieten. Darunter fällt auch die Erkennung von Kontexten wie emotionsgebundene Situationen. Der Bereich, der sich mit diesem Gebiet auseinandersetzt, wird als *Affective Computing* bezeichnet. Das Ziel des Affective Computing ist es, einem Computer menschenähnliche Fähigkeiten bei der Beobachtung und der Interpretation von Emotion und Stimmung beizubringen [27]. Damit ist es eine wichtiger Bereich der Mensch- Maschine Kommunikation, der zur Erkennung Daten wie Gesichtsausdrücke [7, 9], Stimmanalyse [9, 18], Eingabegesten [9, 11] oder die Interaktion in soziale Medien [8, 22] analysiert. Im Kontext dieser Ausarbeitung beschränkten wir uns dabei allerdings auf den Bereich der Aufnahme von Daten über Sensoren des Smartphones. Mit der Zeit haben sich aus den verschiedenen Erkennungstypen drei Hauptansätze herauskristallisiert: *Emotion*

Sensing(Emotionserkennung), *Mood Sensing*(Stimmungserkennung) und *Sentiment Mining*. Sowohl Mood Sensing als auch Emotion Sensing betreffen hierbei die jeweilige Erkennung an einer einzelnen Person. Im Unterschied dazu wird bei Sentiment Mining die Stimmung einer Personengruppe analysiert.

Nachfolgend werden Beispiele zur Gewinnung und Nutzung des Kontextes, wie bspw. das Affective Computing, aufgezeigt.

4.3 Anwendung

Eine Variante, den Kontext eines Probanden zu nutzen, ist die Erkennung der Umgebung, in der sich dieser befindet. Mithilfe einer solchen Information könnte zum Beispiel automatisch das Ausfüllen eines elektronischen Fragebogens verschoben werden, falls der Proband gerade beschäftigt ist.

Fan et al. beschreiben in ihrem Paper 2014 [16] eine audiobasierte Kontexterkennung von öffentlichen Toilettenräumen via Smartphone. Dafür werden nur das Mikrophon sowie der Lautsprecher des Smartphones verwendet (siehe Abbildung 3). Die Erkennung geschieht, indem Sinuswellen mit vordefinierten Frequenzen aus dem Lautsprecher des Smartphones ausgestrahlt und die Impulsantworten aus dem Raum wieder durch das Mikrophon aufgenommen werden. Mit einem Datenset von 103 öffentlichen Toiletten von 49 verschiedenen Gebäuden gelang es ihnen mit der Support Vector Machine eine Klassifikationsrate von >96% zu erzielen. Anwendung könnte eine solche Klassifikation in dem Aufschieben der Abfrage eines Fragebogens finden, falls sich der Proband gerade in einem Toilettenraum befindet.

Abb. 3. Public Restroom Detection via Smartphone - Galaxy Nexus [16]

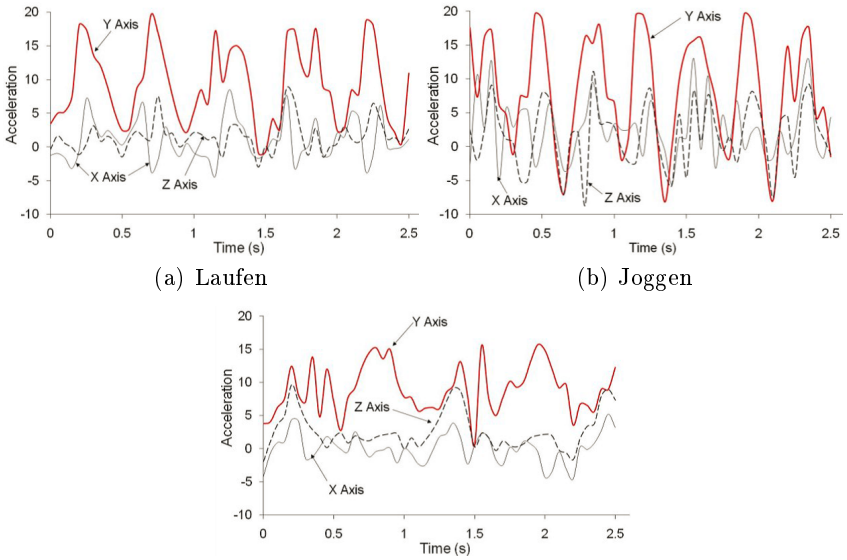
Wie bereits angesprochen werden elektronische Fragebögen häufiger zu bestimmten Zeitpunkten abgefragt. Um an diese Termine zu erinnern, ertönt meist ein Ton - ähnlich wie eines Weckers. Es kann allerdings vorkommen, dass der Ton zu leise gestellt ist, um ihn bspw. in der Hosentasche oder der Handtasche wahrzunehmen. Um diesem Problem entgegen zu wirken, könnte die 2010 von Mulizzo et al. in ihrer Arbeit [20] vorgestellte Methode verwendet werden. In dieser detektieren sie, ob sich das Smartphone in der Hosentasche, einer Tragetasche oder in der Hand befindet. Diese Klassifikation kann verwendet werden, um die Umgebung des Probanden zu erfassen und automatisch die Tonlautstärke anzupassen. In ihrer Ausarbeitung versuchen sie die beste Kombination aus

den Smartphone-Sensoren Beschleunigungssensor, Mikrophon, Gyroskop, Kamera, Lichtsensor und Kompass für die Klassifikation zu finden. Als beste Kombination erwiesen sich dabei der Beschleunigungssensor und der Lichtsensor in Kombination mit der Support Vector Machine als Klassifikator mit einer Genauigkeit von 80%. Weiterhin könnte diese Klassifikation verwendet werden, um bspw. den Zeitpunkt des Fragebogenbeginns nach vorn zu verlagern, falls der Proband gerade sein Smartphone in der Hand hält und der Zeitpunkt innerhalb eines definierten Zeitrahmens liegt.

Eine ähnliche Anwendung könnte die Masterarbeit von Abrehaim im Jahre 2014 liefern [4]. Dieser fokussierte sich auf die Klassifikation von verschiedenen Aktionen mithilfe des Smartphone-Mikrophons. Mithilfe eines künstlichen Neuronalen Netzes unterschied er dabei die Geräusche einer Kaffeemaschine, des Türöffnens oder -schließens eines Fahrstuhls, der Stille, des Laufens sowie das Geräusch des fließenden Wassers während des Händewaschens mit einer Genauigkeit von 98,8%. Angenommen, ein Proband befindet sich, während er sich gerade einen Kaffee zubereitet, in einer ruhigeren emotionalen Phase z.B. in einer Pause, könnte diese genutzt werden, um ihm das Ausfüllen des Fragebogens in dieser Zeit nahezulegen. Doch nicht nur für den Beginn eines computergestützten Protokolls kann die Arbeit von Abrehaim genutzt werden. Nützlich könnte auch eine Erkennung des Kontextes über den Tagesverlauf sein. Beispielhaft könnte in einem Fragebogen die Häufigkeit der Nutzung des Fahrstuhls erfragt werden, um dem Nutzer nach Studienende unter anderem auf die versäumte Bewegung aufmerksam zu machen. Die Menge an Fahrstuhlfahrten kann dabei mithilfe dieser Arbeit erhoben und so dem Benutzer vorweg genommen werden.

Des Weiteren wird häufig in Fragebögen die körperliche Aktivität bzw. Inaktivität des Nutzers erfragt. Problematisch sind dabei meist die ungenauen Zeitangaben des Nutzers aufgrund dessen subjektiver Wahrnehmung. Weiss und Lockhart stellen in ihrer Arbeit eine Personen-abhängige Klassifikation von Bewegungsvarianten über das Smartphone dar [28]. Sie unterscheiden dabei die Klassen 'Laufen', 'Joggen', 'Treppen steigen', 'Sitzen', 'Stehen' und 'Liegen' nur anhand des Beschleunigungssensors mit einer Genauigkeit von 98,7%. Dabei wird angenommen, dass der Nutzer sein Smartphone in der Hosentasche mit sich führt. Die Diskrepanz der Beschleunigungswerte von verschiedenen Klassen wird beispielhaft in Abbildung 4 veranschaulicht. Mit der Ausführung dieser Arbeit könnte dem Nutzer somit die Fragen über die Menge an Bewegung bzw. sitzender Tätigkeit pro Tag in einem Fragebogen abgenommen werden.

Häufig werden in Fragebögen im Bereich des ambulanten Assessments auch die Emotion bzw. die Stimmung des Nutzers erfragt (siehe Kapitel 4.2: „Affective Computing"). Darunter fällt unter anderem der Bereich Emotion Sensing, der die Aufnahme, Verarbeitung und die abschließenden Emotionsklassifikation des Nutzers behandelt. Eine Emotion beschreibt dabei einen kurzlebigen emotionalen Zustand eines Menschen. Diese wird laut Ekman [12] in die sieben Basisemotionen *Trauer*, *Zorn*, *Überraschung*, *Angst*, *Ekel*, *Verachtung* und *Freude* unterteilt. Ein smartphonebasiertes Beispiel liefert hierfür die Arbeit von Rachuri et al. [24]. In dieser wird die smartphonebasierte Plattform 'EmotionSense'



(a) Laufen (b) Joggen

(c) Treppen steigen

Abb. 4. Beispielhafte Bewegungsanalysen durch den Beschleunigungssensor des Smartphones in der Hosentasche [28]

vorgestellt. Mit Hilfe dieser soll es möglich sein, den emotionalen Zustand des Nutzers anhand des Smartphones zu erkennen. Die Klassifikation einer Emotion basiert dabei unter anderem auf der Aufnahme von Audio-Samples mit einem variablen Aufnahmeintervall. Des Weiteren kommen der Beschleunigungssensor zur Unterscheidung von Aktivität oder Nicht-Aktivität, das GPS zur Lokationsbestimmung oder die Textnachrichten zum Einsatz. Ein Beispiel für diese Plattform ist die gleichnamige Android App, siehe Abbildung 5.

In dieser kann der Nutzer beispielsweise nachvollziehen, wie er sich fühlt wenn er sozial aktiv ist. Gemessen werden hierbei unter anderem die Anzahl an Anrufen und Textnachrichten und ob dieser physisch aktiv war, gemessen anhand des Beschleunigungssensors. Dabei liest die Applikation nicht die Nachrichten selbst, sondern zählt nur dessen Anzahl. Ähnlich wird für die Klassifikation das Mikrofon nur für die Umgebungslautstärke und nicht das Gesprochene selbst verwendet. Verwendung finden insgesamt die Sensoren Beschleunigungssensor, Mikrophon und GPS. 'EmotionSense' steht hier beispielhaft für die Erkennung von Emotionen über Smartphones. Diese Erkennung kann genutzt werden, um bei elektronischen Fragebögen bestimmte Fragen automatisch zu beantworten und damit die Eingabemenge an Daten des Nutzers zu reduzieren.

Ein weiterer Bereich des Affective Computing ist die Stimmungserkennung („Mood Sensing"). Hierbei werden die Stimmungsarten unter anderem in *Angespannt*, *Stress*, *Ärger*, *Langeweile*, *Aufgeregt*, *Fröhlich*, *Entspannt* und *Ruhig* unterschieden [19]. Ähnlich der Emotionserkennung kann auch die automatische

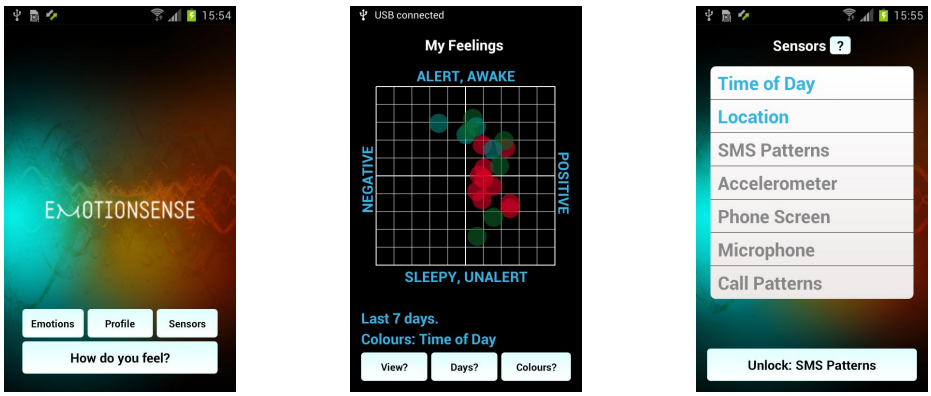


Abb. 5. 'EmotionSense' [3]

Stimmungserkennung dafür genutzt werden, um dem Probanden eine reduzierte Eingabe zu ermöglichen. „AMMON“ [10] ist eine Sprachanalyse Bibliothek zum direkten Analysieren von Stress und mentaler Gesundheit mit dem Smartphone. Die Bibliothek beschränkt sich dabei allerdings auf die Klassifikation von Stresserhöhung und Stressreduktion. Chang et al. erreichen dabei eine Akkuratheit von 93,6%. Weitere Features zur Detektion der Stimmung präsentieren LiKamWa et al. mit ihrem System 'MoodScope' [19]. In diesem zeigten sie die Erkennung anhand von SMS, E-Mail, Telefonanrufen, Anwendungsnutzung, Web-Browsing und der Lokation. Hierbei erreichen Sie mit der personenunabhängigen Klassifikation eine Akkuratheit von nur 66%, welche sich allerdings durch eine zweimonatige Personalisierung auf bis zu 93% verbessern lies. Im Unterschied zu den bisher erläuterten Anwendungen basiert 'MoodScope' auf dem iOS System. Eine weitere Arbeit, die auf iOS basiert, aber auf die Erkennung von Stress abzielt, ist die Arbeit von Muaremi et al. [21]. Hierbei liegt der Fokus auf dem Erkennen von Stress während eines Arbeitstages sowie im Schlaf. Zur besseren Klassifikation verwenden sie zusätzlich zur Smartphonesensorik den Wahoo Brustgürtel, um die Herzratenvariabilität (HRV) als zusätzliches Feature zu gewinnen. Unter den Smartphonesensoren finden die Kategorien Audio (Mikrofon), physikalische Aktivität (Beschleunigungssensor, GPS) sowie die soziale Interaktion (Telefonanrufe, Adressbuch und Kalender) ihren Einsatz. Wie in Abbildung 6 zu sehen ist, werden Fragebögen zum Sammeln der Daten zur späteren Klassifikation an bestimmten Tageszeitpunkten erfragt. Wie bereits zu Beginn dieses Kapitels erläutert, können zur automatischen Verschiebung dieser Zeitpunkte verschiedene Verfahren verwendet werden, falls diese für den Probanden gerade ungünstig sind. Am Ende des Tages wird die Datenaufnahme auf den Wahoo Brustgürtel verschoben, um die HRV über die Nacht zu prüfen. Die Nachtüberwachung hat den physiologischen Sinn der Stressbewältigung und der damit korrelierenden HRV im Schlaf [21]. Schlussendlich gelang Ihnen eine personenabhängige Genauigkeit von 61%.

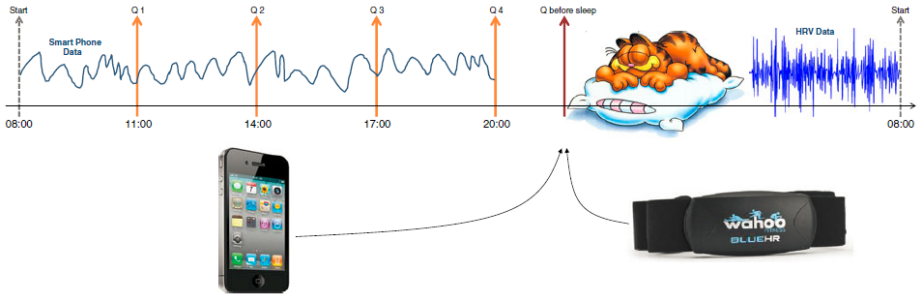


Abb. 6. Datensammelzyklus pro Tag, inkl. Aufnahmezeitpunkte(Q) für den Fragebogen [21]

Abschließend wird auf eine speziellere Sichtweise eingegangen, um an die Information eines Stimmungszustandes zu gelangen. Beispielsweise wäre es möglich von dem Fahrstil des Autofahrers auf die aktuelle Stimmung zu schließen. Johnson et al. haben in ihrem Paper [17] gezeigt, dass es möglich ist, den Fahrstil durch einen binären Klassifikator ('normal' vs. 'aggressiv') zu unterscheiden. Hierfür verwenden sie die Sensoren Beschleunigungssensor, Gyroskop, Magnetometer, GPS sowie die Kamera. Mithilfe einer festen Halterung an der Frontscheibe des Fahrzeugs erreichten sie mit ihrer Arbeit eine Detektion von 97% von aggressiven Fahrevents.

5 Fazit

In dieser Arbeit wurde der Einsatz des ambulanten Assessment mit der Datenaufnahme durch Smartphones thematisiert, wobei der Fokus auf der Reduzierung der nutzerseitigen Eingaben in einen Fragebogen lag. Hierbei wurden verschiedene Assessmentstrategien betrachtet sowie auf die Vorteile des Einsatzes von Smartphones zur Datenaufnahme eingegangen. Die Vorzüge sind unter anderem in den bereits integrierten Sensoren und Schnittstellen zu sehen, da diese neue Kontexte sowie fehlerfreie und angenehmere Übertragungswege der Daten ermöglichen. Beispielhaft wurde gezeigt, wie verschiedenste Kontexterfassungen eine Erleichterung für den Probanden darstellen könnten, wobei auch Kontexte wie Emotion und Stimmung des zu Untersuchenden mit einbezogen wurden. Durch die Umsetzung kann die Akzeptanz und die Compliance des Nutzers während eines Experimentes erhöht bzw. aufrecht erhalten werden, was wiederum eine bessere externe Validität des Experimentes zur Folge hat. Des Weiteren hat diese Arbeit gezeigt, dass speziell für Feldexperimente der Einsatz von Smartphones eine große Erleichterung für den Probanden wie auch für den Untersuchenden bietet. Kritisch zu betrachten ist dabei allerdings der durch das Smartphone entstehende erschwerte Schutz der Privatsphäre auf dem Übertragungsweg sowie auch bei etwaiger Entwendung des Gerätes. Dies gilt insbesondere bei kontinuierlichen Aufnahmen für die Erkennung des Kontextes. Nicht genauer thematisiert

wurde hier die Auswertung von sozialen Netzwerken, welche als weitere Informationsquelle zur Kontexterfassung dienen könnte. Für weiterführende Arbeiten wäre denkbar, die zunehmend verbreiteten Smartwatches oder ähnliche Wearables als Sensorerweiterungen für Smartphones mit einzubeziehen. Diese würden neue Dimensionen zur Kontexterfassung erschließen, was wiederum zu automatisierteren Selbstberichten führen könnte.

Literatur

1. (05 2015), http://de.statista.com/statistik/daten/studie/198959/umfrage/anzahl-dersmartphonenutzer-in-deutschland-seit-2010/
2. (05 2015), http://www.movisens.com/de/produkte/movisensxs/
3. (06 2015), http://emotionsense.org/
4. Abreha, G.T.: An environmental audio-based context recognition system using smartphones (2014)
5. Arziman, z.: Design von Studien mithilfe der Experience Sampling Method p. 24 (2015)
6. Augsburg, A.: Kontexterfassung mobiler Endgeraete (2012)
7. Bartlett, M.S., Littlewort, G., Fasel, I., Movellan, J.R.: Real Time Face Detection and Facial Expression Recognition: Development and Applications to Human Computer Interaction. In: Computer Vision and Pattern Recognition Workshop, 2003. CVPRW'03. Conference on. vol. 5, pp. 53–53. IEEE (2003)
8. Bollen, J., Pepe, A., Mao, H.: Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. arXiv preprint arXiv:0911.1583 (2009)
9. Castellano, G., Kessous, L., Caridakis, G.: Emotion Recognition through Multiple Modalities: Face, Body Gesture, Speech. In: Peter, C., Beale, R. (eds.) Affect and Emotion in Human-Computer Interaction, Lecture Notes in Computer Science, vol. 4868, pp. 92–103. Springer Berlin Heidelberg (2008), http://dx.doi.org/10.1007/978-3-540-85099-1_8
10. Chang, K.h., Fisher, D., Canny, J., Hartmann, B.: How's my mood and stress? An efficient speech analysis library for unobtrusive monitoring on mobile phones. In: Proceedings of the 6th International Conference on Body Area Networks. pp. 71–77. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering) (2011)
11. Coutrix, C., Mandran, N.: Identifying emotions expressed by mobile users through 2D surface and 3D motion gestures. In: Proceedings of the 2012 ACM Conference on Ubiquitous Computing. pp. 311–320. ACM (2012)
12. Ekman, P.: Gefuehle lesen: wie Sie Emotionen erkennen und richtig interpretieren. Spektrum Akad. Verl (2007)
13. Fahrenberg, J.: Handbuch Statistik, Methoden und Evaluation, chap. Ambulantes Assessment, pp. 201 – 212. Hogrefe Verlag (2010), http://www.jochen-fahrenberg.de/uploads/media/Ambulantes_Assessment_2010_01.pdf
14. Fahrenberg, J., Leonhart, R., Foerster, F.: Alltagsnahe Psychologie: Datenerhebung im Feld mit hand-held PC und physiologischem Mess-System. Huber (2002)
15. Fahrenberg, J., Myrtek, M., Pawlik, K., Perrez, M.: Ambulantes Assessment-Verhalten im Alltagskontext erfassen. Psychologische Rundschau 58(1), 12–23 (2007)
16. Fan, M., Adams, A.T., Truong, K.N.: Public restroom detection on mobile phone via active probing. In: Proceedings of the 2014 ACM International Symposium on Wearable Computers. pp. 27–34. ACM (2014)

17. Johnson, D.A., Trivedi, M.M.: Driving style recognition using a smartphone as a sensor platform. In: Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on. pp. 1609–1615. IEEE (2011)
18. Lee, C.M., Narayanan, S.S.: Toward detecting emotions in spoken dialogs. Speech and Audio Processing, IEEE Transactions on 13(2), 293–303 (2005)
19. LiKamWa, R., Liu, Y., Lane, N.D., Zhong, L.: Moodscope: building a mood sensor from smartphone usage patterns. In: Proceeding of the 11th annual international conference on Mobile systems, applications, and services. pp. 389–402. ACM (2013), http://www.ruf.rice.edu/~mobile/publications/likamwa2013mobisys2.pdf
20. Miluzzo, E., Papandrea, M., Lane, N.D., Lu, H., Campbell, A.T.: Pocket, bag, hand, etc.-automatically detecting phone context through discovery. Proc. PhoneSense 2010 pp. 21–25 (2010)
21. Muaremi, A., Arnrich, B., Tröster, G.: Towards measuring stress with smartphones and wearable devices during workday and sleep. BioNanoScience 3(2), 172–183 (2013)
22. Nguyen, T., Phung, D., Adams, B., Venkatesh, S.: Mood sensing from social media texts and its applications. Knowledge and information systems 39(3), 667–702 (2014)
23. Party, J., Perrez, M.: Entstehungs-, Erklaerungs-und Anwendungszusammenhang technologischer Regeln. Feldforschung. Methoden und Probleme sozialwissenschaftlicher Forschung unter natürlichen Bedingungen pp. 17–42 (1982)
24. Rachuri, K.K., Musolesi, M., Mascolo, C., Rentfrow, P.J., Longworth, C., Aucinas, A.: EmotionSense: a mobile phones based adaptive platform for experimental social psychology research. In: Proceedings of the 12th ACM international conference on Ubiquitous computing. pp. 281–290. ACM (2010), http://emotionsense.org/
25. Reuschenbach, B., Funke, J.: Ambulantes Assessment (2011)
26. Schrader, J., Schoel, G., Scheler, F.: Bedeutung der 24-Stunden-Blutdruckmessung in der Diagnostik und Therapie der arteriellen Hypertonie. Klinische Wochenschrift 68(22), 1119–1126 (1990), http://link.springer.com/article/10.1007/BF01798062
27. Tao, J., Tan, T.: Affective Computing: A Review. In: Tao, J., Tan, T., Picard, R. (eds.) Affective Computing and Intelligent Interaction, Lecture Notes in Computer Science, vol. 3784, pp. 981–995. Springer Berlin Heidelberg (2005), http://dx.doi.org/10.1007/11573548_125
28. Weiss, G.M., Lockhart, J.W.: The impact of personalization on smartphone-based activity recognition. In: AAAI Workshop on Activity Context Representation: Techniques and Languages (2012)

Visualisierung von Sensordaten in Augmented Reality

Maximilian Eckert*

Betreuer: Matthias Berning†

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

Zusammenfassung. Mit steigender Anzahl von Sensorik in allen Bereichen des Lebens steigt auch die Menge der anfallenden Daten rasant an. Mittels Augmented Reality kann zwar die Datenmenge nicht reduziert werden, aber die Visualisierung dieser erheblich verbessert werden, sodass die Daten um ein vielfaches effizienter analysiert werden können. Denn die Sensordaten können ad hoc, im Kontext der Datenerhebung ausgewertet werden, was einen erheblichen Mehrwert mit sich bringt. Der Fokus dieser Arbeit liegt darauf, einen Überblick über die verschiedenen Arten und Techniken der Visualisierung in Augmented Reality zu geben und dabei verschiedene Gestaltungsmöglichkeiten aufzuzeigen. Grundsätzlich wird zwischen sensorbasierter Visualisierung und kontinuierlicher graphischer Überlagerung unterschieden. Bei sensorbasierten Systemen werden einzelne Grafiken zu den korrespondierenden Sensoren angezeigt, während bei der graphischen Überlagerung die Messwerte interpoliert und kontinuierlich über den gesamten Messbereich visualisiert werden. Zu Beginn werden zunächst Basistechnologien für Augmented Reality vorgestellt, grundlegende Techniken der Visualisierung betrachtet und dann auf die Gestaltung der Augmented Reality übertragen. Dabei spielt vor allem die menschliche Wahrnehmung und Verarbeitung von Kontrasten, Farben und Formen eine wichtige Rolle. Darüber hinaus werden besondere Herausforderungen, wie Hintergrundinterferenzen, Tiefenwahrnehmung und Sichtbarkeitsprobleme berücksichtigt, die hierbei auftreten. Abschließend zeigt sich, dass sich Augmented Reality ideal eignet, um Sensordaten on-site im Messkontext zu visualisieren. Auch wenn noch standardisierte Visualisierungs-„Guidelines“ fehlen, bringt der Einsatz von Augmented Reality zur Datenvisualisierung einen erheblichen Profit mit sich.

Schlüsselwörter: Augmented Reality, Mixed Reality, Smartphone, Internet der Dinge, Sensoren, Sensordaten, Visualisierung, menschliche Wahrnehmung

1 Einleitung

Im Jahre 2020 werden laut dem Marktforschungs- und Beratungsunternehmen Gartner 26 Milliarden Geräte im Internet der Dinge miteinander verbunden sein [17]. Diese eingebetteten Geräte nehmen einen immer größer werdenden Anteil der digitalen Systeme ein [23]. Die Palette der verschiedenen Geräte reicht dabei von großen Haushaltsgeräten, wie Spülmaschinen oder Kühlschränken, über Wearables, wie Smartwatches oder Fitnessarmbändern, bis hin zu kleinsten Sensorknoten. Jedes dieser Geräte besitzt unzählige Sensoreinheiten, die Unmengen an Daten produzieren. Für 2020 wird die Größe des „digitalen Universums", also der Daten, die jährlich generiert und vervielfältigt werden, auf 44 Milliarden Terabyte prognostiziert. Dies wäre ein Anstieg um das Zehnfache des heutigen Datenaufkommens. Dabei nehmen smarte Gegenstände, eingebettete Systeme und Sensoren einen immer größer werdenden Anteil der digitalen Systeme ein [23]. Will man die Daten von Sensoren visualisieren, geschieht dies mit dem heutigen Stand der Technik meist auf einem externen Rechner, der oft räumlich distanziert von den Messvorrichtungen steht. Doch dabei geht der Kontext, wo bzw. wann die Daten aufgenommen wurden, fast komplett verloren. Hier kommt Augmented Reality (kurz AR) ins Spiel: Mit Augmented Reality Systemen können die anfallenden Daten von verschiedenen Sensoren und Sensortypen in Echtzeit direkt dort visualisiert werden, wo sich der Sensor befindet. Dabei wird die Realität des Nutzers mit virtuellen Objekten (hier Sensordaten) angereichert. Die Einsatzgebiete sind vielfältig: Beispielsweise können Daten von Drucksensoren direkt auf dem gemessenen Objekt angezeigt werden (siehe Abbildung 8 (b)) und somit können mögliche Bruchstellen direkt auf dem Objekt lokalisiert werden. Ein weiteres Anwendungsbeispiel findet sich im Darstellen von Temperatur- und Luftfeuchtigkeitsdaten, die über eine größere Fläche hinweg erhoben werden und anschließend grafisch eingeblendet werden können (siehe Abbildung 9). Die Daten können also ohne einen PC oder Ähnliches als Zwischensystem und direkt im Kontext der Datenerhebung betrachtet werden [25]. Durch AR werden völlig neuartige Darstellungen möglich, zugleich müssen aber zusätzlich zu den bekannten Erkenntnissen der klassischen 2D-Datenvisualisierung weitere Faktoren berücksichtigt werden. Diese Arbeit soll im Folgenden - nach einigen grundlegenden Ausführungen über Augmented Reality - einen Überblick über den Stand der Technik bei der Visualisierung mit AR geben. Dabei werden besondere Herausforderungen, die bei der Visualisierung zu beachten sind und die dafür existierenden Lösungsansätze erläutert. Darauf basierend werden verschiedene Visualisierungstechniken zur Darstellung von Sensordaten präsentiert, die in AR ihren Einsatz finden.

2 Grundlagen

2.1 Grundlagen von Augmented Reality

Zunächst sollen einige grundlegenden Begrifflichkeiten definiert und Technologien vorgestellt werden. *Augmented Reality* stellt eine „Variation einer virtuellen

Umgebung dar“ [3], wobei der Nutzer sich nicht in einer komplett virtuellen Welt bewegt, sondern die reale Welt mit virtuellen Objekten angereichert sieht. Somit befindet sich AR im Bereich der „mixed reality“, wo virtuelle und reale Umgebungen miteinander vermengt werden (Abbildung 1). Die unten stehende Grafik zeigt ein Realität - Virtualität Kontinuum mit den zwei Extrema Realität und Virtualität. Dazwischen befindet sich der Bereich der Mixed Reality, wo sich neben der Augmented Reality noch die Augmented Virtuality ansiedeln lässt, bei der reale Objekte in eine vorwiegend virtuelle Umgebung integriert werden. Das Ziel bei AR ist es also die Realität zu bereichern und das möglichst so, dass kein Unterschied in der Wahrnehmung von virtuellen und realen Objekten besteht.

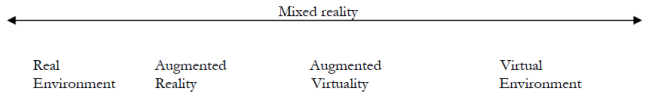


Abb. 1. Reality - Virtuality Kontinuum nach Milgram und Kishino. Quelle: [13]

Ronald T. Azuma nennt drei grundlegende Charakteristika, die ein solches AR-System beschreiben [3]:

1. Kombination von realer und virtueller Umgebung
2. Interaktivität in Echtzeit
3. Wahrnehmbar in drei Dimensionen

Insbesondere sind die Echtzeitfähigkeit und die Dreidimensionalität ein existenzielles Kriterium für ein AR-System. Das heißt, zum einen muss die Berechnung so schnell erfolgen, dass der Benutzer interaktiv mit den virtuellen Elementen arbeiten kann und zum anderen müssen die eingeblendeten Objekte perspektivisch so angezeigt werden, dass sie sich nahtlos in die dreidimensionale Realität einfügen. Zusätzlich zur visuellen Anreicherung der Umgebung können mit Augmented Reality auch andere Sinne wie Hören, Riechen, oder Tasten angesprochen werden [2]. Dies soll allerdings im Folgenden nicht näher behandelt werden, da es für die Visualisierung von Sensordaten von untergeordneter Bedeutung ist.

Zur Umsetzung eines AR-Systems existieren verschiedene Hardwareansätze, wobei man in zwei Dimensionen kategorisiert. Erstens spielt die Position des Displays eine Rolle, auf dem die AR dargestellt wird. Und zweitens unterscheidet man nach der Technik, die zur Darstellung auf diesem Display verwendet wird.

Abbildung 2 zeigt drei mögliche Positionen für das verwendete Display. Bei einem *„head-mounted display“(HMD)* wird das Display (meist in Form einer Brille o.Ä.) direkt vor die Augen des Benutzers gehalten. Die Einblendung der AR kann auf zwei verschiedene Arten erfolgen: Bei einem *„optical see-through display“* sieht der Benutzer die Realität durch das semi-transparente Display und die virtuellen Objekte werden auf dieses Display projeziert, sodass sie die Realität überlagern (Abbildung 3). Beim Einsatz eines *„video see-through displays“*

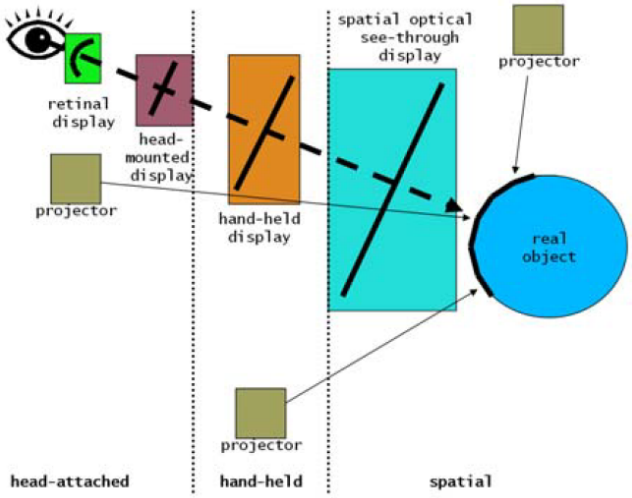


Abb. 2. Positionierung des AR Displays und Anzeigetechniken. Quelle: [6]

wird die reale Welt mit Hilfe einer Kamera aufgezeichnet, die Aufzeichnung mit AR-Elementen angereichert und dem Benutzer auf einem Display angezeigt (Abbildung 3).

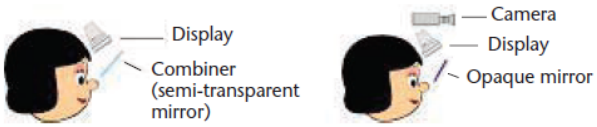


Abb. 3. Gegenüberstellung optical see-through und video see-through. Quelle: [2]

Eine weitere Möglichkeit ist die Verwendung eines *„hand-held displays“*. Hier kommen zum Beispiel Smartphones oder Tablets zum Einsatz, die meist als „video see-through displays“ verwendet werden. Bei dieser Umsetzung ist besonders hervorzuheben, dass sie das beste Potenzial besitzt, um Augmented Reality dem Massenmarkt zugänglich zu machen [24]. Heutige Smartphones zeichnen sich durch eine ausreichend hohe Auflösung von sowohl Bildschirm, als auch Kamera aus, sind für einen Großteil der Bevölkerung intuitiv zu bedienen und kostengünstig. Das verdeutlichen auch Studien, zum Beispiel von Rauhala et. al. [15]. Im weiteren Verlauf der Arbeit werden deshalb weitgehend mobile Endgeräte als Plattform für AR betrachtet.

Neben den bereits genannten „head-mounted“- und „hand-held“-Displays, die direkt am Körper getragen bzw. gehalten werden, existieren im Kontrast dazu Systeme, bei denen sich das Display, losgelöst vom Benutzer, räumlich entfernt in die Umgebung integriert ist [6]. Hier kommen entweder die bereits erwähnten optical- bzw video-see-through Techniken zum Einsatz, oder aber die virtuellen Objekte werden mittels Projektoren direkt auf physikalische Oberflächen projiziert [6].

Für die Platzierung der eingeblendeten Objekte ergeben sich drei Möglichkeiten [4]:

- **Head-stabilized:** Informationen werden relativ zum Sichtfeld des Nutzers fixiert (zum Beispiel immer in der rechten oberen Ecke des Blickfeldes).
- **Body-stabilized:** Hier werden die Objekte relativ zum Körper des Nutzers fixiert. Dazu muss die Orientierung der Blickrichtung bezüglich des Körpers getracked werden.
- **World-stabilized:** Objekte werden bezüglich realer Gegenstände fixiert. Diese Option stellt die erste Wahl für die Visualisierung von Sensordaten dar, da die Messwerte einen direkten Bezug zur Umgebung bzw. zu ihrem Sensor besitzen.

Um die augmentierten Daten korrekt darstellen zu können, müssen die Positionen des Benutzers und der Objekte, die für die AR von Bedeutung sind, bekannt sein [2]. Nur so können die virtuellen Einblendungen transformiert und an der richtigen Stelle dargestellt werden. Auch hier gibt es mehrere Ansätze, die hier nur kurz erwähnt werden sollen. Tabelle 1 zeigt eine Übersicht einiger Technologien.

Technologie	**Reichweite (m)**	**Setup (h)**	**Auflösung (mm)**	**Zeit bevor Drift auftritt (s)**	**Umgebung**
Accelerometer	1000	0	100	1000	Innen/Außen
Optisch, Marker-basiert	10	0	10	∞	Innen/Außen
Optisch, ohne Marker	50	0-1	10	∞	Innen/Außen
Hybrid	10	10	1	∞	Innen
GPS	∞	0	1000	∞	Außen
Wi-Fi	100	10	1000	∞	Innen/Außen

Tabelle 1. Übersicht über ausgewählte Trackingverfahren. Quelle: [14]

Eine sehr einfache und kostengünstige Variante stellt das Marker-basierte Tracking dar. Hierbei werden Markierungen an die Stellen angebracht, an denen später die grafischen Elemente eingeblendet werden sollen. Auch markerlose optische Systeme existieren, benötigen jedoch komplexere Objekterkennung. Für

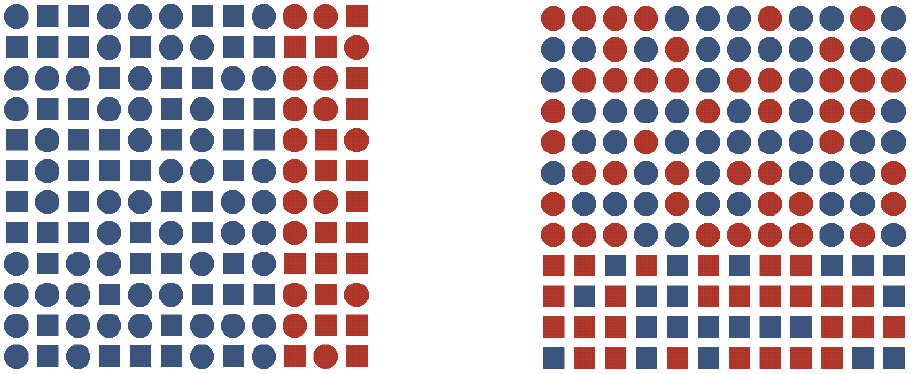
den Außeneinsatz bietet sich GPS an, da es hier keine Beschränkung hinsichtlich der Reichweite gibt (auch wenn es eine erheblich geringere Auflösung besitzt). Denkbar sind auch hybride Lösungen, bei denen mehrere Trackingverfahren kombiniert werden.

2.2 Grundlagen der Visualisierung

Mit den in Kapitel 2.1 vorgestellten Grundlagentechnologien lassen sich AR-Systeme entwickeln, mit denen sich Daten von verschiedensten Sensoren grafisch veranschaulichen lassen. Das Ziel einer Visualisierung ist es immer, grafisch repräsentierte Daten zu untersuchen, um daraus Erkenntnisse über diese zu gewinnen [19]. Doch um beurteilen zu können, ob eine Visualisierung gut bzw. schnell oder schlecht verständlich ist, muss man zunächst wissen, wie der menschliche Wahrnehmungsapparat arbeitet. Denn das menschliche Sehsystem ist ein komplexes und mächtiges Werkzeug unseres Körpers, das wir uns bei der Darstellung zunutze machen können [19]. Bei der menschlichen Wahrnehmung läuft sehr viel unterbewusst ab: Aufgaben wie Mustererkennung, Gruppierungen, Suchen nach bestimmten (einzigartigen) Elementen oder Abschätzungen können extrem schnell (unter 200ms) und akkurat auf niedriger Ebene der Wahrnehmung durchgeführt werden (auch „preattentive Processing“ genannt) [10]. Diese Art der Perzeption kann ausgenutzt werden um dem Betrachter eine schnelle Untersuchung der Visualisierung zu ermöglichen und so Tendenzen oder Auffälligkeiten auf den ersten Blick zu erkennen [10].

Der Mensch ist darauf trainiert, Muster zu erkennen. Dabei spielt eine Rolle, dass er Lichtunterschiede, also Kontraste, besser wahrnehmen kann, als absolute Helligkeitswerte. Außerdem können Kanten und Konturen besser erkannt werden, als langsam variierende Helligkeitsverläufe. Das kommt der Visualisierung vor allem im Computergrafik-Bereich und auch bei AR zugute. Grafische Displays können zwar nicht physikalisch realitätsgetreue Helligkeiten darstellen, aber sehr ähnliche Muster bzw. Kontraste [20]. Dadurch entstehen vor allem bei video see-through Displays (z.B. Smartphones) keine großen Probleme bezüglich Kontrast und Helligkeitsumfang, da hier Realität und Virtualität auf dem gleichen Bildschirm dargestellt werden und so denselben Limitierungen unterliegen. Bei optical see-through Systemen kann es jedoch zu Komplikationen kommen, da hier die Einblendung auf einem begrenzten Display erfolgt und somit kann es beispielsweise bei sehr heller Umgebung dazu kommen, dass AR-Objekte „ausgewaschen“ wirken [3].

Um die Wahrnehmung einer Visualisierung zu verbessern und weitere Informationen in eine Grafik zu codieren können verschiedene visuelle Merkmale wie Farbe, Größe und Form verwendet werden. Dadurch können auch Daten mit mehr als zwei Dimensionen auf einem zweidimensionalen Medium visualisiert werden. Bei der Verwendung solcher Merkmale muss darauf geachtet werden, dass diese eine gegenseitige Einwirkung aufeinander besitzen. Das kann sich zum einen positiv auf die präattentive Verarbeitung auswirken, falsch eingesetzt aber auch die Erkennung verlangsamen (Abbildung 4).



(a) (b)

Abb. 4. In (a) kann die Farbgrenze trotz zufälliger Form schnell (präattentiv) erkannt werden. Hingegen bei (b) interferiert die Erkennung der Formgrenze sehr stark mit der Einfärbung. Quelle: [10]

In Abbildung 4 erkennt man, dass bei der Erkennung von visuellen Grenzen die Farbe eines Elementes eine wichtigere Rolle spielt, als die Form der Objekte [7]. Allgemein kommt Farbe eine große Bedeutung bei der Visualisierung zu. Mit ihr können zusätzliche Informationen in eine Visualisierung codiert werden. Zur Darstellung des verwendeten Farbraumes werden häufig der RGB- bzw. der HSV-Raum benutzt. Beim RGB-System wird eine Farbe durch den Anteil der Rot-, Grün- und Blauwerte definiert. Der HSV-Raum dagegen basiert auf den drei Dimensionen Farbton, Sättigung und Helligkeit. Hinsichtlich der Farbauswahl sollte einiges beachtet werden: Es stellt sich die zentrale Frage, welche Farben gewählt werden sollten, um eine gute Abgrenzung bzw. Erkennung der Datenelemente zu gewährleisten [10]. Da die Größe einer eingefärbten Fläche ihre Farbwahrnehmung beeinflusst, sollten helle, satte Farben für kleinere Flächen und weniger satte Pastelltöne für größere Flächen bzw. Hintergründe verwendet werden [20]. Auch die Bedeutungen von Farben sollten nicht unterschätzt werden. Die Aussagekraft von Farben kann gezielt genutzt werden, um den Fokus des Betrachters auf bestimmte Bildbereiche zu lenken: Rottöne werden instinktiv als Warnfarbe wahrgenommen und ziehen deshalb eine gewisse Aufmerksamkeit auf sich, während hingegen grün gefärbte Bereiche dem Betrachter eher das Gefühl geben hier sei „alles in Ordnung“. Die Wahrnehmung ist aber auch hier erneut stark vom Kontext abhängig. Nimmt man die Visualisierung von Temperaturen, so wird Rot intuitiv als „heiße“ Farbe, blau dagegen eher als „kalt“ wahrgenommen, obwohl physikalisch korrekt betrachtet blau heißer als rot ist (aufgrund der kleineren Wellenlänge) [19].

Doch die endgültige Farbwahrnehmung hängt neben dem intuitiven Eindruck auch entscheidend vom Hintergrund ab. Bei diesem Phänomen - auch Simultankontrast genannt - können zwei identisch eingefärbte Bereiche als komplett ver-

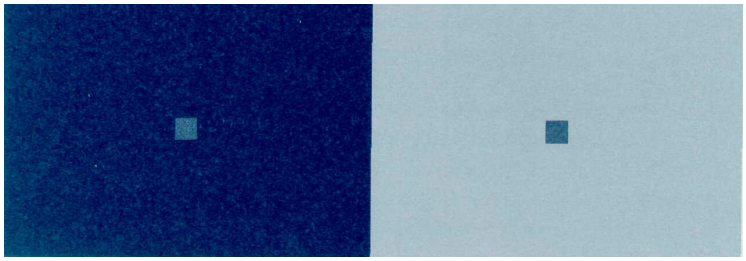


Abb. 5. Simultankontrast. Quelle: [21]

schieden wahrgenommen werden, einzig und allein durch das Vorhandensein von unterschiedlichen Umgebungen [21]. Zu sehen ist diese Illusion in Abbildung 5. Die kleinen Quadrate im Inneren besitzen die gleiche Farbe, das Rechte scheint jedoch viel dunkler zu sein, durch den helleren Hintergrund.

3 Visualisierung der Sensordaten

Sensoren messen physikalische oder chemische Messgrößen über einen bestimmten Zeitraum hinweg. Die gemessenen Daten können dabei vorübergehend zwischengespeichert oder direkt zur Verarbeitung weitergeleitet werden. Interessiert man sich lediglich für den aktuellen Wert, so genügt es, nur diesen graphisch anzuzeigen (z.B. mit Glyphen, siehe Kapitel 3.2). Will man dagegen die Trends und Muster der Daten analysieren, so werden Visualisierungsmethoden benötigt, die den Messwert und seine Änderung im Laufe der Zeit oder über einen größeren Raum hinweg darstellen. Die gemessenen Daten bei einer solchen Zeitreihe sind von diskreter Natur, da der Sensor nur an endlich vielen Punkten Messwerte erfassen bzw. speichern kann. Um den Verlauf der Messgröße hervorzuheben, können die einzelnen Datenpunkte miteinander verbunden (interpoliert) werden, um eine kontinuierliche Darstellung zu erhalten. Eine weitere Eigenschaft, die für die Auswahl der Darstellung eine entscheidende Rolle spielt, ist die Dimensionalität der Daten [19]. Für eindimensionale Daten bieten sich x-y Plots, Säulendiagramme oder Listen an, bei denen ein Messwert einem bestimmten Zeitpunkt zugeordnet wird. (Klassische) Visualisierungstechniken für mehrdimensionale Daten sind zum Beispiel (gestapelte) Flächendiagramme, Matrizen oder sog. „stream graphs“ [12]. Im Folgenden sollen zunächst Probleme aufgezeigt werden, die speziell bei der Visualisierung mit AR auftreten. Anschließend werden die verschiedenen Techniken zur Visualisierung in Augmented Reality näher betrachtet.

3.1 Besondere Herausforderungen bei der Visualisierung mit Augmented Reality

Im vorangegangen Kapitel wurde aufgezeigt, welche grundlegenden Faktoren eine Rolle bei der Visualisierung spielen. Doch zusätzlich zu diesen Erkenntnissen

kommen bei Augmented Reality noch weitere Faktoren ins Spiel, denn hier werden die Daten auf einem dynamischen Hintergrund visualisiert. Da im Voraus nichts über die Umgebung bekannt ist, muss damit gerechnet werden, dass er mit dem grafischen Overlay interferieren kann und so die Wahrnehmung der Visualisierung erschwert wird, z.B., indem die Wahrnehmung der Farbe variiert (siehe Kapitel 2.2). Um Überlagerungen durch einen farbigen Hintergrund zu umgehen, schlagen Gunnarsson et. al. eine Desaturierung des Hintergrunds vor. Hierbei werden die Farben entsättigt, wodurch die Umgebung weniger bunt wirkt. So können die Störungen durch Hintergrundfarben beseitigt werden und gleichzeitig bleibt der Hintergrund noch als Orientierung und Kontext erhalten [8]. Als weitere Option nennt M. Rauhala das (vorübergehende) Ausblenden des kompletten Hintergrunds (sofern ein video see-through display verwendet wird). So können zwar die Interferenzen beseitigt werden, der Kontext zur Umgebung geht aber verloren. Die grafischen Überlagerungen der AR können neben den genannten Interferenzen aber auch problematisch werden, wenn sie zu viel des Hintergrunds überdecken. Dadurch kann der explizit erwünschte Kontext zu Umgebung/Umwelt verloren gehen. Um dies zu verhindern, nutzen Gunnarsson et. al. beispielsweise transparente Einblendungen oder „quadified visualisations“ (siehe Kapitel 3.2) mit dem zusätzlichen Ziel, den Benutzer stärker von der Illusion der augmentierten Realität zu überzeugen [9].

Weiterhin muss die Sichtbarkeit des zu augmentierenden Objektes geprüft werden, bzw. ermittelt werden, ob die einzublendenden Visualisierungen überhaupt sichtbar wären. Dies kann z.B. mit einer Tiefenkamera geschehen, die misst, ob an der Stelle (Tiefe), an der die Elemente eingeblendet werden sollen, nähere Objekte existieren. Die korrekte Verdeckungsberechnung ist außerdem wichtig, um beim Anwender eine richtige Tiefenwahrnehmung der Objekte zu erzeugen [18]. Zusätzlich mit einer Schattenvisualisierung kann damit der Nutzer die Position der virtuellen Objekte im Raum deutlich besser einordnen [11]. Wie man in Abbildung 6 erkennen kann, ist die Position des Schattens eines Objekts entscheidend für die Identifizierung der Lage des Objekts im Raum.

Werden die virtuellen Elemente über einen größeren Bereich der Realität verteilt, passen oftmals nicht alle Objekte ins Sichtfeld des Benutzers. Um zu verhindern, dass der Nutzer fortwährend seinen Kopf bewegen/drehen muss, ist ein adäquater Sichtbereich des AR-Systems nötig in dem es die virtuellen Objekte darstellen kann [18]. Da die meisten heutigen Systeme nicht das komplette Sichtfeld des Menschen abdecken können, sondern meist nur etwa 20 bis 60 (optical see-through) bzw. 90 (video see-through) Grad, ist hier eine Limitierung gegeben, die bei der Erstellung einer Visualisierung beachtet werden muss [18]. Doch beachtet man, dass das foveale Sichtfeld (Bereich des schärfsten Sehens) des Menschen ohnehin nur wenige Grad abdeckt und die Wahrnehmungsfähigkeit bei 10 Grad bereits rapide abfällt, so empfiehlt es sich ohnehin, die Augmentierung auf den Kernbereich des Sichtbereiches zu beschränken [26]. C. Ware empfiehlt zum Beispiel Bereiche, in denen Text vorkommt, nicht größer als 18 Grad des Sichtfeldes zu dimensionieren. Hieraus folgt, dass es unvermeidbar ist, den Kopf bzw. das AR-Display zu drehen, um größere augmentierte Bereiche zu erfassen.

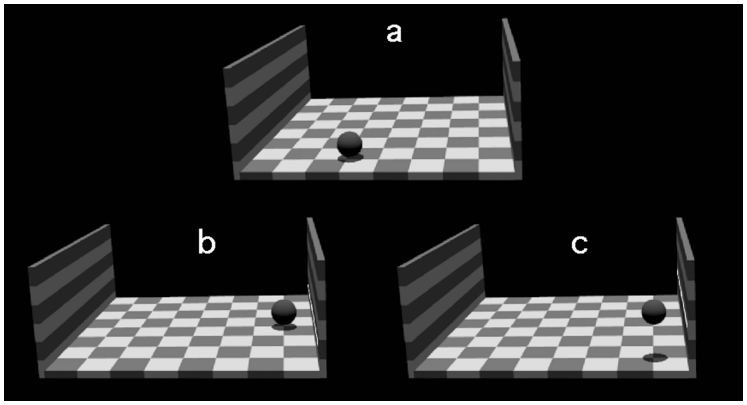


Abb. 6. Einfluss von Objektschatten auf die Positionswahrnehmung. Kugel in Bild (b) scheint auf dem Boden zu liegen, während sie in Bild (c) über dem Boden zu schweben scheint. Quelle: [11]

Doch hierbei muss die Latenz des Systems niedrig genug sein, damit die „Illusion", dass die virtuellen Objekte Teil der Umgebung und dort fixiert sind, nicht verloren geht. Denn ist die Latenz zu groß, können die Objekte nicht schnell genug richtig positioniert werden und „schwimmen" in der Umgebung [18]. Eine mögliche Lösung für diese Herausforderung wäre beispielsweise eine Prädiktion der Kopf-/Displaybewegung [2].

Um unterschiedliche Bereiche der Umgebung zu erfassen nutzt der Mensch darüber hinaus seine Fähigkeit die Sehschärfe an verschiedene Entfernungen anzupassen. Da die Optik (Distanz, an der die virtuellen Objekte projiziert werden) oft fixiert ist (v.A. bei optical see-through Displays) gibt es nur eine Tiefe, an der das AR-Bild und Realität beide im Fokus sind [26]. Dadurch ist es dem Anwender häufig nicht möglich beides gleichzeitig zu fokussieren, wodurch die augmentierte Realität und die physikalische Realität keine Einheit mehr bilden [3]. Wenn die grafische Überlagerung ein reales Bezugsobjekt besitzt, so sollte also versucht werden die Brennebene des virtuellen Bildes auf die typische Tiefe der realen Umgebung zu setzen. Hiermit lassen sich die beiden Bilder einfacher gleichzeitig wahrnehmen, was zu einer besseren Verschmelzung führt. Für den Fall, dass die AR-Überlagerung kein direktes Bezugsobjekt besitzen sollte, empfiehlt es sich die fokale Distanz der Einblendung, also die Distanz, in der die Einblendung im Fokus ist, näher als die der Realität zu setzen. Das reduziert visuelle Interferenzen [26].

3.2 Visualisierung in Augmented Reality

In der klassischen Visualisierung ist man bei der Darstellung auf ein zweidimensionales Medium beschränkt, obwohl der Mensch im alltäglichen Leben an eine dreidimensionale Wahrnehmung gewohnt ist. Edward R. Tufte schreibt über diese Limitierung: *„Escaping this flatland is the essential task of envisioning*

information-for all the interesting worlds [..] that we seek to understand are inevitably and happily multivariate in nature. Not flatlands.“ [21] Mit AR kann aber genau diese Beschränkung aufgehoben werden, also dem Flachland der Visualisierung entkommen werden: Die Daten können als 3D-Objekte in die Realität eingebunden werden. Das bringt natürlich den großen Vorteil mit sich, dass sich die Daten aus verschiedenen Ansichten/Blickwinkeln betrachten lassen. Die Daten können intuitiv durch „herumlaufen“ um die Visualisierung, in genau dem Blickwinkel betrachtet, gezoomt und analysiert werden, wo interessante Auffälligkeiten zu erkennen sind. So können die ausschlaggebenden Aspekte der Visualisierung wesentlich besser und schneller verstanden/wahrgenommen werden [19]. Durch die hohe Integration in die physikalische Welt lassen sich noch viele weitere Formen der Interaktion mit den virtuellen Objekten realisieren, was in dieser Arbeit aber nicht näher vorgestellt werden soll.

Grundsätzlich zeichnen sich zwei verschiedene Visualisierungsmodi bei der Gestaltung der AR in der Literatur ab: Einblendung von einzelnen Visualisierungen zu jedem Sensor oder Einblendung einer graphischen Überlagerung aus den gesammelten Daten der verfügbaren Sensoren. Abbildung 7 zeigt die unterschiedlichen Möglichkeiten, Sensorwerte einzubetten.

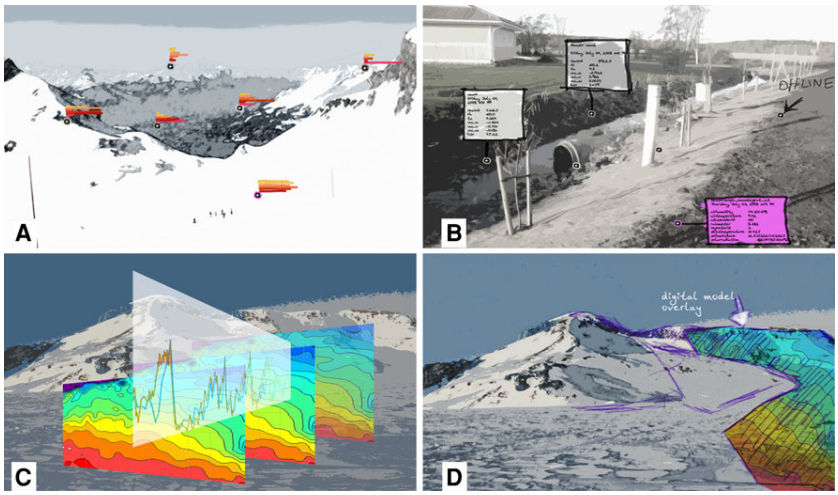
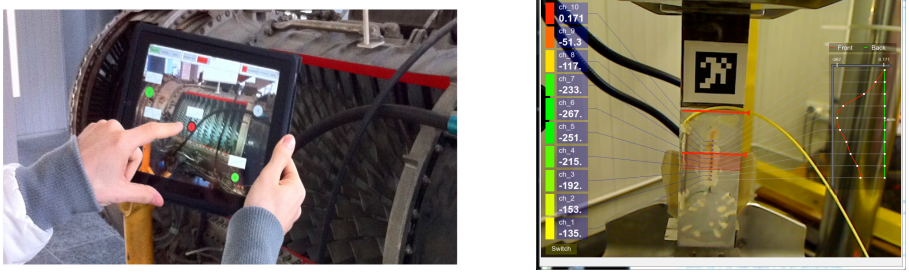


Abb. 7. Übersicht der verschiedenen Visualisierungsmöglichkeiten. Oben: Sensorbasierte Visualisierung mit „miniplots“ (a) und Beschriftungen (b). Unten: Graphische Überlagerung. Quelle: [25]

Bei der *Sensor-basierten Visualisierung* wird zu einem bestimmten Objekt eine einzelne Grafik angezeigt. Das kann lediglich der Messwert selbst sein, aber auch Grafiken die den Messwert oder den Verlauf der Messungen enthalten.

Wie in Abbildung 8 (a) dargestellt kann zu jedem Sensor der Messwertverlauf mit einem Graphen repräsentiert werden. Um die Positionen der Sensoren



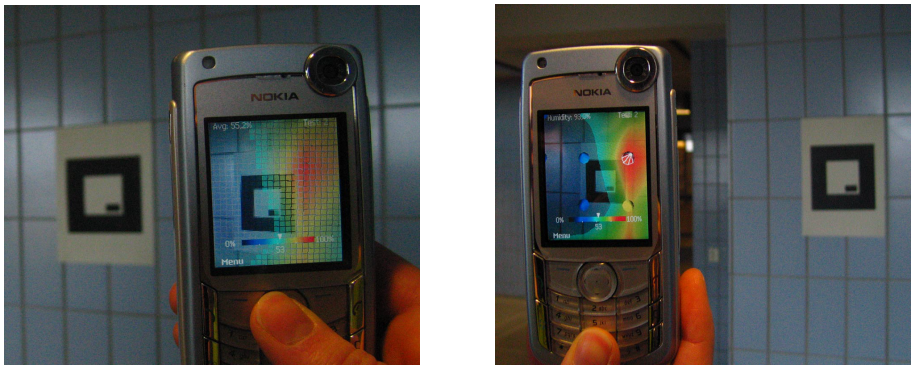
(a) Augmentierung einer mit Sensoren ausgestatteten Turbine. Quelle: [5]

(b) Ar im Einsatz bei Material-Ermüdungstests. Quelle: [16]

Abb. 8. Visualisierung durch Einblendung von Informationen zu einzelnen Sensoren.

zu markieren, benutzt M. Berning farbige Markierungen. Hierbei kann es aber zu Problemen kommen, wenn die einzelnen Grafiken zu klein sind bzw. zu viele Grafiken auf geringem Platz angezeigt werden müssen. Einen etwas anderen Ansatz zeigt G.M. Re, der in Abbildung 8 (b) zu sehen ist. Hier wird zu jedem Messpunkt am linken Bildschirmrand der zugehörige Messwert angezeigt, also nicht direkt am jeweiligen Sensor. Zusätzlich wird ein Verlauf der Werte über das Werkstück hinweg visualisiert. Verbindungslinien werden hier benutzt, um den Sensoren eindeutig ihre Messwerte zuzuordnen. Zu beachten ist hierbei jedoch, dass die Menge der eingeblendeten Objekte nicht zu groß wird, was zu einer unübersichtlichen Darstellung führen kann [2]. Abhilfe vor einer Überflutung der Visualisierung kann z.B. eine Filterung nach der Blickrichtung des Endanwenders schaffen [1]. Dabei wird der Blick des Nutzers verfolgt und so die Bereiche detektiert, auf denen die Hauptaufmerksamkeit liegt. In diesen Bereichen werden dann AR-Objekte visualisiert, während die Objekte im restlichen Blickfeld herausgefiltert werden.

Interessiert man sich eher für den Verlauf einer Messgröße über einen bestimmten Raum hinweg, als für die Werte der einzelnen Sensorknoten, so bietet sich an, die Augmentation als *kontinuierliche Visualisierung* zu gestalten. Da (noch) nicht so viele Sensoren eingesetzt werden können, sodass eine Eins-zu-eins-Abbildung zwischen Sensorwert und Pixelfarbe möglich wäre, müssen die Messdaten interpoliert werden, um eine kontinuierliche Überlagerung zu generieren. Dazu ist eine Vorverarbeitung der Daten nötig, in der die Werte z.B bilinear (für 2D) oder trikubisch (3D) interpoliert werden und die Grafik berechnet wird [8]. Dabei werden die Messwerte auf eine korrespondierende Farbe abgebildet (Abbildung 9). Zusätzlich können Markierungen benutzt werden, um erkennbar zu machen, wo die realen Sensoren positioniert sind, wie in Abbildung 9 (b) zu sehen ist. Die Markierungen können benutzt werden, um dem Benutzer einen Zugriff auf die originalen Messdaten der einzelnen Sensoren zu ermöglichen [15].



(a) „Quadified“ Overlay

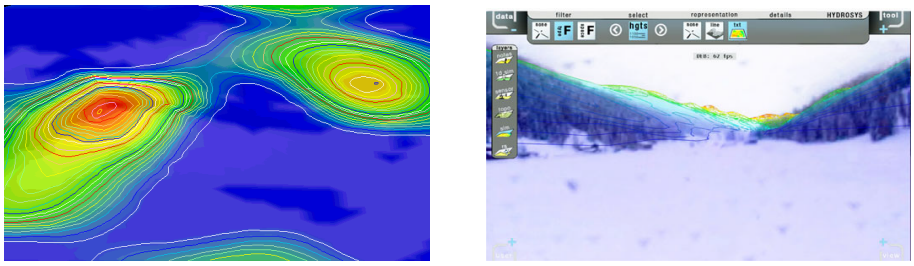
(b) Kontinuierliche Überlagerung mit Markierungen an dern Sensorpositionen.

Abb. 9. Visualisierung durch Überlagerung der Realität mit einem graphischen Overlay. Quelle: [15]

Zusätzlich zur kompletten Überlagerung der Wirklichkeit gibt es noch die Möglichkeit der bereits im vorangegangenen Kapitel erwähnten „quadratifizierten“ Visualisierung, wie in Abbildung 9 (a) zu sehen ist. Hierbei werden in der interpolierten Grafik mehrere Pixel zu einem quadratischen Block zusammengefasst, der nach der Durchschnittsfarbe der Pixel koloriert wird. Durch den dadurch erzeugten höheren Grad an Immersion, kann diese Technik zu einer verbesserten Erfassung der Daten führen [8].

Eine weitere kontinuierliche Visualisierungstechnik ist die Konturierung, bei der die Farbabbildung um Konturlinien erweitert wird. Dabei können die Konturlinien entweder zusätzlich zum Hintergrund eingeblendet werden oder als alleinige Visualisierung dienen. Dieses Konzept kann helfen, die unterschiedlichen Farbbereiche schneller in klar abgegrenzte Bereiche zu separieren, was unmittelbar aus den Erkenntnissen zur Wahrnehmung aus Kapitel 2.2 folgt.

Um aus den Sensordaten eine farbige Visualisierung zu gewinnen, müssen den Messwerten Farbtöne zugeordnet werden. Diese Abbildung kann entweder durch eine Transferfunktion oder über eine Zuordnungstabelle realisiert werden. Mit dieser Technik können eindimensionale Werte visualisiert werden, aber auch höher-dimensionale Objekte um einen zusätzlichen Informationsgrad angereichert werden [19]. Bei der Abbildung von Messwerten zur Pixelfarbe der Visualisierung spielen die verwendeten Farbtöne eine entscheidende Rolle. Standardmäßig wird oft eine Regenbogenskala (von Blau nach Rot) verwendet, wie auch in Abbildung 9 und Abbildung 11 geschehen. Rot signalisiert hier den höchsten Wert und mit Blau werden niedrigere bzw. weniger interessante Werte gekennzeichnet, was den Überlegungen zum Farbeinsatz aus Kapitel 2.2 entspricht. Zusätzlich zur Abbildung auf Farbwerte kann eine Transparenzabbildung hinzugefügt werden. Somit können weniger wichtige Bereiche weitgehend ausgeblendet werden, um den Nutzer nicht unnötig abzulenken [15]. Der Nach-



(a) Konturlinien zusätzlich zum eingefärbten Hintergrund. Quelle: [8]

(b) Kolorierte Konturen ohne Hintergrund. Quelle: [25]

Abb. 10. Colormapping mit Konturlinien.

teil der Regenbogenskala besteht jedoch darin, dass zum einen der Farbton als primärer Ordnungsfaktor verwendet wird und die Skala nicht linear ist. Da über den Raum der Farbtöne keine (natürliche) Ordnung definiert ist, sollte (zusätzlich) die Helligkeit als Indikator für die Wichtigkeit verwendet werden. Zusätzlich sollte man monoton steigende Helligkeitswerte benutzen, um dem Nutzer eine lineare Wahrnehmung der Skala zu ermöglichen [20]. Damit der Anwender schließlich die dargestellten Farbwerte der Visualisierung interpretieren kann, ist der Einsatz einer Legende essenziell.

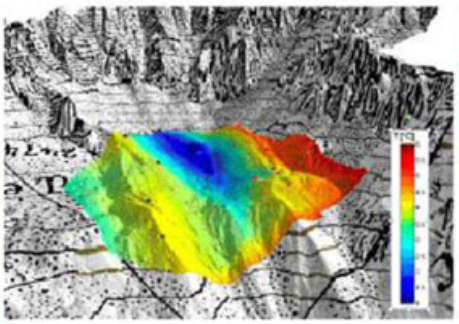


Abb. 11. Farbabbildung nach der Regenbogenskala mit nebenstehender Legende. Quelle: [25]

Wie bereits in Abbildung 9 (b) können Glyphen benutzt werden, um die Position eines Sensors zu markieren. Eine Glyphe ist „ein Objekt, das durch seine Eingabedaten beeinflusst ist“ [19]. Das kann ein zwei- oder dreidimensionales geometrisches Objekt sein, das eine Messgröße darstellt, etwa als Länge, Größe, Richtung oder Farbe. Bereits in Abbildung 9 (b) wurden Glyphen eingesetzt, um weitere Daten in ihnen zu kodieren. Hier wird zusätzlich der Messwert des Sensors als Farbe der Glyphe visualisiert. Ferner können sie benutzt werden, um

komplette Visualisierungen zu realisieren (Abbildung 12), was dann als „hedgehog“ bezeichnet wird [19]. In Abbildung 12 werden in beiden Grafiken Muskelfasern des Herzens visualisiert. Die Ausrichtung der Fasern wird als Farbe kodiert (zum Beispiel sind Rote Boxen entlang der X-Achse orientiert). Das rechte Bild zeigt die selben Daten mit einer verbesserten Skalierung der Glyphen. Dadurch kann die Ausrichtung der Körper besser erkannt werden. Auch bei dieser Technik gibt es einige Komplikationen zu beachten. Zum einen kann die Verwendung von einer zu großen Menge an Glyphen die Visualisierung überladen, sodass einzelne dargestellte Dimensionen nur noch schwer erkennbar sind. Dieser Effekt lässt sich auch gut im linken Teil von Abbildung 12 beobachten. Die Ausrichtung der würfelförmigen Glyphen ist kaum zu erkennen. Lediglich aus der Farbe kann der Betrachter noch Rückschlüsse ziehen. Zum anderen kann die Darstellung zu einer erheblichen Irreführung der Wahrnehmung der Daten führen, da die Größe der Glyphengrafik oft nicht direkt proportional zur Größe der dargestellten Messwerte ist [22]. Wählt man etwa bei einer Kugel einen doppelten Radius für einen verdoppelten Messwert, so entsteht durch die Vervierfachung der Oberfläche eine Verzerrung in der Wahrnehmung [19].

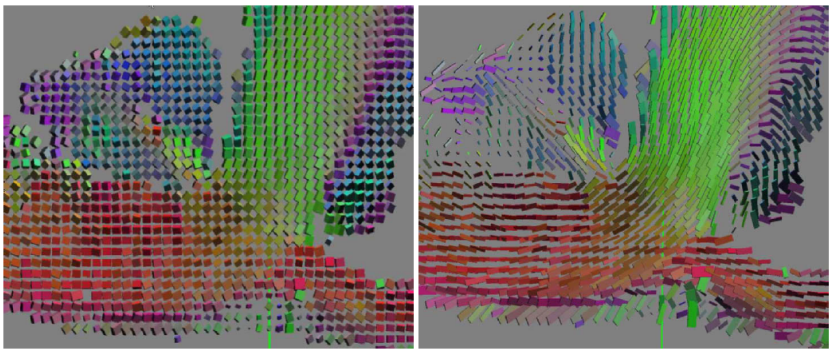


Abb. 12. Visualisierung von Herzmuskel-Fasern mit Hilfe von Glyphen. Quelle: [27]

4 Zusammenfassung

Die weltweite Vernetzung des Internets der Dinge wird in den nächsten Jahren immer mehr voranschreiten und sowohl in unser alltägliches Leben (Smart Homes) als auch in die Arbeitswelt (Industrie 4.0) Einzug erhalten, wobei extrem große Mengen an Daten generiert werden. Gerade in diesem Anwendungsgebiet ist Augmented Reality ideal, um die Sensordaten zu visualisieren: Das Hauptmerkmal des Internets der Dinge ist es ja gerade reale, physikalische Gegenstände zu vernetzen. Will man die Daten dieser Objekte nun auf einem Computer/Workstation visualisieren, so geht genau der besagte Kontext zur realen,

physischen Welt verloren. Es ist also nur logisch, die Daten von solchen Sensoren direkt in der Realität, da wo sie aufgenommen wurden, anzuzeigen. Wie in den vorangegangenen Kapiteln aufgezeigt stehen bei der Visualisierung verschiedene Optionen zur Verfügung, die je nach Anwendungsfall ihre Vorzüge besitzen. So können zur Darstellung von Daten einzelner Sensoren sensorbasierte Visualisierungstechniken eingesetzt werden, bei denen zum jeweiligen Sensor der aktuelle Messwert oder Verläufe dargestellt werden. Zusätzliche Informationen können in Glyphen codiert werden. Will man Sensordaten über größere Bereiche hinweg visualisieren, so bieten sich graphische Überlagerungen an. Hierbei werden die Sensordaten interpoliert, um kontinuierliche Daten zu generieren und anschließend mithilfe eines Mappings auf Farben abgebildet. Neben den Darstellungstechniken selbst wurden auch Herausforderungen bzw. Probleme bei der Visualisierung mit AR aufgezeigt und einige Lösungsansätze präsentiert. Besonders zu beachten ist stets, dass Interferenzen mit dem dynamischen, im Vorfeld nicht bekannten Hintergrund auftreten können. Mit dem Einsatz von AR zur Darstellung von Sensordaten bleibt auf der einen Seite der Messkontext erhalten und es entfallen Kosten für Zwischenschritte, die vorher zwischen dem Aufnehmen und dem Auswerten der Messdaten angefallen wären. Im Kontext der Industrie 4.0 kann also der Anlagentechniker bei der Überwachung/Betreuung und Instandhaltung der Anlagen durch den bereitgestellten höheren Informationsgrad unterstützt werden. So kann die Effizienz um bis zu 10 bis 40 Prozent gesteigert werden [5]. Auf diese Weise können die mit Messdatenanalyse verbundenen Arbeiten schneller und vor allem einfacher durchgeführt werden [3].

Aus den vorangegangenen Darstellungen konnte man ferner sehen, dass bereits sehr gute Ansätze zur Visualisierung von Sensordaten existieren. Diese Ansätze sind jedoch meist sehr spezielle Lösungen für wenige individuelle Anwendungsfälle (z.B. Luftfeuchtigkeitsvisualisierung bei M. Rauhala oder Materialermüdungs-Visualisierung bei G.M. Re). Jedes der genannten Systeme benutzt ganz eigene Arten der Darstellung. Ähnlich wie bei Smartphone-Apps bräuchte es meiner Meinung nach Visualisierungs-„Guidelines“ und standardisierte Widgets für AR-Anwendungen. Nur so können AR-Systeme den Durchbruch zur kommerziellen Nutzung in Industrie und Privatleben schaffen. Zwar wird Augmented Reality bereits des Öfteren benutzt, beispielsweise bei Abseitslinien-Einblendungen im Fußball, jedoch ist den wenigsten dabei weder bewusst, dass es sich hierbei um eine Form der augmentierten Realität handelt, noch hat sich AR in anderen Bereichen wirklich etabliert.

Im Vergleich zur reinen Virtual Reality scheint aber die Akzeptanz von Augmented Reality viel größer zu sein, dadurch, dass hier virtuelle Objekte von einem realen/gewohnten Kontext umgeben sind und nicht alles computer-generiert ist und somit abstrakt und künstlich wirkt [15]. Durch die direkte Kopplung mit der Realität ist das allgemeine Verständnis von AR sehr hoch, vor allem wenn als Hardware ein Smartphone benutzt wird [15]. Gerade Smartphones und Tablets eignen sich meines Erachtens als ideale Hardware-Plattform für AR Anwendungen: Man hat es fast immer dabei, es ist immer an und für einen Großteil der Bevölkerung intuitiv bedienbar. Augmented Reality muss erst noch vollständig

sozial akzeptiert werden und Ängste (vor z.B Verlust von Privatsphäre) müssten abgebaut werden. Aber dadurch, dass die Gesellschaft schon teilweise vertraut damit ist und als Plattform vertraute Geräte verwendet werden können, wird dies in absehbarer Zukunft geschehen [2].

Aber nicht nur im Bereich der Sensordatenvisualisierung, sondern auch in anderen Bereichen kann AR einen Mehrwert schaffen: Im industriellen Bereich können Reparaturarbeiten, das Konstruieren/Designen von neuen Produkten oder die Montage unterstützt werden. Im Infotainment-Bereich finden sich mögliche Anwendungsgebiete bei Head-up-displays für Autos, Videospielen und beim Informieren über in der nähe befindliche Gebäude/Objekte. Und auch im medizinischen Bereich bei der Visualisierung von Körperscans und als Unterstützung bei OPs kann Augmented Reality von außerordentlichem Nutzen sein [3,14]. Gerade wegen dieser Vielfalt und Anpassungsfähigkeit an die verschiedensten Anforderungen wird AR immer stärker in unser Leben integriert werden.

Literatur

1. Ajanki, A., Billinghurst, M., Gamper, H., Järvenpää, T., Kandemir, M., Kaski, S., Koskela, M., Kurimo, M., Laaksonen, J., Puolamäki, K., et al.: An augmented reality interface to contextual information. Virtual reality 15(2-3), 161–173 (2011)
2. Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S., MacIntyre, B.: Recent advances in augmented reality. Computer Graphics and Applications, IEEE 21(6), 34–47 (2001)
3. Azuma, R.T., et al.: A survey of augmented reality. Presence 6(4), 355–385 (1997)
4. Barfield, W., Caudell, T.: Fundamentals of wearable computers and augmented reality. CRC Press (2001)
5. Berning, M., Riedel, T., Ding, Y., Miyaki, T., Fantana, N., Beigl, M.: Augmented service in the factory of the future. In: Adjunct Proc. of the International Conference on Networked Sensing Systems (2012)
6. Bimber, O., Raskar, R.: Modern approaches to augmented reality. In: ACM SIGGRAPH 2006 Courses. p. 1. ACM (2006)
7. Callaghan, T.C.: Interference and dominance in texture segregation: Hue, geometric form, and line orientation. Perception & psychophysics 46(4), 299–311 (1989)
8. Gunnarsson, A.S., Rauhala, M.: Visualisation of sensor data using handheld augmented reality (2006)
9. Gunnarsson, A.s., Rauhala, M., Henrysson, A., Ynnerman, A.: Visualization of sensor data using mobile phone augmented reality. In: Proceedings of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality. pp. 233–234. IEEE Computer Society (2006)
10. Healey, C.G.: Perceptual techniques for scientific visualization. In: SIGGRAPH'99 Course. Citeseer (1999)
11. Kersten, D., Mamassian, P., Knill, D.C., et al.: Moving cast shadows induce apparent motion in depth. PERCEPTION-LONDON- 26, 171–192 (1997)
12. Kirk, A.: Data Visualization: a successful design process. Packt Publishing Ltd (2012)
13. Milgram, P., Kishino, F.: A taxonomy of mixed reality visual displays. IEICE TRANSACTIONS on Information and Systems 77(12), 1321–1329 (1994)

14. Papagiannakis, G., Singh, G., Magnenat-Thalmann, N.: A survey of mobile and wireless technologies for augmented reality systems. Computer Animation and Virtual Worlds 19(1), 3–22 (2008)
15. Rauhala, M., Gunnarsson, A.S., Henrysson, A.: A novel interface to sensor networks using handheld augmented reality. In: Proceedings of the 8th conference on Human-computer interaction with mobile devices and services. pp. 145–148. ACM (2006)
16. Re, G.M., Kharshiduzzaman, M., Bordegoni, M., Bernasconi, A., Anodio, L.F., Comolli, L., Braghin, F.: A mobile augmented reality framework for inspection and visualization during fatigue tests. In: ASME 2014 12th Biennial Conference on Engineering Systems Design and Analysis. pp. V003T10A017–V003T10A017. American Society of Mechanical Engineers (2014)
17. Rivera, J., van der Meulen, R.: Gartner says the internet of things installed base will grow to 26 billion units by 2020. Stamford, conn., December 12 (2013)
18. Rolland, J.P., Fuchs, H.: Optical versus video see-through head-mounted displays in medical visualization. Presence: Teleoperators and Virtual Environments 9(3), 287–309 (2000)
19. Schroeder, W.J., Lorensen, B., Martin, K.: The visualization toolkit. Kitware (2004)
20. Shirley, P., Ashikhmin, M., Marschner, S.: Fundamentals of computer graphics. CRC Press (2009)
21. Tufte, E.R.: Envisioning information. Optometry & Vision Science 68(4), 322–324 (1991)
22. Tufte, E.R., Graves-Morris, P.: The visual display of quantitative information, vol. 2. Graphics press Cheshire, CT (1983)
23. Turner, V., Gantz, J.F., Reinsel, D., Minton, S.: The digital universe of opportunities: Rich data and the increasing value of the internet of things. International Data Corporation, White Paper, IDC_1672 (2014)
24. Van Krevelen, D., Poelman, R.: A survey of augmented reality technologies, applications and limitations. International Journal of Virtual Reality 9(2), 1 (2010)
25. Veas, E., Grasset, R., Ferencik, I., Grünewald, T., Schmalstieg, D.: Mobile augmented reality for environmental monitoring. Personal and ubiquitous computing 17(7), 1515–1531 (2013)
26. Ware, C.: Information visualization: perception for design. Elsevier (2012)
27. Zhukov, L., Barr, A.H.: Heart-muscle fiber reconstruction from diffusion tensor mri. In: Proceedings of the 14th IEEE Visualization 2003 (VIS'03). p. 79. IEEE Computer Society (2003)

Mobiles und smartphonebasiertes Social Sensing

Dominik Galler*
Betreuer: Anja Bachmann†

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

Zusammenfassung. In dieser Arbeit wird ein Überblick über die vorhandenen Forschungsansätze sowie Möglichkeiten im Bereich des Social Sensing gegeben. Social Sensing beschreibt wie Daten, die in direktem Zusammenhang mit menschlichen Aktionen stehen, erfasst und analysiert werden können. Dabei wird zunächst auf die technische Entwicklung eingegangen welche Social Sensing begünstigt hat. Anschließend werden vier verschiedene Studien betrachtet. Hierbei werden die Ansätze der Studien erläutert und auf den Bezug zu Social Sensing eingegangen. Die erste Studie beschäftigt sich damit, wie Nutzer in sozialen Netzwerken als Sensoren eingesetzt werden können. Anschließend wird betrachtet, wie man Vorhersagen über das Verhaltensmuster von Personen oder Personenkreisen durch Social Sensing präzisiert. Die nächste Arbeit beschäftigt sich damit, wie die Anpassungsfähigkeit von Softwaresystemen mittels Social Sensing optimiert werden kann. In der letzten Arbeit wird darauf eingegangen, wie man den Einfluss von Freunden und Bekannten auf eine Person messen kann. Im Anschluss zu den Arbeiten werden die Probleme und Hürden unter technischen, wie humanen Gesichtspunkten analysiert. Abschließend folgt eine Einschätzung der Arbeiten sowie ein Ausblick.

Schlüsselwörter: Social Sensing, Context Sensing, Mobiles Sensing, Smartphonebasiertes Sensing

1 Einleitung

Bereits 2015 werden rund 7 Milliarden Mobilfunkanschlüsse erwartet[12]. Dabei sind heutige Mobilfunkgeräte, wie Handys und Tablet-PCs, mit einer Vielzahl an Sensoren, Kommunikations- und Ortungssystemen ausgestattet. Dazu zählen unter anderem Beschleunigungssensoren, Bluetooth, GPS sowie permanenter Internetzugang durch W-LAN oder Mobilfunk.

Der Nutzer kann also über sein Mobilgerät eine große Menge an Daten generieren und mitteilen. Dazu zählen beispielsweise sein Standpunkt oder die Geschwindigkeit mit der er sich bewegt. Über Bluetooth kann erkannt werden mit wie vielen anderen Geräten (und somit meistens auch Menschen) sich der Benutzer in einem Raum aufhält, womit Rückschlüsse auf seinen sozialen Kontext

gewonnen werden können. Immer mehr Nutzer haben darüber hinaus die technischen Möglichkeiten eines nahezu dauerhaften Internetzugangs [3]. Dadurch werden Soziale Netzwerke auch von unterwegs genutzt und der Nutzer kann gerade gewonnene Eindrücke sofort mit seinem Freundeskreis oder der ganzen Welt teilen. Auch diese Inhalte lassen Rückschlüsse auf den sozialen Kontext zu.

Die Analyse des sozialen Kontextes einer Person oder einer Personengruppe und die Gewinnung von Erkenntnissen aus Daten, die diese generieren, sowie das Schaffen eines Mehrwerts für den Einzelnen, oder einer Gruppe, ist unter dem Begriff Social Sensing zu verstehen. Diese Mehrwerte könnten zum Beispiel für Jugendliche oder junge Erwachsene mit Depressionen oder Persönlichkeitsstörung darin bestehen, dass es ihren Psychologen möglich ist ihr Verhalten und ihren sozialer Umgang zu kontrollieren. Damit wäre es diesen möglich, die Gefühlslage des Patienten zu identifizieren, denn wie von Grünerbl [7] aufgeführt wird, unterscheidet sich das Sozialverhalten von Menschen mit einer manisch-depressiven Erkrankung je nach Phase. Diese Informationen könnten dann dazu beitragen Rückschlüsse auf den Erfolg der Therapie zuzulassen und diese gegebenenfalls anzupassen. Zum anderen könnte ein Mehrwert für eine ganze Gesellschaft geschaffen werden, indem beispielsweise ein Nutzer über ein just stattfindendes Erdbeben twittert und somit die anderen Mitglieder warnen kann [11].

Es stellt sich nun die Frage, wie diese Daten gewonnen werden können, welche Möglichkeiten diese Daten bieten und welche Schlüsse sie zulassen. Diese Arbeit soll hierzu einen Überblick über vorhandene Ansätze und Strategien liefern und versuchen, diese einzuordnen.

Dazu wird zunächst in Kapitel 2 die technische Entwicklung untersucht, welche Social Sensing begünstigt hat. Danach wird in Kapitel 3 eine Arbeit betrachtet, die sich mit Social Sensing durch Twitter beschäftigt. Anschließend widmet sich Kapitel 4 den Arbeiten, bei denen Social Sensing durch Funknetzwerke und Sensoren durchgeführt wird. Darauf hin werden in Kapitel 5 die Grenzen des Social Sensings sowohl unter technischen Gesichtspunkten als auch unter dem Aspekt der Abhängigkeit vom Menschen diskutiert. Letztendlich folgt mit Kapitel 6 eine abschließende Einschätzung.

2 Technische Entwicklung des Social Sensing

Social Sensing befasst sich damit, die sozialen Kontexte der Benutzer zu erfassen und zu analysieren. Dazu zählen beispielsweise alle Interaktionen welche in einem beliebigen sozialen Netzwerk, wie dem Arbeitsplatz, Facebook oder Twitter stattfinden.

Nach Aggarwal [1] gab es in den letzten Jahren einige technische Entwicklungen, welche Social Sensing vor allem durch mobile Endgeräten begünstigt hat. So besitzen mobile Endgeräte immer vielfältigere und bessere Sensoren. Zu diesen zählen unter anderem GPS, Kompass, Bluetooth und Beschleunigungssensor. Des Weiteren müssen die damit gewonnen Daten auch übertragen und gespeichert werden. Aus diesem Grund wird von Aggarwal weiter angeführt, dass

erst die größere Übertragungsbandbreite in Funknetzwerken es ermöglicht hat, die viele Daten in Echtzeit zu übertragen. Betrachtet man dazu die Entwicklung der Mobilfunkstandards wird diese Aussage sehr deutlich. Während unter GPRS am Anfang des 21. Jahrhunderts eine Datenübertragungsgeschwindigkeit von etwa 50 Kbit/s erreicht wurde, können heute mittels LTE und LTE-Advance Upload-Geschwindigkeiten von bis zu 1,5 Gbit/s erzielt werden. Darüber hinaus ist das Speichern der Daten durch immer größere und günstigere Speichermedien stark vereinfacht worden.

3 Echtzeiterkennung von Ereignissen durch Soziale Netzwerke

Takeshi Sakaki et al. befassen sich in ihrer Arbeit „Eathquake Shakes Twitter Users: Real-time Event Detection by Social Sensors"[11] damit, wie man Twitter, respektive dessen Nutzer, als Sensoren einsetzen kann. Die Studie kommt aus Japan. Dies ist nicht ungewöhnlich, denn in Japan finden zum einen sehr viele Erdbeben statt, zum anderen ist Japan sehr urban geprägt und hat viele Twitter-Nutzer [5].

Ansatz Das Ziel der Arbeit ist es, ein Frühwarnsystem für Erdbeben oder andere Umweltereignisse wie Tropenstürmen zu schaffen, welches Nutzer schnell und rechtzeitig über ein Erdbeben informiert, um Schutzmaßnahmen zu treffen. Ein Erdbeben bewegt sich mit 3-7 km/s fort, womit eine 100 Kilometer entfernte Person circa 20 Sekunden Zeit hat, bevor das Erdbeben bei ihr ankommt, wie die Verfasser bemerken.

Datenerfassung Die Autoren betrachten in ihrer Studie jeden Twitter-Nutzer als Sensor. Dieser gibt über seine Beiträge bei Twitter (Tweets) seine Beobachtungen wieder. Ein Tweet ist dabei auf 140 Zeichen begrenzt. Da Twitter als Microblog-Service betrachtet werden kann variieren die Inhalte, welche von Nutzern geteilt werden, stark. Bei der Betrachtung der Daten ist den Verfassern aufgefallen, dass dennoch die meisten Tweets einen Bezug auf ein Ereignis haben. Diese Ereignisse können allerdings von Erdbeben oder Tropenstürmen bis hin zu einem gebrochenen Fußzeh oder neuem Videospiel reichen. Es ist nun also notwendig die Ereignisse zu filtern und zu lokalisieren.

Ereigniserfassung Um ein Ereignis zu identifizieren, müssen zunächst die Tweets ausgewertet werden. Hierfür haben Sakaki et al. eine Support Vector Machine (SVM) derart trainiert, dass sie Schlagwörter wie etwa „earthquake" oder „shaking", die Anzahl der Zeichen und etwaige Hash-Tags eines Tweets filtert und klassifiziert. Dabei ist es wichtig, ebenso den Kontext der Nachricht zu berücksichtigen, denn diese Worte können auch in Nachrichten eingebettet werden, welche keinen Bezug zu einem aktuellen Ereignis haben. Die Autoren bemerken

dabei, dass sehr kurze Nachrichten ein guter Indikator für Aktualität seien können, denn die Tweet-Verfasser sind überrascht und hätten keine Zeit zum Ausformulieren. Die Lokalisierung eines Ereignisses findet entweder über die GPS Koordinaten der Tweets (in der Twitter Smartphone-Applikation integriert) oder über die Ortsangabe der Nutzer bei der Registrierung statt. Damit haben die Autoren es geschafft, die Erfassung von Ereignissen auf das Erfassen eines Objekts und dessen Ortes zu reduzieren. Somit ist es möglich, die Erkenntnisse aus den Tweets mithilfe von Algorithmen und Filtern, wie beispielsweise Partikel-, Kalman- oder Bayes-Filtern, zu verfeinern und zu präzisieren.

Ein weiterer hilfreicher Punkt bei der Ereigniserkennung ist, dass kurze Zeit nach einem Ereignis eine große Menge an Tweets folgen, welche sich auf das Ereignis beziehen. So werden die ersten Nachrichten über ein Erdbeben bereits nach einer Minute vom System der Autoren erkannt. Dabei vermuten sie, dass der Tweet noch schneller geschrieben wurde, da eine Verzögerung durch das Twitter-System sowie die Filterprozedur berücksichtigt werden muss.

Ergebnis Das resultierende Softwaresystem „Toretter“ von Sakaki et al. kann ein Erdbeben über Twitter erkennen und im Zeitraum von 20 Sekunden bis zu einer Minute Nachrichten an registrierte Benutzer per E-Mail verschicken. Auf diesem Wege können sie den Nutzern auch noch weitere Informationen übermitteln, beispielsweise wie man sich bei einem Erdbeben richtig verhält. Damit ist das Frühwarnsystem der Autoren, wie sie anmerken, rund 5 Minuten schneller, als das der Japan Meteorological Agency (JMA). Die JMA kann erst nach zirka sechs Minuten die Zivilbevölkerung mittels Fernseh- und Radioberichten informieren. Außerdem merken die Autoren an, dass sie ihr System bereits ebenfalls erfolgreich auf Tropenstürme ausgeweitet haben. Sakaki et al. haben mit ihrer Software somit ein System geschaffen, welches eine große Hilfe beim Retten von Menschenleben und beim Reduzieren von Schäden sein kann.

4 Social Sensing durch Funknetzwerke und Sensoren

4.1 Social Sensing im Alltag zur Erkennung von Verhaltensmustern

Um Daten von Personen in echten sozialen Netzwerken zu erfassen, beschreiben Nathan Eagle und Alex (Sandy) Pentland in ihrer Arbeit „Reality mining: sensing complex social systems“[4] folgendes Vorgehen:

Ansatz Da moderne Mobiltelefone sowohl Bluetooth als auch GMS beziehungsweise dessen Nachfolger nutzen, können damit Rückschlüsse auf die Position und die Aktivität eines Gerätes gezogen werden. Neben der Technik der Positionsbestimmung über Funkmasten können auch mittels Bluetooth Informationen über die Umgebung eines mobilen Gerätes erlangt werden. Bluetoothfähige Geräte können hierzu in einem Radius von 5 bis 10 Metern gekoppelt werden, um Informationen auszutauschen. Diese Informationen umfassen unter anderem die Bluetooth MAC Adresse (BTID, 12-stellige Hexadezimalzahl), den Gerätenamen und eine dreistellige Zahl, welche den Gerätetyp beschreibt.

Datengewinnung Eagle und Pentland haben zum Erfassen dieser Informationen die Software BlueAware entwickelt, welche im Hintergrund auf Mobilgeräten läuft und die BTIDs speichert. Neben BlueAware haben sie außerdem Bluedar eingesetzt. Dies ist ein Bluetooth-Chip, welcher mit einem WiFi-Sender verbunden ist. Das Bluedar System hat eine Reichweite von bis zu 25 Metern. Es scannt zyklisch nach Bluetoothgeräten und schickt die BTIDs über das Netzwerk zur Auswertung an einen Server.

BlueAware wird in dieser Studie dafür eingesetzt, die soziale Interaktion der Probanden zu erkennen, da es auf den Geräten der Teilnehmer läuft. Bluedar hingegen hat die Aufgabe in Gebäuden nach Geräten zu scannen und so festzustellen, welches Gerät sich wann im Einzugsbereich befindet. Es dient somit als weitere Hilfe zur Positionsbestimmung von Geräten.

Schließlich wurde 100 Probanden ein Nokia 6600 Smartphone überreicht. Das Nokia 6600 ist ein Triband Handy und benutzt das Betriebssystem Symbian 7. Es hat eine 104 MHz CPU, sowie Bluetooth, Infrarot und GPRS. Auf dem Smartphone waren einige Softwareapplikationen vorinstalliert. Die Autoren nennen dabei lediglich die Software ContextPhone[13] und implizieren auch eine Installation von BlueAware.

Darüber hinaus wurde der Bluedar Bluetooth-Scanner in den Büroräumen des MIT Media Laboratory aufgestellt. Außerdem kann auch die wiederkehrende Erkennung stationärer Bluetoothgeräte, wie einem Desktop-PC oder Drucker, Informationen über den aktuellen Aufenthaltsort liefern.

Mit Hilfe der ContextPhone Software können Informationen über den sozialen Kontext des Benutzers erlangt werden. Dazu werden Daten aus Sensoren wie Funkzelle, GPS und Telefongebrauch erfasst. Diese geben Aufschluss über die Interaktion mit anderen Menschen; durch die Anzahl an BTIDs, die erkannt werden, oder die Anzahl an Telefonaten und SMS Nachrichten, die versandt wurden. Anschließend können die Protokolle mittels GPRS, Bluetooth, SMS und MMS versendet werden.

Datensatz Die Daten, welche von Eagle und Pentland erhoben wurden, umfassen Anrufprotokolle, Bluetoothgeräte in der näheren Umgebung, die IDs der aktuellen Funkzelle, Informationen über den Gebrauch von Anwendungen sowie allgemeine Telefondaten wie Akkuladestand und Inaktivität. Mit Hilfe der Smartphones und Bluedar wurden so über ein akademisches Jahr die Daten der 100 Teilnehmer erfasst. Diese können unter [8] eingesehen werden.

Studie Um zu zeigen, dass durch die gesammelten Daten Verhaltensmuster zu erkennen sind, wurden zunächst drei Klassifikationen (zuhause, auf der Arbeit und „anderswo“ (elsewhere)) für die Aufenthaltsorte geschaffen.

In vielen Fällen ist es durch klare, wiederkehrende Aktivitätsmuster des Teilnehmers leicht die Daten zu interpretieren sowie Vorhersagen über das Verhalten zu treffen. Schwieriger wird dies bei Menschen die keine klar abgrenzbaren Aktivitätsmuster zeigen oder bei komplexeren Routinen. Hier versuchen Eagle und Pentland die Zahl der vorhersehbaren Handlungen im Leben eines Menschen

an die informationstheoretische Entropie zu knüpfen. In der Informationstheorie korrespondiert die Zufälligkeit eines Signals schließlich mit seiner Entropie. Dies bedeutet, dass aus einer höheren Entropie eine höhere Zufälligkeit des Signals folgt. Deswegen sind die Handlungen von Menschen mit hoher Entropie, so die Autoren, schwieriger vorherzusagen als die Handlungen von Menschen mit niedriger Entropie. Dabei kann die Entropie nicht nur für die Vorhersagbarkeit einzelner Personen benutzt werden. In der Arbeit wird sogar festgestellt, dass ganze Personengruppen durch unterschiedliche Entropie-Niveaus zu identifizieren sind. Dies beruht darauf, dass sich Personengruppen meist durch gemeinsame Charakteristika abgrenzen lassen.

Erkenntnisse Erfasst man nun also die Positionen zu bestimmten Zeiten, die Aktivitäten des Smartphones, sowie die Nähe zu anderen Benutzern kann man damit Rückschlüsse auf das Verhalten in komplexen sozialen Systemen ziehen. Dafür können Daten von Personengruppen, die eng zusammen arbeiten, dazu benutzt werden, Korrelationen zwischen den auf dem Telefon gesammelten Daten und der Position der Mitglieder herzustellen. Die damit gewonnen Einblicke können Aufschlüsse über das Verhalten von großen Personengruppen geben. Da das Verhalten von solchen Gruppen Ähnlichkeiten mit dem epidemiologischen Verhalten einer Gesellschaft hat, könnten die Erkenntnisse auch zur Entwicklung besser Modelle für die Ausbreitung von Krankheiten beitragen.

Durch die gesammelten Daten war es den Verfassern auch möglich, mit einer 90 prozentigen Wahrscheinlichkeit, vorherzusagen wann Personen mit niedriger Entropie mit anderen Personen interagieren. Aber auch mit weniger Daten ließen sich die sozialen Strukturen erkennen. Um in den Interaktionen der Probanden Muster zu erkennen wurde ein Gaussian Mixed Model (GMM) trainiert. Die Trainingsdaten wurden in einer Umfrage unter den Teilnehmern erhoben. Angegeben werden konnte mit welchen Personen man auf der Arbeit Zeit verbringt, wer zu seinem Bekanntenkreis und wer zum Freundeskreis gehört. Mit dem trainierten Modell konnte für Personen mit niedriger Entropie diese Personengruppen ebenfalls mit einer 90 prozentigen Wahrscheinlichkeit bestimmt werden.

Eagle und Pentland stellen mit ihrer Arbeit also dar, dass es effektive Ansätze gibt, Verhaltensmuster im Alltag zu erkennen.

4.2 Social Sensing zur Softwareanpassung am Beispiel eines Fahrzeugassistenzsystems

Die Arbeit von Ali et al. „Social Sensing: When Users Become Monitors“[2] beschäftigt sich damit, wie die Anpassungsfähigkeit von Software durch den Benutzer als zusätzliche Informationsquelle in Form eines Beobachters gesteigert oder gewährleistet werden kann. Von solchen anpassungsfähigen (Software-) Systemen wird erwartet, dass sie sich an die wechselnden Bedingungen ihrer Umgebung anpassen können. Dabei soll durch eine schnelle und effektive Anpassung vor allem gewährleistet werden, dass die Anforderungen der Benutzer erfüllt werden.

Des Weiteren sind insbesondere effektive Anpassungen für Softwaresysteme, aufgrund mangelnder Technologien zur Informationsbeschaffung, häufig schwierig vorzunehmen.

Motivation Der Adaptionszyklus sieht dabei folgende Schritte vor: Analyse von Veränderungen in der Systemumgebung, Auswahl einer Anpassung, Anwendung der Optimierung und letztendlich der Entscheidung, in wie weit die Anpassung erfolgreich war. Wobei sich „erfolgreich“ nicht zwingend auf die Bedürfnisse des Nutzers beziehen muss. Die Überwachung dieses Zyklus ist durch rein technische Methoden nicht immer erfolgreich oder möglich. Der Benutzer kann dann in der Überwachung dieses Zyklus eingesetzt werden und somit zusätzliche Einschätzungen zu den Anpassungsschritten geben. Er kann diese dadurch aktiv beeinflussen.

Die Punkte, in denen der Benutzer Eingriffe in das System vornehmen, oder es mit zusätzlichen Informationen beliefern kann, sind die Bereiche der Kontext- und Qualitätsattribute. Dabei entspricht ein Kontextattribut einem Baustein, welcher direkte Auswirkungen auf die Validität einer Entscheidung des Systems hat. Qualitätsattribute hingegen beeinflussen die Korrektheit einer Lösung nicht direkt. Hierunter sind eher Eigenschaften in der Repräsentation der Daten oder in der Kommunikation zwischen dem Anwender und dem System zu verstehen.

Anwendungsbeispiel - Fahrzeugassistenzsystem Am Beispiel des Fahrzeugassistenten bedeutet dies konkret, dass das Verkehrsaufkommen in einem bestimmten Gebiet ein Kontextattribut darstellt. Dieses Attribut hat einen direkten Einfluss auf die Entscheidung zu welchem Parkplatz der Fahrer geführt werden soll. Das Verkehrsaufkommen kann aber, wegen fehlenden Informationssystemen, nicht zuverlässig bestimmt werden. Allerdings besteht die Möglichkeit, dass eine Gemeinschaft an Fahrern aus dem bestimmten Gebiet, Informationen dazu bereitstellen. Somit ist es für den einzelnen Anwender möglich von der Masse der Anwender zu profitieren, denn das System kann, aufgrund der gelieferten Informationen, das Verkehrsaufkommen besser abschätzen und eine Lösung bereitstellen, welche den Bedürfnissen des Benutzers gerechter wird. Um dies zu erreichen, muss es aber auch möglich sein, dass die Fahrer und das System kommunizieren können. Die Auswahl des Kommunikationsweges stellt ein Qualitätsattribut dar. Hier kann die Methode, je nach unterschiedlichen Anwendervorlieben oder Situationen, von einer Sprachführung bis zu einer Straßenansicht reichen.

Erkenntnisse Weiter können Anwender auch dazu beitragen, dass Qualitätsattribute, welche von den Entwicklern zum Zeitpunkt der Entwicklung nicht erkannt oder berücksichtigt wurden, in das Softwaresystem integriert werden. Auch hier können folgende Anwender demnach erneut von dem Bedürfnis der Masse profitieren. Dies lässt den Schluss zu, dass die Wahrscheinlichkeit, dass die Bedürfnisse eines einzelnen Anwenders erfüllt werden, recht hoch ist, wenn

eine Menge an Personen eine bestimmte Umsetzung eines Qualitätsattributs bevorzugt. Somit besteht die Möglichkeit, dass die Benutzerzufriedenheit durch die Benutzer selbst optimiert werden kann.

Damit wird die Einbeziehung der Nutzer, wie Maalej et al. [9] erörtern, zu einem wichtigen Gesichtspunkt in der Entwicklung einer Software. Der Entwicklungsprozess entwickelt sich von einer rein technischen hin zu einer sozialen Betrachtungsweise. Diese erfordert, dass der Benutzer von Anfang an in die Softwareanpassung eingeplant und als Teil der Systemausführung wahrgenommen wird. Um dies zu ermöglichen kann Social Sensing als Entwicklungstechnik eingesetzt werden, da sie diese Schnittstellen und Möglichkeiten definiert. Denn zum einen kann ein einzelner Benutzer durch seine Wahrnehmung Einfluss auf das System als Ganzes nehmen, zum anderen profitiert im Gegenzug die Gesamtheit der Benutzter von diesen Informationen. So kann der von Ali et al. verfolgte Ansatz, die Benutzerzufriedenheit zu erhöhen, gewährleistet werden. Die Autoren beschreiben weiter das Ziel, die technischen Grenzen von solchen Systemen durch die Hinzunahme des Benutzers als Sensor zu verschieben. Auch etwaige Unsicherheiten respektive Unvollständigkeiten in einem Systementwurf könnten durch die dargestellten Möglichkeiten gelöst werden. Damit spielt Social Sensing eine entscheidende Rolle in der Beeinflussung der genannten Faktoren.

4.3 Social Sensing zur Messung der Beeinflussung der Fitness, Ernährung und Gesundheit durch soziale Kontakte

Die Arbeit von Madan et al. „Social Sensing: Obesity, Unhealthy Eating and Exercises in Face-to-Face Networks“[10] versucht, mithilfe von Social Sensing, zu ergründen, inwiefern diese Interaktion das eigene Verhalten bei Ernährungsgewohnheiten und Sportaktivitäten beeinflusst. Das Interesse an der Thematik ergibt sich aus der zunehmenden Anzahl von übergewichtigen und adipösen Menschen [14]. Dies ist vor allem auf ungesunde Ernährung sowie Veränderungen im Lebensstil zurückzuführen.

Ansatz Diese Studie nutzt Social Sensing um den Einfluss persönlicher Kontakte auf Gesundheitsbewusstsein und einen gesunden Lebensstiel zu beobachten. Dazu wird versucht mit den, durch Sensoren gewonnenen, Daten die Veränderungen des BMI (Body Mass Index) zu erklären. Die Erkenntnisse könnten dazu dienen bessere Modelle und Maßnahmen gegen die Ausbreitung des kollektiven Übergewichts zu entwickeln.

Datenerfassung Um die Daten von 70 freiwilligen Studenten aus einem Wohnheim zu erheben wurden diese nach ihren Ernährungsgewohnheiten, ihrem sozialen Umfeld und ihrem Gesundheitszustand befragt. Auf diesen Daten beruhend wurde der BMI ermittelt. Die Befragung wurde während des akademischen Jahres 2009 im März, April und Juni durchgeführt. Um nun die Rolle des sozialen Umfeldes zu messen mussten die Studenten mit Sensoren ausgestattet werden.

Dafür wurde an jeden Teilnehmer ein Smartphone verteilt. Die auf den Smartphones installierte Software ermöglicht es, mithilfe von Bluetooth, andere Geräte in der Umgebung zu erkennen. Dabei wird ein Hash-Wert der MAC Adresse des gefundenen Gerätes gespeichert. Weiter wurden mittels WLAN umliegende WLAN-Sender gesucht. Hier wurde ebenfalls ein Hash-Wert der MAC Adresse des Zugriffspunkts gespeichert. Beide Scan-Vorgänge fanden zyklisch alle 6 Minuten statt. Darüber hinaus wurden auch die relevanten Informationen für Anrufe und SMS gespeichert.

Datensatz Um nun Aufschluss über die Rolle von Freunden, Bekannten und persönlichen Kontakten auf die Gewichtsveränderung der Teilnehmer zu erhalten, wurden die Bluetoothverbindungen folgendermaßen unterteilt:

- „Total bluetooth exposure“ - Gesamte Verbindung mit anderen Bluetoothgeräten.
- „Late-Night & Early-morning Bluetooth exposure“ - Verbindung mit anderen Geräten zwischen 21 Uhr und 9 Uhr am Folgetag.
- „Weekend Bluetooth exposure“ - Verbindung mit anderen Geräten am Wochenende.
- „Total Phone and SMS exposure“ - Gesamte Kommunikation über Telefonie und SMS.
- „Weekend Phone and SMS exposure“ - Kommunikation über Telefonie und SMS am Wochenende.

Erkenntnisse Es konnte somit festgestellt werden, dass 17% der Veränderungen des BMI der Teilnehmer sich darauf zurückführen ließen, dass die Personen engen Kontakt mit adipösen Bekannten in „total bluetooth exposure“ und „Late-Night & Early-morning Bluetooth exposure“ hatten. Insgesamt 25% der BMI-Zunahme war auf Kontakt mit übergewichtigen und adipösen Personen im selben Zeitraum zurückzuführen. Die größte Korrelation in der Veränderung des BMI konnte dabei bei der Personengruppe beobachtet werden, welche zwei Kilogramm oder mehr in der Zeit von März bis Juni zunahmen. Hier konnten die Veränderungen des BMI eines Teilnehmers, welcher mit Personen dieser Gruppe Kontakt pflegt, mit ungefähr 35% erklärt werden. Neben der Zunahme des BMI wurde die Veränderung in der sportlichen Aktivität und einer gesunden Ernährung gemessen. Hier konnten 17% der Veränderung in den abhängigen Variablen auf den Kontakt mit adipösen Personen, sowie 23% auf den Kontakt mit Adipösen und Übergewichtigen zurückgeführt werden. Bemerkenswert ist, so die Autoren, dass die oben genannten Korrelationen nicht für Personen zu gelten scheinen, welche im Untersuchungszeitraum zwei oder mehr Kilogramm abgenommen haben. Es gebe also andere Dynamiken im Zusammenhang mit der positiven Beeinflussung des Verhaltens.

Mit Hilfe dieser Studie konnten die Autoren Erkenntnisse darüber gewinnen, welchen Einfluss das Verhalten von Menschen mit ungesunden Ernährungseinstellungen, geringer Bewegung und Übergewicht auf andere Personen hat.

5 Grenzen des Social Sensing

5.1 Technische Grenzen

Positionsbestimmung In der Arbeit von Eagle und Pentland [4] wird auf folgendes Problem zur Lokalisierung von Nutzern eingegangen: In urbanen Gebieten gibt es eine Vielzahl an Funkmasten, welche sich häufig überschneiden. Dadurch kann nicht mit Sicherheit bestimmt werden, in welchen Funkmast sich ein Gerät an einem bestimmten Ort einwählt. Des Weiteren kann die Funkqualität in Gebäuden beeinträchtigt sein, so dass ein Mobilfunkgerät keinen Empfang hat. Die Positionsbestimmung mittels GPS funktioniert durch technische Gegebenheiten nur im Freien zuverlässig. Auch die Reichweite von Bluetooth kann in geschlossenen Räumen stark beeinträchtigt sein und somit auch nicht immer eine zuverlässige Positionierung ermöglichen. Aus diesen Gründen kann es hilfreich sein, stationäre Geräte wie Desktop-PCs oder Drucker, die ebenfalls über Bluetooth verfügen, zur präzisieren Standortbestimmung mit einzubeziehen. Dennoch muss auch hier der genaue Standort dieser Geräte bestimmt werden bevor eine Datenauswertung möglich ist.

Bluetooth Wie Eagle und Pentland aufführen [4] kann die Reichweite von bis zu 10 Metern von Bluetooth, sowie die Fähigkeit der Funkwellen durch Wände zu dringen, fehlerhafte Ergebnisse liefern, wenn man so versucht die Nähe zu anderen Personen zu bestimmen. Weiter führen sie an, dass es eine Wahrscheinlichkeit von 1 bis 3% gibt, mit der ein Gerät während eines Scans nicht erkannt wird. Bei entsprechend großen Datensätzen könnte diese die Ergebnisse durchaus verfälschen. Da die Bluetoothscans außerdem nur alle 5 Minuten durchgeführt werden, werden kurze Treffen, wie beispielsweise ein kurzes Gespräch an der Kaffeemaschine, unter Umständen nicht erfasst.

Komplexität Nach Aggarwal [1] erfordern viele Anwendungen von Social Sensing eine große Menge an Daten in Echtzeit. Diese werden stellenweise von unterschiedlichen Sensoren gleichzeitig erfasst, beziehungsweise benötigt. Damit ist eine große Herausforderung an den Entwurf und die Entwicklung der Anwendung gestellt, denn sie muss gewährleisten, dass die Daten gleichzeitig erfasst und korrekt übermittelt werden können.

Datenkorruption Daten werden bei einigen Ansätzen auf Flash-Speichern, wie SD-Karten oder dem internen Speicher von Smartphones, zwischengespeichert [4,10]. Allerdings haben Flash-Speicher nur eine begrenzte Anzahl an Schreib-Lese-Zyklen, wodurch es - wie in der Arbeit von Eagle und Pentland [4] - zu Datenverlusten kommen kann.

Fehlerhafte Daten Alle Probleme, die beim Erfassen von Daten auftreten können, wie unzulängliche Nutzerinformationen, Fehler bei der Datenübertragung oder Probleme in den Sensoren selbst, können zu fehlerhaften Datensätzen führen [1]. Diese wiederum können zu falschen Schlüssen oder Erkenntnissen führen.

5.2 Der Faktor Mensch als Grenze

Menschen als Sensor Menschen sind in ihrem Verhalten variabel. Darum beschreiben Sakaki et al. in ihrer Arbeit, dass einige menschliche Sensoren aktiver als andere sind [11]. Außerdem setzt die Funktionsfähigkeit der Menschen in bestimmten Situationen aus. D ies ist beispielsweise der Fall, wenn sie schlafen oder arbeiten. Weiter können Menschen auch Vergangenes reflektieren, wie ein Erdbeben, welches einige Tage zurück liegt, so die Autoren. Damit geben sie nicht zwingend Auskunft über neue Ereignisse. Ebenso könnten menschliche Sensoren auch zu falschen Alarmen neigen. Dies ist insbesondere dann problematisch, wenn die Sensoren, wie im Beispiel von Twitter, nicht unabhängig sind. Das selbe Problem wird von Ali et al. beschrieben [2].

Menschliche Unzulänglichkeit Da die Testpersonen zur Überwachung meistens mit einem Sensor in Form eines Smartphones ausgestattet wurden [4,10], mussten sie dieses immer bei sich tragen. Allerdings erkennen Pentland und Eagle [4], dass zum Beispiel zur Mittagspause das Handy am Arbeitsplatz liegen gelassen wurde. Weiter führen sie an, dass der Nutzer in bestimmten Situationen auch die Möglichkeit hatte sein Gerät auszuschalten. Diese Möglichkeit wird sich Nutzern in einer meistens Studie bieten.

Ali et al. [2] führen weiter an, dass Menschen auch unterschiedliche Einschätzungen der selben Situation liefern können. Somit kann ein bestimmtes Ereignis unterschiedlich bewertet werden. Die Daten sind also nicht unbedingt objektiv.

5.3 Privatsphäre und Datenschutz

Wie von Aggarwal [1], aber auch Ali et al.[2] und Madan et al. [10], aufgeführt wird, spielt die Privatsphäre eine große Rolle. Die Menschen benötigen Vertrauen, so die Autoren, um überhaupt erst ihre Daten zu teilen. Dieses Vertrauen sei nur zu bekommen, wenn man die gewonnenen Daten vertrauenswürdig behandelt. Beispielsweise wählen Madan et al. in ihrer Arbeit den Ansatz, die Daten, welche Rückschlüsse auf Personen - wie die Bluetooth MAC-Adresse - zulassen, zu hashen und somit zu anonymisieren. Aggarwal hingegen wendet hier ein, dass der GPS-Sensor, selbst mit anonymisierten Daten, Rückschlüsse auf die Person zulässt. Von ihm wird aufgeführt, dass beispielsweise durch PoolView[6] Sensordaten gewonnen werden können, während die Privatsphäre gesichert bleibt.

6 Fazit

Es wurde zunächst die technische Entwicklung aufgezeigt, welche Social Sensing begünstigt hat. Anschließend wurde eine Arbeit beschrieben, welche Social Sensing im Kontext der sozialen Netzwerke einsetzt, um Nutzer vor Erdbeben zu warnen. Wie Sakaki et al. [11] am Ende ihrer Arbeit beschreiben, versuchen sie gerade ihre Anwendung weiter zu optimieren, so dass sie auch Ereignisse erkennen kann, welche nicht nur zu einer bestimmten Zeit an einem bestimmten Ort

auftreten, wie Erdbeben oder Tropenstürme (wie von den Autoren angenommen wird), sondern häufiger. Sie sprechen hier von multiplen Ereignissen wie Staus oder Regenbögen.

Mit dieser Arbeit wurde gezeigt, dass Soziale Netzwerke durchaus dafür eingesetzt werden können, soziale Kontexte zu erkennen, beziehungsweise diese in einen Mehrwert umzuwandeln. Zum einen kann dafür die Nutzergemeinschaft, wie in der betrachteten Arbeit, als Sensorsystem aufgefasst werden. Zum anderen wäre es aber auch denkbar, die Beziehungen, welche in einem Sozialen Netzwerk - wie Facebook - erkennbar sind, dafür zu nutzen, auch den sozialen Kontext außerhalb dieses Netzwerkes besser zu modellieren und erkennen zu können.

Im Anschluss wurden drei Arbeiten präsentiert, welche alle Sensor- und Funktechnologien eingesetzt haben, um die sozialen Kontexte von Menschen zu erfassen oder Informationen über diese zu erlangen. Auch hier gibt es eine Vielzahl an Anwendungsgebieten von Social Sensing. Diese reichen vom rein wissenschaftlichen Bereich, wie beispielsweise dem Erkennen von Verhaltensmustern, zur Entwicklung besserer Modelle für komplexe Sozialsysteme [4] bis hin zu kommerzielleren Zwecken, wie beispielsweise dem Ansatz von Ali et al. [2], um die Nutzerzufriedenheit und die Validität von Softwarelösungen zu verbessern.

Ich bin der Meinung, dass die technische Entwicklung im Bereich der mobilen Geräte und Sensoren noch nicht abgeschlossen ist. Es ist vorstellbar, dass weitere Sensoren in mobilen Geräten die Datenmenge steigern und so noch mehr Felder mittels Social Sensing betrachtet werden können. Die Entwicklung in den Wearables macht es bereits heute möglich Vitalwerte, wie Blutdruck oder Puls, zu messen und zu beobachten. Damit wäre beispielsweise eine Messung des Stresses in bestimmten sozialen Kontexten möglich, um daraus weitere Rückschlüsse auf das menschliche Sozialverhalten zu erlangen. Außerdem wird die Dichte der Daten weiter erhöht, da die Nutzer zum Beispiel ihre Smartwatch vermutlich häufiger bei sich tragen als ihr Smartphone. Auch die zunehmende Vernetzung mit Alltagsgegenständen, wie Kühlschrank oder Heizung, wird die Menge der zu erfassenden Daten weiter ansteigen lassen.

Werden allerdings immer vielfältigere Daten erfasst, bleibt es dem Menschen nicht immer selbst überlassen, welche Daten er kommuniziert. Es setzt ein hohes Maß an technischem Verständnis voraus zu erkennen, welche Informationen andere schon alleine mit den GPS- und Bluetooth-Sensordaten erlangen können. Außerdem wird die Gesellschaft durch Datenskandale wie die NSA Affäre zunehmend für den Bereich der Privatsphäre und des Datenschutzes sensibilisiert.

Weiterhin muss berücksichtigt werden, dass die Arbeiten von Pentland und Eagle [4] sowie von Madan et al. [10] zur Erfassung des sozialen Kontextes auf das Erkennen von Bluetoothgeräten in der Nähe setzen. Dies lässt aber vermutlich nur beschränkt Rückschlüsse auf den tatsächlichen sozialen Kontext zu, denn es wird vorausgesetzt, dass Bluetooth bei den sozialen Kontakte ebenfalls aktiviert ist. Aus Sicherheitsbedenken oder Energiespargründen ist es aber durchaus nachvollziehbar und häufig vertreten, dass Bluetooth deaktiviert ist. Somit können diese Interaktionen nicht erfasst werden. Ich denke, bezogen auf meine eigenen

Erfahrungen, dass ein deaktiviertes Bluetooth sogar eher die Regel als die Ausnahme darstellt.

Es müssten also auch alternative Wege zur Messung des sozialen Kontextes erprobt werden um die Messungen zu präzisieren. Dafür können sich andere Datenquellen, wie beispielsweise die Kamera mit Gesichtserkennung, eigenen sofern das Mobilgerät nicht in einer Tasche steckt. Auch weitere Sensoren, wie Infrarot oder das Modellieren eines Access Points, könnten neben Bluetooth noch zur Erkennung von Personen beziehungsweise Geräten in der Nähe geeignet sein. Dafür muss Social Sensing aber weiter genutzt und durch Forschung und Einsatz in der Wirtschaft kontinuierlich verbessert werden. Die Risiken im Datenschutz und Bedenken in der Privatsphäre sollten keinen Grund darstellen, auf die immensen Hilfestellungen und Vorteile zu verzichten, welche Social Sensing bieten kann.

Literatur

1. Aggarwal, C.C., Abdelzaher, T.F.: Social sensing. In: Aggarwal, C.C. (ed.) Managing and Mining Sensor Data, pp. 237–297. Springer (2013), http://dblp.uni-trier.de/db/books/collections/mmsd2013.html#AggarwalA13
2. Ali, R., Solis, C., Salehie, M., Omoronyia, I., Nuseibeh, B., Maalej, W.: Social sensing: When users become monitors. In: Proceedings of the 19th ACM SIGSOFT Symposium and the 13th European Conference on Foundations of Software Engineering. pp. 476–479. ESEC/FSE '11, ACM, New York, NY, USA (2011), http://doi.acm.org/10.1145/2025113.2025196
3. Bitcom: Nicht ohne mein Smartphone. http://www.bitkom.org/de/presse/81142_79922.aspx (2014), [Online; 17. Juni 2015]
4. Eagle, N., Pentland, A.S.: Reality mining: sensing complex social systems. Personal Ubiquitous Comput. 10, 255–268 (March 2006), http://dx.doi.org/10.1007/s00779-005-0046-3
5. eMarketer: Japan, India Boast Largest Twitter Audiences in APAC. http://www.emarketer.com/Article/Japan-India-Boast-Largest-Twitter-Audiences-APAC/1011917, [Online; 14. September 2015]
6. 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
7. Grünerbl, A., Muaremi, A., Osmani, V., Bahle, G., Ohler, S., Troster, G., Mayora, O., Haring, C., Lukowicz, P.: Smartphone-based recognition of states and state changes in bipolar disorder patients. IEEE Journal of Biomedical and Health Informatics 19(1), 140–148 (2015)
8. Lab, M.H.D.: Reality Commons. http://reality.media.mit.edu, [Online; 9. Juni 2015]
9. Maalej, W., Happel, H.J., Rashid, A.: When users become collaborators: Towards continuous and context-aware user input. In: Proceedings of the 24th ACM SIGPLAN Conference Companion on Object Oriented Programming Systems Languages and Applications. pp. 981–990. OOPSLA '09, ACM, New York, NY, USA (2009), http://doi.acm.org/10.1145/1639950.1640068

10. Madan, A., Moturu, S.T., Lazer, D., Pentland, A.S.: Social sensing: obesity, unhealthy eating and exercise in face-to-face networks. In: Wireless Health 2010. pp. 104–110. ACM (2010)
11. Sakaki, T., Okazaki, M., Matsuo, Y.: Earthquake shakes twitter users: Real-time event detection by social sensors. In: Proceedings of the 19th International Conference on World Wide Web. pp. 851–860. WWW '10, ACM, New York, NY, USA (2010), http://doi.acm.org/10.1145/1772690.1772777
12. Statista2015: Anzahl der Mobilfunkanschlüsse weltweit von 1993 bis 2015 (in Millionen). http://de.statista.com/statistik/daten/studie/2995/umfrage/entwicklung-der-weltweiten-mobilfunkteilnehmer-seit-1993/ (2015), [Online; 2. Juni 2015]
13. Universität Helsinki, L.f.C.: ContextPhone. http://www.cs.helsinki.fi/group/context/ (2005), [Online; 2. Juni 2015]
14. WHO: Obesity and overweight. http://www.who.int/mediacentre/factsheets/fs311/en/ (2015), [Online; 22. Juni 2015]

Interaktion in Datenanalyse und Augmented Reality: Ein Vergleich

Daniel Sachsenmaier*

Betreuer: Matthias Berning†

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

Zusammenfassung. Da es immer schwerer wird sich in den riesigen Datenmengen zurecht zu finden gibt es viele verschiedene Formen der Datenanalyse. Die Formen der Datenanalyse bringen auch verschiedene Arten der Visualisierung mit sich. Ebenso bringt die Datenanalyse auch verschiedene Arten der Interaktion mit. In diesem Paper soll ein Überblick über die verschiedenen Visualisierungen in der Datenanalyse und Augmented Reality gegeben werden. Es wird aufgezeigt welche Formen der Interaktion es für Benutzer gibt. Interaktionen können in der Datenanalyse und auch der Augmented Reality eingesetzt werden um Sensor- und Statistikdaten näher zu untersuchen.
Des Weiteren wird untersucht ob Interaktionen aus der Datenanalyse auch in der Augmented Reality zu finden sind, dazu wurden die einzelnen Interaktionen miteinander verglichen.
Das Ergebnis zeigt, dass viele Interaktionen die es in der Datenanalyse gibt auch in ähnlicher Art in Augmented Reality angewendet werden. Es zeigt ebenso, dass einige Interaktionen die es nur in der Datenanalyse gibt, mit den Mitteln der Augmented Reality umsetzbar wären.

Schlüsselwörter: Visualisierung, Datenanalyse, Augmented Reality, Sensordaten, Interaktion, Vergleich

1 Einleitung

In der heutigen Zeit wird es immer schwerer Messdaten zu durchsuchen oder zu bearbeiten. In der folgenden Arbeit, wenn von Daten die Rede ist, sind damit Sensor- und Statistikdaten gemeint. Dadurch, dass Sensoren immer kleiner und günstiger werden, und allgegenwärtig sind, produzieren diese eine regelrechte Datenflut. Will man mit diesen Daten arbeiten muss man sich einen Überblick verschaffen und die Daten eingrenzen oder übersichtlich visualisieren.
Sowohl in der Datenanalyse als auch in Augmented Reality (erweiterte Realität) gibt es verschiedene Techniken um Daten darzustellen und mit ihnen zu interagieren. Da Augmented Reality ein neueres Werkzeug ist welches auch zur Datenanalyse eingesetzt werden kann liegt es nahe Zusammenhänge zur allgemeinen Datenanalyse zu suchen. Dies ist so zu verstehen, dass sowohl in Augmented

Reality als auch Datenanalyse Sensordaten visualisiert und untersucht werden können. In der Augmented Reality jedoch kann die Visualisierung und Interaktionen mit den Daten, in Echtzeit vor Ort durchgeführt werden [11]. Aus diesem Grund ist es interessant zu untersuchen wie stark Datenanalyse und Augmented Reality zusammenhängen.
In dieser Arbeit soll ein Vergleich der Interaktionsmethoden durchgeführt werden. Es wird geprüft wie stark die Methoden zusammenhängen und ob es Methoden in der Datenanalyse gibt die in Augmented Reality noch nicht umgesetzt sind aber umgesetzt werden könnten.
Daten zu visualisieren, vor Allem wenn es sich um sehr große Datensätze handelt, ist nicht immer trivial. Um trotzdem Visualisierungen zu realisieren gibt es verschiedene Darstellungsarten. Die Darstellungsarten richten sich dabei nach den verschiedenen Datentypen wie zum Beispiel zwei- oder mehrdimensionale Daten. Eine genauere Erläuterung der Visualisierungsarten findet sich in Abschnitt 2.1.
Um mit Daten zu arbeiten gibt es mehrere Möglichkeiten. Die Naheliegenste ist die Datenanalyse die in Abschnitt 2.2 beschrieben wird. Hier gibt es mehrere Arten der Datenanalyse, die sich meist auf die Art der Darstellung der Daten beziehen.
Neben der „normalen“ Datenanalyse gibt es noch die Datenanalyse mit Hilfe der Augmented Reality. Die Augmented Reality ist keine reine Form der Datenanalyse, wird aber als Werkzeug dazu genutzt. In der Augmented Reality kann ebenfalls mit Daten gearbeitet werden indem sie visualisiert werden, was in Abschnitt 2.3 gezeigt wird. Hier werden kurz die verschiedenen Arten der Darstellung vorgestellt. In der Augmented Reality wird die Darstellung nach Displays unterschieden. Des Weiteren wird in sogenannte Dimensionen unterschieden in denen sich ein Nutzer zum darstellenden Display befindet.
Um Daten effizienter zu bearbeiten gibt es verschiedene Formen der Interaktion mit den Daten. Zum Beispiel gibt es die Möglichkeit bestimmte Bereiche näher zu betrachten, um sich in einem großen Datensatz zurechtzufinden. Die verschiedenen Interaktionsarten der Datenanalyse werden in Abschnitt 3.1 beschrieben.
Auch in der Augmented Reality kann mit den Daten und deren Visualisierung interagiert werden. Dazu gibt es, wie auch schon in der Datenanalyse, verschiedene Interaktionstechniken wie zum Beispiel die direkte oder Okklusionsselektion, die in Abschnitt 3.2 erläutert werden.
Da sowohl in der Datenanalyse als auch in der Datenanalyse mit Hilfe der Augmented Reality mit Daten interagiert wird, liegt es nahe Verbindungen der Interaktionstechniken aus beiden Bereichen zu suchen. In Abschnitt 3.3 werden die verschiedenen Techniken gegenübergestellt und auf vorhandene Gemeinsamkeiten untersucht.
Abschließend werden die Erfahrungen und Ergebnisse aus dem Vergleich in Abschnitt 4 bewertet und zusammengefasst.

2 Grundlagen

Um den Vergleich der Interaktionen in Datenanalyse und Augmented Reality durchzuführen ist es nötig ein paar wenige Grundlagen zu kennen. Zuerst werden in Abschnitt 2.1 die verschiedenen Möglichkeiten der Visualisierung aufgezeigt. Es werden die verschiedene Dimensionen von Daten erläutert. Außerdem werden die verschiedenen Visualisierungsarten für die Dimensionen gezeigt.
Dann wird in Abschnitt 2.2 beschrieben was eine Datenanalyse auszeichnet. Es wird aufgezeigt welche Formen und Methoden der Datenanalyse existieren.
Zuletzt wird noch, in Abschnitt 2.3, Augmented Reality beschrieben. Die verschiedenen Möglichkeiten der Darstellungen mit Hilfe von Displays werden, unter Beachtung der Dimensionen von Betrachter und Display, kurz aufgezeigt. Ebenso wird kurz erläutert wieso es sinnvoll ist Augmented Reality einzusetzen um eine Datenanalyse durchzuführen.

2.1 Visualisierung:

Meistens sind die Datensätze die visualisiert werden sollen sehr groß. Um Daten zu Visualisieren gibt es verschiedene Arten. Die Art in der Daten visualisiert werden hängt von der Art der Daten ab, wie von Keim, D.A. in „Information visualization and visual data mining“ beschrieben wird [14].
In Abb. 1 ist dargestellt wie die verschiedenen Visualisierungstechniken mit Hil-

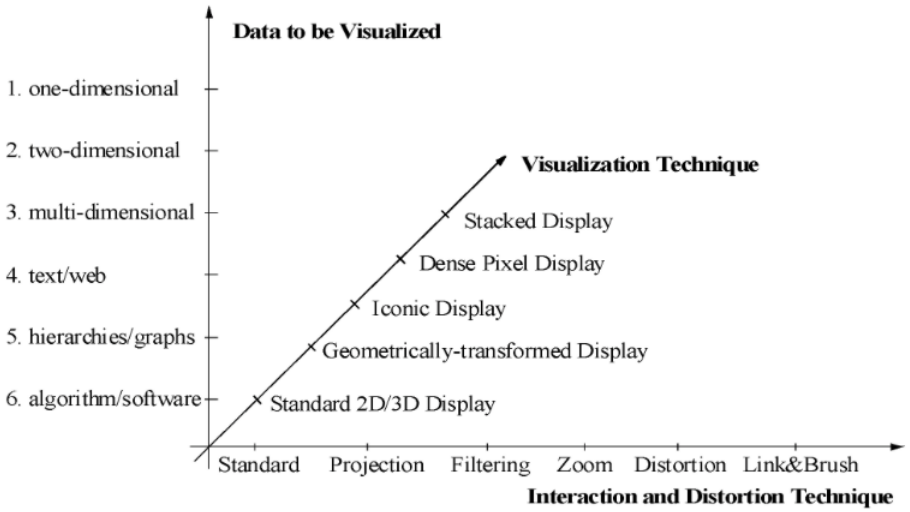


Abb. 1. Übersicht über die vorhandenen Dimensionen von Daten und deren Darstellungsarten [14].

fe von drei Kategorien klassifiziert werden können [14]: 1. die zu visualisierenden

Daten, 2. die genutzte Visualisierungstechnik und 3. die eingesetzte Interaktionstechnik.
Die zu visualisierenden Daten werden nach ihrer Dimension unterschieden. Die Dimension beschreibt die Anzahl der Attribute die ein Datenobjekt besitzt. Es existieren demnach sechs verschiedene Datenarten.
Um die Daten zu visualisieren sind geeignete Visualisierungstechniken notwendig. In der Abbildung sind fünf grundlegende Techniken dargestellt. Bei den Techniken handelt es sich um die Standard 2D/3D Darstellung wie z.B. Balkendiagramme. Eine geometrisch transformierte Darstellung wie z.B. die „Parallel Coordinate Technique“ [12]. Die ikonische Darstellung bei der die Daten mit Hilfe von sogenannten „Icons“, wie z.B. Sternen, dargestellt werden. Mit Hilfe der Dense Pixel (dichte Pixel) Darstellung werden die Dimensionswerte auf verschiedene Farbpixel abgebildet und dann den Farben entsprechend sortiert. Zuletzt noch die gestapelte Darstellung bei welcher z.B. Koordinatensysteme ineinander verschachtelt werden um mehrdimensionale Daten darzustellen.
Die dritte und letzte Kategorie, um die Techniken zu klassifizieren, sind die Interaktionstechniken. Die, in der Abbildung, dargestellten Techniken sind spezielle Techniken die die Techniken aus Abschnitt 3.3 für bestimmte Visualisierungstechniken nutzen. Für den Vergleich wurden deshalb nicht die speziellen sondern die einfachen, einzelnen Techniken verwendet. Bei der Projection soll der Abstraktionsgrad dynamisch geändert werden um multidimensionale Daten zu untersuchen. Wie der Name schon sagt wird Filtering genutzt um einen bestimmten Teil eines großen Datensatzes herauszustellen (vgl. Filter Abschnitt 3.1). Mit Hilfe von Zooming lässt sich ebenfalls wie durch Projection der Abstraktionsgrad verändern. Bei Zooming jedoch ganz einfach mit mehr oder weniger detaillierten Informationen (vgl. Abstract/Elaborate Abschnitt 3.1). Mit Distortion sollen ausgewählte Teile von Daten mit vielen Details und der Rest mit wenigen Details dargestellt werden. Dies ist eine Mischung aus Explore und Abstract/Elaborate aus Abschnitt 3.1. Link&Brush soll mehrere Visualisierungstechniken miteinander vereinen um den größtmöglichen Vorteil daraus zu gewinnen.
Die hier dargestellten drei Dimensionen sind orthogonal zueinander. Orthogonalität bedeutet hier, dass irgendeiner der Datentypen in Verbindung mit irgendeiner der Visualisierungstechniken und irgendeiner der Interaktionstechniken genutzt werden kann.

Datenarten: Die Art der Daten hängt von den Dimensionen, oder Variablen, der Daten ab. Keim, D.A. unterscheidet Daten in diese sechs Dimensionen [14] und beschreibt sie wie folgt:

Eindimensionale Daten: Eindimensionale Daten haben, wie der Name sagt, meist eine dichte Dimension. Zeitabhängige Daten sind ein Beispiel für eindimensionale Daten. Bei zeitabhängigen Daten werden mehrere Attribute einem Zeitpunkt zugeordnet, was eine hohe Dichte der Daten verursacht.
Eindimensionale Daten werden oft mit Hilfe der „dichten Pixel Darstellung„ (Dense Pixel) visualisiert. Diese Art der Darstellung wird im Verlauf dieses Ab-

schnittes, unter Visualisierungsarten, näher beschrieben.

Zweidimensionale Daten: Zweidimensionale Daten haben zwei getrennte Dimensionen. Das bedeutet, dass diese Daten zwei verschiedene Attribute haben nach denen sie geordnet werden können. Geographische Daten sind ein Beispiel für zweidimensionale Daten. Die beiden Dimensionen stellen den Längen- und Breitengrad dar.

Multidimensionale Daten: Multidimensionale Daten sind die am häufigsten vorkommenden Datensätze. Multidimensionale Daten sind durch ihre vielen Attribute schwerer darzustellen als ein- oder zweidimensionale Daten. Grund dafür ist, dass sie nicht einfach auf die zwei Dimensionen eines Bildschirms gemappt werden können. Relationale Datenbanken sind ein Beispiel für multidimensionale Daten. Diese Daten werden oft vereinfacht oder aggregiert um sie darzustellen. Es gibt aber auch die „Parallel Coordinate Technique“ [12] um eine Visualisierung umzusetzen. Ein Beispiel für eine Darstellung mit der Parallel Coordinate

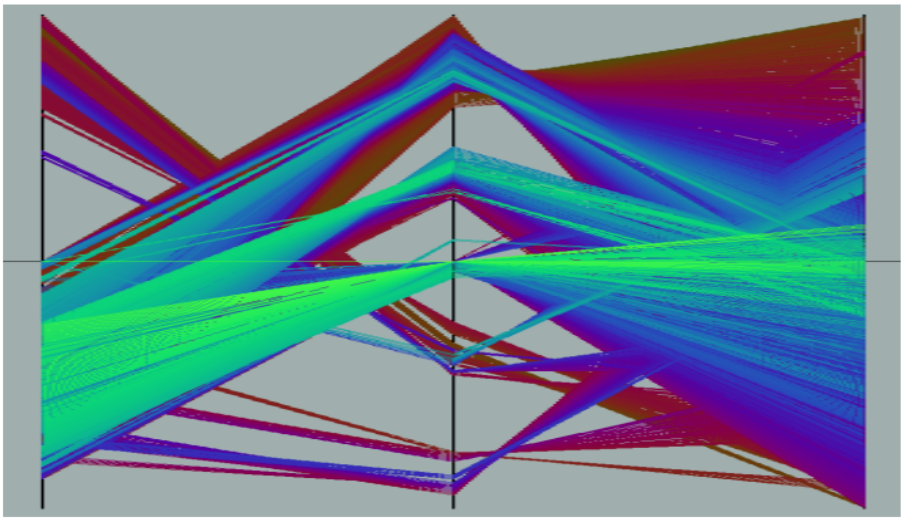


Abb. 2. Beispiel für eine Visualisierung multidimensionaler Daten mit Hilfe der „Parallel Coordinate Technique“ [12].

Technique wird in Abb. 2 gezeigt. Bei dieser Technik werden Datenitems als polygonale Linie dargestellt. Diese Linie schneidet die horizontale Dimensionsaxe an der, dem Datenwert und Dimension, entsprechenden Stelle. Einfacher ausgedrückt bedeutet es soviel, dass die Werte für die einzelnen Attribute jeweils an horizontalen Linien dargestellt werden. Ein Diagramm hat für jedes Attribut, also Dimension, eine horizontale Linie. Für jedes Datenitem verläuft eine Linie von links nach rechts und schneidet die horizontalen Linien.

Text- oder Webbasierte Daten: Heutzutage können Daten nicht immer von Zahlen beschrieben werden. Zum Beispiel bei Text- oder Hypertext-Daten von Webseiten. Um diese Daten visualisieren zu können werden diese meist erst zu Vektoren konvertiert und dann mit einer Technik für multidimensionale Daten dargestellt. Zum Beispiel mit der oben genannten Parallel Coordinate Technique.

Hierarchien- oder Graphendaten: Wenn Daten Beziehungen zu anderen Daten haben werden diese meist durch Graphen oder Hierarchien dargestellt. Ein Graph besteht aus einem Satz von Objekten, dies sind Knoten und Kanten die diese verbinden. Die Hyperlinks im Web sind ein Beispiel für Graphdaten. Es existiert eine Reihe von speziellen Visualisierungstechniken für Hierarchie- und Graphdaten [8]. Zum Beispiel können hierarchische Website-Daten mit Hilfe eines „Hyperbolic Views“ dargestellt werden. In Abb. 3 ist ein Hyperbolic View

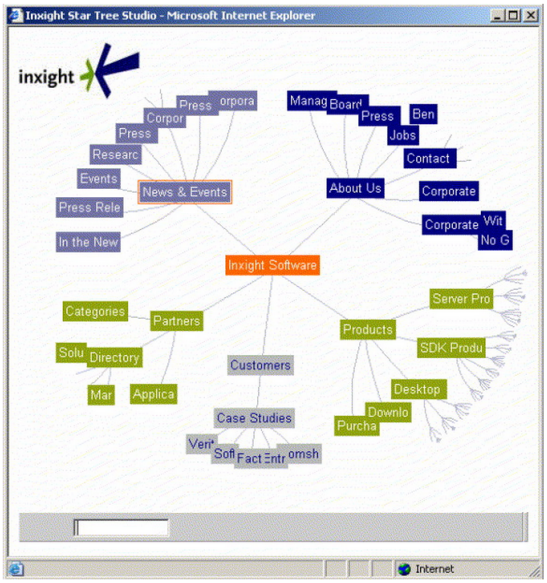


Abb. 3. Beispiel für die Visualisierung von Hierarchiedaten mit Hilfe eines sogenannten „Hyperbolic Views“ [26].

der Website-Map der „Inxight Software“ dargestellt [26]. Der Hauptknoten ist, hier in orange dargestellt, die Startseite. Davon ausgehend gibt es einige Unterknoten die die Unterseiten der Webseite darstellen. Diese Technik erleichtert es einen Überblick über große Hierarchie-Daten zu erlangen. In der Grafik zum Beispiel sieht man welche Seiten die meisten Unterseiten besitzen und kann sich diese genauer ansehen. Visualisiert wurde das ganze mit *inxight*[1], also der ange-

[1] www.inxight.com - Inxight Software

botenen Software der Seite.

Algorithmen- oder Softwaredaten: Die letzte Datenklasse sind die Daten von Algorithmen und Software. Hierbei geht es darum Daten zu visualisieren um den Verlauf eines Algorithmus oder den Fluss einer Software zu verdeutlichen. Dafür wird der Sourcecode visuell dargestellt und ist somit einfacher zu verstehen. Zum Beispiel kann der Sourcecode durch Visualisierung einfacher gedebuggt und Fehler einfacher gefunden werden.

Visualisierungsarten: Um die verschiedenen Arten von Daten zu visualisieren gibt es verschiedene Möglichkeiten [14]. Die folgenden vier wesentlichen Visualisierungsarten werden von Keim, D.A. vorgestellt:

Geometrisch transformierte Darstellung: Eine Technik zur geometrisch transformierten Darstellung ist die bereits genannte "Parallel Coordinate Technique"(Abb. 2). Hier wird versucht multidimensionale Daten so zu vereinfachen, dass die interessanten Stellen hervorgehoben werden. Eine genaue Erklärung findet sich im Abschnitt 2 multidimensionalen Daten.

Ikonische Darstellung: Bei der ikonischen Darstellung werden die Attributwerte von multidimensionalen Datenitems auf Icons gemappt. Icons können alles mögliche sein, wie zum Beispiel kleine Gesichter, Sterne oder Nadelköpfe. Ein Beispiel für eine ikonische Darstellung ist in Abb. 4 gegeben. Auf der Ab-

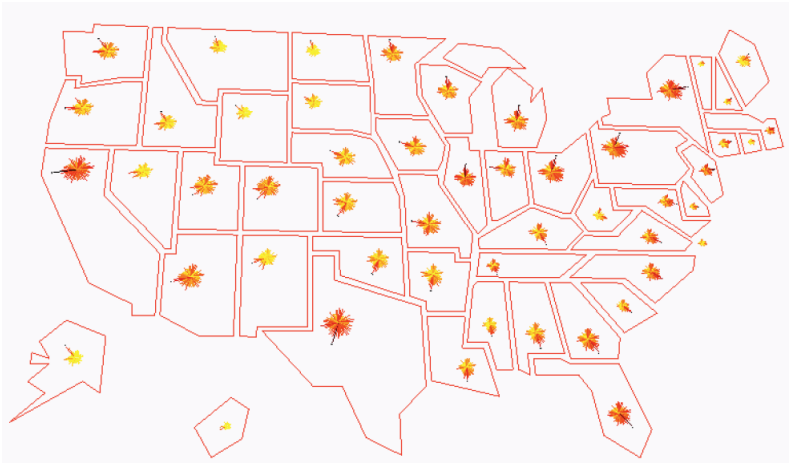


Abb. 4. Beispiel einer ikonischen Visualisierung [1].

bildung ist Nordamerika mit den Bundesstaaten abgebildet. Die farbigen Sterne

stellen die Anrufdichte zwischen den verschiedenen Staaten dar [1]. Unterschieden wird hier durch verschiedene Farben, Längen und Winkel der „Zacken“ der Sterne (Icons).

Dichte Pixel Darstellung: Bei der sogenannten dichte Pixel oder „Dense Pixel“ Darstellung wird jeder Dimensionswert auf ein Farbpixel abgebildet. Die Farbpixel werden dann entsprechend gruppiert und in zusammen liegenden Bereichen angeordnet. In Abb. 5 ist ein Beispiel für eine Visualisierung mit dichten

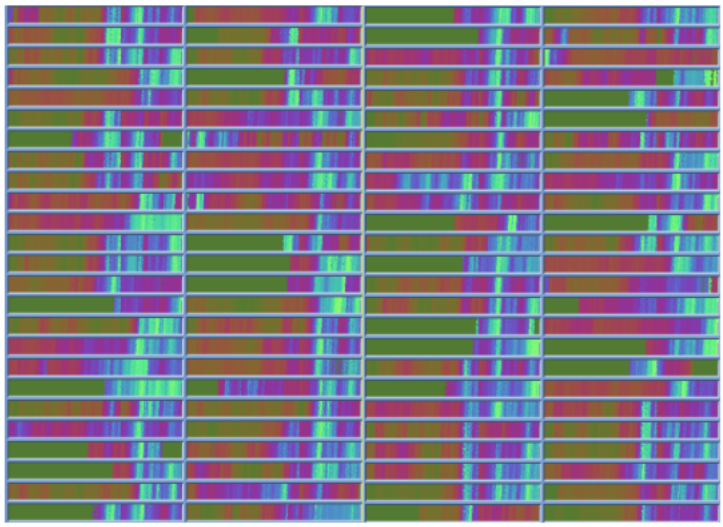


Abb. 5. Beispiel einer Dense Pixel Visualisierung [14].

Pixeln dargestellt. Es sind verschiedene Farbgruppen zu sehen. Die Farbgruppen stellen jeweils Datenitems mit gleichen Attributen dar. In der Abbildung sind zeitabhängige finanzielle Daten dargestellt. Es sind die täglichen Stockpreise von 20 Jahren der 100 Stocks des Frankfurter Stock Indexes dargestellt.

Gestapelte Darstellung: Bei einer Visualisierung mit der gestapelten oder „stacked“ Darstellung werden Daten partitioniert in einer Hierarchie dargestellt. Die Idee ist es, in ein Koordinatensystem ein weiteres Koordinatensystem einzubetten. Zum Beispiel könnten zwei Attribute das äußere Koordinatensystem darstellen und zwei weitere Attribute stellen das Innere dar. Abb. 6 zeigt eine gestapelte Darstellung in einem Koordinatensystem aus Koordinatensystemen. Bei den visualisierten Daten handelt es sich um Daten der Ölförderung. Die äußere x- und y-Achse stellen die Längen und Breitengrade dar. Die innere x- und y-Achse stellen die Tiefe und den Erzgehalt dar. So kann übersichtlich dargestellt werden, an welchen Orten welche Mengen gefördert wurden.

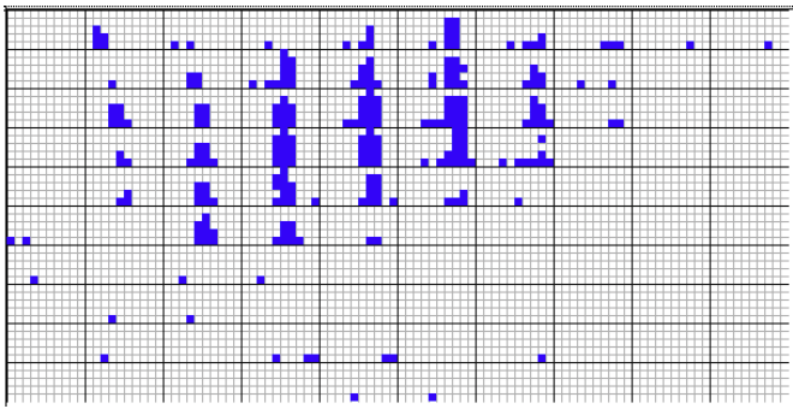


Abb. 6. Beispiel für eine Visualisierung von Daten mit Hilfe der gestapelten Darstellung [14].

2.2 Datenanalyse

Um einen Datensatz zu analysieren gibt es verschiedene Arten. Je nachdem wie die Darstellung aussehen soll ,oder der Datensatz beschaffen ist, bietet sich eine andere Analyseart an. Die folgenden Arten gibt es:

Explorative Datenanalyse: Oft möchte man in einem großen Satz von Daten nach bestimmten Attributen suchen [7]. Um entweder schnell etwas zu finden, oder um die oben beschriebenen Visualisierungen zu vereinfachen oder einzuschränken.
Die meistgenutzte Technik der Datenanalyse ist die explorative Datenanalyse. Diese Technik wir in zwei Hauptziele geteilt [7], in Datenbeschreibung und Modellformulierung. Bei der Datenbeschreibung wird damit begonnen die Daten zusammenzufassen um die wichtigsten Features auszusuchen. Es gibt aber Situationen in denen dies nicht so einfach möglich ist, auf Grund von schlechter Datenqualität.
Beim Formulieren von Modellen werden Hypothesen erstellt um dann eine geeignete statistische Methode, zum Analysieren des gegebenen Datensatzes, zu bestimmen.

Deskriptive Datenanalyse: Bei der deskriptiven Datenanalyse werden die Daten gesammelt und übersichtlich dargestellt. Die Darstellung erfolgt in Form von Tabellen oder Graphen [23].

Inferenzielle Datenanalyse: Im Vergleich zur deskriptiven Datenanalyse wird bei der inferenziellen Datenanalyse untersucht ob Unterschiede zwischen mehreren Proben bestehen [23]. Des Weiteren wird untersucht ob es nötig ist diese Unterschiede für die aktuelle Untersuchung herauszustellen.

2.3 Augmented Realtity

Augmented Reality wird als eine Variation der virtuellen Umgebung oder auch der virtuellen Wirklichkeit definiert [2]. In einer virtuellen Umgebung werden die Nutzer völlig in die Umgebung integriert und können währenddessen die Realität nicht sehen. Im Gegensatz zur Augmented Reality, wo ein Benutzer virtuelle Objekte in der realen Welt sieht. Das bedeutet, dass durch Augmented Reality die Realität ergänzt statt ersetzt wird. Virtuelle Objekte sollen für den Benutzer real erscheinen und mit den wirklich realen Objekten koexistieren.
Augmented Reality kann also als eine Mischung aus komplett real und komplett virtuell angesehen werden. Durch die virtuellen Objekte können Informationen dargestellt werden die der Benutzer mit seinen Sinnen nicht wahrnehmen kann. Ein virtuelles Objekt kann zum Beispiel Sensordaten darstellen, mit welchen der Nutzer interagieren kann. Dies hat einen erheblichen Vorteil bei der Datenanalyse, weswegen Augmented Reality als Werkzeug zur Datenanalyse verwendet wird. Virtuelle Objekte sind in verschiedenen Dimensionen darstellbar [3]. Abb.

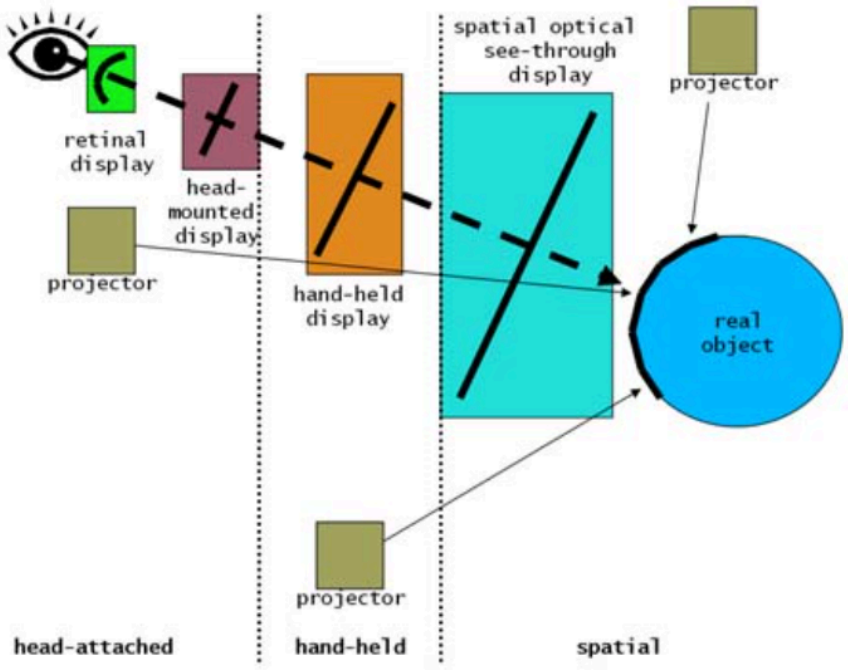


Abb. 7. Dimensionen zur Darstellung von Augmented Reality [3]

7 zeigt welche Darstellungsdimensionen es gibt und wie diese vom Auge bis zum dargestellten Objekt angeordnet sind.

Die Dimensionen lassen sich aufteilen in „head-attached“, „handheld“ und „spatial“. Entweder stellen Displays die reale Welt gemischt mit den virtuellen Objekten dar, oder Projektoren stellen virtuelle Objekte auf realen Objekten dar.
Die drei Dimensionen können noch in verschiedene Arten der Darstellung unterteilt werden. Bei den „head-attached“ Displays gibt es die Darstellung mit Hilfe von Retina-Displays. Retina Displays projizieren das Bild direkt auf die Netzhaut, somit ist die Netzhaut das Display. Das zweite head-attached Display ist das „head-mounted“ Display. Beim head-mounted Display wird eine Brille aufgesetzt und auf den Brillengläsern die Visualisierung von realer und virtuellen Objekten durchgeführt.
Für die Dimension hand-held gibt es das hand-held Display. Hierbei handelt es sich um ein Display (meist ein Smartphone) welches eine Armlänge vom Betrachter entfernt ist.
Die letzte Dimension sind die räumlichen Darstellungsarten. Hier gibt es die Möglichkeit mit Hilfe von spatial optical see-through Displays die Objekte zu visualisieren. Die Displays befinden sich hierbei eingebettet in der realen Umgebung.

3 Interaktion in Datenanalyse und Augmented Reality

Informationen, sowohl in der allgemeinen Datenanalyse als auch in Augmented Reality, sollen nicht nur visualisiert werden. Es soll außerdem möglich sein mit den Informationen oder visualisierten Informationen zu interagieren. Für diese Interaktionen gibt es in der Datenanalyse und Augmented Reality verschiedene Ansätze, die in den nachfolgenden Abschnitten erläutert werden.

3.1 Möglichkeiten in der Datenanalyse

Die Interaktion mit den Daten oder Informationen erfolgt zwischen dem Nutzer und dem System. In Yi, Ji Soo, et al. „Toward a deeper understanding of the role of interaction in information visualization“ [27] werden die verschiedenen, aktuellen, Interaktionsarten erläutert.
Der Nutzer untersucht die Daten genauer um spezielle Informationen oder einen tieferen Einblick in die Daten zu erhalten.
In der Informationsvisualisierung gibt es sieben weitverbreitete, verschiedene, Möglichkeiten mit den Informationen zu interagieren.
Shneiderman beschreibt in „The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations“ [20] das „Visual Information-Seeking Mantra“ von dem sich die folgenden Interaktionsarten ableiten lassen:

Select: Wie der Name schon sagt hat bei dieser Art der Interaktion der Nutzer die Möglichkeit ein bestimmtes Datenitem oder mehrere Datenitems auszuwählen und zu verfolgen. Dies hat den Vorteil, dass Nutzer ausgewählte Datenitems auch in sehr großen Visualisierungen verfolgen können, da diese hervorgehoben werden. Select wird dabei meistens in Zusammenhang mit anderen

Techniken benutzt, um diese zu bereichern.

Explore: Wenn Nutzer Daten visualisieren, um diese zu untersuchen, ist es meist nur möglich einen gewissen teil der Daten anzusehen. Auf Grund von oft riesigen Datensätzen und limitierenden Displays oder Sichtfeldern. Deshalb verschaffen sich Nutzer Verständnis über einen Teil von Daten und gehen dann weiter, um den nächsten Teil zu untersuchen. Hierfür wird bei der Explore-Technik meist ein Teil der Daten aus dem Sichtfeld geschoben und ein neuer Teil kommt hinzu.

Reconfigure: Durch diese Technik kann der Nutzer eine andere Art der Datenanordnung auswählen. Um die Daten aus anderen Perspektiven zu sehen werden von den Anzeigetools die Daten umgeordnet. Durch das Umsortieren, nach einem gewissen Attribut, können dem Nutzer einfacher Datenzusammenhänge (mit einem anderen Attribut) angezeigt werden.

Encode: Im Vergleich zur Reconfigure-Technik wird bei der Encode-Technik eine andere Art der Visualisierung gewählt (z.B. Größe oder Farben). Ein einfaches Beispiel für Encode ist die Repräsentation von einem Tortendiagramm zu einem Balkendiagramm zu ändern. Eine Technik des Encodings ist das Farbencoding. Beim Farbencoding kann der Benutzer ein Farbspektrum für eine bestimme Variable anwenden und dynamisch anpassen, bis sie für ihn passt.

Abstract/Elaborate: Mit dieser Technik ist es dem Benutzer möglich den Grad der Abstraktion einer Datenvisualisierung einzustellen. Der Benutzer kann, angefangen bei einer Übersicht, immer tiefer in gewisse Teilbereiche eintauchen um sich die Daten detaillierter anzeigen zu lassen.

Filter: Mit Hilfe der Filter-Technik können Benutzer einstellen welche Daten sie aus einem gewählten Datensatz sehen möchten. Um die Anzeige einzuschränken werden Anzeigeeinschränkungen definiert. Der Benutzer kann einen Darstellungsbereich oder Darstellungsbedingungen festlegen und nur Datenitems die in diesem Bereich liegen oder die Bedingungen erfüllen werden dem Benutzer angezeigt. Alle Datenitems die ausserhalb des Bereichs liegen oder die Bedingungen nicht erfüllen werden ausgeblendet.

Connect: Mit der letzten Technik, der Connect-Technik, kann sich ein Benutzer Beziehungen zwischen Datenitems anzeigen lassen. Durch die Connect-Technik werden auch relevante aber eigentlich ausgeblendete (durch Filter) Datenitems eingeblendet.

3.2 Möglichkeiten in Augmented Reality

Wie auch bei der Datenanalyse in Abschnitt 3.1 beschrieben, gibt es für die Interaktion in Augmented Reality verschieden Arten. Diese werden in [4] grundlegend

in Selektion, Manipulation und Navigation aufgeteilt. Für die Interaktionsarten gibt es jeweils noch verschiedene Unterarten [19]. Im nachfolgenden Abschnitt werden die state-of-the-art Interaktionen der Augmented Reality beschrieben.

Selektion Mit Hilfe der Selektion ist es möglich Objekte, virtuell oder real, direkt auszuwählen. Die Interaktionstechnik der Selektion wird noch in folgende drei Unterarten unterteilt:

Direkte Selektion: Wie der Name schon sagt wird bei dieser Art der Selektion das Objekt direkt ausgewählt. In der Augmented Reality gibt es die Möglichkeit hierfür die Hand als Cursor zu nutzen. Hierbei handelt es sich um das sogenannte Magic-Mouse System [25]. Durch einen Marker, der an der Hand angebracht wird, kann diese als Cursor fungieren. Für die Magic-Mouse wurde zum Einen das ARToolkit[2] eingesetzt und damit die Werte der verschiedenen Achsen ausgewertet. Das heisst das ARToolkit generiert eine Transformationsmatrix welche die Hand darstellt. Zum Anderen werden Funktionen verwendet die die Positionen und Rotationen von den Achsen X, Y und Z aus der Matrix extrahieren und sie dann nach dem Normalisieren auf eine Variation aus Operationen abbilden.
Bei den Operationen kann es sich um Robotersteuerung z.B. handeln aber auch um Navigation in 2D und 3D. Für die Interaktionen hier ist die Navigation des Cursors in 2D und 3D interessant. Das größte Problem bei dieser Art eine direkte Selektion umzusetzen ist, dass es nicht einfach möglich ist eine Auswahl zu bestätigen. Also z.B. den Druck auf einen Button.
Ein weiteres Beispiel für die direkte Selektion ist die *Virtuelle Hand* [6]. Hier kann die Hand dazu verwendet werden um Objekte, die ausgewählt werden sollen, virtuell zu greifen. Die Bedingung um die virtuelle Hand umzusetzen sind Marker an einem Handschuh. Somit ist die virtuelle Hand nur dazu geeignet Objekte in Greifnähe, die auch mit der realen Hand gegriffen werden könnten, zu greifen. Das ganze nennt sich hier FingARtips System [6]. Es handelt sich hierbei nicht um ein reines Selektionswerkzeug sondern um eine Kombination aus den Interaktionen: Greifen, Zeigen, Navigieren, und Kommandogesten. Für die Beschreibung der direkten Selektion ist aber nur die Interaktion Greifen interessant. Mit der Greifen-Interaktion kann ein Objekt gegriffen werden und somit selektiert werden. Danach kann das Objekt manipuliert werden (siehe direkte Manipulation im folgenden Abschnitt 3.2). In Abb. 8 ist eine Darstellung der virtuellen Hand gegeben. Hier sieht man auf der linken Seite wie die Finger der Hand erkannt werden um zu greifen. Auf der rechten Seite ist die virtuelle Hand in Aktion zu sehen. Es wird gezeigt wie ein virtuelles Objekt mit zwei Fingern gefasst und angehoben wird.

Okklusionsselektion: Bei der Okklusionsselektion ist es möglich mehrere Objekte einfach auszuwählen. Hierfür kann zum Beispiel ein Tablet über einen Bereich geführt werden indem sich die Objekte befinden. Dabei werden, mit

[2] http://artoolkit.org/

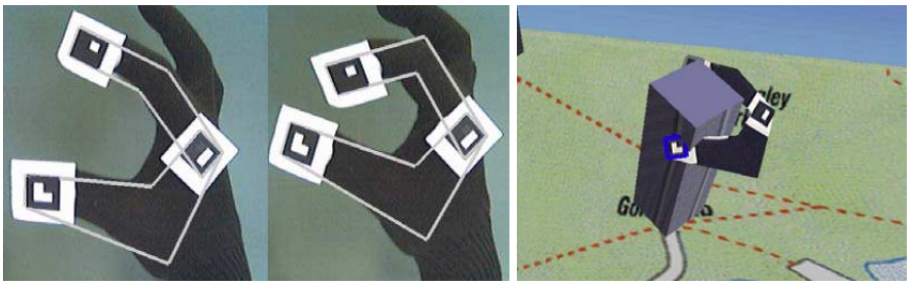


Abb. 8. Darstellung einer direkten Selektion mit Hilfe der *Virtuellen Hand* [6].

Hilfe eines sogenannten Fischnetz-Werkzeugs, die Objekte „eingefangen“ und selektiert [21].
Ein weiterer Ansatz eine Okklussionsselektion durchzuführen wird in Lee et al. [16] beschrieben. Die Selektion soll in der Arbeit von Lee et al. mit Hilfe von vordefinierten Markern durchgeführt werden. Hierfür werden mehrere Marker an ein Objekt angebracht um ein sicheres Tracken des Objekts zu gewährleisten. Die Auswahl wird realisiert indem Marker für eine gewisse Zeit verdeckt werden. Nach Ablauf dieser definierten Zeitspanne gilt das oder die Objekte als ausgewählt.

Distanzselektion: In den zuvor beschriebenen Selektionstechniken werden Objekte in der Nähe selektiert. Zum Beispiel mit der virtuellen Hand mit welcher Objekte nur eine Armlänge entfernt sein können da die virtuelle Hand an die reale Hand gebunden ist. Bei der Distanzselektion sollen auch weiter entfernte Objekte ausgewählt werden können. Hierzu lässt sich der Blickfokus eines Benutzers verwenden [22]. Dazu wird angenommen, dass der Fokus jeweils in der Bildmitte liegt und dadurch die, sich dort befindlichen, Objekte ausgewählt werden können.
In Abbildung 9 ist eine Distanzselektionsmethode dargestellt die mit Raycasting arbeitet. Hierbei wird ein Objekt durch das Ausschicken von Lichtsignalen, z.B. ein Lichtkegel oder ein Laserpointer, selektiert. Zu sehen ist die AR Mask [10] mit einem Ausgabegerät auf dem Kopf und Eingabegeräten in den Händen. Mit Hilfe des Ausgabegerätes können, wie bereits erwähnt, Strahlen auf auszuwählende Objekte gesendet werden. Nach Empfang der zurückkommenden Strahlen über das Eingabegerät sind die Objekte selektiert. Links in der Abbildung ist eine Variante der AR Mask bei dem Ein- und Ausgabe getrennt sind, und rechts die Variante bei der Ein- und Ausgabe verbunden sind.

Manipulation Genau betrachtet hängen Manipulation und Selektion zusammen, da man meist ein Objekt zuerst auswählt und es dann manipuliert. Die Aufgaben einer Manipulation sind die Veränderung von Form, Skalierung, Position und Orientierung eines Objektes welches vorher selektiert wurde [19]. Auch die verschiedenen Manipulationstechniken sind unterteilbar. Die Aufteilung wird

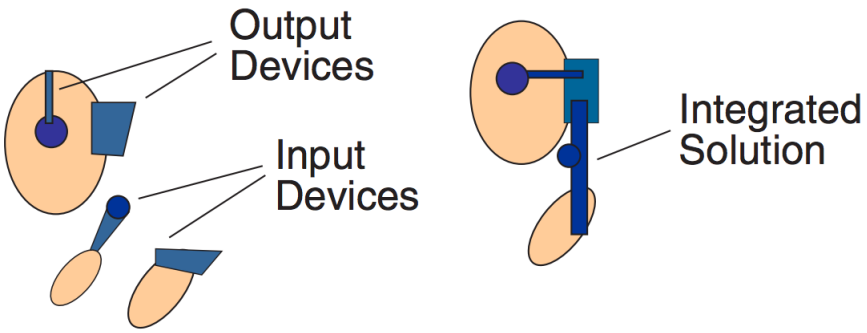


Abb. 9. Distanzselektion mittels Raycasting mit der AR Mask [10].

wie folgt vorgenommen:

Direkte Manipulation: Zuvor ausgewählte Objekte können mit der direkten Manipulation einfach verändert werden. Zum Beispiel kann die Skalierung geändert werden. Ein Beispiel hierfür ist wieder die *Virtuelle Hand* [6] mit der ein ausgewähltes Objekt mit Handbewegungen verändert werden kann, also direkt. Zu sehen ist dies in Abb. 8 auf der rechten Seite. Das Ausgewählte Objekt wird von der Hand gegriffen und kann z.B. gedreht werden.
Oft werden bei der direkten Manipulation Einschränkungen eingesetzt. Die Einschränkungen werden mit der vorhergehenden Selektion verbunden um die Manipulationsmöglichkeiten schon vorher einzuschränken. Zum Beispiel kann realisiert werden, dass eine Selektion eines Objekts von oben besagt, dass das Objekt nur in der Höhe geändert werden kann. Oder eine Seiten-Selektion die Manipulation dahingehend einschränkt, dass nur die Breite verändert werden kann.

Pointer: Hier gibt es die Möglichkeit Objekte und damit Menüs auf einzelne Finger zu mappen [18]. Durch pressen eines Fingers gegen den Daumen wird das jeweilige Objekt bzw. Menü ausgewählt und kann manipuliert werden. Eine weitere, einfachere Möglichkeit ist das Nutzen eines Laserpointers [5]. Der Laserpointer wird zum Selektieren und dann Manipulieren des Objektes verwendet.

Manipulation am Repräsentanten: Wie der Name schon sagt wird hier ein Repräsentant für ein Objekt verwendet und an diesem die Manipulation ausgeführt. Im Vergleich zur direkten Manipulation oder dem Pointer wird hier nicht mit Hilfe eines grafischen Objekts manipuliert. In Abb. 10 ist eine Manipulation am Repräsentanten dargestellt. Auf dem Bild ist oben links eine Person zu erkennen die ein head-mounted Display trägt. Die Kamera zeigt, dass es ein see-throught Display ist. Auf dem Bild ist das zu sehen was die Person in diesem Moment sieht. Die Person hält eine Unterlage in der Hand auf welcher ein virtuelles Auto dargestellt ist. Mit Hilfe dieser Unterlage ist es der Person möglich das

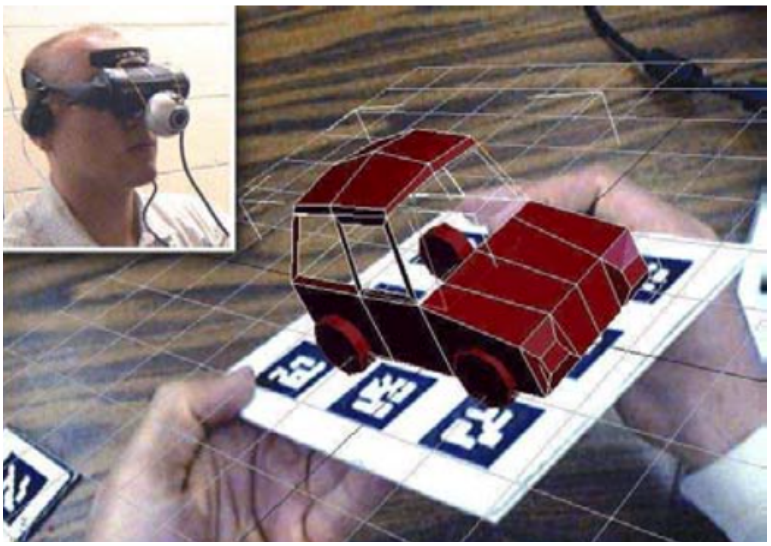


Abb. 10. Darstellung einer Manipulation am Repräsentanten [19].

Objekt (Auto) zu manipulieren. Es wird hier also das reale Objekt manipuliert um das virtuelle Objekt zu manipulieren.

Navigation Die letzte Art der Interaktion in Augmented Reality ist die Navigation oder auch Wegfindung. Wie es der Name schon impliziert handelt es sich hierbei um eine Interaktion durch Bewegung. Auch bei dieser Interaktionsart kann zwischen Mehreren unterschieden werden:

Karten: Durch Bewegungen in Augmented Reality kann der Nutzer mit einer Karte interagieren. Je nach Standpunkt wird ihm die Karte angezeigt. Bewegt sich der Benutzer bewegt sich die Karte mit (adaptive Karten), indem sie sich zum Beispiel rotiert damit sie immer in Bewegungsrichtung zeigt [9]. Da es ungewohnt für den Benutzer ist mit einer starren Karte zu interagieren haben sich in der Augmented Reality die adaptiven Karten durchgesetzt.
Des Weiteren beschreiben Darken und Peterson in [9], dass bei planenden Aufgaben eher starre Karten und bei situativen Szenarien eher adaptive Karten geeignet sind. Aufgaben in Augmented Reality sind eher situative Aufgaben. Bei den Karten bzw. der Navigation allgemein besteht die Aufgabe darin die Bewegung für die aktuelle Situation auszuführen. Deshalb wird die adaptive Karte in der Augmented Reality präferiert eingesetzt.
Die sogenannte „world-in-miniature (WIM)“ [13] ist der adaptiven Karte ähnlich. Der Unterschied ist, dass der Nutzer bei WIM ein head-mounted Display trägt auf welchem ihm eine Karte in Gestalt einer Miniaturwelt angezeigt wird. Die Miniaturwelt enthält Weginformationen mit dessen Hilfe sich der Benutzer in einer unbekannten Umgebung zurecht finden kann. In Abbildung 11 ist eine ad-

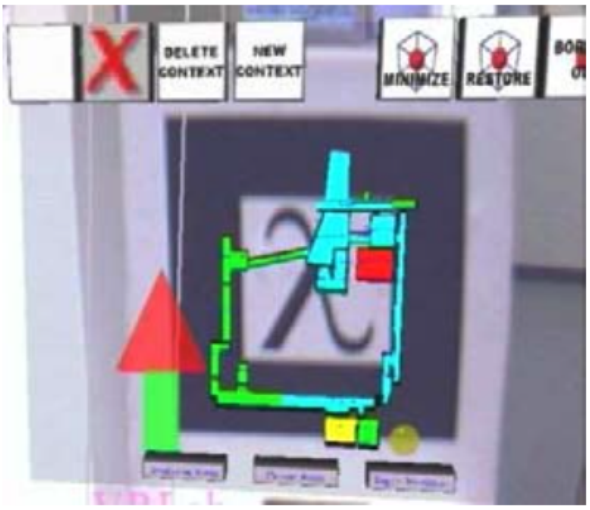


Abb. 11. Darstellung einer WIM Karte [13].

aptive Karte dargestellt. Der Pfeil markiert in welche richtung man gehen muss [13]. Da die Karte adaptiv ist dreht sich diese mit der eigenen Drehung mit. So ist es dem Benutzer möglich immer nach vorne zu gehen, sowohl in der realen Welt als auch auf der Karte.

Markierungen in der Welt: Die zweite Form der Navigation ist mit Hilfe von sogenannten Markierungen in der Welt. Hier sei auch wieder das schon erwähnte WIM System zu nennen. Da WIM neben einer virtuellen Karte auch Markierungen oder Annotationen enthält ist es auch bei Markierungen in der Welt einzuordnen.
Eine weitere Möglichkeit Markierungen in der Welt anzuwenden ist es die echte Welt mit einem virtuellen Weg zu vereinen [17]. Hierbei wird die Straße vor einem Auto abgefilmt und mit Hilfe von Kartendaten mit einem Pfad vereint. Der Pfad zeigt den Weg und der Benutzer muss diesem nur noch folgen. Auch hier werden wie schon bei den Karten die adaptiven Karten eingesetzt. Ein Beispiel für Markierungen in der Welt und WIM ist in Abb. 12 gegeben. Die Abbildung zeigt das Bild das ein Nutzer durch sein head-mounted Display sieht. Der Nutzer sieht die reale Welt kombiniert mit virtuellen Objekten. Bei den virtuellen Objekten handelt es sich um eine Karte, in der unteren Mitte im Bild. Zusätzlich werden die Kanten des Raumes, in dem der Nutzer sich befindet, virtuell nachgezeichnet. Auf der rechten Seite wird eine Tür markiert. Ein Pfeil zeigt auf die Tür und markiert den Weg den der Nutzer nehmen soll. Durch diese WIM-Applikation kann man einfach durch unbekannte Gegenden navigiert werden ohne auf vieles achten zu müssen. Der Nutzer folgt einfach nur dem Pfeil

Abb. 12. Screenshot aus einem Video einer SignPost2 - Vorführung [15].

und kommt somit ans Ziel. Ein Vorteil ist, dass er immer den Weg und die reale Welt im Blick hat. Eine andere Art der Umsetzungen von Markierungen in der Welt ist in Abb. 13 dargestellt. Hier wird dem Fahrer eines Wagens mit Hilfe von Augmented Reality und Markierungen in der Welt eine Gefahr angezeigt [24]. Auf der Frontscheibe wird dem Fahrer über einen Pfeil die Richtung angezeigt in der sich die Gefahr befindet.

3.3 Vergleich von Datenanalyse und Augmented Reality

Nachdem die verschiedenen Arten der Interaktionen in der Datenanalyse und Augmented Reality erläutert wurden, stellt sich die Frage welche Zusammenhänge es zwischen Beiden gibt.

Benutzerintention: Eine Interaktion setzt eine Intention eines Benutzers voraus. Bei der Datenanalyse ist diese Intention das verdeutlichen von Informationen die aus Daten erlangt werden. Ein Benutzer hat einen Datensatz zur Verfügung und möchte genauere Informationen darüber. Um die genaueren Informationen zu erlangen setzt er die Interaktionsarten ein.
Bei Augmented Reality ist die Intention ähnlich. Der Benutzer möchte durch eine Interaktion mehr Informationen über ein Objekt, welches Daten repräsentiert, erlangen. Der Unterschied ist, dass bei Augmented Reality die Interaktion schon einen Schritt früher beginnt. Der Benutzer interagiert bereits um den gewünschten Datenlieferanten auszuwählen in der Augmented Reality. Das bedeutet soviel

Abb. 13. Hinweis einer Gefahr für den Fahrer eins Wagens mit Hilfe von Markierungen in der Welt [24].

wie, dass der Nutzer eine Auswahl von Datenlieferanten hat (z.B. aus mehreren Industriemaschinen mit Sensoren wählt) aus denen er den Untersuchenden wählt.
Bei der Datenanalyse wird zwar auch ein Datensatz ausgewählt, hierbei handelt es sich aber eher um eine normale Interaktion und keine spezifische der Datenanalyse.
Jedoch bei der Augmented Reality ist die Intention des Nutzers die Daten vor Ort auszuwählen und zu untersuchen. Aus diesem Grund ist hier die Intention nicht alleine die Daten einfach zu untersuchen. In Augmented Reality sollen die Daten vor Ort in Echt-Zeit untersucht werden.

Vergleich der einzelnen Interaktionsarten: Sowohl in der normalen Datenanalyse als auch in der Datenanalyse mit Hilfe von Augmented Reality werden gewisse Interaktionen durchgeführt. Diese Interaktionen werden dazu benutzt um z.B. visualisierte Sensordaten genauer zu untersuchen und detaillierter oder auch einen besseren Überblick über diese Daten zu erlangen. Da die Interaktionen von Datenanalyse und Augmented Reality beide zur Untersuchung von Daten angewandt werden können liegt ein Vergleich nahe. Dieser Vergleich soll zeigen inwiefern sich diese Interaktionen ähneln oder die Interaktionen der Datenanalyse in Augmented Reality umsetzbar sind. Aus diesem Grund sollen die einzelnen Interaktionen miteinander verglichen werden.

In Tabelle 1 sind alle Ergebnisse in einer Übersicht zusammengefasst. Es wird jeweils angegeben ob die Interaktionen die gleiche Funktion oder unterschiedliche Funktionen haben. Zur besseren Übersicht sind Interaktionen mit unterschiedlichen Funktionen rot unterlegt. Wenn eine Interaktion aus der Datenanalyse nicht in der Augmented Reality vorhanden ist, diese aber mit einer Interaktion aus der Augmented Reality umsetzbar wäre, wird das jeweilige Feld gelb hervorgehoben. Ebenso wird eine mögliche Umsetzung mit „in AR umsetzbar“

	Select	Explore	Re-configure	Enode	Abstract/ Elaborate	Filter	Connect
Diskrete Selektion	gleiche Funktion	unter-schiedliche Funktion	unter-schiedliche Funktion	unter-schiedliche Funktion	unter-schiedliche Funktion	gleiche Funktion	unter-schiedliche Funktion
Okklusions-selektion	gleiche Funktion	gleiche Funktion	unter-schiedliche Funktion	unter-schiedliche Funktion	in AR umsetzbar	gleiche Funktion	unter-schiedliche Funktion
Distanz-selektion	gleiche Funktion	unter-schiedliche Funktion	unter-schiedliche Funktion	unter-schiedliche Funktion	unter-schiedliche Funktion	unter-schiedliche Funktion	unter-schiedliche Funktion
Direkte Manipu-lation	unter-schiedliche Funktion	unter-schiedliche Funktion	gleiche Funktion	in AR umsetzbar	unter-schiedliche Funktion	gleiche Funktion	unter-schiedliche Funktion
Pointer	gleiche Funktion	unter-schiedliche Funktion	gleiche Funktion	in AR umsetzbar	unter-schiedliche Funktion	unter-schiedliche Funktion	unter-schiedliche Funktion
Manipu-lation am Repräsen-tanten	unter-schiedliche Funktion	unter-schiedliche Funktion	gleiche Funktion	unter-schiedliche Funktion	unter-schiedliche Funktion	unter-schiedliche Funktion	unter-schiedliche Funktion
Karten	unter-schiedliche Funktion	unter-schiedliche Funktion	unter-schiedliche Funktion	unter-schiedliche Funktion	in AR umsetzbar	unter-schiedliche Funktion	in AR umsetzbar
Markier-ungen in der Welt	unter-schiedliche Funktion	gleiche Funktion	unter-schiedliche Funktion	unter-schiedliche Funktion	in AR umsetzbar	unter-schiedliche Funktion	in AR umsetzbar

Tabelle 1. Gegenüberstellung der einzelnen Interaktionsarten und Überprüfung auf Ähnlichkeit oder Umsetzbarkeit.

gekennzeichnet. Interaktionen die sich ähnlich sind sind mit „gleiche Funktion“ markiert und zusätzlich grün unterlegt.
Die erste Spalte zeigt die Interaktionsarten in der Augmented Reality. Zuerst die Selektionsarten mit der direkten Selektion, der Okklusionsselektion und der Distanzselektion. Dann die Manipulationsarten mit der direkten Manipulation, dem Pointer und der Manipulation am Repräsentanten. Zuletzt noch die Navigation mit den Karten und den Markierungen in der Welt. Bei den hier untersuchten Augmented Reality Interaktionsarten handelt es sich um die in Abschnitt 3.2 beschriebenen.
Die erste Zeile zeigt die Interaktionen aus der Datenanalyse. Die sieben untersuchten Interaktionen sind Select, Explore, Reconfigure, Encode, Abstract/ Elaborate, Filter und Connect. Eine detaillierte Erläuterung der Interaktionsarten der Datenanalyse findet sich in Abschnitt 3.1.
Durch diese Matrixdarstellung lassen sich alle Interaktionen miteinander vergleichen und auf Ähnlichkeit überprüfen. Die Untersuchung wurde so durchgeführt, dass jeweils eine Interaktion der Datenanalyse betrachtet wurde. Dabei wurde die Datenanalyse-Interaktion mit jeder Interaktion der Augmented Reality verglichen und geprüft.
Im Folgenden Abschnitt sollen die einzelnen Vergleiche beschrieben und die Ergebnisse erklärt werden.

Select: Zuerst wird das „Select“ aus der Datenanalyse mit allen Interaktionen der Augmented Reality verglichen. Die direkte und Okklusionsselektion aus der Augmented Reality sind sehr ähnlich zum Select aus der Datenanalyse. Mit Select, direkter Selektion oder Okklusionsselektion kann ein oder mehrere Objekte ausgewählt werden um es oder sie näher zu untersuchen. Hier wurde also offensichtlich eine Interaktionsart der Datenanalyse in die Augmented Reality übernommen.
Die Distanzselektion ist eine Erweiterung des Selects. Mit Hilfe der Distanzselektion ist es in der Augmented Reality möglich weit entfernte Objekte direkt auszuwählen. Somit handelt es sich hierbei wie beim Datenanalyse-Select um eine Selektion von Objekten und damit um eine gleiche Funktion. Bis auf den Pointer aus der Augmented Reality hat das Select keine Ähnlichkeiten mehr zu den restlichen Interaktionen. Mit dem Pointer ist es auch möglich, mit Hilfe eines Pointerhandschuhs z.B. ein Objekt auszuwählen. In der Datenanalyse wird ebenso ein Pointer, meistens in Form einer Maus, benutzt. Allerdings ist der Pointer im Vergleich zum Select ein mächtigeres Werkzeug, da hiermit Objekte direkt manipuliert werden können.
Da beim Select Objekte nicht Manipuliert werden sondern nur ausgewählt, besteht hier keine gleiche Funktion. Die meisten Manipulationswerkzeuge haben zwar die Funktion zur Auswahl von Objekten vor der Manipulation, da aber hier nur die Grundfunktionen untersucht werden wurde es hier nicht als gleich markiert. Es handelt sich lediglich um einen Verbund von zwei Grundfunktionen in den Werkzeugen.

Explore: Die Datenanalyse-Interaktion Explore hat Ähnlichkeiten zu zwei Interaktionen aus der Augmented Reality. Mit Explore ist es möglich Teilbereiche eines Visualisierten Sensordatensatzes nacheinander zu betrachten.
Die Okklusionsselektion hat Ähnlichkeit da sich hier verschiedene Bereiche durch Bewegen des Displays betrachten lassen. In der Datenanalyse werden die betrachteten Bereiche mit einem Cursor aus dem Sichtfeld hinaus und die neuen in das Sichtfeld hineingeschoben. Bei der Okklusionsselektion ist dies auch möglich allerdings reicht hier die Bewegung des Displays über die neuen Objekte.
Auch die Markierungen in der Welt aus der Augmented Reality weist Ähnlichkeiten auf. Da man zum Beispiel durch Bewegung durch ein Gebäude immer neue Teilinformationen bekommt ist es ähnlich zu Explore aus der Datenanalyse.
Die Manipulationsinteraktionen aus der Augmented Reality weißen keine Ähnlichkeiten zum Datenanalyse-Explore auf. Mit dem Explore wird nur der angezeigte Bereich verändert, jedoch nicht die Objekte.

Reconfigure: Die Interaktion Reconfigure der Datenanalyse hat lediglich mit den Manipulationsinteraktionen der Augmented Reality Ähnlichkeiten. Mit Reconfigure kann die Anzeige verändert und umsortiert werden. Mit der direkten Manipulation, dem Pointer und der Manipulation am Repräsentanten können selektierte Objekte auch verändert werden. Somit handelt es sich sowohl bei der Datenanalyse-Interaktion Reconfigure als auch bei den Augmented Reality Interaktionen der Manipulation um Manipulationen an den ausgewählten Objekten.
Da mit Den Selektionsinteraktionen der Augmented Reality Objekte lediglich ausgewählt werden besteht hier keine Ähnlichkeit oder keine gleiche Funktion.
Auch die Navigationsinteraktionen der Augmented Reality besitzten keine ähnliche Funktionen mit Reconfigure. Da bei den Karten oder den Markierungen in der Welt keine Veränderung der Objekte vorgenommen wird sondern sich lediglich die Informationen durch Weiterbewegen verändern.

Encode: Es gibt in der Augmented Reality keine einzelne Interaktion die der Datenanalyse-Interaktion Encode ähnlich ist. Mit Hilfe von Encode wird eine einfache Visualisierungsveränderung durchgeführt was auch in Augmented Reality gemacht werden kann. Zum Beispiel wenn Sensordaten dargestellt werden, dass diese zum Beispiel von einem Liniendiagramm zu einem Balkendiagramm geändert werden.
Es müsste möglich sein das Encode in den Manipulationsinteraktionen der Augmented Reality zu nutzen. Am besten wäre die Umsetzung mit der direkten Manipulation oder dem Pointer möglich.
Da es sich hierbei um eine offensichtliche Manipulation handelt kommen auch nur Manipulationsinteraktionen in Frage.
Da mit den Navigationsinteraktionen der Augmented Reality nicht die Darstellungsart sondern die dargestellen Informationen geändert werden, kommen Karten und Markierungen in der Welt hierfür nicht in Frage.

Abstract/Elaborate: Eine Möglichkeit das Abstract/Elaborate der Datenanalyse in Augmented Reality anzuwenden wurde hier in diesen Interaktionen

der Augmented Reality nicht gefunden. Mit Abstract/Elaborate ist es möglich einen vorhandenen Datensatz genauer zu untersuchen. Hiermit wird die Abstraktion verändert, in welcher der Datensatz betrachtet und untersucht werden soll. In der Augmented Reality wäre dies ebenso eine sehr nützliche Interaktion die auch umsetzbar wäre. Zum Beispiel könnte man durch Bewegung zum Objekt dieses genauer untersuchen. Durch Entfernen vom Objekt könnte eine Übersicht geliefert werden. Dies wäre also ein Hinein- und Herauszoomen durch Nähern und Entfernen vom Objekt. Dadurch wäre eine Umsetzung der Interaktion aus der Datenanalyse gegeben.
Bei Abstract/Elaborate ist es möglich tiefer in einen Datensatz zu zoomen um diesen genauer zu untersuchen. Die Umsetzung wäre also mit Hilfe der Okklusionsselektion möglich womit die Abstraktion durch Bewegen des Displays verändert wird. Des Weiteren wäre die Umsetzung mit den Navigationsinteraktionen: „Karten“ und „Markierungen in der Welt“ möglich. Hier kann die Abstraktion durch eine Entfernungsveränderung zum Objekt verändert werden. Also beim Daraufzugehen zum Objekt werden dem Benutzer mehr und detaillierte Informationen zu wenigen Objekten angezeigt. Bewegt sich der Benutzer vom Objekt oder von den Objekten weg werden ihm weniger Informationen angezeigt. Der Nutzer bekommt dann eine Übersicht für mehrere Objekte die sich in seinem Sichtfeld befinden.
Da die Objekte an sich nicht verändert werden, sondern die Entfernung zu den Objekten verändert werden soll um auch die Abstraktion zu verändern, kommen die Manipulationen hier nicht in Frage.

Filter: Bei der Filterinteraktion der Datenanalyse kann der dargestellte Bereich durch gewisse Bedingungen eingeschränkt werden. Bei der direkten Selektion kann die Anzeige eingeschränkt werden in dem man nur gewisse Objekte auswählt. Bei der Okklusionsselektion kann die Anzeige eingeschränkt werden indem man das Auswahldisplay nur über die gewünschten Objekte führt.
Bei der direkten Manipulation kann die Anzeige ebenfalls eingeschränkt werden. Hierbei kommt es darauf an wie Objekte selektiert werden. Diese Einschränkungen beziehen sich auf die Art und Weise wie ein Objekt später manipuliert wird. Zum Beispiel bewirkt ein Selektieren an der oberen Kante eine spätere ausschließliche Manipulation der Höhe des Objekts. Die Filter-Funktion ist also ähnlich zu den oben genannten Interaktionen. Der Unterschied jedoch ist, dass die Interaktion selbst in der Augmented Reality die Anzeige einschränkt und in der Datenanalyse ein vordefinierter Filter dafür zuständig ist.
Diese Interaktion der Datenanalyse ist also nur ähnlich in der Augmented Reality umgesetzt. Es wäre natürlich noch zusätzlich möglich die Funktion des Filters aus der Datenanalyse in eine Augmented Reality - Anwendung aufzunehmen. Allerdings wäre das dann keine Augmented Reality Interaktion sondern eher eine Funktion die in der Anwendung implementiert wäre um die Visualisierung zu verändern.

Connect: Zum Connect aus der Datenanalyse gibt es keine ähnliche Interaktion in der Augmented Reality.
Durch Connect können Beziehungen zwischen Datenitems angezeigt werden. Es können also zwei Items ausgewählt werden um dann in der Visualisierung ihre Verbindung aufzuzeigen. Diese Art der Interaktion wäre auch in der Augmented Reality hilfreich. Durch die Selektionsinteraktionen könnten einzelne Objekte ausgewählt werden um Verbindungen zwischen diesen aufzuzeigen. Denkbar wäre also eine Umsetzung aus zum Beispiel einer Kombination von Selektionsinteraktionen und Navigationsinteraktionen. Beim Bewegen durch eine Welt mit Informationen könnte man sich so Beziehungen von ausgewählten Objekten zu anderen aufzeigen lassen. Da es hier maßgeblich um die Auswahl von Objekten geht, wäre eine Umsetzung mit Hilfe von Manipulationsinteraktionen nicht denkbar.

4 Fazit

Durch die riesigen Datenmengen wird die Visualisierung von Sensordaten immer schwerer. Trotzdem sollen die Daten zur Analyse in irgendeiner Form dargestellt werden. Die Darstellungsform sollte möglichst übersichtlich und einfach zu verstehen sein.
In diesem Paper wurden die verschiedenen Darstellungsformen und die Auswahlkriterien dafür kurz vorgestellt. Es gibt verschiedene Dimensionen von Daten, z.B. eindimensionale oder mehrdimensionale Daten. Je nachdem welche Dimension die zu visualisierenden Daten haben wird eine entsprechende Visualisierungsart ausgewählt. Visualisierungsarten sind z.B. „Stacked Pixel (gestapelte Darstellung)“ oder „Dense Pixel (dichte Darstellung)“.
Es wurden auch zwei Arten vorgestellt wie Daten untersucht werden können. Zum einen können Daten mit Hilfe der normalen Datenanalyse untersucht werden, zum anderen kann Augmented Reality als Werkzeug zur Durchführung einer Datenanalyse eingesetzt werden.
In beiden Analysearten existieren verschiedene Interaktionen um mit den Daten umzugehen. Diese Interaktionen wurden näher untersucht und vorgestellt. Daraufhin wurden die einzelnen Interaktionen der Datenanalyse mit den Interaktionen der Augmented Reality verglichen. Es wurde überprüft ob Interaktionen aus der Datenanalyse in der Augmented Reality verwendet werden oder Ähnlichkeiten aufweisen. Des Weiteren wurde untersucht ob Interaktionen der Datenanalyse in Augmented Reality umgesetzt werden können, falls diese beim Vergleich nicht gefunden wurden. Hierzu wurde jede Interaktionsart aus der Datenanalyse jeder Interaktionsart aus der Augmented Reality gegenübergestellt. Mit einer Vergleichsmatrix wurden so alle einzelnen Interaktionen miteinander verglichen.
Das Ergebnis zeigt, dass die Interaktionen in der Augmented Reality deutliche Ähnlichkeiten zu denen der Datenanalyse aufweisen. Einige Interaktionen, wie zum Beispiel das grundlegende Auswählen („Select“), wurden in der Augmented Reality mit leichten Anpassungen übernommen. Genauso wie das Manipulieren

der Daten, was sowohl in der Datenanalyse als auch der Augmented Reality zu finden ist.
Andere Interaktionen hingegen, wie das „Connect“, das „Encode“ und das „Abstract/Elaborate“ wurden nicht in den Interaktionen der Augmented Reality gefunden. Diese Interaktionen wurden dann darauf hingehend untersucht ob sie in Augmented Reality umsetzbar sind. Das Ergebnis dieser Umsetzungsuntersuchung zeigt, dass aus den Mitteln der Augmented Reality die nicht gefundenen drei Arten umsetzbar sein müssten.
Der Vergleich zeigt deutlich, dass die Interaktionen in Augmented Reality aus den Interaktionen der Datenanalyse abgeleitet sind. Vier von sieben Interaktionen wurden jeweils in mindestens zwei Interaktionen der Augmented Reality wiedergefunden. Drei Interaktionen wurden nicht in Augmented Reality wiedergefunden. Die drei nicht gefundenen wurden allerdings alle als umsetzbar deklariert.

Literatur

1. Abello, J., Korn, J.: Mgv: a system for visualizing massive multidigraphs. Visualization and Computer Graphics, IEEE Transactions on 8(1), 21–38 (2002)
2. Azuma, R.T., et al.: A survey of augmented reality. Presence 6(4), 355–385 (1997)
3. Bimber, O., Raskar, R.: Modern approaches to augmented reality. In: ACM SIGGRAPH 2006 Courses. p. 1. ACM (2006)
4. Bowman, D.A., Kruijff, E., LaViola Jr, J.J., Poupyrev, I.: 3D user interfaces: theory and practice. Addison-Wesley (2004)
5. Broll, W., Lindt, I., Ohlenburg, J., Herbst, I., Wittkämper, M., Novotny, T.: An infrastructure for realizing custom-tailored augmented reality user interfaces. IEEE transactions on visualization and computer graphics 11(6), 722–733 (2004)
6. Buchmann, V., Violich, S., Billinghurst, M., Cockburn, A.: Fingartips: gesture based direct manipulation in augmented reality. In: Proceedings of the 2nd international conference on Computer graphics and interactive techniques in Australasia and South East Asia. pp. 212–221. ACM (2004)
7. Chatfield, C.: Exploratory data analysis. European journal of operational research 23(1), 5–13 (1986)
8. Chen, C.: Information visualisation and virtual environments. Springer Science & Business Media (2013)
9. Darken, R.P., Peterson, B.: Spatial orientation, wayfinding, and representation. Handbook of virtual environments pp. 493–518 (2002)
10. Grasset, R., Looser, J., Billinghurst, M.: A step towards a multimodal ar interface: A new handheld device for 3d interaction. In: Proceedings of the 4th IEEE/ACM International Symposium on Mixed and Augmented Reality. pp. 206–207. IEEE Computer Society (2005)
11. Gunnarsson, A.s., Rauhala, M., Henrysson, A., Ynnerman, A.: Visualization of sensor data using mobile phone augmented reality. In: Proceedings of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality. pp. 233–234. IEEE Computer Society (2006)
12. Inselberg, A., Dimsdale, B.: Parallel coordinates: a tool for visualizing multidimensional geometry; 1990. San Francisco CA pp. 361–375

13. Kalkusch, M., Lidy, H., Knapp, M., Reitmay, G., Kaufmann, H., Schmalstieg, D.: Structured visual markers for indoor pathfinding. In: Augmented Reality Toolkit, The First IEEE International Workshop. pp. 8–pp. IEEE (2002)
14. Keim, D.A.: Information visualization and visual data mining. Visualization and Computer Graphics, IEEE Transactions on 8(1), 1–8 (2002)
15. Knapp, M., Reitmayr, G.: Signpost 2 - mobile ar navigation system (2005), `http://studierstube.icg.tugraz.at/projects/mobile/SignPost2/`
16. Lee, G.A., Billinghurst, M., Kim, G.J.: Occlusion based interaction methods for tangible augmented reality environments. In: Proceedings of the 2004 ACM SIGGRAPH international conference on Virtual Reality continuum and its applications in industry. pp. 419–426. ACM (2004)
17. Narzt, W., Pomberger, G., Ferscha, A., Kolb, D., Müller, R., Wieghardt, J., Hörtner, H., Lindinger, C.: A new visualization concept for navigation systems. In: User-Centered Interaction Paradigms for Universal Access in the Information Society, pp. 440–451. Springer (2004)
18. Piekarski, W., et al.: Interactive 3d modelling in outdoor augmented reality worlds. Citeseer (2004)
19. Schaer, P., Thum, M., et al.: State-of-the-art: Interaktion in erweiterten realitäten (2007)
20. Shneiderman, B.: The eyes have it: A task by data type taxonomy for information visualizations. In: Visual Languages, 1996. Proceedings., IEEE Symposium on. pp. 336–343. IEEE (1996)
21. Szalavári, Z., Gervautz, M.: The personal interaction panel–a two-handed interface for augmented reality. In: Computer graphics forum. vol. 16, pp. C335–C346. Wiley Online Library (1997)
22. Tenmoku, R., Kanbara, M., Yokoya, N.: Annotating user-viewed objects for wearable ar systems. In: Proceedings of the 4th IEEE/ACM International Symposium on Mixed and Augmented Reality. pp. 192–193. IEEE Computer Society (2005)
23. Thompson, C.B.: Descriptive data analysis. Air medical journal 28(2), 56–59 (2009)
24. Tonnis, M., Sandor, C., Lange, C., Bubb, H.: Experimental evaluation of an augmented reality visualization for directing a car driver's attention. In: Proceedings of the 4th IEEE/ACM International Symposium on Mixed and Augmented Reality. pp. 56–59. IEEE Computer Society (2005)
25. Woods, E., Mason, P., Billinghurst, M.: Magicmouse: an inexpensive 6-degree-of-freedom mouse. In: Proceedings of the 1st international conference on Computer graphics and interactive techniques in Australasia and South East Asia. pp. 285–286. ACM (2003)
26. Xiang, Y., Chau, M., Atabakhsh, H., Chen, H.: Visualizing criminal relationships: Comparison of a hyperbolic tree and a hierarchical list. Decision Support Systems 41(1), 69–83 (2005)
27. Yi, J.S., ah Kang, Y., Stasko, J.T., Jacko, J.A.: Toward a deeper understanding of the role of interaction in information visualization. Visualization and Computer Graphics, IEEE Transactions on 13(6), 1224–1231 (2007)

Augmented Reality Development Kits

Cevat Can Undeger*
Advisor: Matthias Berning†

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

Abstract. This paper describes some of the most popular and the most used Augmented Reality(AR) Tool-kits in detail and compare them in order to reveal functional differences among them. The reader of this paper will get an understanding of the augmented reality systems and the toolkits such as Studierstube, ARPA, Wikitude,ARtoolkit,DART and APRIL

Keywords: Augmented reality tools, Studierstube, ARPA, Wikitude,ARtoolkit,DART, APRIL

1 Introduction

This paper describes the current state-of-art of tool kits that are used for developing augmented reality experience. It also describes how some of the most popular augmented reality tool kits work and for whom they are suitable.

What is augmented reality? Augmented reality can be seen as a variation of the virtual reality. It combines the virtual objects and real objects in a real world in real time in an interactive way. The idea behind these systems is to keep track the orientation and the position of the user in order to improve the user's perception of the real world by combining virtual objects(computer generated graphics) related to the real environment with real objects. The placement of virtual objects in real environment can be realized either according to the position of a real object or according to the position of a predetermined spot in the real environment.

At the beginning most of the AR systems were using head mounted displays. With advances in the field of flat screen technology and in the field of smart phones, the AR systems have begun to use flat screens to display the mixed reality. In either way some essential equipments, which provide the important input information, are needed to implement and to run an augmented reality system or application. Hand-held displays provide an interesting way to present the augmented reality to the user. Specially smart phones and other mobile devices are much easier to carry around then the head mounted displays.

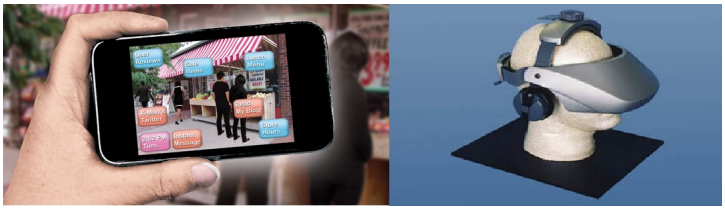


Fig. 1. a head mounted display **(a)** on the left and a handheld device **(b)** on the right which is used to display the AR system.

Additionally it is also much easier to handle the mobile devices than the former devices. Those changes in the field of technology had an impact on augmented reality tool kits. Newly introduced tool kits are much more related to the mobile augmented reality than the old ones. As a result the usage area of augmented reality systems have been widened. In particular, those systems are now integrated both into the video games to make the user feel more real and also they are used in mobile platforms to place a video or any kind of virtual object on a picture for various kinds of purpose. Over the past years the mobile AR applications drew attention of the designers, who don't have any programming experience in any programming language. Subsequently the companies introduced new tool-kits for those designer with no prior programming knowledge or with little programming knowledge. They have very simple user interface like web editors for designers to place the virtual objects into a photograph and create the AR application.

DART(The Designer's Augmented Reality Tool-kit) is developed by Blair MacIntyre, Maribeth Gandy, Steven Dow, and Jay David Bolter in 2003. It is a set of software tools to design and implement rapidly augmented reality experiences and applications. The dart is designed to make the complete development process easier for the designer, from the initial design to the final product and it is built on the Macromedia Director.

Wikitude is a mobile AR software with a development studio Wikitude Studio which is a powerful AR creator and content management tool. It makes the creation of AR applications easier for everyone with no prior programming knowledge. The drag and drop method is used to place the virtual objects into the real environment and the development tool-kit provides image recognition and tracking, 3D model rendering and video overlay and its SDK is available for both android and IOS devices.

APRIL is an XML based language to create content rich augmented reality applications and also interactive presentations. It allows the developer of the presentations to implement them with much less effort than the conventional implementation , while at the same time it supports an increased flexibility and debugging possibilities.

ARToolkit is a software library for creating AR applications. These applications are the applications that involve the overlay of the virtual imagery. This tool-kit has a multi-platform library, a simple graphic library, a complete set of samples and utilities and much more features.

Studierstube system provides a framework for supporting development of complex and distributed AR experiences and applications for multiple users. It is built on the top of the Open Inventor Tool-kit and the Open Tracker library. The tool-kit is composed of a collection of C++ classes and those classes extend the open inventor library and give programmers an application programming interface.

ARPA is a framework for AR on mobile platform which also have a SDK for developers.With this tool it is possible to design location based and marker based AR applications.

Throughout this paper the detailed description of above mentioned AR tool-kits is presented , since they are the most popular and most used tool-kits for developing AR experience.

2 Augmented Reality Tool-kits

2.1 DART

DART (The Designer's Augmented Reality Tool-kit) was introduced by Blair MacIntyre , Maribeth Gandy, Steven Dow and Jay David Bolter in 2003 at Georgia Institute of Technology,USA [6]. The DART is built on the top of the Macromedia Director, which is a popular multimedia development environment among the designers. Creators of the DART collaborated with the designers to address the most significant problems faced by the designers while they are working with the AR in the real world, and the creators tried to fix those problems with the DART. DART is written in interpreted scripting language(Lingo), so that the designers can change the code easily to fulfil their needs.

DART aims to let designers to work with the augmented reality directly and effectively. Macromedia Director is a widely used multimedia environment that has a rich feature set, an interpreted scripting language called Lingo. It's SDK is freely available for everyone, so that it can be extended freely. Besides it is a general purpose authoring tool for Shockwave content. Additionally there are lots of supporting material such as books, Xtras(plug-ins), offical and unoffical web site and mailing lists. For those reasons the AR technologies such as live video, visual marker tracking, overhead tracking information are embedded into Macromedia Director in DART to use the advantages of the director [7]. More importantly, most of the designers don't need to learn a new tool since they are already experienced with the Macromedia Director. Since it is integrated into the director and is aimed to complement the typical development process of the director, the designers can use their usual tools to create contents.

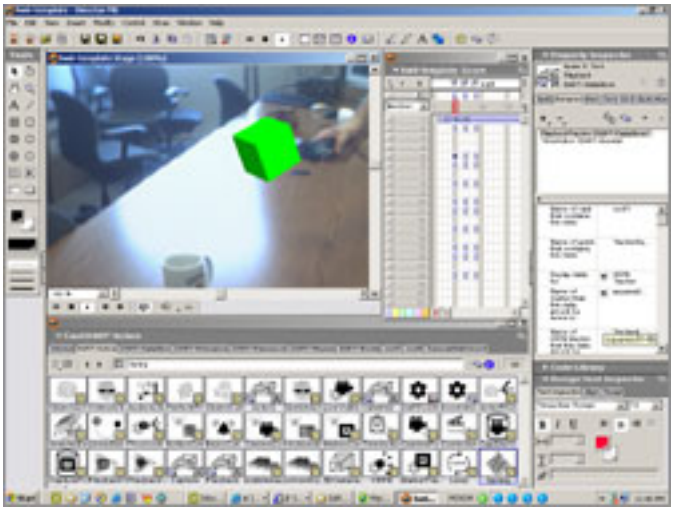


Fig. 2. A caption of the DART tool-kit

In DART it is possible for the designer to choose some of the parts of the tool-kit to use and modify them to suit their needs. The power of the tool-kit comes from a collection of behaviours which are written in interpreted scripting language Lingo. These behaviours can be easily integrated with the other contents of the director and can be easily modified, which allows designers to extend the DART as they need. DART allows designers to create rapidly informal content from 2D storyboards using 3D animatic actors. Animatics are sequences of the 2D storyboards which are often synchronized with audio. On the other hand animatic actors are sequences of sketches which are used as place holders for individual content elements to create an experience prototype prior to the final product. In AR development it is extremely difficult to deal with the multiple unrelated tracking technologies. DART offers an uniform access to these tracking sensor to ease to deal with them. The designers can mix those sensors and match them as they want. Like other components the code that integrates these sensing technologies is written in interpreted scripting language Lingo which allows designers to modify the code as needed if they encounter specific situations. In DART every actor(content element) is defined as an independent entity that is linked to other actors and to the sensing and tracking hardware, so that the modular experience development is encouraged by the tool-kit. Most of the tool-kits do not provide a support for managing the relationship between virtual objects and the physical objects in the real world environment. However, DART provides that support to manage the relationship between those object, which make it possible to position the virtual objects relative to the physical ones and both can interact using the underlying physic engine.

One of the most important feature of the DART is the ability to capture and to playback synchronized videos and sensor information, which allows designers to work with the application off-site, anywhere they want and test their experience against the snippets of the captured data.

Working with the AR in real time is difficult for most of the designers. Since everything are happening in a small amount of time, and the sensor data which are captured by the sensor hardware, could be noisy, the debugging and testing could also be extremely difficult. DART has a solution for that, DARTClock which is an abstract wrapper around the Macromedia Director clock. All time based scripts in DART are built on the DARTClock. The designers can stop the time of the experiences, change its speed or even rewind or move the time forward to a specific point. This gives the designer full control on the captured data and make their implementation process much more easier. Only the live video and sensor data cannot be paused or stepped through but the designers have the possibility to pause the rest of the experience. On the other hand if the designer captures the experiences, then DART allows them to control the time as the designers wanted.

Last researches showed that it is important to test the design ideas of AR system in the target site as soon as possible and as often as possible to create a stable system. With the DART's clock feature it is possible to realise in-situ experience test by combining informal(animatic) content with the captured video. Those animatic content helps to move the design storyboards into the real 3D world quickly and with them the designer can easily change or modify the design storyboards as the implementation processes. Having video playback feature the designer working with the DART can test, refine their design ideas without being in the target environment. This feature actually encourage the designers to work in situ because of working with the captured data is easier than working without such input.

To understand how DART is working, it is essential to know how the Macromedia Director works. Macromedia Director has an object oriented language called Lingo and the director is based on the stages in which the content is placed. The environment in the director is consist of a stage, multiple casts , where all the content are stored such as videos, 3D content, Lingo scripts, a score, which describes the timeline of the experience, and sprites which presents the cast members that are placed in the stage or in the score. As mentioned before Lingo is a interpreted scripting language and those scripts can be assigned to the elements in the score such as cast members, the stage,sprite or frames. It is director's job to generate graphical interface to edit the behaviour properties based on structured comments in the script. Score is the main mechanism in the director, which controls the timeline of the application. Adding behaviours to the sprites can be done by dragging the behaviour script on to the sprites that are placed on the score. In particular it is also possible to place scripts on individual frames as well as on an entire application. The idea behind the placing behaviours on the sprites is to modify the sprite's appearance or its reaction to inputs such as mouse buttons. The output of the application is represented in

the stage, which allows designers to modify their application easily. The most important part of the director is the possibility to modify the initial behaviours using the interpreted scripting language Lingo. The tool-kit provides some initial behaviours which can be changed by the developers via scripts and the designers have also the possibility to create their own behaviours from scratch using scripts written in Lingo. This approach allows both developing applications rapidly by modifying sprites directly and also advanced programming via scripting. In particular it also makes it easier to extend or modify the application during the test by simply dragging and moving elements on the score and dropping the behaviour scripts on the sprites. With the use of the director's automatically created property pages, the designers can set parameters to the behaviours such as parent object in the scene, the linkage of an object to incoming tracker data. Having behaviours added to the sprites can make the object to react an event. Behaviours are also used to configure the trackers and cameras, and to control the video and playback data.

The main mechanism to use the DART is the DART Xtra which is a C++ based director plug-in, that provides an interface to create AR applications. This includes real time live video capture, the ability to feed a live video stream into the texture memory of a Shockwave3D scene , support for visible marker tracking on live video streams and an interface to a wide collection of other tracking and sensing systems. The DART Xtra cannot be accessed directly but the framework provides those functionality to the designers. The designers using this framework are able to control the 3D camera in Shockwave3D world and they can also set the camera parameters for live video capture. They can also decide whether to do marker tracking and the VRPN tracking or not.

Currently DART has support for 3D models, audio, video based content, and animatics. DART has currently 3 initial actor to create or to modify the content, "Object Actor", "Sketch Actor", "Video Actor". The "Object Actor" is used to add a 3D object to the real environment. In DART the physical objects are rendered into the z-buffer before rendering the virtual objects, which allows physical objects in the real world to occlude virtual objects. With DART-Physics behaviours it is possible to specify the physics properties of actors. The "Video Actor" creates a visual representation of video onto the polygon in the 3D environment. Using some of the Directors feature allows the frames to be rewound rapidly with the alpha information intact. With the "Sketch Actor" the implementation of the animatic character can be realised, which is a key feature for rapid design exploration. A "Sketch Actor" shows a sequence of sketched image with a timing information which is provided by the designer.

DART has both a set of event based actor model that are commonly used in the virtual reality systems and also additional event that are suitable to the set of actors that are currently supported. The DART-Event behaviours consist of two part: "cues" and "actions". "Cues" are the events that broadcast when some conditions are fulfilled and "actions" are the actions that will be fired in response to a certain "cue" or in some basic condition. For example a cue could fire when a key is pressed on the keyboard. Also other commonly used cue and

action sets are predefined, which makes the implementation of the cue-action pair easier. The designer also have the possibility to create new cue-action pair other then these cue behaviours.

DART handles the tracking data in two interesting way. First of all the marker tracker tracks the markers and gives the position of the marker in world coordinates, but not relative to the camera. Hence no matter how the camera is positioned, the objects that are attached to markers, are positioned in the stable coordinate system. It is both easier for the designer and for the application to track of the object, since they are fixed relative to the world coordinates. Second of all DART provides a mechanism which allows the designer to specify the transformation that are applied to the reports from the tracking devices before they reaches to the application. Since every tracking system reports the tracking information in different format, in this way the designers are able to adjust the format for all tracking information.

2.2 Wikitude

Wikitude is a company which provides mobile augmented reality technologies[12]. It is founded in Salzburg, Austria in 2008. Its initial focus was to provide location based augmented reality experiences via their app Wikitude World Browser which is the first mobile application that uses the location based approach to augmented reality. In 2012 the company changed their focus with the release of the Wikitude SDK which is a framework to develop augmented reality experiences utilizing image recognition and tracking, and geo-location technologies. The core product of the Wikitude company is the Wikitude SDK which allows designer to create AR apps and is published in 2012. The Wikitude all in one augmented reality solutions includes image recognition and tracking, 3D model rendering, video overlay and location based AR [12]. The SDK has everything that the designer need to build their augmented reality applications. It has support for IOS, Android and Smart glasses. With this kit the designers can create their app based on the well known web standard such as HTML 5, Javascript and CSS3. Adding videos, 3D models,images and entire HTML code snippets to the AR application is easy to achieve with this tool-kit. The SDK can work with various of plug-in and framework which ease the development process.It has extensions for Apache Cordova, Appcelerator's Titanium and Xamarin.Although the Wikitude documentation provides information for all aspects of the SDK to help the designers with their creation, it is hard to find any additional supporting document on the internet.

The SDK has four versions : SDK Lite, SDK Pro, SDK Pro+, SDK Pro+ Unlimited. The lite version has a price of 590 euros, which has the features such as location based services, augmentations and visualization without 3D support,extended functionality such as "take a screen-shot", "camera zoom", "front and back camera support"; support for all devices except smart glasses, additional tools such as ADE.js for desktop debugging; target management tools, extension plug-ins, support for the development except premium support and free upgrade of SDK. Additionally it is licensed for one app on one platform

and if you want to use it for android and ios, then the price doubles itself. The SDK Pro version has all of the feature of the lite version and has some additional features such as marker recognition and tracking,augmentation and visualization for 3D content, support for smart glasses, 3D model encoding tool. This version has a one time price of 990 euros. The Pro+ version has the additional features such as premium support and free upgrade of SDK. However this version has a price of 1980 euros per annum and it can used for one application for two platforms. The only difference between the pro+ and pro+ unlimited version is that the unlimited version can be used for unlimited number of application of both platforms and has a price of 4490 euros per year.

The Wikitude has also another tool kit to create mobile augmented reality experiences called Wikitude Studio which is accessible via http://studio.wikitude.com/ [11]. The Wikitude Studio is a web browser editor that is used to create AR experiences.

Fig. 3. Shows the wikitude studio working on an augmented reality experience

The designers who use this studio, do not need to know how to code. It is a web based editor, so it is not required to install any software to your desktop. The drag-and-drop method is used in the editor to move the content. The created AR experience can be published within the Wikitude App. Additionally the designers have the opportunity to create their own application to publish their AR experience by combining Wikitude Studio with the Wikitude SDK. The Wikitude Studio has monthly and yearly plans that allows designers to create more than one experience according to their subscription.

2.3 APRIL

APRIL is a high level framework for creating authoring augmented reality presentations, which is presented by Florian Ledermann and Dieter Schmalstieg at

Vienna University of Technology in 2005 [10]. It uses XML based language to create interactive AR content. The used XML based language acts as a bridge between the parts that are already exist. The hardware setup, the content of the experience and its interactive capabilities can be described by the XML elements that are provided by APRIL. The main component of the experience are divided into four top level groups: setup, cast, story, behaviours and interactions and they can be modified easily as the developer needs.

The story is the representation of the temporal structure of the experience which includes the individual scenes. The scene includes actors which executes the predefined sequence of behaviours. In order to advance the story the change of the scenes is triggered by user interactions. With the use of the hardware description it is possible to hide the details of the underlying system from the user, thus using various kind of hardware descriptions allows the experience to run on different hardware without changing the content. In APRIL the actor can be a virtual 3D object as well as it can be a video clip or sound track. The behaviours describes the changes of the actors during the execution of the experience. Some parts of the behaviour can be predefined by the developer and some other parts of the behaviours can be modified during the execution in response to user interactions. Stages in APRIL are defined as the top level spatial containers for the content of a presentation. The author of the experience can define the spatial relationship to other stages and to the world as well as he or she can set the rendering techniques. To increase the portability of the experience the roles are used in APRIL. Each stage has a role which is assigned to it from a list of roles, that describes the functions of that stage. The geometry of the real world can be defined with the "world" container in APRIL and it can be realised by modelling or by scanning with 3D scanners.

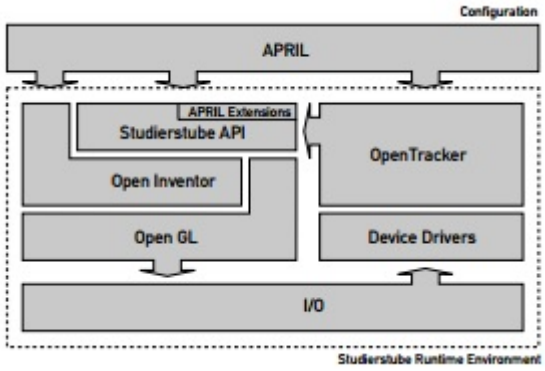


Fig. 4. Shows the runtime environment of the APRIL

For configuring the tracking devices the XML based language OPENTracker is used. It only sets tracking devices and defines their relationship but do not defined how to transform the collected data for the application. Additionally there is an element called "pointer" which is provided by APRIL,that allows developer to choose a pointing technique from several pointing techniques. Thus the developer has the opportunity to use a pointing technique other than ray picking. Another feature of the APRIL is to let developer to choose whether the content should be rendered in 3D or as a 2D texture. The developer can also decide whether the content should be placed in according to the world coordinate system or it should be located at a fixed position. Transition between scenes are triggered by the user interactions and APRIL provides built in high level user interactions, that starts the transition when some conditions are met. Additionally there are also automatically called "pseudo interaction" such as time-out, always, that trigger or disable certain transitions.

APRIL applications are consist of independent components which can be used in other applications easily. These components should defined outside of the experience and they are called templates. Templates are written using existing ASCII based host language to describe the intended content plus additional XML-Markup to define the interface of the component. An APRIL component includes two main part : the interface of the component and one or multiple implementations of the component. The interface definition provides the input and output fields of the component. Having interface definition the components can be re-used in different applications without changing their implementation. APRIL separates the content of the application from the description of the hardware configuration on which the application should run, which makes it possible to run the application on different systems without changing the content of the application. The developers who are using the APRIL, can realize the experience with much less effort than with a conventional approach, at the same the the flexibility and the debugging possibilities are increased. The component that are defined in APRIL are itself platform independent and thus it can make use of any language. Another important aspect of APRIL is that the behaviours are bounded to the scenes and each scene can have an arbitrary number of behaviours that are arranged on a local timeline which allows behaviours to be performed sequentially. The work flow in APRIL can be divided into separate parts that allows different developer to contribute to the implementation of application. An augmented reality application that is encoded in APRIL format is transformed into files in the Studierstube complaint format.

2.4 ARtoolkit

ARtoolkit is the world's most widely used tracking library for augmented reality systems[4]. It is a C/C++ software library which makes it possible for developers to implement augmented reality systems easily [5]. It was introduced by Hirokazu Kato of Nara Institute of Science and Technology in 1999 and was released by the University of Washington HIT Lab. The source code of the ARtoolkit can be found on GitHub as well as the compiled SDK can be found on the ARtoolkit

web page. The supporting documents are also available on that web page and a community forum provides help for the developers.

One of the most difficult part while developing an AR system is that to determine precisely the viewpoint of the user to align the virtual objects in the real world. The ARtoolkit uses a computer vision technique to determine the position of the camera and orientation relative to the shapes and gives developer the possibility to overlay the virtual objects. The toolkit currently has support for 2D barcode, multimarker, feature tracking and classical square marker. The tool-kit can used on multiple platform such as Windows, Mac OS X, Linux, iOS and Android, while each tool-kit for specific platforms has the similar functionalities and the performances of the tool-kits are depended on the hardware specifications of the platform. Additionally it is also possible to port the ARtoolkit to new and experimental platforms. Despite the some other tool-kits ARtoolkit supports video and optical see through augmented reality.

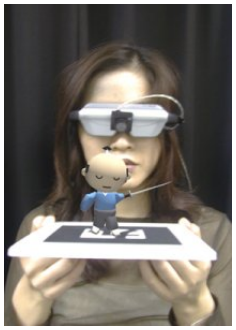


Fig. 5. An augmented reality experience that is created with ARtoolkit.

Video see through describes the augmented reality systems on which virtual images are overlaid the live video of the real world. On the other hand optical see through AR describes systems on which the computer generated graphics overlaid directly the view of the real world and most of them requires an head mounted display. The ARtoolkit has much more features than the other augmented reality tool-kits. The tool-kit can overlay 3D virtual objects on real markers. It is a multi platform library with multiple input sources, multiple format,multiple camera tracking, GUI initializing interface. It has an extensible markers pattern approach, an 3D VRML support, a fast rendering based on OpenGL, a simple and modular API(in C), a simple graphic library, an easy calibration routine and a complete set of samples and utilities. It is an open source with GPL licence for non-commercial usage.

2.5 Studierstube

Studierstube is a framework which supports the development of complex, distributed augmented reality experiences for multiple users[8]. The Studierstube is built on the top of the Open Inventor Tool-kit and the OpenTracker library. It is published by Dieter Schmalstieg, Anton Fuhrmann, Gerd Hesina Zsolt Szalava ri,L. Miguel Encarnacao,Michael Gervautz,Werner Purgathofer in 2002. The initial studierstube system which was described by Schmalstieg,Fuhrmann, Szalava'ri, and Gervautz (1996) and Szalava'ri, Fuhrmann, Schmalstieg, and Gervautz (1998), was one of the first collaborative AR system. More than one user can use the system at the same time sharing the virtual space that is populated with 3D data[1]. The system renders the same virtual space for all users according to their viewpoint using the tracking data.

The Studierstube which means in german "student room", is a project to create 3D environment and allow multiple user to interact with the system simultaneously using virtual reality devices. The rich development graphical environment allows developer to create new interaction styles[9]. OpenInventor is an object oriented 3D tool-kit which is used to create interactive 3D-graphics application.The tool-kit provides a library of objects that can be modified or extended by the developer. The OpenInventor kit is used for rendering 3D objects. The OpenTracker library provides an abstraction layer for input devices which include drivers for a wide range of devices such as magnetic, ultrasonic and inertial trackers. With the use of the library the transformation of the data that are retrieved by the tracking devices, is done in order to met the requirements of a particular application. To develop an new AR application with the Studierstube the developers need to write subclasses that inherit from C++ classes, and to compile them as shared objects that can be further dynamically loaded into runtime framework. The implementation of the application logic is consist of a set of compiled classes. The Studierstube system does not provide any scripting method for specific behaviour of applications. To create an advanced augmented reality system with this tool-kit the developer should have experience in advanced programming and it is also essential to have an understanding of the underlying libraries.

2.6 ARPA

ARPA is a augmented reality platform which allows the overlay of the digital information on the real world environment, using a comprehensive detection, recognition and tracking solutions in videos in execution time[2]. It is published by the ARPA Solutions company. The ARPA SDK is an augmented reality software for developing interactive experiences around the user. The mobile applications that are created with the SDK are used for various kind of purposes such as augmented world geo apps, advertising, marketing campaigns, museum,games etc. It supports image detection, recognition and tracking, 3D face tracking and detection, GPS and sensor tracking. With the use of the ARPA SDK for smartphones, tablets and AR glasses the developers can build AR projects in an easy

way. Additionally the product ARPA plug-in for Unity 3D provides a powerful tool to create AR games for desktops,smart-phones,tablets and AR glasses.

ARPA SDK is available for both android and iOS. With the SDK GPS version it is possible to create augmented reality applications based on geo-location. Thus the applications are able to find point of interest in the near of the users position and show them information about how to get there using three operating modes : AR mode, list mode, map mode. The AR mode show the points of interests over the camera image in real time in order to let the user to interact with the point of interests(POIs) easily.

On the other hand list mode shows a list of the POIs ordered by distance. Lastly the map mode display the POIs with a marker to show the location of each POI and the location of the user.

ARPA SDK is available in four different versions. The AR sdk trial can be used freely and it has the following features: Image recognition, detection and tracking,multi-detection and multi-tracking. The pro version has only additionally 6 months support and it costs 499 dolars per application. With the use of the AR GPS SDK version it is possible to develop AR application based on the geo-location. However this version does not have the following features: Image recognition, detection and tracking,multi-detection and multi-tracking. On the other hand it is free of charge and is limited to 15 POIs. There is also a Glass SDK version which has only support for android[3]. It can detect the markers on images, however it is not suitable for geo-location based applications. Furthermore it can used freely until 15 POIs.

3 Comparison of AR Development Tool-kits

In this section AR development toolkits that are described in this paper, will be compared based on different features.

3.1 Platform Comparison

Artoolkit, ARPA SDK, Wikitude SDK support developing augmented reality applications for iOS, Windows and Android platforms.

Platforms	Artoolkit	ARPA SDK	Wikitude Sdk	Studierstube	DART
IOS	✓	✓	✓	×	×
Android	✓	✓	✓	×	×
Windows	✓	✓	✓	✓	✓
Mac OSX	✓	×	✓	×	×

Fig. 6. Platform comparison of the tool-kits

Artoolkit and Wikitude SDK have also support for Mac OSX platform. Studierstube and DART have only support for Windows platform.

3.2 License

ARPA SDK, Wikitude SDK and DART are not open source tool-kits. ARPA SDK has a free version as well as versions that require payment for use. DART and the ARtoolkit are available for free to the developers. Wikitude has four different versions which are not free.

Licence Type	Artoolkit	ARPA SDK	Wikitude Sdk	Studierstube	DART
Open Source	✓	×	×	✓	×
Free	✓	✓	×	×	✓
Commercial SDK	✓	✓	✓	✓	×

Fig. 7. License comparison of the tool-kits.

3.3 Tracking and Overlapping

ARtoolkit supports square marker tracking, multiple marker tracking and fast 6D marker tracking. However it does not have support for face detection and tracking, image recognition and geo-location based tracking. With ARtoolkit it is possible to overlap 2D and 3D content into the real environment.

Tracking and Recognition	Artoolkit	ARPA SDK	Wikitude Sdk	Studierstube	DART
Marker Tracking	✓	✓	✓	✓	✓
Geo-location based tracking	×	✓	✓	×	×
Marker Tracking on live video Stream	✓	×	×	×	✓
Image Tracking	×	✓	✓	×	×
Image Recognition	×	✓	✓	×	×
Multi Marker Tracking	✓	×	✓	✓	×
Face Detection and Tracking	×	✓	×	×	×

Fig. 8. Features of the tool-kits.

Overlapping Features	Artoolkit	ARPA SDK	Wikitude Sdk	Studierstube	DART
2D Object	✓	✓	✓	×	✓
3D object	✓	✓	✓	✓	✓

Fig. 9. Overlap features of the tool-kits.

Wikitude SDK and ARPA SDK have some common features. They both have support for image tracking and recognition, geo-location based tracking

and marker tracking. Additionally ARPA SDK has the ability to detect and track the face which is newly introduced in 2015. On the other hand Wikitude SDK has support for multiple markers.

DART has only support for marker tracking and marker tracking on live video streams, if the tracking and recognition features are considered. It can also overlap 2D and 3D into the real environment

Studierstube can place 3D object into the real environment but not 2D contents. Its multiple marker ability is a key feature of this tool-kit.

3.4 Advantages and Disadvantages

ARtoolkit Advantages:

Artoolkit has many supporting document available on the internet and its camera calibration routine is easy to achieve. Since the ARtoolkit is an open source, there are many new AR framework that are created using this SDK.

Disadvantages: The Artoolkit is written in C/C++, thus the developer should have an experience with programming languages C/C++. Additionally the marker tracking of this tool-kit it is not accurate even when the camera and marker are not moving. More importantly it does not have support for geo-location based tracking.

ARPA SDK Advantages: The support for GPS-based tracking and face recognition and tracking is a plus for this SDK, since none of the other SDKs have support for face detection and recognition.

Disadvantages: It is hard to find supporting document online and the web site of the owner company is not stable at all.

Wikitude SDK Advantages: Multiple marker tracking support is an advantage for this SDK. Additionally the people who don't have any programming experience, can use the web-browser editor Wikitude Studio.

Disadvantages: All of its versions require payment to use it.

Studierstube Advantages:

It is possible to create augmented reality experience for multiple user. This is a unique feature for this tool-kit. It is built on the top of the OpenInventor Tool-kit and OpenTracker library so it has a rich graphical support.

Disadvantages:

Overlapping 2D content into the real environment is not possible with Studierstube and it is only available for the Windows platform. More importantly it is essential to understand the underlying libraries.

DART Advantages:

The designers who are already familiar with the macromedia director, do not need to learn a new platform. The components of this tool-kit are written in

scripting language Lingo, so the designers can easily modify them. Additionally it is possible to capture and playback the captured data to test the experience.

Disadvantages:

It is only available for the windows platform and it does not have any support for image recognition and tracking.

4 Conclusion

With the advances in the technology the augmented reality system or applications are getting more and more popular and the tool-kit are evolving themselves to keep up with the newly introduced technologies. In the near future the technology will even more rapidly evolve thus tool-kits will change along or new tool-kit will be introduced. Many researches are focussed on creating authoring tool-kit to let people with no prior programming language like the Wikitude and ARPA do. The tool-kit that requires advanced programming skills will also be more popular among the developer as the AR tracking and displaying technologies develop, to create more complex AR experience for the users. Even though most of the development kits for the mobile platforms use drag and drop technology and no programming skills needed, but the other tool-kits like ARtoolkit requires them, it is hard to categorize the tool-kit according to these features so the developers should know what is best for them and choose the suitable one that met their requirements.

References

1. Presence, Vol. 11, No. 1, chap. pages 33-54. the Massachusetts Institute of Technology (2002)
2. ARPA: Arpa sdk (2015), http://www.arpa-solutions.net/en/ARPA_SDK
3. ARPA: User guide arpa glass sdk (2015), http://www.arpa-solutions.net/Developer_Guide/SDKGoogleGlass/Index.html
4. ARtoolkit: Artoolkit documentation, http://artoolkit.org/documentation/
5. ARtoolkit: Introduction to artoolkit, http://www.hitl.washington.edu/artoolkit/documentation/userintro.htm
6. Blair MacIntyre, Maribeth Gandy, S.D., Bolter, J.D.: Dart: A toolkit for rapid design exploration of augmented reality experiences. 4 ACM 1-58113-957-8/04/0010 (2004)
7. DART: Dart information, http://ael.gatech.edu/dart/
8. Dieter Schmalstieg, Anton Fuhrmann, G.H.Z.S.L.M.E.M.G.W.P.: The studierstube augmented reality project. Vienna University of Technology (2004)
9. Doxygen: Studierstube 4 documentation, http://studierstube.icg.tugraz.at/doc/stb4/
10. Ledermann, F., S.D.: April: A high-level framework for creating augmented reality presentations. In: Virtual Reality
11. Wikitude: Wikitude studio (2012), http://www.wikitude.com/products/studio/
12. Wikitude: Wikitude sdk (2014), http://www.wikitude.com/products/wikitude-sdk/

Effiziente dynamische Softwareupdates in Java

Lukas Hofmann*
Betreuer: Martin Alexander Neumann†

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

Zusammenfassung. Die zunehmende Auslagerung sensibler Daten in vernetzte Systeme und deren wachsende Präsenz im Alltag erfordern, dass diese Systeme sicher gegenüber Softwarefehlern und Angriffen sind. Die gleichzeitig steigende Nachfrage nach hoher Verfügbarkeit führt dazu, dass der klassische Ansatz, Prozesse während Softwareupdates zu beenden, nicht mehr ausreicht. Stattdessen sind zunehmend Methoden für dynamische Softwareupdates (DSU) gefragt, mit welchen Software ohne Zustandsverlust und ohne signifikante Unterbrechung zur Laufzeit aktualisiert werden können soll. Um einen Überblick zu aktuellen DSU-Systemen zu geben, stellt diese Arbeit drei Systeme für Java vor, die sich zum Ziel gesetzt haben, dynamische Softwareupdates effizient zu realisieren: JavAdaptor kann von außen über Schnittstellen der JVM einen darin laufenden Prozess aktualisieren, Jvolve nimmt als selbständige JVM direkt Änderungen am Prozess vor und Rubah kombiniert eine durch das zu aktualisierende Programm nutzbare API mit einem Hilfstool. Während die drei Systeme unterschiedliche Ansätze verfolgen, erreichen alle einen nur geringen Overhead während der Programmausführung und kurze Verzögerungen während des Updatevorgangs.

Schlüsselwörter: Dynamische Softwareupdates, Java, JVM

1 Einleitung

Das regelmäßige Bekanntwerden von Sicherheitslücken - nicht selten mit öffentlichkeitswirksamen Namen wie Heartbleed oder Shellshock - erinnert immer wieder daran, dass Software Fehler enthält, die teilweise weitreichende Auswirkungen haben können [5]. Umso wichtiger ist, dass diese Fehler zeitnah behoben und neue Softwareversionen ausgeliefert werden. Deren Installation läuft dabei meist nach dem selben Schema ab: Der fragliche Prozess wird beendet, der geänderte Programmcode eingespielt und der Prozess neu gestartet. Die damit verbundene Wartezeit und der Verlust des Programmzustandes - im Fall, dass dieser nicht zuvor gespeichert wird - kann bei der meisten Software für Endanwender und in unkritischen Umgebungen in Kauf genommen werden, was diesen aus softwaretechnischer Sicht einfachen Ansatz praktikabel macht. Anders ist es bei Software, die hochverfügbar sein muss, u.a. bei manchen Netzwerkdiensten

oder Betriebssystemen [3]. Ein Unterbrechen des zu aktualisierenden Prozesses kommt hier häufig nicht in Frage. Um dennoch das Installieren von Updates ohne Zustandsverlust und Unterbrechung zu ermöglichen, wird im Serverbereich häufig redundante Hardware eingesetzt, bei der z.B. eingehende Anfragen per Load-Balancer auf synchronisierte Rechner verteilt werden. Dabei können die Server nacheinander angehalten und aktualisiert werden, während der Dienst von außen weiterhin verfügbar bleibt [7]. Dieser hardwarebasierte Ansatz ist jedoch kaum auf Fälle außerhalb des Serverbereichs anwendbar und mit hohem Aufwand verbunden, weshalb eine Softwarelösung wünschenswert ist.

Hier setzt das Prinzip der dynamischen Softwareupdates (DSU) an. Dabei wird ein Prozess zur Laufzeit aktualisiert, d.h. dessen Programmcode wird durch eine neuere Version ersetzt und die Prozessausführung in dieser fortgesetzt, wobei der Prozesszustand, also z.B. Variablen und Datenstrukturen, erhalten bleibt. Dabei muss dieser ggf. in eine Form überführt werden, die äquivalent zur vorherigen und dabei kompatibel zu dem neuen Code ist.

Gegenstand dieser Abhandlung ist es, verschiedene DSU-Systeme für Java vorzustellen, die für sich beanspruchen, Updates effizient einspielen zu können, wobei speziell Updategeschwindigkeit und Ausführungsoverhead betrachtet werden. Dazu soll im zweiten Abschnitt zuerst auf Besonderheiten eingegangen werden, die beim Einsatz von Java auftreten, darunter das Java Virtual Machine Tool Interface (JVM TI), mit welchem der Zustand eines in der JVM ablaufenden Prozesses untersucht und gesteuert werden kann [2], sowie im Speziellen der HotSwap-Mechanismus.

Im dritten Abschnitt soll zuerst ein Überblick über die Kategorisierung von DSU-Systemen gegeben werden, die üblicherweise nach zwei Gesichtspunkten erfolgt: Ein Kriterium ist der Ablauf des Zustandsübergangs (lazy/eager) [10]. Außerdem wird zwischen einer Anwendungsebene und einer VM-Ebene unterschieden, d.h. zwischen Systemen, die von außen in die JVM und den Prozessablauf eingreifen (Anwendungsebene), und solchen, die selbst als JVM agieren (VM-Ebene) [12,15]. Darauf aufbauend werden mit Rubah, Jvolve und JavAdaptor drei konkrete DSU-Systeme vorgestellt, die einen effizienten Aktualisierungs- und Programmablauf zum Ziel haben [12,14,15]. Mit dieser Auswahl soll ein guter Querschnitt durch DSU-Systeme in Java erreicht werden: Rubah arbeitet auf Anwendungsebene und bietet einen lazy- sowie einen eager-Algorithmus an [12], JavAdaptor läuft ebenfalls als separate Anwendung und liefert einen eager-Algorithmus mit [14]. Jvolve dagegen wird als Erweiterung einer Java-VM implementiert und überführt den Programmzustand ebenso mittels eines eager-Algorithmus [15]. Mit Rubah (2014 [12]) und JavAdaptor (2011 [13], 2012 [14]) wurde dabei darauf geachtet, zwei aktuelle DSU-Systeme zu behandeln.

Im vierten Abschnitt wird näher auf den Aspekt der Effizienz eingegangen. Hier liegt der Fokus auf der Updategeschwindigkeit sowie auf Veränderungen in der Laufzeitgeschwindigkeit vor und nach einem Update.

2 Besonderheiten bei der Verwendung von Java

Wie die meisten verbreiteten Programmiersprachen enthält Java keinen integrierten DSU-Mechanismus. Anders als z.B. C oder C++ bietet Java jedoch Möglichkeiten, die die Implementierung von DSU-Systemen vereinfachen. Um diese vorzustellen, wird die Java-Architektur im Folgenden vor allem mit C verglichen.

2.1 Gegenbeispiel: C

Unter C geschriebener Programmcode wird typischerweise in nativen Maschinencode übersetzt, d.h. er läuft (wenn man von Betriebssystemfunktionen wie Speicherverwaltung und Scheduling absieht) ohne dazwischenliegende Abstraktionsschichten direkt auf der CPU. Diese Autonomie des Prozesses erschwert dabei durch fehlende Kontroll- und Steuermechanismen das Ändern des Programmcodes zur Laufzeit. Dynamische Software-Updates sind in C und ähnlichen Sprachen dennoch möglich. Ein Mechanismus, der dabei genutzt werden kann, ist Dynamic Linking [7]. Dabei können Codeteile, die im Vorfeld in Objektdateien ausgelagert worden sind, zur Laufzeit geladen werden. Ein weiteres Mittel ist das Einfügen von zusätzlicher Indirektion, z.B. Zwischenschritten, die durch den Compiler automatisch vor Funktionsaufrufen eingefügt werden und auf die jeweils aktuelle Version der Funktion verweisen [11].

2.2 Die Java-Plattform

Anders als bei den meisten verbreiteten kompilierten Sprachen werden Java-Programme nicht in nativen Maschinencode übersetzt, sondern in sogenannten Java-Bytecode. Dieser wird von der Java Virtual Machine (JVM) ausgeführt, einem abstrakten Computer, der im Gegensatz zu verbreiteten Prozessorarchitekturen Abstraktionen wie Arrays, Objekte und Klassen kennt [16].

Der Speicheraufbau eines Prozesses in der JVM ist z.B. mit dem eines C-Programms vergleichbar. Er ist in mehrere Bereiche aufgeteilt, zu denen die folgenden zählen [9]:

- **PC:** Ist threadspezifisch und speichert die Adresse des atuell ausgeführten Befehls.
- **Stack:** Jeder Thread des laufenden Prozesses hat einen eigenen Stack, der u.a. lokale Variablen enthält. Er ist vergleichbar mit dem Stack eines C-Programms.
- **Heap:** Wird von allen Threads eines Prozesses gemeinsam genutzt. Der Heap speichert z.B. Klasseninstanzen und Arrays.
- **Method Area:** Entspricht dem Text-Segment eines Systemprozesses und enthält den Code der Methoden des Programms sowie klassen- und methodenspezifische Daten.

Im Heap werden Speicherbereiche explizit reserviert (z.B. beim Instanziieren einer Klasse), aber automatisch wieder freigegeben. Dies erfolgt durch den in der JVM integrierten Garbage Collector, welcher nicht mehr benötigte Objekte aus dem Speicher entfernt [16].

Die von der JVM auf Maschinenebene bereitgestellte Objektorientierung kann von DSU-Systemen genutzt werden, um beim Update-Vorgang eine erhöhte Modularität zu erreichen: Während bei C-ähnlichen Sprachen z.B. komplette Programme [4] oder Objektdateien [7] betrachtet werden, können hier Klassen als funktionale Einheiten behandelt werden. Ein Werkzeug, welches zu diesem Zweck genutzt werden kann, wird im Folgenden vorgestellt.

2.3 Das Java Virtual Machine Tool Interface

Mit dem Java Virtual Machine Tool Interface (JVM TI) ist es möglich, Java-Programme von außen zu untersuchen und zu kontrollieren. Die Schnittstelle wird für C und C++ bereitgestellt und bietet u.a. Möglichkeiten, die Ausführung von Threads zu beeinflussen, eine Garbage Collection zu starten oder Operationen auf Instanzen einer Klasse auszuführen. [2]

Programme, welche die Schnittstelle benutzen, werden als sogenannte Agents in Form von Laufzeitbibliotheken implementiert, die beim Start der JVM geladen werden können. [2]

Ein Beispiel für einen Dienst, der über das JVM TI angeboten wird und zu einem gewissen Grad das Ändern eines Programms zur Laufzeit ermöglicht, ist der HotSwap-Mechanismus. Dieser ist zwar nicht standardisiert, aber dennoch in den verbreitetsten JVM-Implementierungen enthalten. Mit HotSwap kann eine Methode zur Laufzeit aktualisiert werden, was sich allerdings auf die Implementierung beschränkt. Eine Änderung ihrer Signatur sowie das Hinzufügen oder Entfernen von Methoden ist damit nicht möglich. [14]

3 Vorstellung: DSU-Systeme in Java

3.1 Kategorisierung

Grundsätzlich gibt es zwei verschiedene Ansätze, in Java ein DSU-System zu realisieren: als eigene JVM oder in Form eines zusätzlichen Programms bzw. einer API (Abb. 1).

Systeme wie Jvolve [15] und DCE VM [17] sind in Form einer *eigenen JVM* implementiert, welche ihre Update-Funktionalität fest integriert hat. Dieser Ansatz ermöglicht eine hohe Flexibilität und Freiheit in der Umsetzung, da ein Entwickler eines DSU-Systems nicht auf die Möglichkeiten angewiesen ist, die ihm eine Standard-JVM bietet. Stattdessen kann der laufende Java-Prozess ohne Zwischenschritte direkt manipuliert werden. Ein Nachteil dieses Ansatzes ist, dass keine beliebige JVM mehr genutzt werden kann. Nutzer sind hier auf Plattformen beschränkt, auf denen die spezielle VM läuft.

Eine andere Möglichkeit ist die Implementierung auf *Anwendungsebene* in Form eines zusätzlichen Programms und/oder einer Bibliothek, die von der zu

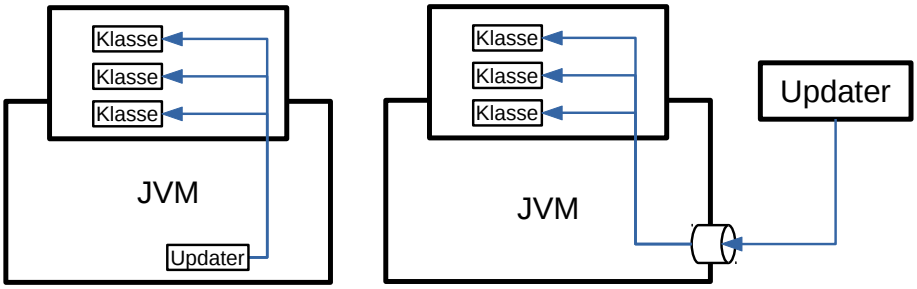


Abb. 1. Zugriff auf einen zu aktualisierenden Prozess durch eine in der VM integrierte Updatefunktionalität (links) oder ein externes Updatetool (rechts). Im zweiten Fall wird eine Schnittstelle der JVM genutzt.

aktualisierenden Software direkt eingebunden werden kann. Dabei kann eine unveränderte JVM genutzt werden, welche für die Steuerung des Java-Prozesses das JVM TI anbietet. Beispiele für DSU-Systeme, die auf diese Art umgesetzt wurden, sind JavAdaptor [14], Rubah [12], und JRebel [8]. Im Gegensatz zum obigen Ansatz ist dieser auf die Möglichkeiten und Schnittstellen beschränkt, die die JVM anbietet. Jedoch ist man hier als Nutzer in der Regel nicht auf eine spezielle JVM angewiesen.

Die Kategorisierung von DSU-Mechanismen kann auch nach anderen Kriterien erfolgen. Häufig wird hier der Ablauf des Updateprozesses herangezogen, im Speziellen die *Zustandsübertragung*. Ein dynamisches Softwareupdate läuft üblicherweise grob in zwei Schritten ab: Zuerst wird der Programmcode aktualisiert, indem dessen neue Version in den Speicher geladen wird. Im zweiten Schritt wird der Programmzustand in eine Form überführt, die den neuen Code berücksichtigt und mit der dieser umgehen kann. Dieser Zustandsübergang kann auf verschiedene Arten ablaufen, wobei zwischen einem *eager*- und einem *lazy*-Ansatz unterschieden wird [6]. Ein eager-Algorithmus aktualisiert den gesamten Prozesszustand in einem Schritt, wobei der Prozess währenddessen angehalten werden muss [12]. Ein lazy-Algorithmus aktualisiert Datenstrukturen und Referenzen erst dann, wenn sie benötigt werden [10]. Die Updatedauer wird hier so weit auf die Laufzeit verteilt, bis alle Daten und Referenzen aktualisiert worden sind, der Updatevorgang also abgeschlossen ist. Damit soll eine lange Wartezeit während des Updates vermieden werden [12].

3.2 JavAdaptor

Das DSU-System JavAdaptor ist in der Arbeit *JavAdaptor:—Flexible runtime updates of Java applications* [14] beschrieben. Es nutzt u.a. den HotSwap-Mechanismus der JVM, um den darauf laufenden Prozess zu aktualisieren.

Code-Update: Da die JVM standardmäßig keine Möglichkeit zur Änderung von Klassen bietet, bei der das Klassenschema geändert wird, aktualisiert JavAdaptor Klassen, indem die jeweils neue Version unter einem veränderten Namen geladen wird und damit neben der alten Version im Speicher liegt.

Zustandsübertragung: Damit der neue Programmcode verwendet wird, müssen alle Referenzen auf eine Klasse so geändert werden, dass die neue Version referenziert wird. JavAdaptor unterscheidet hier zwischen *short-lived* und *long-lived objects*. Mit ersteren sind beispielsweise lokale Variablen gemeint. Um diese auf neue Klassenversionen zu aktualisieren, genügt es, die Implementierung der entsprechenden Methode zu verändern. Dies geschieht durch HotSwap, welches im zweiten Abschnitt vorgestellt wurde.

Um Referenzen auf *long-lived objects* zu aktualisieren, werden mit Hilfe von JVM TI zuerst alle Instanzen der zu aktualisierenden Klasse und Objekte, die diese referenzieren, gesucht. Um außerdem referenzierende Klassen zu finden, die noch nicht instanziiert sind, wird Javassist [1] verwendet. Danach erzeugt JavAdaptor für jede Instanz der zu aktualisierenden Klasse eine entsprechende Instanz der neuen Version, und überträgt den Zustand (also Attribute) in diese. Wo ein automatischer Transfer nicht möglich ist, z.B. bei geänderten Attributtypen, muss manuell in Form einer Funktion ("mapping function"), welche vor dem Update vorbereitet werden kann, festgelegt werden, wie dieser ablaufen soll.

Schließlich müssen die Referenzen auf die Instanzen der neuen Klassenversion geändert werden, was nicht ohne weiteres möglich ist. Da die neue Version unter verändertem Namen geladen worden ist, ändert sich die Signatur der referenzierenden Klasse, denn Attribute, welche zuvor die alte Klassenversion als Typ hatten, verwenden nun die neue. Um dieses Problem zu umgehen, führt JavAdaptor sogenannte *Container* ein. Jede Klasse erhält beim Programmstart ein Attribut vom Typ eines Containers. Der entsprechende Container erhält dann beim Update eine Referenz zur neuen Version der jeweiligen Klasseninstanz. Per HotSwap wird außerdem jede Methode der alten Klasse so modifiziert, dass jeder Aufruf der Methode über den Container auf die entsprechende Methode der neuen Instanz umgeleitet wird.

Dieser Ansatz kann allerdings unpraktikabel sein, wenn Methodenparameter oder Rückgabewerte als Typ eine zu aktualisierende Klasse haben, da dabei die referenzierenden Klassen ebenfalls aktualisiert werden müssen. Für diesen Fall verwendet JavAdaptor sogenannte *Proxies*. Diese sind Unterklassen von jeweils einer zu aktualisierenden Klasse und können deshalb in Rückgabewerten und Parametern anstelle dieser verwendet werden. Ähnlich wie Container verweisen sie wiederum auf die jeweils aktuelle Klassenversion und geben Methodenaufrufe an diese weiter.

Der Zustandsübergang läuft in Form eines *eager*-Algorithmus ab, d.h. dass das gesamte Programm in einem Schritt aktualisiert wird. Werden mehrere Klassen aktualisiert, kann JavAdaptor das Erstellen der neuen Klasseninstanzen und die Übertragung des Zustandes auf diese parallelisieren.

Beschränkungen: Bei Programmen mit mehreren parallel arbeitenden Threads kann es zu Problemen kommen, u.a. da HotSwap aktive Methoden nicht verändert. Greifen z.B. zwei Threads synchronisiert auf das selbe Objekt zu, könnte die entsprechende Methode des einen Threads beispielsweise bereits auf dessen neue Version aktualisiert sein, während die laufende Methode des anderen Threads noch die alte Version nutzt. Diese Situation kann zu einem Deadlock führen.

Ein anderes Problem, welches sich ergeben kann, ist der Verlust des Zustandes. Wird ein Objekt aktualisiert, d.h. eine Instanz der neuen Klassenversion erstellt und der Zustand übertragen, während eine Methode, die dieses Objekt nutzt, auf einem anderen Thread noch läuft, finden dort noch Zugriffe auf die alte Version statt. Etwaige Zustandsänderungen werden dann nicht auf die neue Version übertragen und gehen verloren.

Lösungsansätze für diese Probleme werden allerdings bereits diskutiert. Im Fall von möglichen Deadlocks gibt es z.B. die Möglichkeit, nicht die referenzierten Objekte, auf die synchronisiert zugegriffen wird, zur Threadsynchronisation zu verwenden. Stattdessen kann man dazu spezielle Objekte einführen, auf die die referenzierten Objekte verweisen. Diese Verweise bleiben beim Update einer Klasse erhalten, wodurch ein Deadlock vermieden wird.

3.3 Jvolve

In dem Paper *Dynamic Software Updates: A VM-centric Approach* [15] wird mit Jvolve ein DSU-System vorgestellt, welches eine eigene JVM mitbringt. Es basiert auf der Jikes RVM.

Vorbereitung: Jvolve liefert ein Programm mit, welches die Unterschiede zweier Programmversionen untersucht. Dabei generiert es automatisch Funktionen, welche den Zustand je einer Klasse, bzw. ihrer Instanzen, auf ihre neue Version übertragen. Bei Attributen mit unverändertem Typ wird der alte Wert verwendet, neue Attribute erhalten einen Standardwert. Diese Funktionen können manuell verändert und angepasst werden.

Update-Voraussetzungen: Updates finden hier nicht zu beliebigen Zeitpunkten statt, sondern an *safe points*, also Stellen in der Programmausführung, die bestimmte Kriterien erfüllen: Erstens muss ein *VM safe point* gegeben sein, der dadurch charakterisiert ist, dass der Garbage Collector und Thread Scheduling ausgeführt werden können. Zweitens dürfen bestimmte Methoden sich währenddessen nicht in der Ausführung, d.h. auf einem Stack, befinden. Dazu zählen Methoden, die selbst aktualisiert worden sind oder sich auf aktualisierte Klassen beziehen. Weiterhin können Entwickler selbst Methoden bestimmen, die während des Updates nicht ausgeführt werden dürfen.

Erstere Bedingung - das Erreichen eines VM safe points - wird bereits durch Jikes RVM beim Unterbrechen der Ausführung durch den Scheduler erfüllt, indem die Ausführung nur an bestimmten Stellen angehalten wird, z.B. am Beginn

oder Ende von Methoden. Um weiterhin sicherzustellen, dass währenddessen keine kritischen Methoden ausgeführt werden (zweite Bedingung), baut JVOLVE auf dieser Funktionalität auf: Wird durch ein bevorstehendes Update die Prozessausführung an einem VM safe point angehalten, wird geprüft, ob eine der fraglichen Methoden aktuell auf einem Stack liegt. Falls ja, wird die Ausführung des entsprechenden Threads wieder aufgenommen. Sobald das Ende der Methode erreicht ist, wird der Thread wieder angehalten und die Prüfung erneut durchgeführt. Dieser Vorgang wird so oft wiederholt, bis ein safe point erreicht ist und das Update durchgeführt werden kann oder durch einen Timeout-Mechanismus abgebrochen wird. Bei manchen sich auf aktualisierte Klassen beziehende Methoden ist es stattdessen allerdings auch möglich, den Code auf dem Stack direkt dahingehend zu verändern, dass die neue Version referenziert wird.

Code-Update: Jikes RVM speichert zu jeder Klasse Metadaten, darunter den sogenannten *type information block* (TIB). Jede Methode wird bei ihrer ersten Ausführung in nativen Maschinencode übersetzt. Der TIB enthält dann einen Verweis zum entsprechenden Code.

Ändert sich bei einem Klassenupdate nur die Implementierung von Methoden, werden die Metadaten angepasst und der TIB als ungültig deklariert, damit die Methoden bei der nächsten Ausführung in ihrer aktuellen Version neu übersetzt werden. Bei einem umfangreicheren Update, bei dem sich die Signatur einer Klasse ändert, wird die alte Klassenversion umbenannt und deren Metadaten für den Garbage Collector freigegeben. Die aktuelle Version wird mit neuen Metadaten in die VM geladen.

Zustandsübertragung: Als Garbage Collector verwendet JVOLVE einen in Jikes RVM eingebauten sogenannten "semi-space copying collector". Dieser kopiert alle referenzierten Objekte auf einen neuen Heap, welcher nach der Garbage Collection genutzt wird. JVOLVE setzt hier an und erstellt beim Kopieren eines Objekts in einer alten Klassenversion zusätzlich eine Instanz der neuen Version (Abb 2). Danach werden die Funktionen, welche den Zustand auf die neue Version übertragen (s.o.), ausgeführt. Die nun überflüssigen alten Objekte werden bei der nächsten Garbage Collection entfernt.

Beschränkungen: Der Ansatz, Updates nur an bestimmten Stellen in der Ausführung zu erlauben, kann dazu führen, dass in Extremfällen ein Update nicht ausgeführt werden kann. Dies wurde z.B. im Test mit dem Webserver Jetty und dem CrossFTP-Server festgestellt, wo jeweils eine bestimmte Methode nahezu ständig auf einem Stack liegt. Ein safe point wurde dabei bei einer Version von Jetty nie und bei einer CrossFTP-Version nur bei niedriger Auslastung erreicht.

Eine funktionale Einschränkung von JVOLVE ist, dass in der Vererbungshierarchie einer Klasse keine Klassen vertauscht werden dürfen.

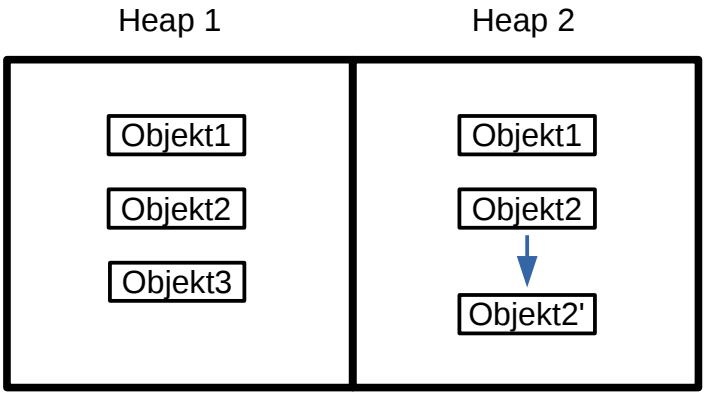


Abb. 2. Verwendung des Garbage Collectors durch JVolve: Im Beispiel wird Objekt 3 bei der Garbage Collection entfernt, also nicht auf den neuen Heap kopiert. Objekt 2 soll aktualisiert werden und wird deshalb zuerst unverändert kopiert. Danach wird eine entsprechende Instanz der neuen Klassenversion erzeugt und der Zustand darin übertragen.

3.4 Rubah

In dem Paper *Rubah: DSU for Java on a Stock JVM* [12] stellen Luís Pina und Luís Veiga das DSU-System Rubah vor, welches neben Hilfstools eine API mitbringt. Mit dieser wird ein Teil der Updatefunktionalität direkt in die zu aktualisierende Software eingebaut.

Vorbereitung: Rubah enthält ein Tool, das die Software analysiert und Daten zurückliefert, welche später beim Updatevorgang genutzt werden. Dazu zählen Informationen darüber, welche Klassen aktualisiert werden können. Dieses Tool wird für jede neue Programmversion ausgeführt. Weiterhin muss das Programm selbst an Rubah angepasst werden. Programmierer müssen manuell Stellen im Code festlegen, an denen ein Updatevorgang erlaubt ist. Beim Update wird außerdem nach der Übertragung des Zustands die Programmausführung wieder aufgenommen, indem alle Threads neu gestartet werden. Hier müssen Entwickler zusätzliche Prüfungen einbauen, damit z.B. Initialisierungscode nicht erneut ausgeführt wird. Programme, die mit Rubah aktualisiert werden sollen, müssen des Weiteren mit einem mitgeliefertem Tool (sog. *driver*), ausgeführt werden.

Code-Update: Wird ein Updatevorgang gestartet, lädt der *driver* zuerst alle Klassen in ihrer neuen Version in die JVM. Wurde eine Klasse nicht oder nur in der Implementierung ihrer Methoden geändert, wird in ihren Instanzen die *_klass*-Referenz, welche auf die Klasse eines Objekts verweist, auf die neue Version abgeändert.

Ist danach außerdem ein im Programmcode festgelegter Punkt erreicht, an dem ein Update erlaubt ist, (s.o.) wird die Überführung des Zustands gestartet, welche in einem *lazy*- oder einem *eager*-Algorithmus durchgeführt wird.

Zustandsübertragung - eager: Der *eager*-Algorithmus kann in einem einzelnen Thread oder parallelisiert stattfinden. Er betrachtet zuerst alle root-Objekte, d.h. Objekte, welche durch statische Felder oder Felder in Thread-Objekten referenziert sind. Für jedes dieser root-Objekte wird zuerst die entsprechende Klasse und deren aktualisierte Version bestimmt. Sind diese nicht identisch, erzeugt RUBAH eine Instanz der neuen Klassenversion. Auf dieser werden alle Transformationsmethoden für die Klasse und alle Elternklassen ausgeführt, um den Objektzustand zu übertragen. Danach wird das jeweilige Objekt markiert und der Vorgang für alle Objekte wiederholt, die durch das aktuelle referenziert werden und noch nicht als aktualisiert markiert sind. So werden iterativ alle Klassen des Programms auf ihre neue Version aktualisiert.

Zustandsübertragung - lazy: Für den *lazy*-Algorithmus verwendet RUBAH *Proxies*. Diese werden immer dann aufgerufen, wenn auf ein Objekt zugegriffen wird, welches noch nicht aktualisiert worden ist. Dies wird umgesetzt, indem für jede Programmklasse ein Proxy als Unterklasse erzeugt wird. Dessen Methoden führen die Zustandsüberführung durch und fahren danach mit dem regulären Methodenaufruf fort. Zugriffe auf Attribute erfolgen ebenso über Methoden. Diese werden ggf. automatisch von RUBAH in das Programm eingefügt.

Nach der Aktualisierung des Programmcodes werden ähnlich wie beim *eager*-Algorithmus die root-Objekte für ein Update ausgewählt. Für jedes zu aktualisierende root-Objekt wird eine leere Instanz der neuen Klassenversion erstellt und als Proxy markiert, d.h. ihr Typ wird auf den zur Klasse passenden Proxy gesetzt. Danach werden die Transformationsmethoden ausgeführt, welche den Zustand der alten Objekte auf die neuen übertragen. Referenzen werden so geändert, dass nicht mehr die Klassen selbst, sondern ihre Proxies referenziert werden. Schließlich wird die Programmausführung wieder aufgenommen.

Wird während des weiteren Programmverlaufs eine Methode eines root-Objekts aufgerufen, aktualisiert dieses Objekt (als Proxy) zuerst alle Objekte, die von ihm referenziert sind und noch in der alten Version vorliegen. Dies findet wieder nach obigem Schema statt, also mit dem Umwandeln der Objekte in Proxies und dem Übertragen des Zustands. Danach ändert der Proxy seinen Typ wieder in seine ursprüngliche reguläre Klasse. Mit diesem Vorgang werden nach und nach alle Klassen des Programms aktualisiert, bis dieses vollständig die neue Version erreicht hat (Abb. 3).

Beschränkungen: Wie oben bereits beschrieben, erfordert RUBAH eine Anpassung des Programmcodes, um dynamische Updates zu ermöglichen. Dies beinhaltet u.a. ein manuelles Festlegen von Updatepunkten. Des Weiteren muss im Code unter Umständen geprüft werden, ob RUBAH sich aktuell im Updatevorgang befindet und die *run*-Methode eines Threads deshalb im Zuge der Wiederaufnahme

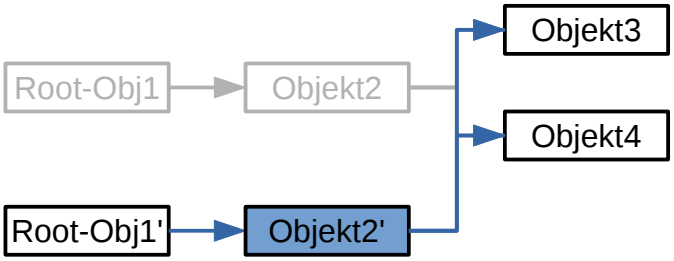


Abb. 3. Update per lazy-Algorithmus in Rubah: Der hier beispielhaft dargestellte Zustand wird erreicht, nachdem das root-Objekt aktualisiert und eine seiner Methoden aufgerufen worden ist. Das referenzierte Objekt 2 ist vom root-Objekt bereits aktualisiert worden und ist nun selbst als Proxy markiert. Beim Aufruf einer in Objekt 2 enthaltenen Methode werden schließlich die noch in der alten Version vorliegenden Objekte 3 und 4 aktualisiert.

der Programmausführung neu gestartet worden ist. Dieser zusätzliche Aufwand ist jedoch relativ gering und muss zum größten Teil nur einmal geleistet werden.

Desweiteren erlaubt Rubah keine blockierende Ein-/Ausgabe, um Updates nicht unnötig zu verzögern. Diese Bedingung erfordert wieder manuelles Eingreifen in den Programmcode und Anpassen an Rubah. Es wird allerdings eine API mitgeliefert, die diesen Vorgang vereinfachen soll.

4 Update- und Laufzeiteffizienz

Die oben vorgestellten DSU-Systeme verfolgen unterschiedliche Ansätze, um dynamische Softwareupdates zu realisieren. Während JavAdaptor umfassend Gebrauch von HotSwap sowie JVM TI im Allgemeinen macht, und damit den Updateprozess von außen durchführt, integriert sich Rubah zusätzlich per eigener API im Programm. Jvolve wiederum muss nicht auf *Proxies* oder *Container* zurückgreifen, welche die fehlende DSU-Unterstützung der JVM zu umgehen versuchen, sondern liefert die benötigte Funktionalität in seiner eigenen JVM mit.

Die vorgestellten Systeme wurden von ihren jeweiligen Entwicklern hinsichtlich der Effizienz untersucht. Die Ergebnisse werden nun im Folgenden zusammengefasst und - soweit möglich - diskutiert. Dabei wird der Schwerpunkt auf den Einfluss auf Ausführungs- und Updategeschwindigkeit gelegt.

4.1 Ausführungsgeschwindigkeit

Mit den präsentierten DSU-Systemen wird in der Regel ein nur sehr geringer Overhead in der normalen Programmausführung erreicht. Beim Test des Webservers Jetty mit Jvolve wurden beispielsweise vor und nach einem Update nur

geringe Geschwindigkeitsunterschiede in Latenz und Durchsatz im Vergleich zu einer unveränderten Jikes RVM festgestellt [15]. Ein Faktor, der dazu beitragen kann, ist die Umsetzung von JVOLVE als JVM. Da hier die Klassen im Speicher direkt manipuliert werden können und nicht auf Beschränkungen einer Standard-JVM Rücksicht genommen werden muss, kann auf zusätzliche Indirektion wie Proxies verzichtet werden. Zu einem solchen Ergebnis sind auch die JVOLVE-Entwickler gekommen [15].

Fü JAVADAPTOR wurde bei einem Test mit HyperSQL nur dann ein messbarer Geschwindigkeitsunterschied gemessen, wenn Proxies eingesetzt wurden [14]. Da Container keine Indirektion durch zusätzliche Funktionsaufrufe verursachen, findet hier wie bei JVOLVE keine Verlangsamung statt. Allerdings war auch der Einfluss von Proxies auf die Geschwindigkeit im Test zwar messbar, befand sich aber im Nanosekundenbereich und ist damit im Vergleich zur gesamten Ausführungsdauer von Methoden vernachlässigbar [14].

In der Ausführungsgeschwindigkeit nach einem erfolgten Update wurde für RUBAH festgestellt, dass die getestete Software nach dem Update etwas langsamer läuft. Dies hat nach Ansicht der Entwickler mit JVM-internen Effekten zu tun, die bei einem Update auftreten [12]. Diese Beobachtung wurde bei JVOLVE nicht gemacht. Bei einem Test mit dem Jetty-Webserver waren hier Durchsatz und Latenz vor und nach einem Update nahezu identisch [15].

Bei RUBAHS *lazy*-Algorithmus wurde ein weiterer Effekt festgestellt. Wird nach dem Beginn eines Updates die Programmausführung wieder aufgenommen, steigt die Geschwindigkeit zuerst an, während die Objekte nach und nach auf die neue Version ihrer zugehörigen Klassen aktualisiert werden. Der negative Einfluss des Updatevorgangs auf die Geschwindigkeit sinkt mit der Anzahl der noch verbleibenden alten Objektversionen. Nach Abschluss eines Updates wird jedoch wie beim *eager*-Algorithmus die ursprüngliche Ausführungsgeschwindigkeit nicht erreicht. Stattdessen haben die RUBAH-Entwickler eine im Vergleich zu diesem sogar etwas größere Verlangsamung festgestellt. Diese wird von den Entwicklern auf Optimierungen durch die JVM zurückgeführt, welche stattfinden, während Objekte noch in Form von Proxies im Speicher liegen. [12]

4.2 Updategeschwindigkeit

Ein weiteres Kriterium bei der Bewertung der Effizienz eines DSU-Systems ist die Geschwindigkeit des Updatevorgangs selbst. *eager*-Algorithmen halten einen Prozess komplett an und aktualisieren währenddessen das gesamte Programm, was je nach Programmgröße zu unterschiedlich langen Verzögerungen führt. RUBAH und JAVADAPTOR mildern diesen Effekt, indem sie ihren *eager*-Algorithmus parallelisieren. [14,12]

Im Vergleich des *lazy*-Algorithmus von RUBAH mit der parallel arbeitenden *eager*-Variante wurde ein starker Unterschied in der Dauer der Unterbrechung festgestellt. In den meisten Fällen wurde diese durch den *lazy*-Ansatz auf weniger als die Hälfte verkürzt, in manchen Fällen liegt sogar eine Beschleunigung mit einem Faktor von zweistelliger Größenordnung vor. [12]

4.3 Ergebnis

Während zusätzliche Indirektion bei anderen Systemen zu Geschwindigkeitseinbußen führen kann (z.B. über 50% bei DuSTM [12]), wird sie bei den hier vorgestellten entweder komplett vermieden oder verursacht nur vernachlässigbaren Overhead.

Eine Wahl zwischen *lazy* und *eager* lässt sich zumindest bei Rubah nicht allgemeingültig treffen. Vielmehr sollte nach den vorliegenden Anforderungen entschieden werden. Während ein *lazy*-Algorithmus zu bevorzugen ist, wenn die Ausführungsunterbrechung möglichst kurz sein soll, ist er nicht zu empfehlen, wenn z.B. zeitliche Garantien über Antwort- und Reaktionszeiten gegeben werden sollen, da hier nach dem Updatebeginn die Ausführung kurzzeitig verlangsamt wird.

5 Zusammenfassung

Es gibt verschiedene Ansätze, um dynamische Softwareupdates in Java umzusetzen. In dieser Ausarbeitung wurde nach einer kurzen Einführung mit Jvolve, Rubah und JavAdaptor ein Querschnitt durch aktuelle DSU-Systeme vorgestellt, und damit ein Überblick über den aktuellen Stand der Forschung ermöglicht.

Weiterhin wurden die Systeme unter dem Aspekt der Effizienz beleuchtet. Der Geschwindigkeitsoverhead ist jeweils sehr gering, was unter anderem damit zusammenhängt, dass das Einführen zusätzlicher Indirektion größtenteils vermieden wird.

Zur weiteren Entwicklung im Bereich der dynamischen Softwareupdates sehen die Entwickler von Rubah und JavAdaptor Möglichkeiten zur weiteren Optimierung. Mit diesen soll die Performance jeweils ihres Systems noch weiter verbessert werden können [14,12]. Diese Ansätze können ein weiterer Schritt zur Entwicklung von DSU-Systemen sein, die zunehmend auch in Produktivumgebungen Verwendung finden können.

Literatur

1. Javassist by jboss-javassist. `http://jboss-javassist.github.io/javassist/`, Zugriff: 2015-06-28
2. Jvm(tm) tool interface 1.2.1. `http://docs.oracle.com/javase/7/docs/platform/jvmti/jvmti.html` (2007), Zugriff: 2015-05-25
3. Arnold, J., Kaashoek, M.F.: Ksplice: Automatic rebootless kernel updates. In: Proceedings of the 4th ACM European conference on Computer systems. pp. 187–198. ACM (2009)
4. Chen, H., Yu, J., Hang, C., Zang, B., Yew, P.C.: Dynamic software updating using a relaxed consistency model. Software Engineering, IEEE Transactions on 37(5), 679–694 (2011)

5. Durumeric, Z., Kasten, J., Adrian, D., Halderman, J.A., Bailey, M., Li, F., Weaver, N., Amann, J., Beekman, J., Payer, M., et al.: The matter of heartbleed. In: Proceedings of the 2014 Conference on Internet Measurement Conference. pp. 475–488. ACM (2014)
6. Gu, T., Cao, C., Xu, C., Ma, X., Zhang, L., Lu, J.: Javelus: A low disruptive approach to dynamic software updates. In: Software Engineering Conference (APSEC), 2012 19th Asia-Pacific. vol. 1, pp. 527–536. IEEE (2012)
7. Hicks, M., Moore, J.T., Nettles, S.: Dynamic software updating, vol. 36. ACM (2001)
8. Kabanov, J., Vene, V.: A thousand years of productivity: the jrebel story. Software: Practice and Experience 44(1), 105–127 (2014)
9. Lindholm, T., Yellin, F., Bracha, G., Buckley, A.: The java® virtual machine specification. `https://docs.oracle.com/javase/specs/jvms/se7/html/` (2013), Zugriff: 2015-06-15
10. Martinez, S., Dagnat, F., Buisson, J.: Prototyping dsu techniques using python. In: HotSWUp'13. pp. https–www (2013)
11. Neamtiu, I., Hicks, M., Stoyle, G., Oriol, M.: Practical dynamic software updating for C, vol. 41. ACM (2006)
12. Pina, L., Veiga, L., Hicks, M.: Rubah: Dsu for java on a stock jvm. In: Proceedings of the 2014 ACM International Conference on Object Oriented Programming Systems Languages & Applications. pp. 103–119. ACM (2014)
13. Pukall, M., Grebhahn, A., Schröter, R., Kästner, C., Cazzola, W., Götz, S.: Javadaptor: unrestricted dynamic software updates for java. In: Proceedings of the 33rd International Conference on Software Engineering. pp. 989–991. ACM (2011)
14. Pukall, M., Kästner, C., Cazzola, W., Götz, S., Grebhahn, A., Schröter, R., Saake, G.: Javadaptor—flexible runtime updates of java applications. Software: Practice and Experience 43(2), 153–185 (2013)
15. Subramanian, S., Hicksy, M., McKinley, K.S.: Dynamic Software Updates for Java: A VM-Centric Approach. Computer Science Department, University of Texas at Austin (2008)
16. Venners, B.: Inside the Java virtual machine. Java masters, McGraw-Hill, New York, NY [u.a.] (1998), includes CD-ROM.; pbk ; : £28.99 : CIP entry (Oct.)
17. Würthinger, T., Wimmer, C., Stadler, L.: Dynamic code evolution for java. In: Proceedings of the 8th International Conference on the Principles and Practice of Programming in Java. pp. 10–19. ACM (2010)