



Hochschule Reutlingen
Reutlingen University



INF

Studiengang
Medien- und
Kommunikationsinformatik



14 Informatics Inside

Human-Centered Computing

Uwe Kloos, Natividad Martínez, Gabriela Tullius (Hrsg.)

Tagungsband

30. April 2014

Hochschule Reutlingen





INF

Studiengang
Medien- und
Kommunikationsinformatik



Hochschule Reutlingen
Reutlingen University

Uwe Kloos, Natividad Martínez, Gabriela Tullius (Hrsg.)

Informatics Inside: Human-Centered Computing

Informatik-Konferenz an der Hochschule Reutlingen
30. April 2014

ISBN 978-3-00-045427-1



9 783000 454271 >

Impressum

Anschrift:

Hochschule Reutlingen
Reutlingen University
Fakultät Informatik
Medien- und Kommunikationsinformatik
Alteburgstraße 150
D-72762 Reutlingen

Telefon: +49 7121 / 271-4002

Telefax: +49 7121 / 271-4042

E-Mail: infoinside@reutlingen-university.de

Internet: <http://www.infoinside.reutlingen-university.de>

Organisationskomitee:

Prof. Dr. Gabriela Tullius, Hochschule Reutlingen
Prof. Dr. Natividad Martínez, Hochschule Reutlingen
Prof. Dr. Uwe Kloos, Hochschule Reutlingen

André Antakli
Thomas Bauer
Olaya De la Rosa Avitia
Matthias Gutekunst
Viktoria Hoffmann
Johannes Kartheininger
René Mangold
Stanislas Mauser
Lars Schneider
Arkadius Weister
Anna Wellerdiek



Hochschule Reutlingen
Reutlingen University

Copyright: © Hochschule Reutlingen, Reutlingen 2014
Herstellung und Verlag: Hochschule Reutlingen
ISBN 978-3-00-045427-1

Vorwort

Herzlich willkommen zur “Informatics Inside”! Diese studentische Konferenz tagt bereits zum sechsten Mal und ist ein fester Bestandteil des akademischen Jahres der Fakultät Informatik der Hochschule Reutlingen. Die Konferenz wird von Studierenden des Masterprogramms der „Medien- und Kommunikationsinformatik“ im Rahmen ihrer Forschungsarbeit organisiert und durchgeführt. Das Masterprogramm sieht vor, dass die Studierenden eine wissenschaftliche Vertiefung in einem Forschungsgebiet ihrer Wahl erarbeiten. Dies kann in einer Firma oder in einem Labor der Hochschule oder in ausländischen Partnerhochschulen stattfinden. Als Teil der Ausbildung müssen die Studierenden ihre wissenschaftliche Arbeit in Form eines Forschungsartikels veröffentlichen, vortragen und mit dem Fachpublikum diskutieren. So entsteht ein doppelter Gewinn: Einerseits können die Studierenden die Fähigkeiten trainieren, ihre Arbeit verständlich zu präsentieren und über den Inhalt zu diskutieren. Andererseits wird dem Publikum ein breites Spektrum der aktuellen Themen in der Informatik angeboten.

Die Konferenz hat sich auch in den letzten Jahren nach außen geöffnet und freut sich über studentische Beiträge von anderen Fakultäten oder von anderen Hochschulen. Diese Beiträge werden in einer Poster-Session präsentiert, die zum offenen Austausch mit den Teilnehmern einlädt. Im Rahmen der Konferenz werden auch andere studentische Projekte in einer kleinen Ausstellung präsentiert. Alles in allem wird die Informatics Inside zum Treffpunkt von Professoren und Studierenden aus allen Fakultäten und bietet Raum für interdisziplinäre Gespräche rund um die Informatik.

Das Motto in diesem Jahr lautet: „Human-Centered Computing“, als Einstimmung auf den Schwerpunkt unseres neuen Masterprogramms. Human-Centered Computing betrachtet die Anwendung der menschlichen Informationsverarbeitung auf Computersysteme sowie deren nutzerzentrierte Gestaltung. Die Beiträge vertiefen beispielsweise Themen der Mensch-Maschine-Interaktion, nutzerorientierte Softwareentwicklung, kollaborative Umgebungen oder virtuelle Realität.

Ein letztes Wort möchte ich der Organisation der Konferenz widmen: Der gesamte Prozess wurde von den Studierenden in mehreren Monaten geplant. Vom Call-for-Papers, Reviews, Proceedings-Erstellung bis zur Finanzierung und Logistik wurde alles äußerst professionell durchgeführt. An dieser Stelle möchte ich den Masterstudierenden mein Glückwunsch aussprechen.

Ich wünsche allen für diesen Tag eine spannende Konferenz, neue Erkenntnisse, neue Kontakte und viel Spaß.

Prof. Dr. Natividad Martínez

Inhaltsverzeichnis

Gestenerkennung & Augmented Reality

Thomas Bauer

Anforderungsanalyse zur computergestützten Erkennung der Deutschen Gebärdensprache..... 8

Matthias Gutekunst

Augmented Reality zur Steigerung der Immersion in virtuellen Umgebungen..... 26

Stanislas Mauser

Analysis of Finger- and Palm-based interaction paradigms for Touch-Free Gesture-Based Control of Medical Devices with the Leap Motion Controller..... 34

Softwaretechnik

René Mangold

Selektion von Szenarien zur Optimierung von Simulationen im präventiven Krisenmanagement..... 46

Arkadius Weister

Language Oriented Programming: Modulare domänenspezifische Sprachen..... 54

Entwicklung Mobiler Anwendungen

Olaya De la Rosa Avitia

Strategy to Test Mobile Apps..... 70

Viktoria Hoffmann

Optimierung der Usability von digitalen Fahrtenbüchern durch automatisches Erfassen von fahrzeugspezifischen Daten..... 80

Johannes Kartheininger

Vergleich der Single Sign On Verfahren SAML und OpenID Connect..... 92

Virtuelle Welten

André Antakli

Umgebungswahrnehmung von agentenbasierten simulierten Menschmodellen in virtuellen Welten im Kontext C3D..... 100

Lars Schneider

Steigerung von Immersion durch die Integrierung von virtuellen Menschen in die virtuelle Welt NeueWelt9..... 112

Anna Wellerdieck

Erkennung eigener Körperbewegungen in animierten Charakteren..... 120

Shortpaper

Frank Bertagnolli

Bestimmung von 3D-Messpositionen durch genetischen Algorithmus 130

Peter Einberger, Christian Hensler

Neue Welt 9: Vorteile und Probleme bei der Migration zur CryEngine 3.5.4 132

Hazel Grünewald, Michael Moser

QUEST 3C - Kooperation, Kommunikati-on, Präsentation ohne Grenzen 134

Sebastian Hirth, Susanne Hönig, Alexander Kunz, Dominic Lyons, Jens Mahler

Technologien verteilter Systeme zur mobilen Kommunikation 136

Patrik Huber, Josef Kittler, Matthias Rätsch

Bottom-up and Top-down Face Analysis based on 3D Face Models 138

Stefan Radicke, Markus Rapp

Is Virtual Reality The Future Of The Video Games Industry? 140

Greg Rauhöft, Florian Strieg, Natascha Stumpp

Oculus Rift mit externem Motion Tracking..... 142

Anforderungsanalyse zur computergestützten Erkennung der Deutschen Gebärdensprache*

Thomas Bauer
Reutlingen University
Thomas.Bauer@Student.
Reutlingen-University.DE

Abstract

Im Rahmen der wissenschaftlichen Vertiefung an der Hochschule Reutlingen befasst sich diese Arbeit mit der Untersuchung der Anforderungen und der Machbarkeit zur computergestützten Erkennung der Deutschen Gebärdensprache - im Folgenden DGS - und des deutschen Fingeralphabets. Die Erkenntnisse aus dieser Arbeit dienen als Grundlage zur Entwicklung eines Systems zur Übersetzung von Gebärden der DGS oder des Fingeralphabets in die deutsche Schriftsprache. Zunächst werden grundlegende Informationen zu Geschichte, Aufbau und Grammatik der DGS und des Fingeralphabets aufgeführt. Die Erkennung der Gebärden soll durch optische Bewegungssensoren erfolgen. Hierfür werden unterschiedliche Sensortypen betrachtet und verglichen. Im weiteren Verlauf erfolgt die Analyse der benutzerspezifischen und technischen Anforderungen. Erstere basieren auf der Befragung einer Fokusgruppe aus gehörlosen und hörenden Menschen aus dem Bereich der Gehörlosen-, Schwerhörigen- und Sprachbehindertenpädagogik. Abgeleitet aus den Informationen der Anforderungsanalyse ergibt sich, bis zu einem gewissen Grad, die Machbarkeit aus technischer und benutzerspezifischer

Sicht. Abschließend erfolgen die Zusammenfassung der Anforderungen, welche an das zu entwickelnde System gestellt werden sowie eine Handlungsempfehlung für die Entwicklung eines Prototyps.

Schlüsselwörter

Gestenerkennung, Deutsche Gebärdensprache, Fingeralphabet, Übersetzung, Dolmetscher, Grammatik, Anforderungsanalyse, Prototyp, Bewegungssensor, Asus Xtion, PrimeSense Carmine, Leap Motion

CR-Kategorien

A.1 [Introductory and survey]; D.2.1 [Requirements/Specifications]: Elicitation methods (e.g., rapid prototyping, interviews, JAD); H.1.2 [User/Machine Systems]: Human information processing; I.4.8 [Scene Analysis]: Sensor fusion, Tracking

1 Thematische Einführung

Neben der Linguistik, welche sich bereits seit Jahrhunderten mit der exakten wissenschaftlichen Erforschung der menschlichen Sprache befasst, charakterisiert die Computerlinguistik (CL) seit Mitte des 20. Jahrhunderts das Fachgebiet der maschinellen Verarbeitung natürlicher Sprache. Sie orientiert sich dabei an den linguistischen Teilbereichen der Phonetik und Phonologie, der Morphologie, der Syntax und Semantik sowie der Pragmatik (vgl. [2], S. 1ff). Ein Anwendungsbereich der Computerlinguistik stellt die maschinelle Übersetzung (MÜ) oder auch Machine Translation (MT) dar.

*
Betreuer Hochschule: Prof. Dr. rer. nat. Gabriela Tullius
Hochschule Reutlingen
Gabriela.Tullius@Reutlingen-
University.de

Informatics Inside 2014
Wissenschaftliche Vertiefungskonferenz 2014
30. April 2014, Hochschule Reutlingen
Copyright 2014 Thomas Bauer

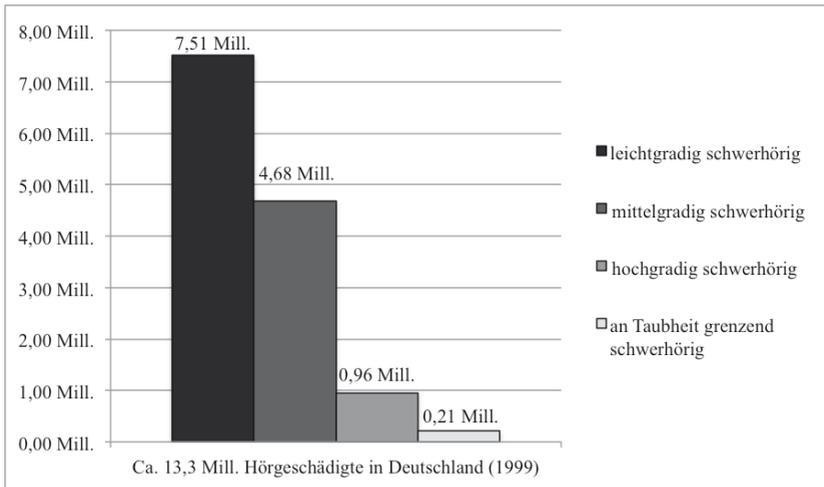


Abbildung 1: Verteilung der Hörschädigung nach Sohn

Quelle: vgl. [15], Abbildung eigen

Die fortschreitende Globalisierung erfordert zunehmend zeitnahe Übersetzungen unterschiedlicher digitaler Dokumente wie z.B. von Handbüchern, Gebrauchsanweisungen oder Websites. Dabei behandelt die zugrunde liegende Übersetzung die Übertragung eines Textes aus einer Quellsprache in die jeweilige Zielsprache. Das Verfahren der maschinellen Übersetzung aus der Computerlinguistik soll hierbei unterstützend eingesetzt werden, indem beispielsweise Übersetzungsprozesse sowie Sprachein- und ausgaben durch eine Recheneinheit vollautomatisch analysiert und übernommen werden. Damit vereint die maschinelle Übersetzung Einflüsse aus der Linguistik, Mathematik, Informatik, Übersetzungswissenschaft und der Elektrotechnik (vgl. [2], S. 642ff).

Auch für Menschen mit Behinderung kann die maschinelle Übersetzung unterstützend eingesetzt werden. So besteht bereits die Möglichkeit der Ein- und Ausgabe am Computer für sehbehinderte bzw. blinde Menschen mittels Lautsprache. Die automatische Spracherkennung (Automatic Speech Recognition - ASR) liefert diesbezüglich bereits gute und brauchbare Ergebnisse, dennoch

steigt die Fehlerquote durch unterschiedliche Faktoren an. So erschweren bspw. Störgeräusche, ein umfangreiches Vokabular oder aber auch Konstruktionsabbrüche, Dialekte und Akzente die korrekte Spracherkennung (vgl. [9], S. 15). Braillezeilen und Brailletastaturen hingegen sind vergleichbar mit einem Monitor und einer Tastatur und somit nahezu fehlerlos. Die Fehleranfälligkeit wird hier durch den Faktor Mensch bestimmt. Die Verarbeitung der maschinellen Übersetzung bezieht sich in diesem Falle ausschließlich auf die Substitution der einzelnen Schriftzeichen eines Textes, da für jedes dieser Zeichen ein passendes Pendant in Brailleschrift existiert und grammatikalische Unterschiede, wie bei verschiedenen Sprachen, nicht existieren.

Ein weitaus komplexerer Anwendungsfall ergibt sich bei Menschen mit einer starken Hörschädigung sowie gehörlosen Menschen. Dabei steht vor allem die Kommunikationsunterstützung im Vordergrund. Der Deutsche Schwerhörigenbund e.V. geht derzeit von rund 80 000 gehörlosen und rund 16 Millionen schwerhörigen Menschen in Deutschland aus (vgl. [5]). Diese Zahlen beruhen auf einer Schätzung, der die wissenschaftliche

Untersuchung „Hörscreening“ von Dr. med. Wolfgang Sohn aus dem Jahre 1999 zugrunde liegt. Dabei wurden Daten von ca. 2000 Probanden aus Deutschland im Alter über 14 Jahren ermittelt und ausgewertet (vgl. [15]). Das Diagramm in Abbildung 1 zeigt die Verteilung nach Grad der Schwerhörigkeit zum Zeitpunkt der Datenerhebung.

Als leichtgradig schwerhörig wird ein mittlerer Hörverlust bis 35dB (Dezibel) bezeichnet, wobei als Faustregel davon ausgegangen werden kann, dass eine Abnahme des Schalldrucks um ca. 10 dB einer Lautstärkenhalbierung entspricht. Bei einem Hörverlust von 30dB entspräche dies etwa $\frac{1}{8}$ des ursprünglichen Schalldrucks. Mittelgradig schwerhörig entspricht einem mittleren Hörverlust zwischen 40 – 65 dB, hochgradig bereits 70 – 85 dB. Ab einem mittleren Hörverlust von 90 dB und größer gilt eine an Taubheit grenzende Schwerhörigkeit (vgl. [15]). Vielen der Betroffenen bleibt für die Kommunikation, aufgrund ihrer Behinderung, ausschließlich der Schriftverkehr oder die Gebärdensprache. Im Gegensatz zur Brailleschrift stellen die Gebärdensprachen eigenständige Sprachen mit eigener Grammatik dar. Die Anforderungen an eine maschinelle Übersetzung der Gebärdensprache sind daher weitaus höher gesetzt und nicht so einfach zu realisieren, da zunächst die Erkennung der Gebärden und im weiteren Verlauf die grammatikalische Übersetzung erfolgen muss.

1.1 Motivation und Stand der Technik

Verschiedene Möglichkeiten wie z.B. Dolmetscher, Ferndolmetschdienste oder aber auch mobile Gebärdensprachlexika ermöglichen bzw. erleichtern die Kommunikation zwischen Gehörlosen und Hörenden. Dennoch ergeben sich daraus diverse Einschränkungen. So ist der Einsatz eines Dolmetschers oder Ferndolmetschdienstes immer von einer realen Person abhängig, welche die eigentliche Übersetzung durchführt. Des Weiteren ist die Nutzung solcher Dienste mit relativ hohen Kosten verbunden. Diese werden jedoch in

der Regel von einem Kostenträger übernommen und belasten somit nicht den Anwender selbst. Gebärdensprachlexika erlauben es dem Benutzer einzelne Begriffe und deren Gebärden nachzuschlagen, jedoch wird hier keine Übersetzung ganzer Sätze angeboten. Eine computergestützte Übersetzung könnte dabei eine höhere Verfügbarkeit für den Anwender und ggf. gleichzeitig eine Kostenreduktion für den Kostenträger bedeuten.

Es gibt einzelne Ansätze zur computergestützten Erkennung und Übersetzung von Gebärdensprachen. Sowohl Eingabetechniken als auch die Methoden zur Erfassung und Übersetzung der Gebärdensprache oder des Fingeralphabets variieren hier. In früheren Ansätzen fanden für die Eingabe meist Datenhandschuhe Verwendung. So beschreiben Janguin et al. in [11] bereits 1998 einen Ansatz zur Erkennung der Chinese Sign Language (CSL) unter Verwendung eines Datenhandschuhs. Mittlerweile werden jedoch meist rein visuelle Ansätze verfolgt (vgl. [3], S. 1). Diese reichen von der einfachen Webkamera bis hin zu dreidimensionalen Tiefenkameras wie z.B. die Microsoft Kinect. So beschreiben Dos Santos Anjo et al. in [6] einen Ansatz zur Erkennung einiger statischer Gesten des Fingeralphabets der Brazilian Sign Language (Libras) mit Hilfe der Kinect.

Ebenfalls auf Basis der Microsoft Kinect Technologie entwickelten Forscher der Chinese Academy of Science und der Beijing Union University in Zusammenarbeit mit Microsoft Research Asia einen Prototyp zur Gebärdensprachübersetzung. Der Software-Prototyp VisualComm ermöglicht die Kommunikation zwischen gebärdensprachorientierten und lautsprachenorientierten Menschen sowie eine Übersetzung zwischen unterschiedlichen Gebärdensprachen (vgl. [4], S. 1ff). Den Entwicklern war es wichtig, eine Zweiwegekommunikation anzubieten. So werden einerseits Gebärden in Text oder Sprache übersetzt und andererseits Wörter und Sätze durch einen 3D Avatar in Gebärdensprache dargestellt. Zur Eingabe der

Gebärden wird der optische Tiefensensor Kinect verwendet. Mit einem Wortschatz von 370 chinesischen Gebärden und einer Erfassungsquote von ca. 95% beweist dieser Ansatz, dass die Erkennung gestenbasierter Sprachen mit bereits verfügbaren und kostengünstigen 2D und 3D-Sensoren möglich ist (vgl. [3], S. 1f).

Im Vergleich hierzu bietet das britische Unternehmen Technabling Ltd. bereits ein plattformunabhängiges Softwaresystem zur Echtzeitübersetzung gestenbasierter Sprachen. Der Einsatz des Portable Sign Language Translator (PSLT) soll auf den meisten unterschiedlichen Geräten wie z.B. Smartphones, Tablets und Laptops möglich sein. Voraussetzung ist eine einfache Kamera um die jeweiligen Gesten aufzuzeichnen. Aktuell beschreibt Technabling Ltd. die Möglichkeit einer Übersetzung in geschriebenen Text aus der British Sign Language (BSL), der Chinese Sign Language (CSL) und aus Makaton¹(vgl. [20]).

Die Ansätze von Microsoft und Technabling Ltd. wurden gegen Ende 2013 veröffentlicht und befinden sich derzeit in unterschiedlichen Entwicklungsstufen. Dies lässt auf das noch junge Thema der Gebärdenspracherkennung und –übersetzung schließen, welches weiterhin viel Raum für Forschungsmöglichkeiten bietet.

1.2 Zielsetzung

Im Zuge einer Master-Thesis soll ein Prototyp zur computergestützten Übersetzung der Deutschen Gebärdensprache und/oder des deutschen Fingeralphabets entstehen. In einem ersten Schritt werden daher in dieser Arbeit sowohl erste benutzerspezifische als auch technische Anforderungen ermittelt und analysiert. Aus den daraus gewonnen Informationen lässt sich die Machbarkeit des Projekts ableiten. Hinsichtlich der benutzerspezifischen

¹Makaton ist ein Förderansatz welcher ein Kernvokabular von Alltagsbegriffen in Form von Gebärden und Symbolen bereitstellt (vgl. [13]).

Anforderungen werden sowohl Gehörlose und Schwerhörige als auch Hörende in einem Interview befragt. Hier gilt es unter Anderem die Relevanz festzustellen, ob eine Übersetzung der Deutschen Gebärdensprache und/oder des deutschen Fingeralphabets erfolgen soll. Wie bereits in Abschnitt 1.1 aufgeführt, werden mittlerweile meist nur visuelle Eingabetechniken für die Erfassung der Gebärdensprache verwendet. Daher werden für die technischen Anforderungen die derzeit aktuellen optischen Bewegungssensoren Leap Motion, Asus Xtion Pro Live sowie die PrimeSense Carmine 1.09 verglichen und auf ihre Einsatztauglichkeit untersucht.

2 Gebärdensprache und Fingeralphabet

In den folgenden Abschnitten werden die Grundlagen und Besonderheiten der Deutschen Gebärdensprache und des deutschen Fingeralphabets aufgeführt.

2.1 Deutsche Gebärdensprache (DGS)

Woher Gebärdensprachen stammen und wann sie entstanden sind, lässt sich nicht genau erklären. Dies ist darauf zurückzuführen, dass in den vergangenen Jahrhunderten gehörlose, schwerhörige und sprachbehinderte Menschen meist unterdrückt und als Minderheit ausgeschlossen wurden, weshalb es auch nur wenig Zeugnisse dieser Zeitgeschichte gibt. Es wird jedoch angenommen, dass Gebärdensprachen in einfacher Form schon vor der Entstehung von Lautsprachen verwendet wurden (vgl. [10], S. 4). Durch die Entwicklung und Verbreitung der Lautsprachen wurden die Gebärden nur noch in abgeschwächter Form zur Unterstützung eingesetzt oder zum Teil ganz verdrängt. Erst Mitte des 16. Jahrhunderts erkannte der Mönch Pedro Ponce de Leon, dass die bis dahin als „geistarm“ geltenden gehörlosen und sprachbehinderten Menschen ebenfalls lernfähig sind. Er unterrichtete Jungen in Sprechen, Schreiben und Lesen. Bis ins späte 19. Jahrhundert entwickelten sich aus dieser Kenntnis die ersten Schulen für Gehörlose

in Frankreich, den USA und Deutschland. 1880 erfolgte der Umbruch beim Mailänder Kongress. Insgesamt 255 Lehrer aus Europa und Amerika – darunter nur zwei Gehörlose – entschieden über die Entwicklung des Gehörlosen-Unterrichts und verboten ihn. Erst 1988 wurde durch das Europäische Parlament der Entschluss zur Anerkennung nationaler Gebärdensprachen gefasst. Seit dem 1. Mai 2002 ist die Gebärdensprache in der Bundesrepublik Deutschland als eigenständige Sprache anerkannt (vgl. [14]).



Abbildung 2: Gebärdensraum

Quelle: [1], S. 23

Gebärdensprache ist nicht gleich Gebärdensprache. Oftmals wird davon ausgegangen, dass sich gebärdensprachorientierte Menschen auf der ganzen Welt miteinander verständigen können. Dem ist jedoch nicht so. Wie bei den Lautsprachen haben sich kulturell unterschiedliche Sprachen und Dialekte entwickelt. In Deutschland spricht man daher von der Deutschen Gebärdensprache (DGS). Sie ist wie alle anderen Gebärdensprachen eine rein visuelle Sprache und besitzt eine eigene Grammatik. Neben den Armen und Händen werden vor allem das Gesicht, dessen Mimik und Mundbild sowie der gesamte Oberkörper zur Zeichenproduktion eingesetzt. Im sogenannten Gebärdensraum, welcher sich im Bereich des Oberkörpers befindet, werden die Gebärden in dreidimensionaler Form dargestellt. In seltenen Fällen kann der Gebärdensraum auch verlassen werden wie z.B. bei einer Form der Gebärde für Hund, bei welcher mit der flachen Hand zweimal auf den Oberschenkel geklopft wird (vgl. [7], S.34f).

Auch in Bezug auf die Grammatik unterscheidet sich die DGS von der deutschen Lautsprache. In beiden Sprachen werden zur Bildung eines einfachen aktiven Satzes mindestens die Satzglieder Subjekt (S) und Prädikat/Verb (V) benötigt – oft auch mit der Erweiterung durch ein Objekt (O). Ist unklar, ob die Sprache die Satzfunktionen Subjekt und Objekt aufweist, wird stattdessen von Agens (A) und dem Gegenüber Patiens (P) gesprochen. So auch bei der DGS. Die Anord-

nung der Satzglieder der beiden Sprachen ist meist unterschiedlich. Während die deutsche Lautsprache in der Regel einer Anordnung von AVP bzw. SVO folgt und somit das Prädikat als bindendes Glied zwischen Agens und Patiens dient, steht das Prädikat in der DGS meist am Ende eines Satzes (vgl. [7], S. 175ff).

Lautsprache:

Das Mädchen - putzt - das Auto

DGS:

Mädchen - Auto – putzen

Aber auch hier gibt es unterschiedliche Wortstellungsvarianten sodass, neben der grundlegenden Variante APV, auch die Varianten AVP oder PAV vorkommen können. Dies hängt von unterschiedlichen Faktoren ab (vgl. [7], S. 178).

Nicht nur die Syntax beeinflusst den grammatikalischen Unterschied der DGS zur deutschen Lautsprache. Beispielsweise existieren, wie im obigen Beispiel zu sehen, auch keine Artikel (der, die, das) oder Kopulae². Der grammatikalische Aufbau setzt

²Kopula ist eine Verbform von „sein“, „bleiben“ und „werden“, welches Subjekt und Prä-

sich zudem aus unterschiedlichen Elementen zusammen. So spielen neben dem Produktionstempo und der Modalitätsunterschiede der Gebärden auch die Nutzung des Gebärdensraums sowie unterschiedliche Formen der Hände, des Oberkörpers oder der Mimik eine elementare Rolle in der Grammatik einer visuell-gestischen Sprache. Im Folgenden werden daher nur die einzelnen Bereiche gestreift, welche für die vorliegende Arbeit relevant sind. Eine ausführliche Beschreibung des grammatikalischen Unterschieds der DGS zur deutschen Lautsprache bieten Boyes, Braem und Penny in [1].

Modalitätsunterschiede beziehen sich primär auf die lineare Anordnung der gesprochenen Sprache im Vergleich zur räumlichen Anordnung der Gebärdensprache und sind eng mit dem Produktionstempo verknüpft. Ein Versuch von Bellugi und Fischer aus dem Jahr 1972 vergleicht dabei die Geschwindigkeiten der beiden Sprachen bezogen auf einen vorgegebenen Text im Vergleich zum Ausdruck eines eigens gefassten Gedanken bzw. Satzinhaltes. Die Ergebnisse, dargestellt in Tabelle 1, überraschten die Forscher. Während beim vorgegebenen Text das Produktionstempo der Lautsprache annähernd doppelt so schnell war wie bei der Gebärdensprache ist das Tempo um den Inhalt in eigenen Worten auszudrücken annähernd gleich (vgl. [1], S. 46f). Die Annahme der Forscher, dass Satzinhalte in Lautsprache schneller zu übermitteln sind als in Gebärdensprache wurde damit wiederlegt.

Tabelle 1: Vergleich Produktionstempo

Quelle: ([1], S. 47)

nach Bellugi und Fischer 1972

Gebärdensprache	Lautsprache
2,37 Gebärden/Sek.	4,7 Wörter/Sek.
1,27 Sek./Satzinhalt	1,47 Sek./Satzinhalt

dikat zu einer Aussage verbindet. So ist bei dem Satz „Er ist krank“ das Wort „ist“ als Koppula anzusehen. In der DGS würde somit nur „Er krank“ bestehen (vgl. [1], S. 45).

Wie ist es dennoch möglich, dass bei einer zeitlich doppelt so langen Produktion der Gebärden, Satzinhalte in gleicher oder sogar schnellerer Zeit übermittelt werden können? Dies liegt am Aufbau und der Form der Sprache. Während die Lautsprache durch eine sukzessive Organisation und Verkettung linguistischer Formen einen grammatikalischen Aufbau in linearer Form verfolgt, können in der Gebärdensprache durch die räumliche Anordnung grammatikalische Formen gleichzeitig produziert werden. Auch die Wahrnehmung spielt dabei eine große Rolle. Das menschliche Gehör kann einzelne aufeinanderfolgende Elemente einfacher erfassen als eine Überlagerung der Elemente. Die visuelle Wahrnehmung hingegen erlaubt die simultane Erfassung mehrerer Elemente. So können bei der Gebärdensprache alle Komponenten (Handform, Bewegung, Mimik) gleichzeitig wahrgenommen und verarbeitet werden. Je nach Ausführung bzw. Abwandlung einer Gebärde können dadurch unterschiedliche Inhalte übermittelt werden, für die in der Lautsprache mehrere Wörter notwendig wären. Folgende Abbildung zeigt ein Beispiel der Gebärdensprache in welchem zum Einen dargestellt wird, dass in manchen Fällen nur eine Gebärde für die Übermittlung eines ganzen Satzes benötigt wird und zum Anderen, dass die geringe Abwandlung der gleichen Gebärde eine etwas andere Information übermittelt (vgl. [1], S. 47ff).



Abbildung 3: Modifizierung von Gebärden

Quelle: vgl. [1], S. 51

Der dreidimensionale Gebärdenraum stellt hier ein unabkömmliches Werkzeug zur grammatikalischen und semantischen Satzbildung dar. So können unter Anderem auch Ausgangsort und Ziel einer Handlung oder aber auch Zeitangaben durch die Position und die Bewegungsrichtung der Gebärde im Gebärdenraum übermittelt werden (vgl. [1], S. 74). Allgemein lässt sich also sagen, dass die Gebärdensprache ein Konstrukt einzelner Gebärden ist und aus Bewegungen und Formen der Hände, der Arme, des Oberkörpers und des Kopfes besteht. Zudem müssen Mimik und Mundbild sowie die eigene Syntax berücksichtigt werden. Durch den Einsatz dieser Elemente bzw. durch die Modifikation einer Gebärde durch diese Elemente lassen sich unterschiedliche Bedeutungen eines Satzes ausdrücken.

2.2 Deutsches Fingeralphabet

Das Fingeralphabet stellt keine eigenständige Sprache dar. Es ist ein Teilbereich der Gebärdensprache und ermöglicht die Darstellung von Buchstaben mittels einzelner Handformen. So ist jedem Buchstaben des Alphabets eine eigene Handform zugewiesen. Meist dient das Fingeralphabet nur unterstützend zur Beschreibung von Eigennamen oder unklaren Wörtern in der DGS (vgl. [1], S. 146f).



Abbildung 4: Beispiel deutsches Fingeralphabet

Quelle: www.visuelles-denken.de/Schnupperkurs3.html

Das Buchstabieren von Wörtern erfolgt in der Regel wie bei der Gebärdensprache innerhalb des Gebärdenraumes vor der Brust. Es ist möglich ganze Sätze zu buchstabieren. Dies ist jedoch über einen längeren Zeitraum so-

wohl für den Produzent als auch für den Zuseher anstrengend, sodass durchschnittlich nur 56% der gezeigten Buchstaben klar verstanden werden können (vgl. [17] zitiert nach [1], S. 146). Wie die Gebärdensprache selbst, unterscheiden sich Fingeralphabete über Ländergrenzen hinweg oft sehr stark. Das deutsche Fingeralphabet wird beispielsweise einhändig buchstabiert während das britische Fingeralphabet durch beide Hände dargestellt wird (vgl. [1], S. 146f).

3 Erfassung durch optische Bewegungssensoren

Bewegungssensoren ermöglichen im Allgemeinen die Erfassung von kinematischen Vorgängen innerhalb des zu überwachenden Raumes. Dies können Lageänderungen, Geschwindigkeiten oder auch Beschleunigungen sein. Für jede dieser Größen existieren spezielle Sensoren (vgl. [18], S. 182). Im Bereich der Mensch-Maschine-Interaktion werden Bewegungssensoren meist zur Erfassung von Gestik und Mimik eingesetzt. Dies setzt voraus, dass Position und Lageveränderungen von Händen, Kopf, Augen, Augenbrauen oder Lippen erkannt werden können. Bei der Bewegungserfassung von Händen und Fingern werden derzeit überwiegend optische Bewegungssensoren oder Datenhandschuhe eingesetzt. Für die Erfassung der Gesten der DGS und des Fingeralphabets eignen sich Datenhandschuhe. Sie bieten im Vergleich zu den optischen Systemen eine exaktere Erfassung der Anatomie und Bewegung von Händen und Fingern. Dies wird über Dehnungssensoren entlang der Finger und Hände sowie durch Beschleunigungssensoren ermöglicht. Dadurch lassen sich die Freiheitsgrade von Hand- und Fingergelenken sowie deren Bewegungen genau abbilden. Optische Bewegungssensoren hingegen können bei Verdeckungen keine zuverlässige Erfassung, wie sie bspw. bei einer Faust vorkommen können, ohne weiteres gewährleisten (vgl. [21], S. 1). Datenhandschuhe sind jedoch im Vergleich zu optischen Eingabegeräten umständlicher und nicht leicht adaptierbar. Im folgenden Abschnitt werden daher zwei Arten von optischen Bewegungssen-

soren in ihrem Aufbau und der Funktionsweise beschrieben, welche für die Gebärdenspracherkennung eingesetzt werden könnten. In Abschnitt 4 werden diese Sensoren auf ihre technischen Anforderungen und deren Einsetzbarkeit überprüft.

3.1 Asus Xtion Pro Live / PrimeSense Carmine

Sowohl die Asus Xtion Pro Live als auch die PrimeSense Carmine gehören zu der Familie der RGB-D Sensoren oder auch Tiefenkameras. Sie bieten die Möglichkeit einer dreidimensionalen Erfassung von Räumen und Objekten und unterscheiden sich prinzipiell nur im Design voneinander. Vergleichbar ist die Asus Xtion Pro Live mit der PrimeSense Carmine 1.08. Beide Kameras besitzen dieselben Spezifikationen und sind für Entfernungen zwischen 0,8 – 3,5 m ausgelegt. Parallel hierzu bietet PrimeSense noch zusätzlich die Carmine 1.09 für kurze Distanzen zwischen 0,35 – 1,4 m an (vgl. [16], S. 4).



Abbildung 5: PrimeSense Carmine 1.09

Quelle: www.faceshift.com/wordpress/wp-content/uploads/2013/01/PrimeSense_sensor.jpg

Die Erfassung des Raumes und dessen Objekte wird mittels zweier integrierter CMOS-Sensoren sowie einer Infrarotlichtquelle realisiert. Dabei sendet der IR-Emitter ein codiertes Lichttraster in den Raum. Das reflektierende Raster wird von einem der beiden CMOS-Sensoren aufgezeichnet. Aus den so gewonnenen Daten lässt sich ein Tiefenbild erzeugen. Der zusätzliche CMOS-Sensor ermöglicht die Aufnahme eines einfachen Kamerafarbbildes (vgl. [16], S. 2).

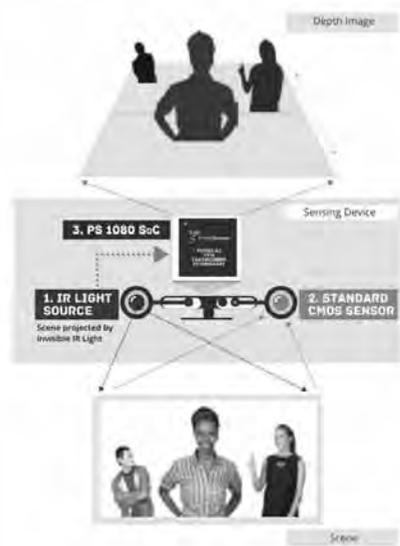


Abbildung 6: Erzeugung eines Tiefenbildes

Quelle: [16], S. 2

Auf Basis der Daten des Tiefenbildes kann dann, beispielsweise durch Software, eine Identifikation von Personen und deren Bewegungen erfolgen, eine Klassifikation von Objekten, eine Messung der Größe oder des Volumens sowie die Lage von Wänden und Boden ermittelt werden (vgl. [16], S. 2). Mit dem OpenNI (Open Natural Interaction) Framework der gleichnamigen Organisation wird eine Schnittstelle geboten, über welche mit der Hardware kommuniziert und Daten ausgelesen werden können. OpenNI ist ein quelloffenes Projekt und kann somit durch weitere Organisationen oder auch einzelne Entwickler beliebig erweitert und verändert werden (vgl. [8], S. 9).

3.2 Leap Motion

Im Juli 2013 brachte das Startup-Unternehmen Leap Motion seinen gleichnamigen Bewegungssensor auf den Markt. Dahinter verbirgt sich eine kleine Box, in der Größe eines Feuerzeugs.



Abbildung 7: Leap Motion Kontroller

Quelle: www5.pcmag.com/media/images/393268-leap-motion-controller.jpg?thumb=y

Platziert zwischen Display und Anwender erlaubt der Kontroller eine dreidimensionale Bewegungs- und Gestenerkennung der Hände und Finger innerhalb eines ca. 60 cm hohen und rund 150° großen Feldes über dem Kontroller. Leap Motion bewirbt sein Produkt mit der Erkennung aller zehn Finger und einer Genauigkeit von 0,001 mm (vgl. [12]). Technische Details werden Seitens Leap Motion nicht veröffentlicht. Nach Informationen der Firma SparkFun Electronics³ sind nur einfache elektronische Komponenten verbaut. Lediglich zwei CMOS Bildsensoren mit einer optischen Linse sowie drei Infrarot-LEDs sind Hauptbestandteil des Kontrollers. Die genaue Funktionsweise ist dabei nicht bekannt. SparkFun Electronics vermutet, dass die beiden Kameras auf jeden Fall zur stereoskopischen Bildverarbeitung dienen. Wird einer der beiden Sensoren verdeckt, funktioniert der Kontroller nicht mehr. Des Weiteren ist nur noch die USB Anschluss-technik verbaut. Dies lässt darauf schließen, dass die Verarbeitung der Daten überwiegend vom Rechner selbst und der mitgelieferten Software erledigt werden muss (vgl. [19]). Über die Airspace Oberfläche lassen sich Anwendungen aus unterschiedlichen Bereichen verwalten. Diese können über den Airspace Store, überwiegend kostenpflichtig, bezogen werden. Aktuell stehen ca. 170 Anwendungen zur Verfügung, wobei der größte Teil auf die Kategorie „Games“

³SparkFun Electronics ist ein Elektronikhändler welcher neben Herstellung und Verkauf von MicroController-Entwicklerboards, neue Gadgets wie die Leap Motion hinsichtlich technischer Bestandteile und Funktionen untersucht (vgl. [19]).

fällt. Über die Leap Motion API ist es Entwicklern möglich, eigene Anwendungen für den Bewegungssensor zu implementieren.

4 Anforderungsanalyse

Im folgenden Abschnitt werden sowohl die benutzerspezifischen als auch die technischen Anforderungen hinsichtlich des zu entwickelnden Systems zur Gebärdenspracherkennung aufgeführt. Die benutzerspezifischen Anforderungen basieren dabei auf den Aussagen von betroffenen bzw. fachnahen Personen aus dem Bereich der Gehörlosen-, Schwerhörigen- und Sprachbehindertenpädagogik. Hierzu wurden Interviews mit den entsprechenden Personen durchgeführt. Diese eignen sich vor allem für die Ermittlung der funktionalen und nicht-funktionalen Anforderungen aus Benutzersicht. Die funktionalen Anforderungen beschreiben dabei, welche Aufgabe bzw. welches erwartete Verhalten an das System gestellt werden soll. Nicht-funktionale Anforderungen beziehen sich unter Anderem auf die Qualität, Performance oder Bedienbarkeit des Systems. Hierbei hilft die Befragung der betroffenen und fachnahen Personen, da diese, als zukünftige Anwender, ein qualifiziertes Urteil bezüglich der benutzerspezifischen Anforderungen abgeben können.

Die technischen Anforderungen beziehen sich auf Auflösung, Geschwindigkeit und Bewegungsfreiheit der optischen Sensoren sowie auf das Produktionstempo der Gebärden. Hierfür wurden von den gehörlosen Personen jeweils zwei Sätze in eine Kamera gebärdet bzw. buchstabiert. Durch diese Aufnahmen erfolgt eine Zeitmessung der Gebärden pro Sekunde. Aus den gewonnenen Informationen der Anforderungsanalyse lässt sich bis zu einem gewissen Maß die Machbarkeit eines Systems zur Gebärdenspracherkennung bzw. zur Erkennung des Fingeralphabets ableiten. Daraus ergibt sich auch die abschließende Entscheidung, welcher Ansatz weiterhin in der Master-Thesis verfolgt wird.

4.1 Benutzerspezifische Anforderungen

Für die Ermittlung der benutzerspezifischen Anforderungen wurden zwei Interviews durchgeführt. Als Gesprächsgrundlage wurde ein Szenario in Form einer Alltagssituation vorgegeben. Dabei sollten sich die Interviewpartner einen Arztbesuch vorstellen, bei welchem kein Dolmetscher vorhanden ist und die Kommunikation zwischen Gehörlosem und Arzt direkt stattfindet. Im ersten Interview fand die Befragung mit einer hörenden Person statt. Die Fokusgruppe des zweiten Interviews bestand aus insgesamt drei gehörlosen Personen im Alter zwischen 30 – 60 Jahren. Dieses Interview entwickelte sehr schnell eine Eigendynamik und wurde zur Gruppendiskussion mit teils unterschiedlichen Ansichten.

Zusammengefasst sind alle befragten Personen in der Lage mit Hilfe der DGS zu kommunizieren, wobei es starke Unterschiede gibt, seit wann, wie lange und intensiv die DGS gelernt und eingesetzt wurde. Auch unter den Gehörlosen ist es möglich, dass diese nicht DGS-orientiert aufgewachsen sind und die Sprache teilweise erst spät erlernt haben. Grundsätzlich lässt sich jedoch sagen, dass die DGS für Gehörlose als Muttersprache anzusehen ist.

In Bezug auf die Selbsteinschätzung über die Kenntnis und Erfahrung von technischen Hilfsmitteln wie E-Mail, SMS, Mobiltelefon und Computer gehen alle Interviewpartner von einem guten bis sehr guten Kenntnisstand aus. Der Unterschied in der Nutzung von technischen Hilfsmitteln zwischen den gehörlosen und hörenden Personen beschränkt sich lediglich auf die Nutzung des Telefons.

4.1.1 DGS oder Fingeralphabet

In Bezug auf die Frage, ob eine Übersetzung der DGS in die deutsche Schriftsprache oder die Übersetzung des Fingeralphabets sinnvoller wäre, sind alle Interviewpartner gleicher Meinung. Hierbei steht die DGS eindeutig im Vordergrund. Das Fingeralphabet ist nur als Hilfsmittel zur Erklärung einzelner Wör-

ter oder Eigennamen anzusehen. Meist werden schnell neue Eigengebärden erfunden. Zudem erfordert das Gebärden und Verstehen des Fingeralphabets mehr Konzentration und oftmals werden nur Bruchstücke verstanden. Ganze Sätze werden in der Regel nicht durch das Fingeralphabet gebärdet. Die DGS bietet hier eine weitaus einfachere und vor allem schnellere Möglichkeit der Kommunikation. Die Erkennung des Fingeralphabets wäre demnach nur eine zusätzliche Option.

4.1.2 Übersetzungsrichtung

Innerhalb der Fokusgruppe wird eine Übersetzungsrichtung von der DGS in die deutsche Schriftsprache bevorzugt. Die Interviewpartner beschreiben dies damit, dass DGS-orientierte Menschen in der Regel beide Sprachen lernen. Auch wenn bei der deutschen Schriftsprache Defizite existieren, ist die Verständlichkeit für geschriebene Texte relativ groß. Dem gegenüber wird das Verfassen eigener Texte entsprechend schwieriger empfunden. Verglichen wird dies mit einer anderen Fremdsprache wie z.B. Englisch. Meist wird auch hier mehr verstanden wie selbst artikuliert. Einige DGS-orientierte Menschen können zudem viele Informationen über das Mundbild gesprochener Sätze aufnehmen. Hörende Menschen hingegen verstehen kaum etwas von der Gebärdensprache, es sei denn sie wurde erlernt. Somit steigt die Relevanz einer Übersetzung aus der DGS in die deutsche Schriftsprache. Dennoch sollte auch der umgekehrte Weg nicht ganz außer Acht gelassen werden. Vor allem bei Gruppengesprächen eines Gehörlosen mit mehreren hörenden Personen sinkt die Aufnahmefähigkeit der gehörlosen Person stark. Hier kam der Vorschlag seitens der Interviewpartner, die gesprochene Sprache über eine virtuell gebärdende Person in die DGS übersetzen zu lassen.

4.1.3 Grammatikalische Übersetzung

Über die grammatikalische Übersetzung der DGS in die deutsche Schriftsprache herrscht eine geteilte Meinung innerhalb der Fokusgruppe. Zwei Interviewpartner sind der Mei-

nung, dass zunächst eine Übersetzung der Inhalte wichtig ist, da der Kontext dennoch verstanden werden kann, auch wenn der jeweilige Satz nicht in grammatikalisch richtigem Schriftdeutsch vorliegt. Viel wichtiger als eine grammatikalische Übersetzung wäre eine Erkennung der richtigen Gebärde auf Basis der gesamten Körpersprache. Dazu zählt auch die Beachtung der Mimik, des Mundbildes und der Körpersprache. So wäre es beispielsweise auch möglich, statt der DGS aus der LBG zu übersetzen. Die LBG beschreibt dabei lautsprachen begleitende Gebärden, welche sich an der deutschen Schriftgrammatik orientieren. Dies würde jedoch für DGS-orientierte Menschen bedeuten, dass sie entgegen ihrer Muttersprache die Sätze in deutscher Grammatik bilden müssten. Für die anderen Interviewpartner ist eine grammatikalische Übersetzung wichtig und sollte auf jeden Fall durchgeführt werden. Dennoch sehen alle Interviewpartner auch eine gewisse Schwierigkeit der grammatikalischen Übersetzung, da hier mehrere Faktoren eine Rolle spielen. Eine Übersetzung mit einfachen grammatikalischen Regeln, wie das bereits in Abschnitt 2.1 aufgeführte AVP \Rightarrow APV, sollten für den Anfang verfolgt und umgesetzt werden.

4.1.4 Persönliche Erwartungen

Alle Interviewpartner können sich ein System zur Übersetzung der DGS vorstellen und würden dies auch effektiv einsetzen. Vor allem in Alltagssituationen wie beim Einkaufen, wo keine komplexen Themeninhalte gesprochen werden, sehen sie ein solches System als sinnvolle Unterstützung an. Bei komplexen Themenbereichen wie bei Ärzten oder Ämtern ziehen die gehörlosen Interviewpartner jedoch immer noch einen realen Dolmetscher vor. Dies liegt ein wenig in der Vorstellungskraft der Interviewpartner, da sich derzeit keiner ein voll funktionsfähiges und vor allem fehlerfreies System vorstellen kann.

Ein weiterer wichtiger Punkt bezieht sich auf die Größe und damit die Einsetzbarkeit des Systems. Es muss klein, handlich und

praktikabel sein. Als Beispiel wird hier das Handy bzw. Smartphone genannt. Die Übersetzung muss annähernd in Echtzeit erfolgen. Kleine Latenzen werden akzeptiert, dürfen jedoch nicht störend sein. Neben der reinen Übersetzung aus der DGS in die deutsche Schriftsprache werden Erweiterungen wie z.B. eine Sprachein- und -ausgabe, eine Speichermöglichkeit oder auch Mustertexte vorgeschlagen. Auch sollte es möglich sein, neue Gebärden in das System übertragen zu können.

4.2 Technische Anforderungen

Bezüglich der technischen Anforderungen wurden die bereits aufgeführten Bewegungssensoren mit Hilfe ihrer Demoanwendungen auf ihre Einsatzmöglichkeit hinsichtlich der Gebärdensprach- und Fingererkennung getestet. Hier spielen die Auflösung und damit die Genauigkeit der Systeme sowie die Bewegungsfreiheit und die Bewegungsrichtung eine ausschlaggebende Rolle. Auch die Ermittlung des durchschnittlichen Produktionstempos einer Gebärde ist entscheidend, da hiermit eine Einschätzung gegeben werden kann, mit welcher Mindestgeschwindigkeit das System reagieren sollte.

4.2.1 Produktionstempo

Unter dem Begriff des Produktionstempos versteht man, wie bereits in Abschnitt 2.1 angesprochen, den zeitlichen Zusammenhang hinsichtlich der Erzeugung des sprachlichen Ausdrucks durch die Gebärden oder durch die Darstellung der Buchstaben des Fingeralphabets (vgl. [1], S.46f). Die Ermittlung der durchschnittlichen Zeit pro Gebärde bestimmt die Mindestgeschwindigkeit, mit welcher ein System zur Gebärdenspracherkennung agieren muss. Werden Gebärden erst später oder langsamer erfasst, entstehen Latenzen, welche sich nachteilig auf die Benutzerfreundlichkeit auswirken. Dies bedeutet gleichzeitig, dass die Akzeptanz eines solchen Systems stark abnehmen würde. In Bezug auf den Versuch von Bellugi und Fischer (1972) wurden folgende Sätze von den Gehörlosen der Fokusgruppe in eine

Kamera gebärdet:

„Hallo, mein Name ist V-o-r-n-a-m-e-N-a-c-h-n-a-m-e.“

„Ich arbeite bei der Paulinenpflege in Winnenden.“

Im ersten Satz sollten dabei Vorname und Nachname im Fingeralphabet buchstabiert werden. Die Aufnahme erlaubt, bedingt durch das Aufnahmeformat, eine zeitliche Messung mit einer Genauigkeit von bis zu 25 Bildern pro Sekunde (fps). Dies entspricht 40 Millisekunden (ms). Die Auswertung ergab dabei die in Tabelle 2 aufgeführten Ergebnisse.

Um beide Sätze, inklusive dem Buchstabieren des Namens, zu gebärden, benötigten die Testpersonen im Schnitt 9,4 sek. Betrachtet man lediglich die Gebärden der DGS, so werden ca. 0,55 sek pro Gebärde und ca. 1,87 sek pro Satzinhalt benötigt. Die minimale Abweichung im Vergleich zum Messergebnis von Bellugi und Fischer ist auf die geringe Anzahl von Sätzen und gebärdenden Personen zurückzuführen. Dennoch ähnelt das Ergebnis der Studie von 1972 (vgl. [15]). Betrachtet man die durchschnittlich benötigte Zeit pro Gebärde im Fingeralphabet, fällt auf, dass diese mit ca. 0,37 sek deutlich schneller durchgeführt werden kann. Dies liegt schätzungsweise am kleineren Gebärdenbereich und der damit geringeren Bewegungsstrecke. Für ein System zur Gebärdenpracherkennung müsste demnach die Erfassung der jeweiligen Gebärde mindestens innerhalb von ca. 500 ms in der DGS und ca. 350 ms im Fingeralphabet erfolgreich durchgeführt und verarbeitet werden.

4.2.2 Räumliche Auflösung und Erfassungsgeschwindigkeit

Mit der räumlichen Auflösung und der Erfassungsgeschwindigkeit lässt sich die Genauigkeit des Bewegungssensors beschreiben, mit welcher Objekte erkannt sowie deren

Bewegungen erfasst und verarbeitet werden können. Vor allem bei der Erkennung der Hände und einzelner Finger sollte eine möglichst hohe Auflösung gegeben sein. Tabelle 3 zeigt die Auflösungen der Bewegungssensoren sowie deren Erfassungsgeschwindigkeiten im Vergleich.

Die aufgeführten Werte sind Angaben der Hersteller und können nicht messbar überprüft werden. Dennoch bietet die Leap Motion eine deutlich höhere Auflösung und Erfassungsgeschwindigkeit. Sie wurde primär für die Erkennung der Hände und Finger entwickelt, während die Asus Xtion als auch die Sensoren von PrimeSense primär zur Erkennung des gesamten Körpers konzipiert wurden. Für den Einsatz zur Erkennung des Fingeralphabets steht demnach die Leap Motion an oberster Stelle. Aber auch die Sensoren von Asus und PrimeSense ermöglichen, mittels der speziellen API des amerikanischen Unternehmens 3Gear Systems, die Erkennung beider Hände und aller zehn Finger. Die API basiert dabei auf einer Erweiterung des OpenNI Frameworks. Da es mit der Leap Motion nicht möglich ist ein Körpertracking zu realisieren, wird im Anschluss lediglich das Fingertracking betrachtet und getestet.

Ein messbarer Vergleich der Hardware ist mit den vorhandenen Mitteln nicht zu realisieren. Daher wird die Hardware auf Basis der dazugehörigen Demoanwendungen getestet und beurteilt. Als Testumgebung dient ein Mac Book Pro mit einem 2,4 GHz Intel Core i5 Prozessor, 8GB RAM und dem aktuellen Betriebssystem OS X in der Version 10.9.1. Hinsichtlich des Fingertrackings wird die Leap Motion mit der PrimeSense Carmine 1.09 und deren jeweils aktuellsten Softwareversionen verglichen.

Die Leap Motion wird standardmäßig mit dem sogenannten Visualizer ausgeliefert. Diese Demoanwendung erlaubt die Erfassung und grafische Darstellung der Hände und Finger. Im Einstellungsfeld der Leap Motion lässt sich während der Nutzung die aktuelle Daten-

Tabelle 2: Messergebnis Produktionstempo

Messwert	Gesamt Sätze bzw. Name		Durchschnitt pro Gebärde	
	fps	ms	fps	ms
Gebärden DGS	93,66	3746,66	13,73	549,33
Fingeralphabet	141,33	5653,33	9,15	365,86
Gesamt	234,99	9399,99	-	-

Tabelle 3: Auflösung und Geschwindigkeit*Quelle: Vgl. [12] und [16]*

	Leap Motion	Asus Xtion/Carmine 1.08	Carmine 1.09
Räumliche Auflösung	0,001 mm	3,4 mm	0,9 mm
Tiefenauflösung	0,001 mm	12 mm	1 mm
Geschwindigkeit	>200 fps	max. 60 fps	max. 60 fps

übertragungsrate sowie die Verarbeitungszeit anzeigen. Durchschnittlich stellte sich während des Tests eine Datenübertragungsrate von ca. 90 fps und einer Verarbeitungszeit von ca. 8 ms ein. Vergleichbar hierzu wird das Fingertracking der Carmine 1.09 mit der Demoanwendung HandViewer aus der bereits erwähnten API von 3Gear Systems getestet.

Ergebnisse übertreffen dennoch die Anforderungen des in Abschnitt 3.2.1 ermittelten Produktionstempos deutlich und könnten somit zur reinen Erfassung von Gebärden der DGS und des Fingeralphabets eingesetzt werden. Zum Test selbst werden nacheinander verschiedene Handformen und Bewegungen durchgeführt und deren visueller Eindruck beurteilt.

**Abbildung 8: Darstellung Visualizer (oben), HandViewer (unten)**

Bezüglich der Geschwindigkeit ist ein deutlicher Unterschied, vor allem bei schnellen Bewegungen, zu verzeichnen. Hier bietet die Leap Motion mit ihrer höheren Datenübertragungsrate einen flüssigeren und annähernd realen Bewegungsablauf. Über den HandViewer wirken die Bewegungen etwas langsamer und stockend. Schnelle Bewegungen der Finger können teilweise nicht erfasst werden.

Bei unterschiedlichen Handformen fällt im Visualizer der Leap Motion auf, dass eng beieinander liegende Finger schnell zu einer Erfassungsschwierigkeit und damit zum Verschwinden der Finger führt. Dies wird gerade bei einer Faust oder einer Rotation der Hand, bei welcher sich die Finger überlagern, stark wahrgenommen. Im HandViewer in Verbindung mit der Carmine 1.09 werden

Während des Tests stellte sich hier eine durchschnittliche Datenübertragungsrate von ca. 59 fps ein. Eine Verarbeitungszeit wird seitens der Anwendung nicht gegeben. Beide

solche Überlagerungen teilweise kompensiert. Selbst eine zur Kamera orthogonal stehende Hand wird dargestellt und bei einer Bewegung einzelner Finger werden diese trotz Überlagerung erfasst. Allerdings sind auch hier Erfassungsfehler zu verzeichnen. Beispielsweise wird bei einer Faust, die den Daumen unter oder über den restlichen Fingern hat, kein Unterschied wahrgenommen. In Bezug auf die Bewegungsauflösung ist die Leap Motion deutlich filigraner. Kleinste Bewegungen des Zeigefingers im Millimeterbereich werden erfasst. Hier kann es bei der Carmine 1.09 vorkommen, dass solch geringe Bewegungen erst gar nicht erfasst werden können.

Hinsichtlich der Gebärden des Fingeralphabets, bei welchem die Finger oft eng beieinander liegen, verzeichnen beide Systeme deutliche Schwächen bei der Auflösung. Hier entstehen in der Regel stärkere Erfassungsfehler. Dennoch ist es denkbar, dass diese Fehler durch die Entwicklung bestimmter Algorithmen kompensiert werden könnten. Ein Beispiel hierfür bieten Zagel, Süßmuth und Bodendorf in [22]. In ihrer Arbeit „*Automatische Rekonstruktion eines 3D Körpermodells aus Kinect Sensordaten*“ wird ein dreidimensionales Körpermodell aus dem Tiefenbild des Kinect Sensors durch einen speziellen Algorithmus errechnet. Der verwendete Algorithmus gleicht dabei Artefakte wie z.B. Rauschen oder Quantisierungsartefakte, welche durch die limitierte Auflösung des Sensors entstehen, aus (vgl. [22], S. 554f). Eine Aussage darüber, ob und wie gut die Erfassung mit beiden Systemen erfolgen kann, ist zum aktuellen Zeitpunkt nicht möglich, da dies einen konkreten Test mit einer Software zur Erkennung der Gebärden voraussetzen würde. Dennoch ist der Einsatz beider Systeme hinsichtlich der räumlichen Auflösung und der Erfassungsgeschwindigkeit denkbar.

4.2.3 Bewegungsfreiheit und Bewegungsrichtung

Sowohl die Gebärden der DGS als auch die des Fingeralphabets werden innerhalb

des Gebärdenraums in Sprachrichtung zum Zuseher gebärdet. Dies erfordert eine frontale Erkennung der Gebärden. Die Asus Xtion sowie die PrimeSense Carmine sind in technischer Hinsicht für diesen Frontaleinsatz konzipiert. Leap Motion bietet hingegen nur eine Erkennung der Hände und Finger über dem Controller. Bezüglich der Erfassung von Gebärden der DGS kann die Leap Motion nicht eingesetzt werden, da der Erfassungsraum zu klein und damit kein Körpertracking möglich ist. Sowohl die Asus Xtion als auch die Sensoren von PrimeSense ermöglichen bereits, mittels OpenNI Framework, ein Körpertracking. Hier kann beispielsweise der gesamte Körper oder aber auch nur der Oberkörper erfasst werden. Durch eine Erweiterung des Frameworks könnte demnach die gewünschte Gebärdenspracherkennung realisiert werden.

Für die Erfassung des Fingeralphabets müsste die Leap Motion um 90° gedreht werden und sich die gebärdende Person sehr nahe am Controller befinden, da die Erfassungstiefe lediglich max. 60 cm beträgt. Ein Test mit dem Visualizer zeigt, dass dies nicht so einfach zu realisieren ist. Die reine Erfassung der Hände bei einem gedrehten Controller funktioniert, solange sich keine Fläche oder kein Objekt hinter den Händen befindet. Da das Fingeralphabet jedoch vor dem Körper in Brusthöhe gebärdet wird, kann hier keine Erfassung erfolgen. Beim Fingertracking in Verbindung mit der PrimeSense Carmine 1.09 erfolgt das Fingertracking von oben. Auch hier muss der Sensor zunächst gedreht werden. Die Frontalerfassung ist hier nur eingeschränkt möglich, wenn sich die Hände vor einem ebenen Hintergrund befinden. Vor dem Körper steigt die Fehlerrate jedoch stark an. Somit können beide Sensorarten nicht für eine Frontalerkennung des Fingeralphabets eingesetzt werden.

5 Zusammenfassung

Durch die mit der Fokusgruppe geführten Interviews wird deutlich, dass aktuell immer noch Probleme in der Kommunikation zwischen gehörlosen und hörenden Menschen

bestehen. Es gibt nur wenig effektive Hilfsmittel, sodass meist ein realer Dolmetscher zum Einsatz kommt. Die Idee einer computergestützten Erkennung und Übersetzung der Gebärdensprache findet Anklang bei der Fokusgruppe. Allerdings begrenzt sich die Vorstellung derzeit noch auf ein System, welches nur für einfache Gespräche verwendet wird. Für komplexe Gespräche, wie z.B. bei Ärzten oder Ämtern, wird weiterhin noch der reale Dolmetscher bevorzugt. Das System selbst sollte natürlich portabel, einfach zu nutzen und schnell sein. Geringe Latenzzeiten werden akzeptiert. Zunächst wird die Übersetzungsrichtung aus der DGS in die deutsche Schriftsprache gewünscht. Aber auch der umgekehrte Weg sollte nicht außer Acht gelassen werden. Weitere Optionen könnten eine Sprachein- und ausgabe, eine Speicherfunktion sowie die Möglichkeit zur Erweiterung des Wortschatzes durch neue Gebärden sein.

In technischer Hinsicht zeichnet sich der Trend zu den Sensoren von Asus und PrimeSense ab. Trotz der geringeren räumlichen Auflösung sowie der langsameren Erfassungsgeschwindigkeit, lassen sich diese Sensoren vielseitiger einsetzen. Auch die Möglichkeit mit den Sensoren eine Erkennung der Mimik realisieren zu können, spricht stark für den Einsatz bei der computergestützten Übersetzung aus der DGS in die deutsche Schriftsprache. Alle ermittelten Anforderungen werden nochmals in Tabelle 4 aufgeführt.

6 Handlungsempfehlung

In Anbetracht der technischen und benutzerspezifischen Anforderungsanalyse wird für die Entwicklung eines Systems zur Erkennung und Übersetzung der DGS in die deutsche Schriftsprache folgende Handlungsempfehlung ausgesprochen. Im ersten Schritt sollte die Erfassung der Gebärden im Vordergrund stehen. In technischer Hinsicht werden dafür die Sensoren Asus Xtion oder PrimeSense Carmine empfohlen. Die Software muss daher zunächst auf einem stationären PC oder Laptop lauffähig sein. Dies begründet sich durch die vorgeschlagenen Bewegungssensoren,

welche für den mobilen Einsatz aufgrund ihrer Größe ungeeignet sind. PrimeSense entwickelt aktuell diesbezüglich den Bewegungssensor Capri 1.25, welcher für den mobilen Einsatz vorgesehen ist (vgl. [16], S. 3). Die Entwicklung einer mobilen Version könnte dann, nach erfolgreichem Testlauf des stationären Systems, erfolgen. Aus Entwicklersicht müssen, für die Erfassung der Gebärden, eigene Gesten programmiert werden, welche beim Erfassungsvorgang ggf. über eine Datenbank abgeglichen werden können. Ist die Erfassung mehrerer Gebärden möglich, kann im zweiten Schritt die grammatikalische Übersetzung implementiert werden. Zunächst sollten hier jedoch nur einfache grammatikalische Regeln in Betracht gezogen werden. Ziel für einen Prototyp ist die Möglichkeit zur Erkennung von ausgewählten Gebärden, aus welchen bspw. zwei unterschiedliche Sätze erzeugt und übersetzt werden können.

7 Danksagung

Mein Dank gilt zunächst meiner Lebensgefährtin Frau Manuela Schreiber, ohne die ich vermutlich nicht auf dieses interessante und vor allem wichtige Thema gestoßen wäre. Sie ermöglichte mir auch die einfache und problemlose Kommunikation und Organisation mit den Interviewpartnern der Paulinenpflege Winnenden e.V. Ebenfalls geht mein Dank an die Schulleiterin der Schule beim Jakobsweg Frau Beate Löffler sowie die stellvertretende Schulleiterin Frau Christiane Sättler-Adel, welche mir zunächst die Durchführung des Interviews an ihrer Schule ermöglichten. Auch möchte ich mich diesbezüglich für die Bereitstellung des Raumes und der Verpflegung bedanken sowie für den Einsatz der Dolmetscherin Frau Sonja Lewandowsky und ihrer Praktikantin Antje Voss. Ganz großer Dank gilt natürlich den Interviewpartnern Frau Gabi Braig, Herrn Christian Hermann und Herrn Martin Lindenberger von der Paulinenpflege Winnenden e.V. sowie Herrn Martin Müller von der Stiftung St. Franziskus Heilbronn. Allen zusammen nochmals ein herzliches Dankeschön für Ihre Mitarbeit.

Tabelle 4: Ermittelte Anforderungen

Benutzerspezifische Anforderungen	Übersetzung	<i>DGS</i>
	Übersetzungsrichtung	<i>DGS ⇒ dt. Schriftsprache</i>
	Grammatik	<i>Zunächst einfache Regeln (AVP ⇒ APV)</i>
	Persönliche Erwartungen	<ul style="list-style-type: none"> - <i>Portabel</i> - <i>Praktikabel</i> - <i>Einfache Nutzung</i> - <i>Annähernd Echtzeitübersetzung</i>
Technische Anforderungen	Produktionstempo/ Erfassungsgeschwindigkeit	≤ 500 ms/ <i>Gebärde (DGS)</i>
	Auflösung	<i>Die Auflösung wird bestimmt durch die Erfassungsgenauigkeit und die Erfassungsgeschwindigkeit. Je höher die Auflösung, desto genauer können Erfassungen durchgeführt werden.</i>
	Bewegungsrichtung/ Bewegungsfreiheit	<ul style="list-style-type: none"> - <i>Frontal zum Anwender</i> - <i>Gebärdenraum muss vollständig erfasst werden</i>
	Empfohlener Sensortyp	<ul style="list-style-type: none"> - <i>RGB-Tiefensensor</i> - <i>Asus Xtion</i> - <i>PrimeSense Carmine</i>

Literatur

- [1] P. Boyes-Braem. *Einführung in die Gebärdensprache und ihre Erforschung*, volume Bd. 11. Signum, Hamburg, 2. korrigierte auflage edition, 1992.
- [2] K.-U. Carstensen, C. Ebert, C. Ebert, S. Jekat, R. Klabunde, and H. Langer. *Computerlinguistik und Sprachtechnologie: Eine Einführung*. Spektrum, Akad. Verl., Heidelberg, 3., überarb. und erw. aufl edition, 2010.
- [3] X. Chai, G. Li, X. Chen, M. Zhou, G. Wu, and H. Li. Visualcomm: A tool to support communication between deaf and hearing persons with the kinect. In *ASSETS'13*, pages 1–2.
- [4] X. Chen, L. Hanjing, T. Pan, S. Tansley, and M. Zhou. Kinect sign language translator expands communication possibilities, 2013.
- [5] Deutscher Gehörlosen-Bund e.V. *Faq - gehörlosigkeit*, 2014.
- [6] M. Dos Santos Anjo, E. Brigante Pizzolato, and S. Feuerstack. A real-time system to recognize static gestures of brazilian sign language (libras) alphabetu sing kinect. In C. Maciel and Interaction, ACM Special Interest Group on Computer-Human, editors, *IHC'12*,

- pages 259–268, [S.l.], 2012. Brazilian Computer Society.
- [7] H. Eichmann, M. Hansen, and J. Heßmann. *Handbuch deutsche Gebärdensprache: Sprachwissenschaftliche und anwendungsbezogene Perspektiven*, volume Bd. 50 of *Internationale Arbeiten zur Gebärdensprache und Kommunikation Gehörloser*. Signum, Seedorf, 2012.
- [8] S. Falahati. *OpenNI Cookbook*. Packt Publishing, Birmingham, 2013.
- [9] S. Fünfer. *Mensch oder Maschine? Dolmetscher und maschinelles Dolmetschsystem im Vergleich*, volume Bd. 2 of *Transkulturalität - Translation - Transfer*. Frank & Timme, Berlin, 2013.
- [10] M. Hillenmeyer. *Dgs - deutsche gebärdensprache 1: Grundkurs für anfangler*, 2000.
- [11] W. Jiangquin, G. Wen, S. Yibo, and L. Wei. A simple sign language recognition system based on data glove. In Institute of Electrical and Electronics Engineers, editor, *International Conference on Signal Processing (ICSP)*, pages 1257–1260. I.E.E.E. Press, 1998.
- [12] Leap Motion, Inc. Leap motion - produktionformationen, 2014.
- [13] MAKATON Deutschland. Was sie über makaton wissen sollten, 2014.
- [14] J. Muhs. Die geschichte der gebärdensprache - von der römerzeit bis heute, 2014.
- [15] T. D. Pahlke. Interpretation der studie „hörscreening“ aus sicht des deutschen schwerhörigenbundes, 2001.
- [16] PrimeSense. Primesense 3d sensors: Produkt- und technologiebeschreibung, 2014.
- [17] P. A. Reich and M. Bick. An empirical investigation of some claims made in support of visible english. *American annals of the deaf*, 121(6):573–577, 1976.
- [18] W. Roddeck. *Einführung in die Mechatronik: Mit 9 Tabellen*. Lehrbuch. Teubner, Wiesbaden, 3., überarb. und erg. aufl edition, 2006.
- [19] SparkFun Electronics. Leap motion tear-down, 2014.
- [20] Technabling Ltd. Pslt - portable sign language translator, 2014.
- [21] A. P. Vicente and A. A. Faisal. Calibration of kinematic body sensor networks: Kinect-based gauging of data gloves “in the wild”. In IEEE, editor, *2013 IEEE International Conference on Body Sensor Networks (BSN)*, pages 1–6, 2013.
- [22] C. Zigel, J. Süßmuth, and F. Bodendorf. Automatische rekonstruktion eines 3d körpermodells aus kinect sensordaten: Wirtschaftsinformatik proceedings 2013, paper 35, 2013.

Augmented Virtuality zur Steigerung der Immersion in virtuellen Umgebungen *

Matthias Gutekunst
Reutlingen University
Matthias.Sebastian.Gutekunst
@Student.Reutlingen-University.DE

Abstract

In mehreren Untersuchungen hat sich gezeigt, dass sich die Wahrnehmung des eigenen Körpers in einer virtuellen Umgebung positiv auf die Wahrnehmung der gesamten Umgebung auswirkt. Für diese Untersuchungen wurden der Körper einer Person, oder Teile davon, als animierter Avatar aus der Ego-Perspektive dargestellt. Im Kontext der Informatikkonferenz *Informatics Inside 2014* an der Hochschule Reutlingen soll in dieser Arbeit eine andere Möglichkeit der Darstellung untersucht werden. In einer prototypischen Augmented Virtuality Anwendung soll die virtuelle Umgebung um realen Inhalten erweitert werden. Es soll einer Person ermöglicht werden, Teile ihres eigenen Körpers nicht als Avatar, sondern auf Basis einer Kameraaufnahme als realistische Repräsentation wahrzunehmen. Die Arbeit beschreibt hierbei die gesetzten Ziele, sowie Aufbau und Funktionsweise der prototypischen Anwendung und deren derzeitigen Stand.

Schlüsselwörter

Bildsegmentierung, Objekterkennung, Augmented Virtuality, Selbstwahrnehmung, Head-Mounted Display

*

Betreuer Hochschule: Prof. Dr. rer. nat. Uwe Kloos
Hochschule Reutlingen
Uwe.Kloos@Reutlingen-
University.de

Informatics Inside 2014,
Wissenschaftliche Vertiefungskonferenz
30.April 2014, Hochschule Reutlingen
Copyright 2014 Matthias Gutekunst

CR-Kategorien

I.4.6 [Image Processing and Computervision]: Segmentation; I.4.8 [Image Processing and Computervision]: Scene Analysis—*Object recognition*

1 Einleitung

Neue Entwicklungen im Bereich der Head-Mounted Displays öffnen den Bereich der immersiven virtuellen Umgebungen für eine breite Masse an Nutzern und Anwendungsszenarien. Produkte wie die von OculusVR¹ entwickelte Oculus Rift stellen eine kostengünstige Möglichkeit dar, um interaktive, virtuelle Umgebungen mit einem hohen Grad an Immersion zu präsentieren. Ein Sichtfeld mit weitem Blickwinkel und ohne erkennbare Ränder, sowie Echtzeit-Headtracking ermöglichen es eine Person zu großen Teilen von der realen Umgebung abzuschotten. Der Person wird somit ein Eintauchen in die virtuelle Umgebung ermöglicht. Sie kann sich in der virtuellen Welt umsehen und je nach verwendetem Verfahren natürlich bewegen. Die Abschottung von der Realität stellt jedoch zur gleichen Zeit ein Hindernis dar, wenn es darum geht in der virtuellen Welt zu interagieren. Ein Grund dafür ist, dass eine Person nicht in der Lage ist ihren eigenen Körper als Referenz wahrzunehmen. Basierend auf der Aussage von Gibson [4] geht Linkaeger davon aus, dass ein Mensch nicht seine Umgebung wahrnimmt, sondern die Beziehung zwischen sich selbst und seiner Umgebung [6, S.1]. Der Grad an Immersion einer virtuellen Umgebung hängt also direkt von der Selbstwahrnehmung einer Person innerhalb dieser ab.

¹Oculus VR: <http://www.oculusvr.com/>

1.1 Verwandte Arbeiten

Die Darstellung des eigenen Körpers oder Teilen davon und die damit verbundenen Auswirkungen auf die Wahrnehmung einer virtuellen Welt wurden in mehreren Experimenten untersucht. Linkenauger et al. [6, S. 3] untersuchten den Einfluss der Größe eines Avatars auf die Größenwahrnehmung von anderen im Raum befindlichen virtuellen Gegenständen. Der Avatar war hierbei eine virtuelle Hand- und Arm Repräsentation, deren Größe in mehreren Versuchen angepasst wurde. Es hat sich dabei gezeigt, dass die Größenwahrnehmung anderer virtueller Objekte direkt mit der Größe des Avatars zusammenhängen. Aus diesem Versuch und mehreren Kontrollversuchen wurde geschlossen, dass der eigene Körper bzw. eine Repräsentation davon eine Hauptrolle bei der Größenwahrnehmung von virtuellen Objekten einnimmt [6, S. 8]. Die Testpersonen assoziieren den Avatar demnach als Körperteil und verwenden diesen als Referenz für die Größenerkennung. Eine weitere Versuchsreihe von Cream-Regehr et al. [2] beschäftigt sich mit den Auswirkungen eines Avatars auf die Abschätzung von Distanzen in einer virtuellen Umgebung. Hierbei wurde festgestellt, dass Personen weniger Fehler bei der Distanzabschätzung in einer virtuellen Umgebung machen, wenn sie zuvor die Möglichkeit hatten sich selbst als Avatar in einer solchen Umgebung wahrzunehmen.

1.2 Zielsetzung

Basierend auf den Ergebnissen der Untersuchungen in Kapitel 1.1 soll geprüft werden, in wie fern sich die Wahrnehmung einer Person ändert, wenn statt den in diesen Experimenten verwendeten Avataren eine möglichst realitätsnahe Darstellung des eigenen Körpers in die virtuelle Umgebung integriert wird. Zu diesem Zweck ist das Hauptziel dieser Arbeit eine Möglichkeit zu erforschen, um es Personen zu ermöglichen, ihre eigenen Körperteile innerhalb einer virtuellen Umgebung aus der Ego-Perspektive direkt wahrzunehmen. Anstatt einen Avatar einzusetzen, welcher lediglich die Bewegungen einer Person in die virtuelle Welt überträgt, sollen bei diesem An-

satz die Körperteile einer Person von einer Kamera erfasst werden und die virtuelle Umgebung soll mit diesen Bildern erweitert werden. Die Annahme ist dabei, dass sich eine Person durch eine realitätsnahe Darstellung des eigenen Körpers noch besser mit dieser Darstellung identifizieren kann und dadurch die Grenzen zwischen virtueller und realer Umgebung weiter verwischen. Dadurch soll ein höherer Grad an Immersion erreicht werden.

2 Vorgehensweise

Basierend auf der Zielsetzung wurden zunächst Anforderungen für eine Augmented Reality Anwendung definiert, durch die es ermöglicht werden soll, Kameraaufnahmen von ausgewählten realen Objekten in eine virtuelle Umgebung zu integrieren. Im Fall des Prototyps sollen dabei die Hände einer Person aus einer Aufnahme aus der Ego-Perspektive extrahiert und anschließend mit der virtuellen Welt zu verschmelzen werden. Neben dieser funktionellen Anforderung, ist eine Hauptanforderung eine hohe Performanz, sowohl bei der hardwareseitigen Umsetzung als auch bei der Verarbeitung der Kameradaten. Dies ist notwendig, da eine niedrige Bildwiederholungsrate und ein damit verbundener Versatz zwischen einer Bewegung und deren Darstellung sensorische Konflikte hervorrufen oder verstärken kann. Diese wirken sich negativ auf die Immersion aus und können Symptome der Simulatorkrankheit auslösen [5, S. 50]. Des Weiteren soll die Wahrnehmung des eigenen Körpers nicht durch andere Objekte gestört werden, weshalb eine weitere Anforderung ist, die Verarbeitung der Kameradaten ohne die Verwendung von Markern durchzuführen.

Gemäß den zuvor genannten Anforderungen wurde eine prototypische Anwendung entwickelt, welche es dem Nutzer ermöglicht, seine Hände in einer virtuellen Umgebung wahrzunehmen. Dieser Prototyp teilt sich in eine Software- und eine Hardwarekomponente. Es wurde daher zunächst nach geeigneter Kamerahardware, für die Verwendung in Kombination mit dem Oculus Rift Development Kit, gesucht. Interessante Kandidaten waren

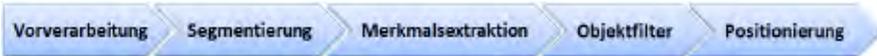


Abbildung 1: Verarbeitungsschritte der Anwendung

neben der für die Versuche verwendeten Logitech C525, eine IDS UI-3240CP, eine VRmC-16/C COB, die RGB Kamera aus dem Microsoft Kinect System sowie eine PS3 Eye Kamera. Mit Ausnahme der PS3 Eye Kamera verfügen alle diese Kameras über eine Auflösung, die der Ausgabe der Oculus Rift nahezu entspricht. Des Weiteren weisen alle Kameras eine hohe Bildwiederholungsrate von mindestens 30Hz und bis zu 60Hz auf. Für den Prototyp wurden mehrere Aufbauten, sowohl mit einer Kamera als auch stereoskopisch, erstellt. Der in Abbildung 2 dargestellte Prototyp wurde hierbei die meiste Zeit während der Entwicklung der Anwendung eingesetzt. Es handelt sich um einen nicht stereoskopischen Aufbau mit einer Logitech C525 Webcam, bei dem das Kamerabild für die Ausgabe dupliziert wurde. Die Kamera wurde wegen einer hohen Bildwiederholungsrate sowie einer relativ gut zur Ausgabe der Oculus Rift passenden Auflösung verwendet. Ein weiterer vielversprechender Ansatz für einen stereoskopischen Hardwareprototypen von Steptoe[7] wurde erst gegen Ende der Entwicklungsphase gefunden und konnte daher in dieser Arbeit nicht mehr betrachtet werden. Bei diesem Ansatz wurde eine Logitech C310 Kamera mit einem Weitwinkel Objektiv ausgerüstet, um das große Sichtfeld der Ausgabe der Oculus Rift schon bei der Aufnahme abdecken zu können. Die prototypische Anwendung wurde mit Hilfe der Open Source Bibliotheken OpenCV² und OpenGL³ als Plugin für die Unity⁴ Entwicklungsumgebung entworfen. Das Ziel war es hierbei eine Anwendung zu schaffen, welche ein Objekt (im speziellen die Hände einer Person) aus einer Aufnahme extrahiert und innerhalb einer virtuellen Welt positioniert.



Abbildung 2: Hardware Prototyp: Oculus Rift + Logitech C525

3 Funktionsweise der prototypischen Anwendung

Abbildung 1 zeigt die einzelnen Verarbeitungsschritte, die verwendet werden, um Objekte aus dem Kamerabild zu extrahieren und diese in einer virtuellen Umgebung zu platzieren. Im aktuellen Stand des Prototyps werden so die Hände einer Person in einer virtuellen Umgebung dargestellt. Die Vorverarbeitung erzeugt zunächst aus dem Originalbild ein binäres Referenzbild auf dessen Basis die Bildsegmentierung durchgeführt wird. Zur Erzeugung des Referenzbild wird ein doppeltes Schwellwertverfahren angewendet, welches es erlaubt einen bestimmten Farbbereich zuzulassen oder diesen auszuschließen. Das Verfahren wird dabei auf den kompletten Farbraum eines Bildes angewendet um einen möglichst genauen Bereich definieren zu können. Es entstehen dadurch sechs Grenzwerte die einzeln eingestellt werden können.

²OpenCV: <http://opencv.org/>

³OpenGL: <http://www.opengl.org/>

⁴Unity 3D: <https://unity3d.com/>

Für ein Pixel an der Stelle i im RGB-Farbraum funktioniert das Schwellwertverfahren folgendermaßen:

```
dst(i) = lowerB; <= src(i); <=upperB; &&  
        lowerG; <= src(i); <=upperG; &&  
        lowerR; <= src(i); <=upperR;
```

Für jeden Farbwert R,G und B im Pixel wird geprüft ob sich dieser innerhalb der vorgegebenen Grenzwerte befinden. Nur wenn dies auf alle drei Farbwerte zutrifft, wird das Pixel des Referenzbildes gesetzt (weiß). Ansonsten bleibt dieses schwarz. Durch dieses Vorgehen finden bei der Erzeugung des Referenzbildes auch schon eine Vorsegmentierung statt, bei der einige irrelevante Objekte durch die eingestellten Schwellwerte ausgeblendet werden. Zur Unterdrückung von Bildrauschen wird auf das Binärbild anschließend noch ein morphologisches Opening angewendet [3, S. 179]. Diese Kombination von Erosion und Dilatation glättet die Ränder der im Binärbild gesetzten Bereiche und entfernt einzelne gesetzte Pixel, welche meist durch Bildrauschen entstanden sind.

Für die Segmentierung wurden zwei verschiedene Verfahren untersucht. Das erste ist die markerbasierte Watershed Transformation. Bei diesem Verfahren werden die Daten eines Bildes als Höheninformationen interpretiert. Markante Kanten bilden hierbei Berge und farblich einheitliche Bereiche Täler. Der Watershed Algorithmus [1, S. 295f] flutet anschließend diese Täler, bis nur noch die Bergkämme erhalten bleiben, welche die Täler umgeben. Diese definieren als sogenannte Wasserscheiden die Grenzen zwischen verschiedenen Regionen im Bild. Da niedrigere Bergkämme teilweise auch überflutet werden, werden mehrere ähnliche Regionen, zu größeren Regionen zusammengefasst. Dennoch entsteht bei der Watershed Transformation oftmals eine Übersegmentierung des Bildes. Das bedeutet es werden zu viele kleine Regionen ermittelt. Um dies zu vermeiden, kann im Vorfeld definiert werden, welche Bereiche im Bild zusammengehören. Dies

wird mit sogenannten Markern gemacht, die dem Watershed Algorithmus Anhaltspunkte liefern, wie die Gebiete geflutet werden sollen. Das im Vorfeld erzeugte Binärbild stellt diese Marker zur Verfügung. Die Watershed Transformation findet also auf dem Originalbild statt und verwendet dabei Informationen aus dem erzeugten Binärbild um die Segmentierung zu optimieren. Das zweite untersuchte Verfahren ist eine kantenbasierte Segmentierung auf dem Binärbild mittels des Canny Kantendetektors. Diese von Canny im Jahr 1986 vorgestellte Methode [1, S. 151] versucht aus möglichen Kanten Konturen zu bilden. Es werden dabei alle in Frage kommenden Pixel auf einen unteren und einen oberen Grenzwert hin überprüft. Sollte ein Pixel über dem oberen Grenzwert liegen, so wird es als Pixel einer Kante akzeptiert, liegt es unter dem unteren Grenzwert wird es abgelehnt. Wenn das Pixel zwischen oberem und unterem Grenzwert liegt, wird es akzeptiert sofern es mit einem bereits akzeptierten Pixel verbunden ist [1, S. 152]. Durch dieses Vorgehen entstehen Konturen, welche die einzelnen Objekte im Bild umschließen. Bei Versuchen in der Anwendung war es jedoch oft nicht möglich geschlossene Konturen zu bilden und die einzelnen Objekt zuverlässig zu trennen. Die Resultate konnten zwar durch eine weitere Anwendung von morphologischen Operationen auf das Kantenbild verbessert werden, jedoch war eine Segmentierung mittels Canny Kantendetektor dennoch oft zu instabil. Diese Instabilität führte dazu, dass in der virtuellen Umgebung flackernde Hände dargestellt wurden, da diese manchmal als Objekt erkannt wurden und manchmal nicht. Die Segmentierung mittels der Canny Methode spielte daher bei der Entwicklung der Anwendung eine nachrangige Rolle.

Aus dem segmentierten Bild werden neben der bereits bekannten Objektkontur weitere Merkmale extrahiert. Dadurch wird es ermöglicht irrelevante Objekte im nächsten Verarbeitungsschritt herauszufiltern. Abbildung 3 zeigt die verschiedenen Merkmale, welche aus dem Binärbild extrahiert wurden.

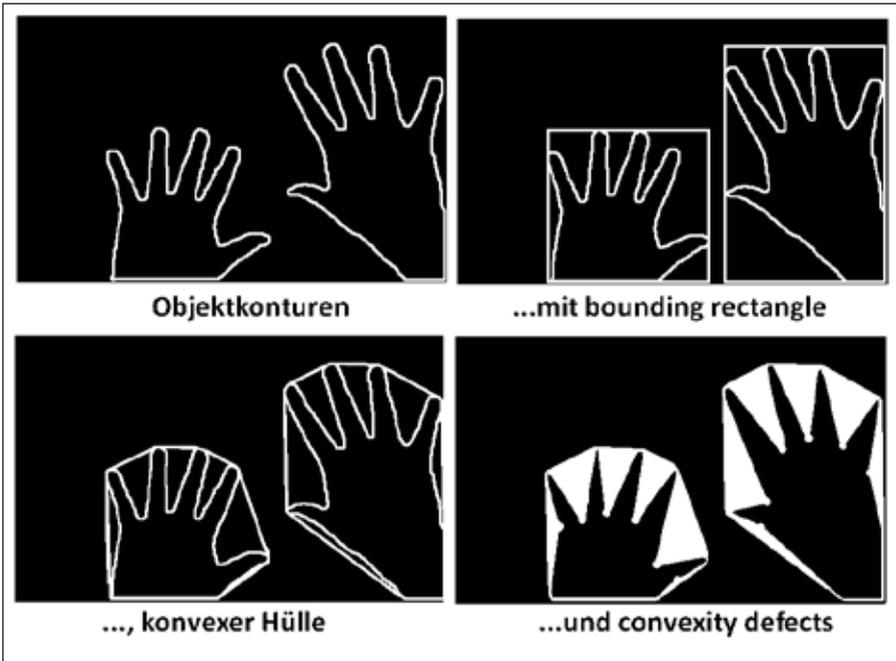


Abbildung 3: Extraktion von Merkmalen erkannter Objekte

Das erste Merkmal ist das bounding rectangle. Es stellt das minimale, nicht gedrehte Rechteck dar, welches die Kontur eines Objekts umschließt. Des Weiteren wird eine konvexe Hülle um das Objekt gezeichnet, welche es ermöglicht die convexity defects (in Abbildung 3 als weiße Dreiecke dargestellt) zu bestimmen. Diese Defekte definieren Bereiche zwischen der Kontur eines Objekts und dessen konvexer Hülle, welche beim Erzeugen dieser Hülle nicht berücksichtigt wurden. Die Defekte werden durch drei Punkte beschrieben: Startpunkt, Endpunkt und entfernter Punkt, wobei letzterer den am weitesten von der konvexen Hülle entfernten Punkt darstellt. Bei der Bestimmung der convexity defects im Prototyp werden nur Defekte berücksichtigt, bei denen der entfernte Punkt einen zuvor festgelegten Mindestabstand zur konvexen Hülle aufweist. Neben den genannten Merkmalen wird zusätzlich die Fläche des Objekts bestimmt. Anhand dieser Merkmale werden Filter an-

gewendet, die irrelevante Objekte entfernen. Es werden nur Objekte zugelassen, deren Fläche sich innerhalb eines vorher definierten Bereichs befindet. Außerdem werden Objekte bezüglich der Anzahl der vorhandenen convexity defects aussortiert. Bei einer gespreizten Hand wie in Abbildung 3 werden beispielsweise sechs Defekte erwartet. Jeweils einer an den Außenseiten von Daumen und kleinem Finger und weitere vier zwischen den Fingern. Für diesen Fall könnte also ein Bereich für erlaubte Defekte von beispielsweise 6 ± 2 festgelegt werden (erwartete Defekte \pm Toleranz). Alle Objekte, mit einer anderen Anzahl an Defekten würden entfernt. Verbleibende Objekte werden einzeln aus dem Kamerabild extrahiert, wobei das bounding rectangle die Begrenzung des einzelnen Objekts darstellt. Die Objekte werden dann in einem neuen Bild, welches dem Format der Ausgabe entspricht, neu positioniert.

Die Neupositionierung ist notwendig, da das Kamerabild und das Ausgabebild nicht das exakt gleiche Format aufweisen. Die Objekte werden daher im Bezug zu ihrer Originalposition im Kamerabild und zu der Größe der Ausgabe an einer neuen Position im Ausgabebild platziert. Dadurch wird sichergestellt, dass eine Person, die beispielsweise ihre Hand an den Rand des Kamerasichtfeldes bewegt, diese auch in der virtuellen Umgebung am Rand des dortigen Sichtfeldes wahrnimmt.

4 Derzeitiger Stand

Der derzeitige Stand der entwickelten Anwendung erlaubt einer Person ihre Hände innerhalb einer virtuellen Umgebung so wahrzunehmen, wie sie es auch in einer realen Umgebung tun könnte. Das bedeutet, es werden nicht nur ihre Bewegungen auf einen Avatar übertragen, sondern sie sieht ihre Hände mit eigener Hautfarbe und sonstigen eventuellen Besonderheiten.

Da der Prototyp zurzeit noch nicht über eine zuverlässige Klassifizierung der extrahierten Objekte verfügt, sondern lediglich einige irrelevante Objekte anhand bestimmter Merkmale und gesetzter Regeln herausfiltert, ist die Anzahl von false-positives bei einem komplexen Hintergrund teilweise noch sehr hoch. Dies führt dazu, dass die Anwendung bisher nur vor einfachen Hintergründen, bei der sich die Hände der Person klar abheben, zuverlässig funktioniert. Ein weiteres Problem beim derzeitigen Stand ist, dass die extrahierten Objekte in ihrer Größe bisher lediglich von der Kameraaufnahme, genauer von ihrer Entfernung zur Kamera und dem Sichtwinkel der Kamera, abhängen und noch keinen Bezug zur Größe der anderen virtuellen Objekte innerhalb der virtuellen Umgebung haben. Dadurch werden die Objekte zwar im Bezug zur Bewegung einer Person korrekt skaliert, können aber gemäß den Untersuchungen von Creem-Regehr [2] zu einer falschen Größenabschätzung innerhalb der virtuellen Welt führen. Obwohl diesem Effekt durch einen möglichst gleichen Sichtwinkel zwischen Kamera und Ausgabegerät entgegengewirkt wird, muss zukünftig dafür Sorge getragen

werden, dass die extrahierten Objekte ähnlich der in Abschnitt 3 beschriebenen Neupositionierung auch zusätzlich im Bezug auf den Maßstab der virtuellen Welt skaliert werden.



Abbildung 4: Reale Hand in einer virtuellen Umgebung

Im bisherigen Prototyp werden auch noch keine 3D Positionsdaten zu den erkannten Objekten ermittelt. Dies führt dazu, dass die Objekte in der virtuellen Umgebung lediglich als Ebene integriert werden. Ein Eindruck von dreidimensionaler Tiefe entsteht daher bisher nur durch die Entfernung der Objekte (im Fall des Prototyps, der Hände) zu der Kamera und der damit verbunden Größenänderung. Zwar entsteht hierbei, vor allem bei stereoskopischen Aufnahmen, auch schon eine gewisse Tiefenwahrnehmung, allerdings ist die 3D Position eines Objektes darüber hinaus auch essentiell für eine mögliche Interaktion mit virtuellen Objekten.

Eine Untersuchung über die Auswirkung auf die Selbstwahrnehmung und damit verbunden auf die Wahrnehmung der virtuellen Umgebung konnte aufgrund der dargestellten Probleme mit der bisherigen Anwendung noch nicht durchgeführt werden.

5 Ausblick

Zum derzeitigen Stand der Entwicklung des Prototyps ist es einer Person nicht möglich mit der virtuellen Umgebung zu interagieren. Weiterhin können die Parameter für Segmentierung und Objekterkennung bisher nur manuell und nur für eine Art von Objekten fest-

gelegt werden. Um einen Vergleich mit anderen Verfahren zu ermöglichen ist geplant den Prototyp an mehreren Punkten zu überarbeiten. Zunächst soll der Hardwareaufbau so erweitert werden, dass eine Bestimmung der 3D Position extrahierter Objekte und damit eine Interaktion mit der virtuellen Umgebung möglich wird. Als Basis wird dafür ein stereoskopischer Aufbau der Hardware, wie von Steptoe [7] beschrieben, untersucht. Des Weiteren müssen die Objekterkennungsmethoden überarbeitet werden. Das bedeutet zum einen, dass weitere Merkmale für verschiedene Objekte bestimmt werden müssen. Zum anderen müssen die bisherigen Filter durch Verfahren zur Objektklassifizierung ersetzt werden. Der Hauptaugenmerk liegt hierbei auf einer Verbesserung der Erkennung von Objekten vor komplexen Hintergründen und auf der Erkennung mehrerer Objekte verschiedenen Typs (beispielsweise Hände, Beine, Torso). Es werden dafür verschiedene Verfahren zum überwachten und nicht überwachten maschinellen Lernen untersucht. Anschließend sollen Untersuchungen durchgeführt werden, um festzustellen, inwiefern Unterschiede bei der Selbstwahrnehmung bestehen, wenn ohne Avatar, mit einem animiertem Avatar oder mit dem hier vorgestellten Ansatz gearbeitet wird.

Literatur

[1] G. R. Bradski and A. Kaehler. *Learning OpenCV: Computer vision with the OpenCV Library*. O'Reilly Media, Inc., first edition edition, 2008.

- [2] S. Creem-Regehr, K. Myszkowski, B. J. Mohler, H. H. Bühlhoff, W. B. Thompson, and S. H. Creem-Regehr. A full-body avatar improves egocentric distance judgments in an immersive virtual environment. In *the 5th symposium*, page 194.
- [3] A. Erhardt. *Einführung in die digitale Bildverarbeitung: Grundlagen, Systeme und Anwendungen ; mit 35 Beispielen und 44 Aufgaben*. Studium. Vieweg + Teubner, Wiesbaden, 1. aufl edition, 2008.
- [4] J. J. Gibson. *Wahrnehmung und Umwelt: D. ökolog. Ansatz in d. visuellen Wahrnehmung*. U-&-S-Psychologie. Urban und Schwarzenberg, München and Wien and Baltimore, 1982.
- [5] J. J. LaViola. A discussion of cybersickness in virtual environments. *ACM SIGCHI Bulletin*, 32(1):47–56, 2000.
- [6] S. A. Linkenauger, M. Leyrer, H. H. Bühlhoff, B. J. Mohler, and J. Witt. Welcome to wonderland: The influence of the size and shape of a virtual hand on the perceived size and shape of virtual objects. *PLoS ONE*, 8(7):e68594, 2013.
- [7] William Steptoe. Camera requirements for a stereo rig for head-mounted video see-through ar for the oculus rift.

Analysis of Finger- and Palm-based interaction paradigms for Touch-Free Gesture-Based Control of Medical Devices with the Leap Motion Controller

Stanislas Mauser

Reutlingen University

stanislas_christophe_yves.mauser@
student.reutlingen-university.de

Abstract

There are several intra-operative use cases which require the surgeon to interact with medical devices. I used the Leap Motion Controller as input device for three use-cases: 2D-Interaction (e.g. advancing EPR data), selection of a value (e.g. room illumination brightness) and an application point and click scenario. I evaluated the Palm Mouse as the most suitable gesture solution to coordinate the mouse and advise to use the implementation using all fingers to perform a click. This small case study introduces the implementations and methods that result those recommendations.

Keywords

Gesture recognition, touch-less interaction, human machine interface, medical device control

CR-Categories

H.5.2 [Information Systems]: User Interfaces

Betreuer Hochschule: Prof. Dr. Oliver Burgert
Hochschule Reutlingen
oliver.burgert@reutlingen-
university.de

Informatics Inside 2014
Wissenschaftliche Vertiefungskonferenz
30. April 2014, Hochschule Reutlingen
Copyright 2014 Stanislas Mauser

1 Introduction

1.1 Goals

During a surgical intervention, a surgeon depends on the availability of patient information such as risk areas of tumor margins, further he has to control surgical devices like endoscope, x-ray, drill etc. Nowadays, this control is either performed by using dedicated sterile equipment like foot pedals or by directing a surgical assistant or nurse. This impedes the execution of the surgery, increases the risk of infection for the patient and is generally associated with a higher expenditure of time (see [1], [2]).

For a subset of the use cases described above, a contact free, gesture based controller, which can be used by the surgeon himself might be helpful. Such a touch-less approach should be reachable for the surgeon while operating and without needing special gloves or additional tools.

In [1], several surgeons were interviewed whether they prefer to instruct other persons to acquire information or access requested information themselves by using e.g. new gesture recognition based system technologies. Over 80% of the surveyed surgeons preferred to search for information themselves and not by instructing another person. Using this information and the fact that the availability of touch-based systems (like smartphones and tablets)

and pose tracking systems like Kinect pave the way towards additional intuitive interaction devices, such a solution is desperately needed.

1.2 State of the art

There exist multiple examples for contact free input devices: In [3], a sterile solution to navigate the planning system MeVis with a Wii Remote Controller is described. This allows gesture recognition, but the controller needs to be touched. Gesture recognition can be performed based on tracked surgical instruments. In [4], such an approach is used to detect surgical gestures performed by a Polaris® navigation system pointer. A Microsoft Kinect is used for contact free interaction with the InVesalius program. Here the surgeon may contact-free interact with the program with one hand, to visualize 2D and 3D images [5].

A commercially available solution is the Mi-Report developed by Fraunhofer HHI for Karl Storz which allows the presentation of patient data with contact free gestures control, specifically for use in the sterile medical field [6]. The camera system is called HHI Handtracker and is based on an infrared camera system, which allows a fast and robust detection and tracking of fingers and gestures. The cameras are mounted to the hand orthogonal to the ceiling. By using infrared-based cameras, the system is also less insensitive to unwanted light of other devices or the sun. Several fingers are recognized and processed in real time (50 Hz). The 3D coordinates are then passed to the application.

The available systems have several drawbacks: Either they require a dedicated input device for gesture recognition (navigation system, Wii controller), or the gestures recognition is relatively coarse. Furthermore, large gestures result in physical fatigue.

2 Methods & Materials

For this solution I used the new Leap Motion Controller: A small device, not bigger than two fingers, but with the capability to track

“ten fingers up to a precision of 1/100th of a millimeter” [7]. The tracking volume of the Leap Motion Controller is about 20 to 600 mm semicircular around the device. I used a development kit including a Leap Motion Controller revision 0.6.5., C# as programming language, and the Visual Studio 2013 IDE.

Development hardware included a Apple Mac Pro running two Intel Xeon Westmere CPUs, 24GB RAM, two ATI 5700 HD graphic cards and Windows 8 in a virtual machine. For evaluation and presentation, a Lenovo W520 running a Intel i7 CPU, 8GB RAM and a nVidia Quadro 2000M graphic card was used.

2.1 Contact-free, gesture-solution prototype for medical devices

The high precision of the Leap Motion offers much more possibilities than similar devices like the Microsoft Kinect. But basically, the problem of finding a gesture which is easy to learn and intuitively to use, is the same as with other devices. There are no commonly accepted gestures for 3D contact-free input devices, yet. Best practices gestures from touch-based systems, like Apple iPhone or similar Point-Gestures can be implemented with the Leap Motion, but must be seen as starting point for the development of truly intuitive 3D input gestures.

In the medical context, I identified different classes of interaction: binary interaction (e.g. a simulated mouse click), simple 2D interaction, like advancing to the next picture or patient record entry, entering a value (e.g. adjusting OR table height or control a light source), point and click for selection of objects, and complex 3D-interaction, e.g. rotating a volumetric patient reconstruction. For the first prototype, I implemented the “simple 2D interaction” and “entering a value” use cases, in a second step, a point and click scenario was realized.

2.2 *Contact-free approaches to control a mouse pointer*

There are no studies which evaluate best practices in contact-free mouse control, yet. Therefore, I developed a prototype which offers the possibility to switch between different implementations to control a mouse pointer using the Leap Motion Controller and to test multiple concepts to emit a left or right click event.

The current version of the prototype has two different implementations: The first uses one finger as mouse with two different methods to perform a left click event. The second concept controls the mouse by palm orientation. For the Palm-Mouse I implemented two different click methods.

2.2.1 Mouse navigation using one finger

This implementation uses one finger to map the mouse movement to the position the user points to. If the Leap Motion Controller API recognizes more than one finger, the foremost one is chosen as reference. The Leap Motion API represents a finger as directional vector. The position in xyz-space of the Leap Motion Coordinate System is mapped linearly to the screens xy-coordinates.

One of the implemented click approaches uses the so called Touch Zone provided by the Leap Motion SDK. This is a defined plane in front of the controller. The SDK can “report whether that pointable object is close-to or touching this imaginary surface” [8]. Whenever a touch event is detected, a mouse click is triggered.

The second click approach uses a simple tap-gesture, which is one of the four default gestures of the Leap SDK: While navigating the mouse with the frontmost finger, the thumb is used to perform a tap-gesture creating a click event.

For the study described in 1.3, I implemented a rule allowing to apply a click even though it is slightly next to a button, if this button is slipped over within the next 600ms.

One problem with this implementation is the precision of the Leap Motion controller. Because the precision of the controller is very high, every trembling of the finger is also detected, which may result in an unstable behavior of the mouse pointer.

2.2.2 Mouse navigation with a single finger and optional precision mode

Reason for implementing this mouse solution was the problem described in the section above. In order to work more precisely, I implemented a so called precision mode.

The general mouse movement using the frontmost finger is identical to the implementation described above. In addition, the mouse movement velocity is reduced to a hundredth of the actual movement by lifting the thumb additionally to the frontmost finger.

The options to perform a click event remain the same.

2.2.3 Palm-Mouse implementation

The navigation with the palm has been used in the Light Control application (see [9]) and is known from gaming applications using the Leap Motion. The tilt of the palm is used to navigate the mouse: The steeper the tilt, the faster the mouse movements. The mouse position does not change if the palm is fully orthogonal to the controller. With e.g. 5° variation from the orthogonal position, the mouse position changes are nearly zero.

This implementation eases the stabilization of the mouse position at a desired position, compared to the finger mouse implementation where the user needs more cognitive concentration to ensure the finger doesn't vary its position heavily. Moreover, all fingers can be used for a finger tap-gesture to perform a left click.

To perform mouse click events I implemented two approaches. One approach uses the simulated touch area from the SDK. If this click method is activated, it is important to interact with the hand in front of the controller. When the mouse pointer is over a desired object, a user fires a click event by moving the palm to the direction of the controller and staying meanwhile orthogonal to the controller.

The second approach uses the built-in tap-gesture of the SDK. With this method it is not important where the palm is, as long as it is in the range of the Leap Motion tracking area. With a tap gesture performed by one arbitrarily selected finger, a left click event is fired.

2.3 Application for a click-case study of the Leap-Mouse implementations

To evaluate the usability, performance and acceptance of each of the implementations, I performed a first case study.

For this study, I developed an application which can use all mouse implementations. I defined five different button sizes, each size is shown six times on the evaluations software GUI. To detect the possible minimal button size the study includes buttons of 10x10 pixel, 15x15 pixel, 30x30 pixel, 50x50 pixel and 80x80 pixel size. While testing the mouse interaction, successively one of the possible 30 buttons is shown to the user. After clicking on a button, another button in a different size appears on another spot of the GUI. The buttons are arranged randomly on the screen, but the same order and position is guaranteed for each implementation and test person. If the user fails to click a button for five times, the current button is skipped.

The application logs the following information:

- Test person ID
- Distance between the button and the mouse pointer at the time the button appears;

- Sum of time the mouse was over the button;
- Sum of failed clicks;
- Sum of time until a button was clicked or skipped;
- And the information whether this button was skipped.

This data is automatically exported at the end of a test run into an Excel file.

Every implementation test followed a same procedure: Before the study begins, I introduced all mouse implementations to the test persons and allowed unlimited time to try each implementation, until they felt ready to start the study. During this warm-up phase, all 30 buttons are visible and clickable. All five implementations described above are part of the study; their order is hard-coded but can be tested also in reversed order. I distinguished the sequence of the implementations for each test by giving each test person having an even personal ID the reversed order of the five implementations. The personal ID is equal to the number of test persons, in their temporal order.

2.4 2D Interaction

2D interaction is realized by performing a swipe gesture to the right or to the left, orthogonal over the Leap Motion Controller. To detect a swipe I implemented a two-staged process: At the beginning, I used the build in Leap SDK swipe detection. On top of that, I defined three threshold states. The first threshold starts the swipe process and defines whether it is a left or right swipe. The next two thresholds and the speed of the finger or hand ensures that the gesture was really intended. The gesture detected can be mapped to a key-press event (e.g. cursor left, cursor right). This allows control of applications which were not developed with contact-free interaction in mind. I implemented a mock-up of an electronic patient record viewer to show the feasibility of the approach.

Evaluation of the detection capabilities and error rate was performed by controlling the PowerPoint presentation of a 90 minutes

lecture on computer science (Informatik II) at Reutlingen University. The contact-free 2D interaction was the only input device for slide advancement. The long duration of 90 minutes and the real lecture situation ensured that the teacher was not concentrating on using the interaction device correctly but on the teaching situation. The professor had no training on the device and started after a 20 seconds test. Afterwards, a video recording of 60 minutes of the lecture was analyzed to verify how many unwanted gestures were performed and how many gestures failed.

2.5 Entering a value

Since there is no commonly accepted interaction scenario for entering a value, yet, I implemented a control interface which allows for mapping different types of gestures to a method which finally passes the calculated value to the target application.

The gestures for entering a value take the best practice gestures from touch devices into account. I used the built-in gestures of the Leap Motion and additional values like pitch, yawl or the y-position of the hand, which can be read out from the Leap Motion. Gestures like “drag and swipe” seem to be intuitive, but detection of start and stop position is difficult. Along with this, the problem of how the given minimum, maximum and current value shall be represented occurs.

2.5.1 Counting fingers

From a technical and logical viewpoint, this is the simplest gesture. Analog to the sign language, the surgeon can modify the value by showing up to five fingers orthogonal over the leap motion Controller. This can be mapped to the values 0-5 or to a percentage of the value range. E.g. for setting of 20% of the maximum value, a surgeon shows only one finger, etc. The problem of this solution is the limitation on only five levels. The termination of the gesture must be defined properly to prevent misinterpretations: After the desired value has been set and the hand is removed from the

detection area, the fingers successively disappear and the value would decrease to one or zero.

2.5.2 Mapping the palm orientation

A rolling gesture of the palm, analogue to an airplane’s wing that rolls down to the right or to the left for navigation, was implemented as an alternative. The vertical palm position, relates to 50% of the maximum value. By rolling the palm to the right, the value increases and when rolling to the left, the value decreases.

Like in the counting fingers scenario, unwanted values sometimes occurred when removing the hand out of the tracking area. This effect happened much less frequently than in the number of fingers gesture, but often enough that it hinders productivity. Therefore, the number of detected fingers on the hand that performs the roll gesture was added as an additional condition. For example, if two fingers are set as the restriction modification, the values will only change when at least two fingers are recognized by the controller. So if the surgeon has set the desired value, he can simply make a fist and remove the hand without concerns out of the detection area.

2.5.3 Mapping the vertical hand position

The third gesture approach is based on the vertical movement of the palm position. In this scenario, the distance between the location of the controller and the measured center of the palm will modify the value. In the prototype, 250 sampling stages were implemented, starting from about 15 cm above the controller and ending at about 40 cm. In the graphical prototype this gesture is mapped to a vertical slider.

In most cases the values rest untouched, when the hand is removed slowly to the side out of the detection area. But in few cases, errors still occurred and moreover, for the use case during the OR course, it can’t be a requirement to remove the hand slowly. To compensate

this, the technique of restriction by a certain number of fingers, as described above, was introduced.

2.5.4 Lock and unlock gesture

In analogy to the Mi-Report [6], a lock-unlock gesture was implemented for this gesture concept. To reduce the possibility of unintended gesture input during the intervention, a defined gesture for locking and unlocking the detection of gestures was implemented. For unlocking the gesture detection, the surgeon has to hold his palm orthogonal over the controller with all fingers extended and as straight as possible for a defined duration. In the prototype these conditions can be adjusted at run time, since it's not entirely clear how precise the check of the gesture must be. Adjustable variables can be for example the length of time, the gesture has to be executed without interruption, the minimum number of fingers it has to recognize, and the maximum pitch and roll valued of the palm.

This gesture toggles the locking and unlocking condition, so that the same gesture can be used for both. But additionally the gesture detection is automatically locked after a defined amount of time, when no changes to the light value occur. The current default value in the prototype is three seconds.

3 Results

3.1 2D Interaction study results

During 60 minutes, 99 gestures were performed: 78 intentional gestures and 21 unwanted gestures. Unwanted gestures resulted from arm movement, putting down chalk, etc. 37 of the intentional gestures were performed to correct unwanted gestures. 58 of the 78 intentional gestures were recognized correctly and triggered a slide change. Thus, ~25% of the intended gestures were not fully recognized because they were not performed in the working space or the gesture was not performed properly.

In an interview, the performing teacher stated that the interaction was simple and fun, but the false positive rate was too high for routine use.

3.2 Click-Prototype study results

Following charts describe the results of the case study for the mouse click prototype. Those results are representing the findings of the study, which has been fulfilled and analyzed with 12 participants (6 female, 6 male, average age is 23.5).

Figure 1 shows the relation between the time until a button was pressed compared to the different button sizes. Recognizable is that with an increasing button size the click appeared earlier. The slowest implementation is the line (C), the fastest is the line (B).

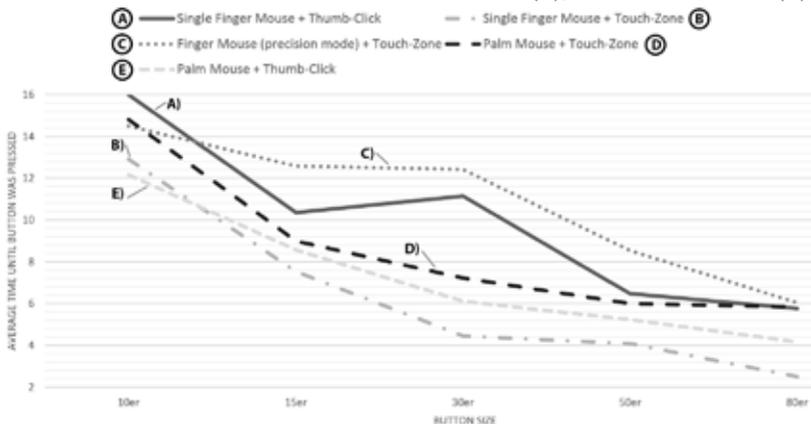


Figure 1: Time until button was pressed

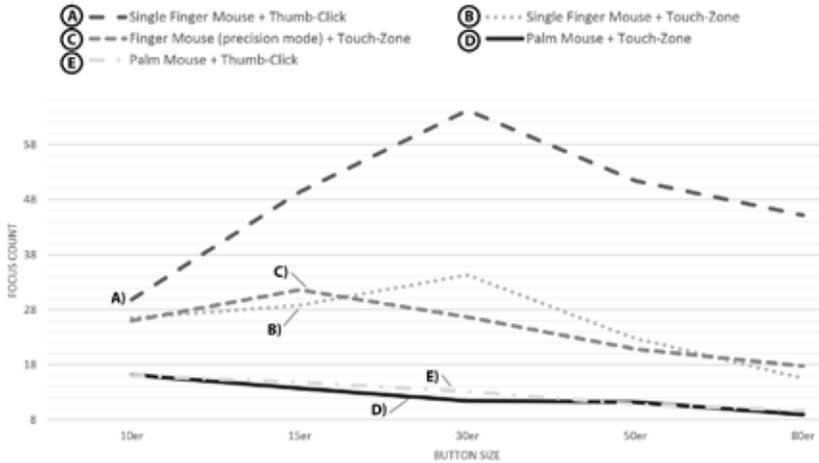


Figure 2: Relation of lost focus on a button to button size

This figure 2 above shows the relation between the sum of focus lost and the different button sizes. Focus lost represents the amount how often the mouse was over a button but lost its focus again before the button was finally clicked. The figure also displays, that there are no significant differences comparing both Palm Mouse Implementations. Generally it is recognizable that the two Palm Mouse implementations achieve the best overall rate of minimum lost focuses compared to all other implementations.

Figure 3 illustrates the relation between the amount of failed clicks and the button size. Failed clicks are mouse clicks beside a button and not direct on a button.

The implementation (E) shows a significant better result to all the other four implementations. As well it shows that the average number of failed clicks of the Palm Mouse Implementation received already with the 15x15 pixel button size a constantly low value of less than five fail clicks per person.

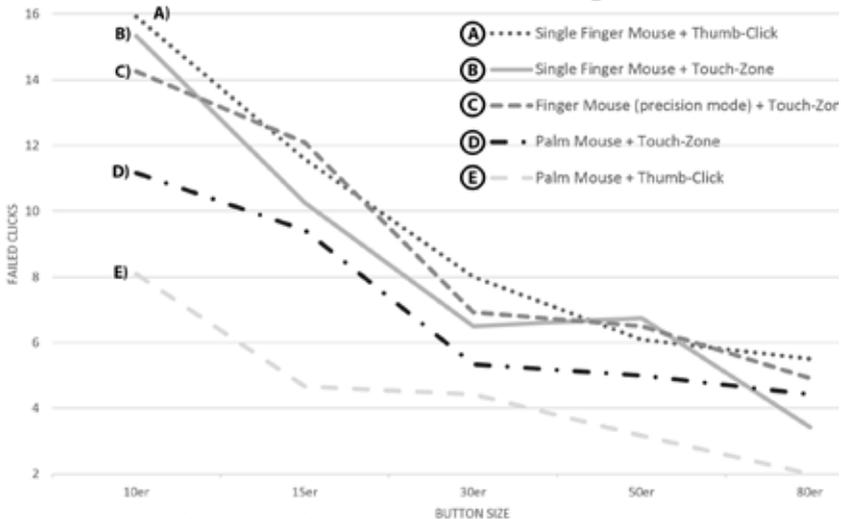


Figure 3: Relation failed clicks beside a button to button size

This graph shows that the three finger mouse implementations had similar results.

Figure 4 visualizes the amount how many buttons per person were skipped. It is consi-

The line (A) shows the best results over all button sizes.

Figure 5 outlines the usability of the button size comparing it for all implementations. The usability is rated through the relation of

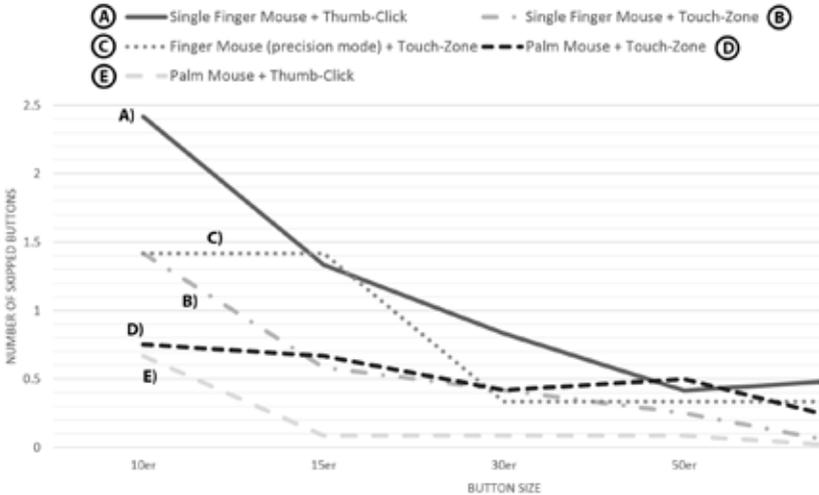


Figure 4: Relation Skipped Buttons to button size

derable to mention the extreme high value of skipped buttons using the Single Finger Mouse with Thumb Click Implementation for the smallest button size. This is due to a slight finger movement when the thumb is moved to trigger a click.

the following data: the amount of skipped buttons, the amount of failed clicks and the average click time per person with an additional priority value to scale the importance of an argument. The faster a person clicked an icon, the better are the results. Therefore this graph shows the best results within the lowest bar.

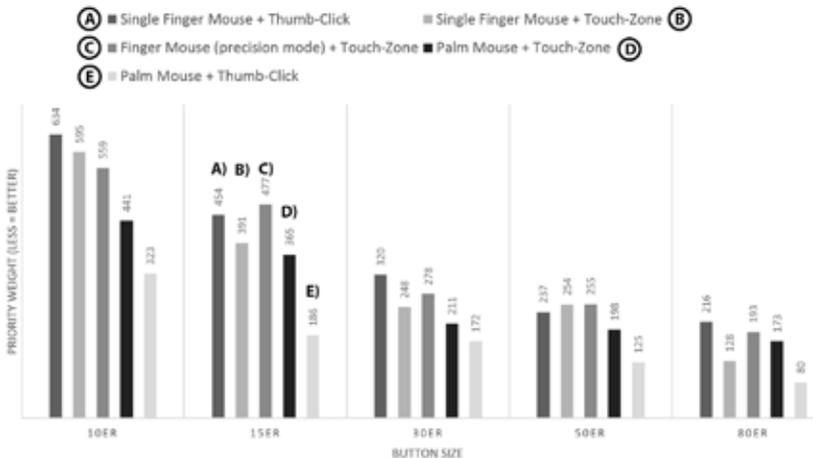


Figure 5: Comparison of usability of all implementations per button size

Goal is to evaluate the test person's favorite button size and implementation. Figure 5 shows that the 30x30 up to 80x80 button sizes have are rated equally. Secondly, I can identify that the line (E) archived better results to all other implementations. Mostly it is twice as good compared to the line (A).

3.3 MDeviceControl gesture prototype

In the prototype, the gestures approaches were all mapped to horizontal or vertical sliders to show potential testers the effect of their gesture movements. Both gestures, the palm roll gesture and the vertical movement gestures are implemented and visualized in the prototype with and without the additional condition of *n* fingers, for restriction of unwanted value modification during the gesture process.

In the prototype, as can be seen in figure 6, the lock- and unlock gesture conditions can be adjusted in the 3rd box. In the second box the detected gesture is visualized in values. These both boxes are just for the prototype and not for the productive working solution

implemented. On the bottom left, the pitch and roll state of the hand is visualized by arrows, which is more intuitive as plain numbers.

First tests with untrained subjects showed that with four or more fingers for three seconds and a maximum pitch/roll-deviation of 15° it is quite sure that this gesture is not unwanted performed during the OR, but can still be carried out quickly and easily.

4 Conclusion & Discussion

This section points out the main consolidated findings I achieved based on this work. The point-and-click study clearly favors the palm-based control method with thumb click over the finger based methods. Further studies which are having a setting closer to the intended clinical use cases can be designed based on this finding and may lead to even better interaction methods. Those studies must be performed with the intended end users of such systems, such as surgeons, interventional radiologists or other clinical staff.

Figure 2 shows that both Palm Mouse Implementations reached the best results in not losing the focus regardless of the button size.

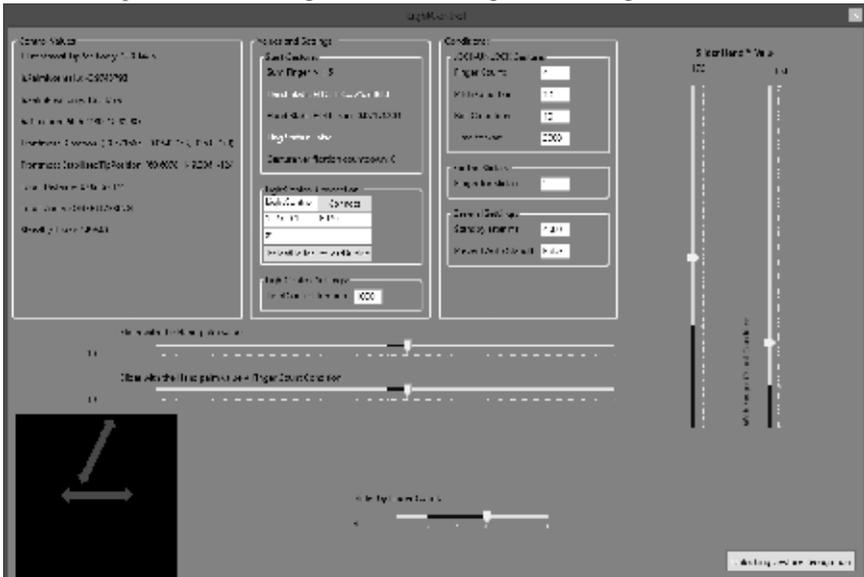


Figure 6: Screenshot of MDeviceControl

Moreover, the Palm Mouse with Thumb-Click got the best results. The Single Finger Mouse with Touch-Zone was the implementation, that allowed to press a button the quickest, the Palm Mouse Implementation reached the second place (see figure 1). Both figures show that with increasingly button size the results are getting proportionally better.

An important measure to evaluate the usability of a contact-free mouse control is the comparison of failed clicks and the button size. The Palm Mouse Implementation received the best score, and for the largest button size of 80x80 all test persons had no failed clicks at all (see figure 3). This is an important aspect for surgeons, who can't spend much cognitive effort for finding the right button.

This interaction paradigm is also the one with the fewest number of skipped buttons (see figure 4). Furthermore, it is constantly low from a button size of 15x15 on whilst other options reach a constant level at a size of 30x30 or even at 50x50.

Figure 5 shows that it gives the best results for all button sizes. The possible minimal button size can be 15x15, for the other implementations it generally starts from a size of 30x30.

Therefore, I expect the Palm Mouse with Thumb-Click interaction to be the most precise and least stressful interaction method for realizing a point-and-click scenario.

Additionally to this technical analysis the results of the questionnaires showed same results: 75% of all test persons prefer the Palm Mouse Implementation. 50% of all test persons favour a 30x30 button and the other 50% decided that a 50x50 button was finally the preferred one size. Besides the analysis results it is important to assess the Leap Motion Controller. This is a stable 3D gesture input device which has potential applications in intra-operative situations. The use-cases have to be carefully selected and the best gestures for each application have to be detected and

implemented. The developed MDeviceControl prototype is a valuable tool for user testing and rapid development.

Based on the studies and tests performed, it is clear, that a lock- and unlock gesture is mandatory, especially if precise and intuitive tool control is crucial.

Still not evaluated are the optimal light conditions for the Leap Motion Controller, since unwanted light and the impairment of its emitted IR light could result in reduced recognition rate. I did not detect interferences with surgical navigation systems using IR light, yet. The next step will be to simulate lighting conditions found in operating theaters with or without daylight access and OR illumination.

5 References

- [1] C. Dressler, T. Neumuth, M. Fischer, O. Abri, G. Strauss: Intraoperative Bedienung einer elektronischen Patientenakte durch den Operateur, In: HNO, vol. 59 issue 9, Universität Leipzig, September 2011, DOI: 10.1007/s00106-011-2331-y, 900-907.
- [2] P. Chojcecki, U. Leiner: Berührungslöse Gestik-Interaktion im Operationssaal, i-com, Vol. 8, Issue 1, Mensch-Computer-Interaktion im Operationssaal, DOI 10.1524/icom.2009.0003, 13-18.
- [3] F. Ritter, C. Hansen, K. Wilkens, A. Köhn, H. Peitgen: Benutzungsschnittstellen für den direkten Zugriff auf 3D-Planungsdaten im OP, i-com, vol. 8 issue 1, Mensch-Computer-Interaktion im Operationssaal, DOI: 10.1524/icom.2009.0005, 24-31.
- [4] S. Franke: Integration intraoperativer Patientendaten in ein TIMMS Patientenmodell am Beispiel der Lokalisation des Sulcus Centralis, Diploma Thesis, Universität Leipzig, 2010.
- [5] G. Ruppert, L. Reis, P. Amorim, T. Moraes, J. Silva: Touchless gesture user interface for interactive image visualiza-

- tion in urological surgery, World Journal of Urology, Ausgabe 5, 2012, DOI: 10.1007/s00345-012-0879-0, 687–691.
- [6] S. Schrader, U. Leiner: Neuartiges System zur Visualisierung von Patientendaten mit berührungsloser Handgestik-Steuerung, In: Workshop-Proceedings der Tagung Mensch & Computer, Berlin, 2009, ISBN: 978-3-8325-2181-3, 302-304.
- [7] Leap Motion official website at <https://www.leapmotion.com/> (last access: 14.06.2013).
- [8] Leap Motion developer website: Touch Emulation; https://developer.leapmotion.com/documentation/java/devguide/Leap_Touch_Emulation.html (last access: 14.06.2013).
- [9] S. Mauser, O. Burgert: Touch-Free, Gesture-Based Control of Medical Devices and Software Based on the Leap Motion Controller, In: Medicine Meets Virtual Reality 21; IOS Press, 2014, DOI: 10.3233/978-61499-375-9-265, 265-270.

Selektion von Szenarien zur Optimierung von Simulationen im präventiven Krisenmanagement*

René Mangold
Reutlingen University
Rene.Mangold@student.
Reutlingen-University.DE

Abstract

Im präventiven Krisenmanagement geht es um die frühzeitige Erkennung von möglichen, unvorhersehbaren Ereignissen. Hierzu zählen beispielhaft Busunfälle, einstürzende Gebäude und ähnliche Großschadensereignisse. Krisen treten meist unerwartet auf und neigen oftmals aufgrund der knapp bemessenen Handlungszeit zu Fehlentscheidungen [3]. Um dies zu verhindern, dient das präventive Krisenmanagement dazu, sämtliche auftretende Ereignisse mittels einer Simulation zuvor durchzuspielen, um im Falle einer realen Krise die notwendigen Schritte bestmöglich einzuleiten. Um Simulationen für das Krisenmanagement zu präzisieren und die Ergebnisse effektiv und vereinfacht zu veranschaulichen, ist es notwendig, eine Vorauswahl an vorhandenen Szenarien für Vergleiche herauszusuchen zu können. Diese Arbeit entstand im Rahmen des FP-7 EU-Projekts *CRISMA (Crisis Management)* [1] und dient zur Evaluation eines Konzepts zur Vorauswahl geeigneter Szenarien, welche in früheren Simulationen entstanden.

*

Betreuer Hochschule: Prof. Dr. Gabriela Tullius
Hochschule Reutlingen
Gabriela.Tullius@Reutlingen-
University.de
Betreuer Firma: Johannes Sautter
Fraunhofer IAO Stuttgart
Johannes.Sautter@iao.
fraunhofer.de

Informatics Inside 2014
Wissenschaftliche Vertiefungskonferenz
30. April 2014, Hochschule Reutlingen
Copyright 2014 René Mangold

Schlüsselwörter

crisis management, simulation, modeling, filtering

CR-Kategorien

I.6.6 [Simulation Output Analysis]; I.6.7 [Simulation Support Systems]; Environments

1 Einleitung

Simulationsanwendungen wie *CRISMA* dienen im präventiven Krisenmanagement dazu, mit Hilfe geeigneter Modelle und Szenarien die Benutzer bei ihrer Entscheidungsfindung zu unterstützen. Simulationen können dadurch optimiert werden, indem eine Vorauswahl von geeigneten Szenarien getroffen wird. Hierbei wird das gewählte Szenario durch den Benutzer überarbeitet, sodass die Ergebnisse der Simulation verbessert werden. Das Problem besteht hier allerdings darin, dass die Benutzer geeignete Szenarien aus einer großen Menge auszuwählen haben. Durch die stetige Anpassungen der Simulationen steigt ebenfalls die Anzahl der verfügbaren Szenarien, was den Auswahlprozess erschwert.

Um den Benutzer nun bei der Auswahl der Szenarien zu unterstützen, wurde ein Konzept aufgestellt, welches die Filtrierung der Eingabeparameter für die Simulation nach bestimmten Indikatoren vorsieht.

Eine Simulation in *CRISMA* liegt einem Modell zugrunde, welches vom Benutzer erstellt und manipuliert werden kann (siehe hierzu [4]). In *CRISMA* existieren verschiedene Modelle. Zum einen bestehen Modelle, die Gebäude beinhalten, die das Abbild der realen

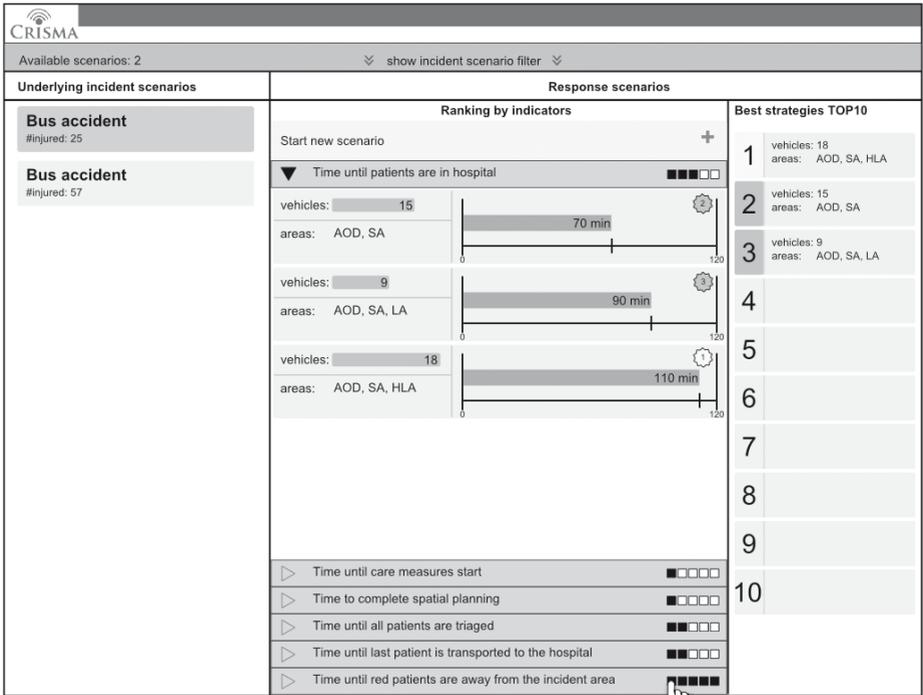


Abbildung 1: Simulations-Anwendung CRISMA

Welt darstellen. Dazu zählen Versorgungsgebäude wie Krankenhäuser, Rettungswachen, Feuerwachen und sonstige wichtige Stützpunkte. Daneben gibt es für die Simulation für Szenarien die Simulationsmodelle, welche auf den *Incident Szenarios* aufbauen. Unter Incident Szenarios werden Krisenereignisse wie Busunglücke, Erdbeben und ähnliches zusammengefasst.

Eine Simulation nutzt das Modell, welches die Versorgungsgebäude beinhaltet, um bei der Berechnung von Anfahrtswegen und dergleichen, Anhaltspunkte für die Dauer der Einsätze zu erhalten. Da sich die Versorgung in ländlichen Gebieten von denen in Ballungszentren stark unterscheidet, spielen geografische Eigenschaften ebenfalls eine wichtige Rolle innerhalb des Modells. Die Anfahrtszeiten von Rettungskräften innerhalb einer Stadt sind in der Regel sehr viel geringer als in einer länd-

lichen Umgebung.

Diese Arbeit konzentriert sich auf einen Teilspekt des Projektes *CRISMA*: Die Ressourcen Planung. In der Ressourcen Planung werden die oben genannten Modelle durch die taktische Einsatzplanung ergänzt, sodass bei Krisen mit Großschäden die Rettungskräfte optimal eingeteilt werden können. Mittels der Ressourcen Planung kann beispielhaft die Anzahl der benötigten Rettungswagen, THWs (Technisches Hilfswerk) und Feuerwehren für das entsprechende Szenario optimal im Voraus kalkuliert werden. Basierend auf den Modellen, können die verschiedenen Szenarien anhand der Ein- und Ausgabeparameter weitestgehend verbessert werden, sodass eine möglichst optimale Anzahl an Ressourcen innerhalb des gewählten Szenarios verfügbar ist. So kann im Falle eines realen Busunglücks auf die bestehenden Szenarien und Simulation zu-

rückgegriffen werden, um die notwendige Anzahl der Rettungskräfte an den Unfallort zu senden, sowie die Dauer, die zum Abtransport der Verletzten ins Krankenhaus benötigt wird, miteinzukalkulieren.

Mit Hilfe unterschiedlicher Filter-Kriterien soll in dieser Arbeit ein Konzept erarbeitet werden, welches dem Benutzer ermöglicht, bereits durchgeführte Simulationen sowohl anhand ihrer Eingabe- als auch ihrer Ausgabeparameter zu filtern, um die Ergebnisse der Simulation zu verbessern. In Abschnitt 2 wird zunächst der grundlegende Aufbau von Modellierungs- und Simulationsanwendungen dargestellt. Abschnitt 3 beleuchtet hingegen das Konzept der Szenarienauswahl innerhalb der Simulationen.

2 Modellierung und Simulation

Durch ein Modell kann ein komplexer Sachverhalt wie ein Unfallszenario auf die wichtigsten Eigenschaften abstrahiert werden, so dass eine vereinfachte Darstellung der Realität entsteht. Anhand des Modells können Vorhersagen über künftige Ereignisse und deren Einflüsse auf das Modell getroffen werden. Ein Modell ist somit ein für Computer darstellbares Abbild der realen Welt.

Mittels einer Simulation werden Sachverhalte durch Computer errechnet, welche zumeist aus finanziellen Gründen in der realen Welt nicht umsetzbar sind. In Bezug auf das präventive Krisenmanagement können somit Katastrophen virtuell erstellt und deren Verlauf anhand des verwendeten Modells betrachtet werden.

Birta et al. [2] beschreiben, dass es ein sehr breites Spektrum an Anwendungsgebieten gibt, in welchen Modelle und Simulationen ihren Einsatz finden. Unter anderem zählen hierzu die folgenden Punkte:

1. "Comparison of control policy options
2. Education and training
3. Engineering design
4. Evaluation of decision or action alternatives

5. Evaluation of strategies for transformation or change
6. Forecasting
7. Performance evaluation
8. Prototyping and concept evaluation
9. Risk/safety assessment
10. Sensitivity analysis
11. Support for acquisition/procurement decisions
12. Uncertainty reduction in decision making"([2, S.4-5])

Im Rahmen dieser Arbeit wird jedoch nur der Bedarf an den Punkten 4 und 5 gedeckt. Das Ziel der Arbeit besteht im Erstellen eines Konzepts, welches zur Unterstützung der Krisenmanager dient. Mit Hilfe der Punkte 4 und 5 wurden innerhalb des Konzepts Hilfen für die Entscheidungsfindungen und Handlungsalternativen für den Benutzer evaluiert. Zusätzlich kann durch das Konzept, wie in Punkt 5 beschrieben, dem Benutzer bei der Vorauswahl von Szenarien mittels Filteroptionen geholfen werden, welches die Optimierung der Simulationen mit sich bringt. Wie in Abschnitt 1 bereits erläutert, geht es im EU-Projekt CRISMA um die Simulation von Szenarien im präventiven Krisenmanagement. In dieser Arbeit wird ein Massenansturm von Verletzten (kurz: MANV) betrachtet. Dabei handelt es sich um ein fiktives Busunglück, welches zu 25 Verletzten führt. Die Simulation dieses Vorfalls wird dazu eingesetzt, um Krisenmanagern bei ihren Entscheidungen behilflich zu sein, so dass die Planung der Rettungskräfte effizienter und effektiver greifen kann. Hierzu ist es notwendig, eine Vielzahl an Simulationen durchzuführen, um die beste Lösung in diesem Notfall herauszukristallisieren.

Um die Benutzer von *CRISMA* eine übersichtliche Darstellung auf die Szenarien und die dazugehörigen Indikatoren zu gewährleisten, dient ein Filter am oberen Rand (Abb. 2) der Anwendung zur Eingrenzung der verfügbaren Szenarien. Mittels dieses Filters, der

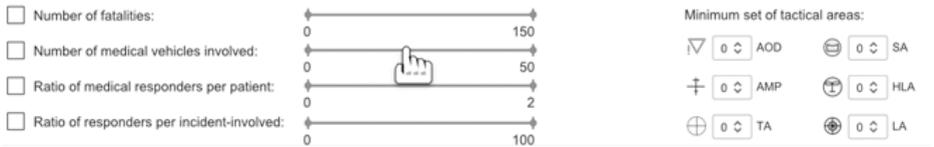


Abbildung 2: Filter der Anwendung

aus *Double Slider* und *Two Spinner* Elementen besteht, kann der Benutzer durch die Auswahl bestimmter Indikatoren die verfügbaren Szenarien eingrenzen. So kann beispielhaft die minimale Anzahl taktischer Bereiche wie Patientenablageflächen oder Hubschrauberlandeplätze definiert werden, sodass in der linken Spalte der Simulationsanwendungen, nur noch diejenigen Szenarien erscheinen, die dieses Kriterium erfüllen. Die Patientenablage definiert hierbei eine Fläche, die nach Sichtung der einzelnen Verletzten zur Ablage deren dient. So werden die Patienten nach der Sichtung und der Klassifizierung anhand ihres Verletzungsgrads direkt vom Unfallort an die Patientenablagestelle gebracht, um von dort aus die notwendigen Behandlungsmaßnahmen einzuleiten.

3 Szenarien Selektion

Beim EU-Projekt *CRISMA* bestehen unterschiedliche Szenarien, die Katastrophenfälle repräsentieren. Der Benutzer muss anfänglich mittels der Filterfunktion am oberen Rand der Simulationsanwendung (siehe Abb.1, Abb.2) verschiedene Parametereinstellungen vornehmen, welche sich auf die Liste der Szenarien auswirken. Entsprechend der getroffenen Vorauswahl kann der Benutzer das Szenario auswählen, welches es zu optimieren gilt.

Zu Beginn dieser Arbeit wurde für die Auswahl der Szenarien das Entwurfsmuster *Tree-map* evaluiert (vgl. [7, S. 336]). Die *Tree-map* stellt eine Visualisierungstechnik zur Darstellung von hierarchisch, strukturierten Daten dar ([6, S.68]). Der Nutzen von *Tree-maps* besteht vorwiegend in der Visualisierung und Verwaltung von großen Informationsmengen ([6, S.69]). Jedoch stellte sich hierbei heraus, dass die *Tree-map* aufgrund ihrer Darstellungsform für die Visualisierung von sowohl

Eingabe- als auch Ausgabeparametern ungeeignet ist. Zudem sind *Treemaps* aufgrund ihrer multidimensionalen Datenansicht oftmals zu komplex, so dass eine schnelle und einfache Übersicht vorhandener Szenarien in diesem Fall nicht möglich gewesen wäre.

Aufgrund dieser Erkenntnisse wurden alle verfügbaren Szenarien in der linken Spalte der Startseite (siehe Abbildung 1) in Form von *Tabs* dargestellt. Durch die zuvor stattgefundenen Filterung werden auf der linken Spalte nur diejenigen Szenarien mittels *Tabs* dargestellt, die den Filterdaten entsprechen. Für den Benutzer ist zur selben Zeit immer nur ein Szenario sichtbar, was durch das Entwurfsmuster der *Module Tabs* realisiert wurde ([7, S. 154-157]). Die bereits durchgeführten Simulationen des gewählten Szenarios werden im mittleren Teil der Anwendung dargestellt. Tidwell [7] schreibt, dass die Funktionsweise der *Module Tabs* niemanden verwirren wird. Dies liegt vor allem daran, dass dieses Entwurfsmuster in vielen Desktop Anwendungen und auch auf Internetseiten häufig zum Einsatz kommt ([7, S. 156]). Über Filter können entsprechende Indikatoren geändert werden, sodass sich die Liste der Szenarien den Filterkriterien anpasst. Dementsprechend reduziert sich die Anzahl der verfügbaren Szenarien auf der linken Spalte, sodass der Benutzer lediglich die Auswahl an Szenarien erhält, die den zuvor gefilterten Indikatoren entsprechen. Mit Hilfe des Filters lassen sich beispielhaft die Anzahl der benötigten Rettungskräfte oder taktischen Einsatzbereichen wie Hubschrauberlandeplätzen eingrenzen (vgl. Abbildung 2).

Als Filterelement für die Szenarien wurde ein *double slider* eingesetzt (vgl. Abbildung 3). Dieser hat die Eigenschaft, dass ein Bereich

mittels minimalem und maximalem Wert angegeben werden kann, welcher die Auswahl eingrenzt. Nach einem Expertenfeedback wurde jedoch klar, dass der *double slider* allein nicht ausreichen wird, um die Bedürfnisse des Benutzers zu befriedigen.



Abbildung 3: *double slider*-Element [7]

Zusätzlich zum vorhandenen Auswahlelement soll künftig das Entwurfsmuster der *two spinners* (siehe Abbildung 4) integriert werden.



Abbildung 4: *two spinners*-Element [7]

Das Resultat könnte wie in Abbildung 5 aussehen und vereint hierbei die guten Eigenschaften beider Einzelelemente ([7, S. 354]). Dies bietet den Vorteil, dass der Benutzer die Justierung der Bereiche auch über die Tastatur eingeben kann. Es bleibt ihm dennoch freigestellt, den Slider wie gewohnt über die Steuerung mit der Maus zu benutzen.



Abbildung 5: Neues Filterelement als Verbindung zwischen *double slider* und *two spinners*

Neben den Slider Elementen, die sich im aufklappbaren Filterbereich am oberen Rand von *CRISMA* befinden (Abb.1, 2), gibt es einen Bereich zum Auswählen von taktischen Bereichen, die auch als *Areale* bezeichnet werden. Im Kontext des präventiven Krisenmanagements werden als *Areale* Bereiche definiert, die beispielhaft zur Ablage von Patienten oder Hubschrauberlandeplätzen dienen. Das Krisenmanagement sieht unterschiedliche

geografische Bereiche vor, um die Übersicht im Falle einer Katastrophe zu gewährleisten. Um die Anzahl der verfügbaren Areale zu bestimmen, wurde in dieser Arbeit das Entwurfselement Spinner eingesetzt. Der Vorteil besteht bei den Spinnern in der Eingabemöglichkeit. Sowohl die Auswahl der Werte mittels Maus, als auch die direkte Eingabe über die Tastatur sind hier möglich.

4 Visualisierung von Indikatoren

Szenarien im präventiven Krisenmanagement beinhalten diverse Indikatoren, bei deren Visualisierung mit Hilfe bekannter Entwurfsmuster [7] vorgegangen wurde. Im erarbeiteten Konzept werden folgende Indikatoren berücksichtigt:

1. Dauer des Patientenabtransports ins Krankenhaus (Time until patients are in hospital)
2. Dauer bis die Behandlungsmaßnahmen eingeleitet werden (Time until care measures start)
3. Dauer der Raumplanung (Time to complete spatial planning)
4. Dauer zur Sichtung der Patienten (Time until all patients are triaged)
5. Dauer zum Abtransport des letzten Patienten (Time until last patient is transported to the hospital)
6. Dauer bis Schwerverletzte von der Unfallstelle entfernt sind (Time until red patients are away from the incident area)

Im Fall eines realen Krisenvorfalles dienen die zuvor festgelegten Indikatoren zur Berechnung des optimalen Szenarios. So kann anhand der Dauer, die es zum Abtransport der Patienten ins Krankenhaus benötigt, die Anzahl der notwendigen Einsatzkräfte berechnet werden. Bei einem Krisenvorfall werden zunächst alle Patienten am Unfallort gesichtet

und nach Kategorien eingeteilt. Die Kategorien entsprechen der Schwere der Verletzungen. So gibt es grüne, gelbe und rote Patienten, welche von leicht- bis schwerverletzt eingeteilt werden.

Um dem Benutzer einen Überblick über die vorhandenen Indikatoren zu ermöglichen, wurde das Entwurfsmuster der Akkordeon-elemente angewandt. Akkordeons haben sich vor allem im Bereich von Webseiten als beliebtes Interaktionselement etabliert ([7, S. 159–161]). In Bezug auf *CRISMA* bieten Akkordeon-elemente eine gute Möglichkeit, den Krisenmanagern zunächst einen allgemeinen Überblick über die vorhandenen Indikatoren zu gewährleisten, ohne Details anzuzeigen. Durch Anklicken des jeweiligen Akkordeon-elementes wird die Detailansicht sichtbar (siehe Abbildung 6). Das erste Element (1) des Akkordeons wird in Abbildung 6 angeklückt, sodass der Detailbereich (2) sichtbar wird. Bei der Umsetzung in *CRISMA* fiel die Entscheidung darauf, nur ein Akkordeon-element zur selben Zeit aufgeklappt zu haben. Dies ermöglicht die sequenzielle Abarbeitung der einzelnen Indikatoren, ohne sich von anderen Parametern ablenken zu lassen.

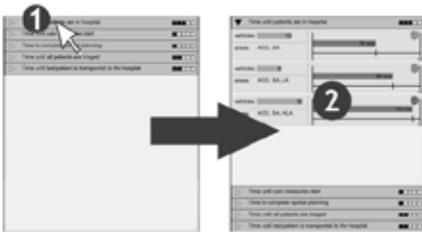


Abbildung 6: Akkordeon

Bei Akkordeon-elementen ist es wichtig, dem Benutzer eine optische Rückmeldung zu geben, welche Elemente aufklappbar sind. Bei *CRISMA*, wurde dies standardmäßig mit Pfeilen auf der linken Seite des Titelelements umgesetzt. Dabei weist ein Pfeil in horizontaler Richtung auf ein geschlossenes Akkordeon-element hin, wobei ein Pfeil mit der Spitze nach unten gerichtet, ein bereits geöffnetes Element darstellt. Über eine Gewichtung auf

der rechten Seite des Elements kann außerdem festgelegt werden, wie stark der jeweilige Indikator in das Ergebnis der Auswahl mit aufgenommen wird.

Innerhalb der Akkordeon-elemente können die Benutzer bereits simulierte Szenarien für das ausgewählte Szenario auswählen. Im Fall des Szenarios Busunfall wurden die Eingabeparameter *Anzahl der Fahrzeuge* sowie eine Liste der taktischen Bereiche genutzt. Die Anzahl der Fahrzeuge wurde hierbei mit einem horizontalen Balkendiagramm visualisiert. Dies ermöglicht eine schnelle Sichtung der Anzahl der Elemente im Vergleich zu den restlichen Szenarien. Die Auflistung der Bereiche ist als reine Aufzählung dieser visualisiert. Die Eingabeparameter sind farblich schlicht und eher zurückhaltend gestaltet, sodass der Benutzer sich auf das Wesentliche konzentrieren kann.

Die wichtigen Merkmale für einen Benutzer sind die Ausgabeparameter. Diese sind farblich von den Eingabeparametern abgehoben, sodass es dem Anwender leichter fällt sich darauf zu konzentrieren. Das horizontale Balkendiagramm auf der rechten Seite im Akkordeon-element für die Kennzahl zu *Time until patients are in hospital* zeigt die Dauer zwischen dem Busunfall und der Zeit, zu der alle Patienten sich im Krankenhaus befinden. Für den Benutzer heißt das, dass die kürzeste Zeit das beste Resultat aufweist. Dadurch ergibt sich eine Rangfolge innerhalb des Akkordeon-elementes für jede einzelne Kennzahl. Diese ist je nach aufgeklapptem Indikator auf- oder absteigend sortiert. Im Fall des oben genannten Indikators ist dies eine aufsteigende Sortierung. Hierbei ergibt sich für den Benutzer eine tabellenartige Struktur mit dem jeweils besten Ergebnis in der ersten Zeile. Mit dieser Kenntnis sieht der Benutzer somit auf einen Blick, dass für den gewählten Indikator das erste Element das beste Ergebnis darstellt. Da dieses Optimum sich nur auf den lokalen Bereich des aktiven Akkordeon-elementes mit gewählter Indikator bezieht, muss dem Benutzer zudem eine globale Entscheidungshilfe für das beste Ergebnis über alle Indikatoren hinweg aufgezeigt werden. Für diesen Zweck wurde auf der rechten Seite eine Top

10 Liste angelegt, welche die beste Strategie zu den entsprechend gewichteten Indikatoren liefert. Die hier aufgelisteten Elemente werden zudem farblich und gewertet im zuvor beschriebenen Akkordeon eingeblendet, sodass der Benutzer direkt erkennen kann, ob das lokale Optimum auch dem globalen entspricht. Um die Top 10 Liste anzupassen, muss der Anwender lediglich die Gewichtung der Indikatoren an seine Bedürfnisse anpassen.

Nachdem nun das optimale Ergebnis feststeht, kann der Benutzer mit der Optimierung der Simulation fortfahren. Dazu wählt er eines der Elemente aus und bekommt somit die Möglichkeit, mit dem Ausgangspunkt dieses Szenarios eine erneute Simulation zu parametrisieren.

5 Fazit

In dieser Arbeit konnte aufgezeigt werden, wie es dem Benutzer ermöglicht wird, eine treffende Auswahl eines Szenarios nach bestimmten Bedingungen zu treffen. Dennoch gibt es einige offene Fragen die noch zu klären sind. Beispielsweise gibt es einige Usability Verbesserungen, welche in den Mockups Optimierungsbedarf haben.

In Microsoft Office ab Version 2007 gibt es eine sogenannte *ribbon* ([5, S. 139]), welche als Menü verwendet wird. Ein solches *ribbon* ist laut Tidwell eine Ansammlung von mehreren Module Tabs ([7, S. 157]). Dieses Element wäre eine Verbesserung für die derzeitig dargestellte Filterfunktion. Über einen Klick auf dieses *ribbon* würde die Filterfunktion geöffnet werden, welche es dem Benutzer erlauben, einige Einstellungen für den Filter zu setzen. Die gewählten Optionen können dann mit einem Element (siehe Abbildung 7) einer Liste im oberen Bereich der Anwendung hinzugefügt werden, sodass der Benutzer jederzeit sehen kann, nach welchen Kriterien er die Vorauswahl getroffen hat.

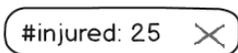


Abbildung 7: Neues Filterelement

Zudem steht eine Optimierung der Gewichtung der Indikatoren an, da die derzeitige Anzeige der Gewichtung zugleich die einzige Möglichkeit ist diese einzustellen. Da das Akkordeon Kopfelement keinerlei Interaktion zulässt, ist es notwendig, dass die Änderung der Gewichtung an anderer Stelle stattfinden kann, bspw. innerhalb des Akkordeonelements selbst oder gar auf der rechten Seite oberhalb der Top 10 Liste. Damit der Benutzer trotz dieser Veränderung über die Gewichtung der Indikatoren gewahrt ist, wird die momentane Interaktionsfläche als reine Anzeigefungieren. Dies ist wichtig, da der Benutzer sonst nicht gleich feststellen kann, nach welchen Kriterien er die globale Liste der Strategien gerade betrachtet.

Literatur

- [1] EU-project CRISMA: Online verfügbar: <http://www.crismaproject.eu/>.
- [2] L. G. Birta and G. Arbez. *Modelling and Simulation*. Springer London, London, 2013.
- [3] U. Kipp. Präventives krisenmanagement: Handlungsfähigkeit des unternehmens sicherstellen – gezielt chancen nutzen und risiken minimieren., 2012.
- [4] R. Marwein. Hierarchische geolokalisierte konfiguration im präventiven krisenmanagement: Die konfiguration der welt. In U. Kloos, G. Tullius, and N. Martínez, editors, *Wissenschaftliche Vertiefungskonferenz 2013*, pages 17–21, 2013.
- [5] C. Moser. *User Experience Design: Mit erlebniszentrierter Softwareentwicklung zu Produkten, die begeistern*. X.media.press. Springer, Dordrecht, 2012.
- [6] A. Schmidt. *Modellgetriebene Informationsvisualisierung auf Basis einer serviceorientierten Architektur*. Diplomica Verlag GmbH, 2008.
- [7] J. Tidwell. *Designing interfaces*. O'Reilly, Sebastopol and CA, 2nd ed edition, 2011.

Language Oriented Programming: Modulare domänenspezifische Sprachen*

Arkadius Weister
Reutlingen University
Arkadius.Weister@Student.
Reutlingen-University.DE

Abstract

Beim Language Oriented Programming (LOP) erstellt der Entwickler eine Programmiersprache, um ein Problem oder eine Aufgabe in einer bestimmten Domäne zu lösen. Dabei wird die Sprache so entwickelt, dass sie das konzeptuelle Modell des Entwicklers ohne Umdenken umsetzen kann. Diese Sprachen nennt man domänenspezifische Sprachen (DSL). Zur Entwicklung dieser Sprachen werden sogenannte Language Workbenches (LWB) verwendet. Diese Arbeit befasst sich mit der Entwicklung von DSLs als ein Mittel zur Umsetzung von LOP. Durch die Nutzung der LWBs kann man DSLs mit relativ kleinem Aufwand erstellen und einsetzen. Im Fokus dieser Arbeit steht die Entwicklung von "Modularen DSLs". Hierbei werden Kriterien und Voraussetzungen für die Modularisierung betrachtet. Zum Abschluss werden drei Konzepte bestehender Systeme anhand dieser Kriterien betrachtet und bewertet.

Schlüsselwörter

Domain-specific Languages, Language Workbenches, Concrete/Abstract Syntax, Design, Generation, Transformation

*

Betreuer Hochschule: Prof. Dr.-Ing. Peter Hertkorn
Hochschule Reutlingen
Peter.Hertkorn@Reutlingen-
University.de

Informatics Inside 2014
Wissenschaftliche Vertiefungskonferenz
30. April 2014, Hochschule Reutlingen
Copyright 2014 Arkadius Weister

CR-Kategorien

D.2.6 [Software]: Programming Environments—*Programmer workbench*; D.2.11 [Software]: Software Architectures—*Domain-specific architectures, Languages*; D.3.2 [Programming Languages]: Language Classification—*Extensible languages, Design languages*

1 Einleitung

Das Konzept des Language Oriented Programming wurde bereits 1994 von M.P. Ward [1] in seinem gleichnamigen Artikel erläutert. Damals hat er es als einen neuen Weg zur Organisierung der Entwicklung von großen Softwaresystemen bezeichnet. Der Ansatz der sprachenorientierten Programmierung beginnt mit dem Entwickeln einer formal spezifizierten und domänenorientierten sehr hohen (very high-level) Programmiersprache. Diese Sprache soll für die Entwicklung in der entsprechenden Domäne zugeschnitten und entwickelt sein [1, S.3]. Um das zu erreichen, wird das Domänenwissen im Design der Sprache so gekapselt, dass der Entwickler der Domäne die Sprache bequem und selbstdokumentierend benutzen kann. Laut Ward ist eine passende Sprache ein gutes Mittel um Domänenwissen greifbar zu machen und den Entwicklungsaufwand dramatisch zu reduzieren. Gleichzeitig wird dadurch die Wartbarkeit und die Wiederverwendbarkeit erhöht. Die Nutzung einer höheren Programmiersprache führt zu weniger Zeilen von Code. Außerdem ist der Code einfacher zu lesen, zu analysieren, zu verstehen und zu modifizieren [1, S.5]. Diese

Sprachen werden "Domänenspezifische Sprachen" genannt (engl: Domain Specific Language, kurz: DSL). Eine Möglichkeit um LOP umzusetzen, ist die Nutzung von Language Workbenches (LWB) [2]. Im nächsten Kapitel wird das Language Oriented Programming detaillierter betrachtet und die grundlegenden Begriffe der Sprachentwicklung erläutert. Die nächsten Abschnitte erläutern die Vorteile und Gründe für domänenspezifische Sprachen sowie Sprachbaukästen (Language Workbenches) und die Modularität bei DSLs. Anhand klassischer Programmiersprachen werden Kriterien für modulare Sprachen ermittelt und erfasst. Im Kapitel Language Workbenches werden die Eigenschaften aufgezählt, die eine LWB besitzen muss, um Modularität bei Sprachen gewährleisten zu können. Hinsichtlich der Kriterien zum Erreichen von Modularität werden drei Konzepte von Language Workbenches betrachtet und bewertet. Das Ergebnis der Bewertung soll Aufschluss darüber geben, ob es möglich ist mithilfe der drei Sprachbaukästen Modulare DSLs zu entwickeln. Es handelt sich dabei um das Jetbrains Meta Programming System (MPS), Xtext und Spoofox.

2 Language Oriented Programming

Die Frage, was Language Oriented Programming ist, lässt sich nicht einheitlich beantworten. Sergey Dmitriev beschreibt LOP als das nächste Programmierparadigma. Er fasst Ansätze wie z.B. Intentional Programming, MDA oder auch generative programming alle unter dem Begriff Language Oriented Programming zusammen [3]. Aber was unterscheidet LOP zu herkömmlichen Programmiermethoden? Bei einer herkömmlichen Programmierung erstellt der Programmentwickler zunächst ein konzeptuelles Modell (mentales Modell), um ein gegebenes Problem zu lösen oder eine Aufgabe zu erfüllen. Dieses Modell muss im nächsten Schritt in einer Programmiersprache abgebildet bzw. implementiert werden. Dazu wählt der Programmierer eine Allzwecksprache (z.B. Java) aus. Der nächste Schritt ist das Übertragen (Mapping)

vom konzeptuellen Modell in die Programmiersprache. Dazu kann der Entwickler nur die Mittel nutzen, die von der Programmiersprache bereitgestellt werden. Dieser letzte Schritt kann sich als sehr schwierig erweisen, da das konzeptuelle Modell in den meisten Fällen nur mit Umwegen übertragen werden kann. Das kann man auf die Vielfalt der Ausdrücke einer natürlichen Sprachen zurückführen. Einem anderen Programmierer kann man das Modell in einer hohen Sprache erklären. Ein Computer benötigt zum Verständnis jedes Detail. Bei LOP baut der Programmierer wie zuvor zunächst das mentale Modell für das gegebene Problem. Anschließend wählt der Programmierer eine spezialisierte domänenspezifische Sprache aus, um das Modell umzusetzen. Sollte es keine passende DSL geben, so entwickelt der Programmierer eine DSL, die zur Lösung des Problems geeignet ist. Die spezialisierte DSL ermöglicht dem Programmierer eine einfache Übertragung vom mentalen Modell in die DSL, da er hier nicht gezwungen ist seine Ideen umständlich in Notationen zu fassen, die von der Sprache verstanden werden können. Zur Umsetzung von LOP benötigt man also DSLs [3, S.1ff]. Demnach werden beim LOP nicht nur Programme geschrieben, sondern auch die Sprachen entwickelt, mit denen sie geschrieben werden.

„Language Oriented Programming will not just be writing programs, but also creating the languages in which to write our programs“

Dmitriev (zitiert nach [3, S.5])

Abbildung 1 zeigt den Ablauf der Entwicklung in der herkömmlichen Programmierung mit einer Allzwecksprache (engl. General Purpose Language, kurz: GPL). In Abbildung 2 kann man den Entwicklungsablauf in LOP sehen. Die Vorgehensweise von LOP soll das Erstellen von spezialisierten DSLs erleichtern, die wiederum das Schreiben von Programmen erleichtern [3, S.6]. Mithilfe von LOP können eine Reihe von Problemen innerhalb großer Softwaresysteme in ihrem Ausmaß verringert werden.

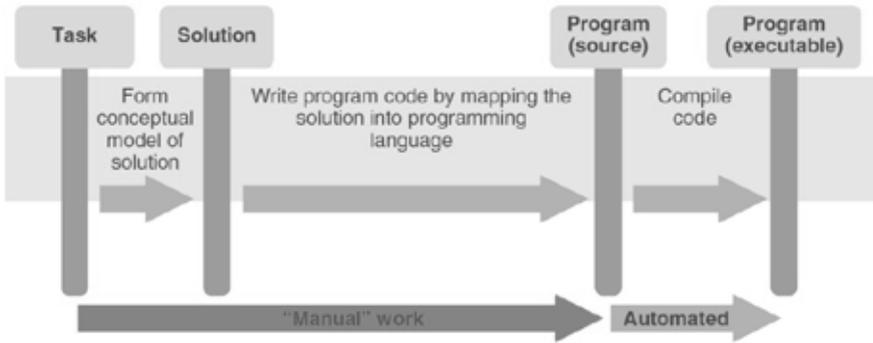


Abbildung 1: Programmierung mit Allzweckssprachen (GPL) [3]

2.1 Was ist eine Sprache?

Die Sprache (engl. Language) ist der zentrale Teil im Language Oriented Programming. Um Sprachen entwickeln zu können, muss man zunächst beantworten, was eine Sprache ist. Nach Anneke Kleppe [4] ist eine Sprache eine Reihe von „Sprachlichen Ausdrücken“: „(Language) A language L is a set of linguistic utterances“ [4, S.23]. Dabei ist ein sprachlicher Ausdruck ein Bestandteil der Sprache, der durch die Sprachspezifikation definiert wird. Es kann sich dabei um ein Modell, ein Programm, ein Datenbankschema, eine Anweisung (engl. Statement) oder aber auch um einen Ausdruck in der Mathematik handeln [4, S.24ff]. Die Sprachspezifikation (engl. language specification) einer Spra-

che ist eine Menge von Regeln für die Ausdrücke dieser Sprache. Dabei wird die Struktur der Ausdrücke beschrieben und mindestens gezeigt, welche sprachlichen Ausdrücke genutzt werden dürfen und gegebenenfalls eine Beschreibung beziehungsweise Definition für diese bereitgestellt [4, S.39f]. Eine Sprache besteht aus einer konkreten Syntax, einer abstrakten Syntax, einer statischen Semantik und einer Ausführungssemantik. Die konkrete Syntax ist die Notation, durch die der Benutzer Programme ausdrücken/schreiben kann [5, S.26]. Die Abbildung der Eingaben eines Benutzers in die abstrakte Syntax wird von der konkreten Syntax bestimmt. Die abstrakte Syntax definiert die Beziehungen der Elemente der Sprache untereinander fest. Oft

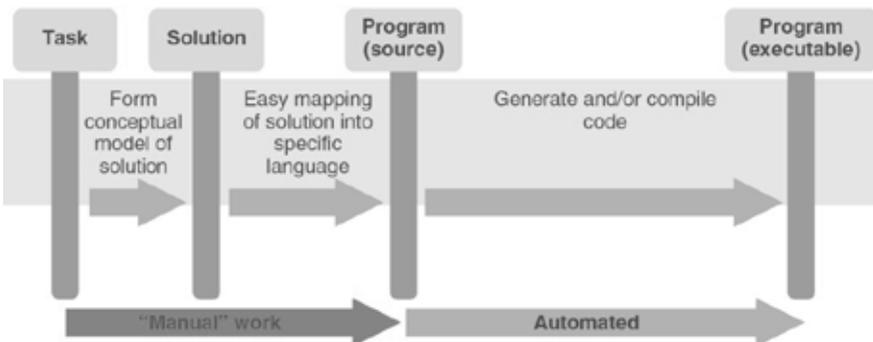


Abbildung 2: Language Oriented Programming mit DSLs [3]

wird die abstrakte Syntax in Form eines Metamodells definiert [6]. Die statische Semantik einer Sprache ist eine Sammlung von Einschränkungen und Regeln, welche von den Programmen erfüllt werden müssen. Zusätzlich muss das Programm eine korrekte Struktur in Bezug auf die konkrete und abstrakte Syntax vorweisen. Die Ausführungssemantik bezieht sich auf Programme, die ausgeführt werden und wird durch eine „execution engine“ realisiert. Eine „execution engine“ kann ein Generator oder Interpreter sein [5, S.26]. Beim LOP wird eine Sprache laut Sergey Dmitriev [3, S.6] mit drei Hauptbestandteilen definiert und kann weitere Aspekte wie Typsysteme und Einschränkungen besitzen:

- **Struktur:** Beschreibt die abstrakte Syntax einer Sprache, die unterstützten Konzepte und die Nutzung dieser.
- **Editor:** Beschreibt die konkrete Syntax einer Sprache. Dazu gehört die Beschreibung, wie sie verarbeitet und genutzt wird.
- **Semantik:** Beschreibt das Verhalten der Sprache. Wie wird die Sprache interpretiert und wie wird sie in ausführbaren Code transformiert.

Wie bereits erwähnt, besteht eine Sprache aus Ausdrücken. Beim LOP werden diese Ausdrücke jedoch Konzepte genannt. Eine Sprache L ist demnach eine Menge/Reihe von Sprachkonzepten C und deren Beziehungen untereinander (siehe Abbildung 3). Der Begriff Konzept (engl. concept) bezeichnet alle Aspekte eines Elements einer Sprache. Dazu gehören alle in diesem Abschnitt besprochenen Bestandteile einer Sprache (z.B. abstrakte Syntax) [5, S.64].

2.1.1 Programm und Domänen

Eine **Domäne** ist ein Wissensbereich, für den man eine Softwareunterstützung bereitstellen möchte. Eine weitere Definition ist, dass eine Domäne durch eine Reihe/Menge von Programmen mit gemeinsamen Charakteristika oder ähnlichem Zweck identifiziert wird. Diese Definition kann man noch spezifizieren. Eine Domäne ist eine Menge von Programmen,

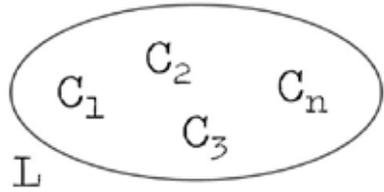


Abbildung 3: Sprache als Menge von Konzepten [5]

die in einer spezifischen Sprache geschrieben wurde. Mit dieser Definition lassen sich DSLs für eine Domäne relativ simpel erstellen, da man die Gemeinsamkeiten der Programme innerhalb der Domäne klar identifizieren kann und damit klar sagen kann, welchen Bereich eine DSL abdecken muss und welcher Code von den Programmen der DSL generiert werden muss [5, S.58f]. Abbildung 4 zeigt eine Domäne PD und zwei DSLs für diese Domäne ($PL1$ und $PL2$). Die Sprache $PL1$ deckt in diesem Beispiel nicht die ganze Domäne ab und $PL2$ deckt zu viel ab und hat Funktionalitäten, welche nicht von der Domäne gebraucht werden.

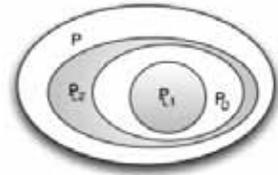


Abbildung 4: Domäne mit Sprachen [5]

Ein **Programm** ist eine konzeptionelle Darstellung einer Berechnung, die auf einem Universalrechner läuft [5, S.57]. Viele Programmierer stellen sich ein Programm als eine Reihe von Instruktionen für den Computer vor. Diese werden dann nacheinander ausgeführt. Dmitriev definiert ein Programm in LOP als eine eindeutige Lösung für ein Problem. Im Genauen heißt das: Ein Programm ist jedes präzise definierte Modell einer Lösung für ein Problem in einer Domäne, ausgedrückt mit Konzepten der Domäne [3, S.4f].

2.2 Ziele

M.P. Ward hat in seinem Artikel von Language Oriented Programming bereits die Probleme großer Softwaresysteme beschrieben und den positiven Einfluss von LOP auf diese gezeigt. Die folgende Liste ist eine kurze Zusammenfassung der Probleme, auf die LOP nach [1, S.2-3, S.20] positiven Einfluss hat:

- **Komplexität:** Große Softwaresysteme führen zu mehr Komplexität innerhalb des Systems. Dies kann zum Beispiel durch Kommunikationsschwierigkeit innerhalb von Teams und Entwicklern hervorgerufen werden. Diese können wiederum zu Kostenüberschreitungen usw. führen. Durch die Größe der Systeme gibt es eine steile Lernkurve für neues Personal und es ist kaum möglich, alle Zustände des Systems ausreichend zu visualisieren. Durch den Einsatz von LOP wird die Komplexität eines Systems in großem Maße reduziert. Die Komplexität der Domäne wird in Form von abstrakten Datentypen und Konstrukten gekapselt. Durch diese domänenspezifische Sprache werden komplexe Funktionen des Systems mit nur wenigen Codezeilen implementiert.
- **Konformität:** Die Nutzung von domänenorientierten Sprachen macht das Erreichen von Konformität einfacher. Zum Beispiel kann eine Sprache entwickelt werden, welche die aktuellen Konzepte der Steuergesetze benutzt. Mit dieser Sprache wird dann ein System zur Lohnberechnung geschrieben, welches die Anforderungen Steuerregulierungen erfüllt.
- **Änderungen:** Die einfache Verständlichkeit und Benutzbarkeit der DSL und auch die geringe Größe des Quellcodes führen dazu, dass die Sprache einfach geändert und erweitert werden kann.
- **Übersichtlichkeit:** In einer hohen domänenspezifischen Sprache wird die

meiste Komplexität innerhalb der Sprache versteckt (Objekte, Konstrukte, Operationen). Dadurch lässt sich das System einfacher Visualisieren und damit besser validieren. Die problemspezifischen Aspekte stehen hier im Vordergrund.

Language Oriented Programming bietet laut [1, S.6ff] eine Reihe von Vorteilen:

- **Trennung der Bereiche:** Das Design des Systems wird vollständig von der Implementierung der Sprache getrennt. Dadurch bleibt das Design der Sprache simpel, leistungsstark und ausdrucksstark. Die Komplexität wird reduziert.
- **Hohe Produktivität in der Entwicklung:** Durch die DSL genügt eine geringe Anzahl von Codezeilen um komplexe Funktionen darzustellen. Es werden nur die Funktionalitäten implementiert, die relevant für die Domäne sind.
- **Hohe Wartbarkeit:** Die Wartbarkeit ist abhängig von der Größe des Softwaresystems. Eine geringe Anzahl von Codezeilen führt zu einer besseren Wartbarkeit.
- **Hohe Portabilität:** Die Sprache und das System lassen sich einfach von einem Betriebssystem zu einem anderen portieren. Es muss darauf geachtet werden, dass die genutzten Sprachen (z.B. Java) auf dem Zielsystem vorhanden sind.
- **Wiederverwendbarkeit:** Die Sprache kapselt ein großes Domänenwissen und kann daher für die Anforderungsanalyse eines neuen Systems genutzt werden. Außerdem kann die Sprache bei ähnlichen Projekten wiederverwendet werden.
- **Erweiterbarkeit des Systems:** Mit einer passenden Schnittstelle zu der Sprache und geeigneten Templates kann der Benutzer neue Funktionen integrieren oder bestehende erweitern.

3 Domänenspezifische Sprachen (DSL)

Der einzige Weg, um Code vollständig und ausführbar aus einem Modell zu generieren ist, wenn die Sprache und die Generatoren domänenspezifisch aufgebaut sind [7, S.23]. Ein Sprachentwickler definiert oder erweitert eine Domänenspezifische Sprache (DSL). Dabei stehen die Bedürfnisse der Benutzer im Vordergrund. Zur Generierung von Code müssen Generatoren für die DSL geschrieben werden oder bereits vorhandene genutzt werden. Dabei wird zum Beispiel der geschriebene Code der DSL in eine Zielsprache mittels Transformation generiert [7, S.24ff]. Laut Markus Völter [5] werden DSLs immer wichtiger in der Softwaretechnik und es gibt Werkzeuge, um DSLs mit relativ wenig Aufwand zu entwickeln. Eine domänenspezifische Sprache ist eine Sprache für eine Domäne, die darauf spezialisiert ist, Programme für die Domäne zu erstellen. Das macht die DSL für die Entwicklung von Programmen der Domäne effizienter als andere Sprachen. Eine DSL wird so konzipiert, dass sie so gut wie möglich in die Domäne passt. Dazu werden Abstraktionen in der Sprache genutzt, die unnötige Details vermeiden, die nicht in den Programmen gebraucht werden [5, S.59f]. Die Wahl der konkreten Syntax einer DSL soll sich nach den Benutzern in der jeweiligen Domäne richten und damit ihre Arbeit innerhalb der Domäne erleichtern [8, S.97]. Eine bekannte DSL ist zum Beispiel die Sprache für Datenbanken „SQL“ (Structured Query Language), die eine Schnittstelle für relationale Datenbanksysteme bietet. Ein weiteres Beispiel ist die Beschreibungssprache HTML für die Erstellung von Webseiten [7, S.25]. Eine DSL sollte folgende Basiskriterien erfüllen [7, S.26]:

- Die Konzepte der Sprache müssen von den Experten der Domäne in ihrem Zweck und ihrer Funktionalität verstanden werden.
- Grafische oder textuelle Notation der Sprache sollte einfach zu benutzen sein.
- Die Regeln und Grammatik der Sprache müssen ausführlich definiert sein.

Bevor eine DSL definiert und entwickelt werden kann, muss eine Domänenanalyse durchgeführt werden. In der Analyse werden die Aktivitäten, Objekte und Operationen von Systemen innerhalb einer Domäne identifiziert und in einem Domänenmodell festgehalten. Die Ergebnisse einer Domänenanalyse helfen bei der Entwicklung und Definition einer neuen DSL und beim Finden von existierenden DSLs [9, S.28]. Ein Domänenmodell besteht aus den folgenden Elementen [9]:

- Definition der Domäne und ihres Anwendungsbereiches.
- Definition einer Domänenterminologie mit den wichtigsten Begriffen.
- Beschreibung der Konzepte in der Domäne.
- Beschreibung der Gemeinsamkeiten, Abhängigkeiten und Unterschiede der Domänenkonzepte .

Zur Definition und Entwicklung einer neuen DSL gehören laut Martin Fowler [2] drei Bestandteile. Einer davon ist das Definieren einer abstrakten Syntax, die das **Schema** der abstrakten Repräsentation ist. Der zweite Bestandteil ist das Definieren eines **Editors**. Dieser definiert die konkrete Syntax und ermöglicht die Bearbeitung der abstrakten Repräsentation über eine Projektion. Der letzte Bestandteil ist das Definieren eines **Generators**. Dieser beschreibt, wie die abstrakte Repräsentation der Sprache in eine ausführbare Repräsentation der Sprache übersetzt wird. Ein Generator definiert also die Semantik einer DSL. Diese drei Bestandteile sind das Minimum und können in ihrer Anzahl nach oben variieren. Es kann zum Beispiel mehr als einen Generator für die DSL geben. Damit können die DSL in mehrere Zielsprachen übersetzt werden (z.B Java und C#). Es kann auch mehrere Editoren geben. Dabei kann jeder Editor für eine Benutzergruppe zugeschnitten werden [2]. Damit sind alle, von Sergey Dmitriev genannten, Bestandteile einer Sprache im Language Oriented Programming durch eine DSL abgedeckt (siehe: 2.1).

3.1 Gründe für DSL

Welche Gründe bestehen für die Nutzung einer DSL? Überwiegen die Vorteile den Nachteilen oder kann keine klare Antwort darauf gegeben werden? Den folgenden Grund für die Benutzung hat bereits die Definition von DSLs geliefert. DSLs erlauben es implizit vorhandenes Expertenwissen in kompakter Form explizit zu formulieren [6]. In Abbildung 5 wird die Entwicklung eines Produktes innerhalb einer Domäne gezeigt. Zunächst besteht eine Idee für die Lösung eines Problems der Domäne. Bei der herkömmlichen Methode wird im nächsten Schritt ein Mapping der Idee in ein Modell oder direkt in Code einer Programmiersprache durchgeführt. Beim Nutzen einer DSL kann jedoch auf das umständliche Mapping verzichtet und die Lösung direkt mit der Domänensprache in einem Domänenmodell dargestellt und daraus mithilfe von Codegenerierung ein fertiges Produkt erzeugt werden [10, S.15f].

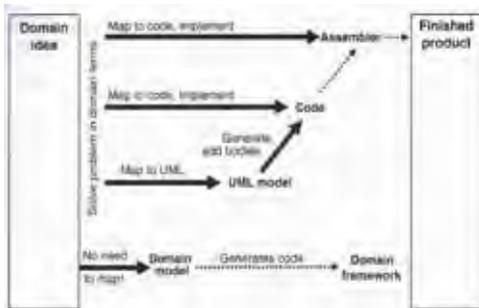


Abbildung 5: Von der Idee zum Code [10]

Die Nutzung von DSLs kann Vorteile und auch Herausforderungen mit sich bringen. In seinem Werk „DSL Engineering“ [5, S.40ff] geht Markus Velter auf die Vorteile und Herausforderungen bei DSL ein. Die folgende Liste zeigt eine Zusammenfassung davon. Die Vorteile der Benutzung von DSL sind:

- *Steigerung der Produktivität:* Da eine DSL wenig Codezeile besitzt, wird die Komplexität gesenkt und die Arbeit effizienter.

- *Steigerung der Qualität:* Durch eine DSL entstehen weniger Fehler, bessere Konformität in der Architektur und keine Duplizierung von Code.
- *Bessere Validierung und Verifikation:* Eine DSL ist semantisch reicher als eine GPL. Dadurch ist sie nicht so überladen und Analysen und Validierung sind einfacher durchzuführen.
- *Langlebigkeit der Daten:* Die Modelle sind in einer hohen Abstraktionsebene ausgedrückt. Basierend auf den Modellen werden Analysen oder Generierungen starten. Das Modell kann auch in andere Repräsentationen transformiert werden.
- *Besseres Verständnis und Kommunikation:* Die Benutzer und Entwickler der DSL verstehen die Domäne und die Sprache sehr gut. Dadurch wird die Teamkommunikation einfacher.

- *Beteiligung:* Die Domänenexperten werden bei der Entwicklung der DSL beteiligt und können diese selbstständig benutzen, da sie wegen der Nähe zur Domäne selbsterklärend ist.

- *Werkzeuge:* Viele DSLs werden mit Werkzeugen (IDE) geliefert und bieten Debugger, Codevervollständigung, Simulatoren, Visualisierungen und vieles mehr.

- *Kein Mehraufwand:* Bei der Generierung von Code wird kein Mehraufwand in der Laufzeit geleistet. Bei Fragen der Leistung und Ressourcen ist das ein wichtiger Faktor.

- *Plattformunabhängig:* Die Anwendungslogik einer DSL ist unabhängig von der Plattform und Implementierung.

Die Herausforderungen bei DSLs sind:

- *Aufwand:* Der Aufwand bei der Entwicklung einer DSL ist hoch. Es wird Expertenwissen gebraucht und es sollte

eine Kostenanalyse durchgeführt werden. Eine DSL sollte so konzipiert werden, dass man sie wiederverwenden kann. Zur Reduzierung der Kosten werden Werkzeuge wie Language Workbenches (siehe Kapitel 5) benutzt.

- *Erfahrung und Fertigkeit*: Die Definition einer guten Sprache benötigt Erfahrung und muss durch Praxis erlernt werden.
- *Probleme bei Vorgängen*: Die Entwickler der Sprache, die Benutzer und die Domänenexperten müssen zusammenarbeiten, damit es nicht zu Problemen in den Prozessen kommt (z.B. werden mehrere Sprachen für eine Domäne entwickelt). Deshalb müssen Prozesse für die Kommunikation untereinander definiert werden.
- *Entwicklung und Wartung*: Durch die schnelle Weiterentwicklung einer DSL kann es zu Wartungsschwierigkeiten kommen. Gleichzeitig muss eine Sprache immer weiterentwickelt werden, damit sie nicht veraltet.
- *Zu viele DSLs*: Es besteht die Gefahr, dass erfahrene Entwickler immer neue DSLs erstellen, anstatt zu prüfen, ob es bereits eine existierende DSL in der Domäne gibt. Das führt dazu, dass viele halb fertige DSLs und keine vollständigen entwickelt werden.
- *Investierung*: Wenn man sich an eine Arbeitsweise gewöhnt hat und effizient darin ist, ist es schwer davon abzulassen und etwas Neues zu probieren.
- *Werkzeuge*: Es gibt grundsätzlich keine Interoperabilität zwischen DSL Tools. Dadurch ist die Entwicklung einer DSL abhängig von einem Tool (Werkzeug).
- *Kultur*: Domänenexperten sehen sich oft nicht als Programmierer und möchten nur in Standards wie z.B. UML modellieren. Außerdem denken viele, dass

die Entwicklung von Sprachen zu kompliziert ist [5, S.45]. Diese und weitere Annahmen erschweren die Einführung von DSLs.

Ob eine DSL genutzt werden sollte kommt auf die jeweilige Situation der Domäne, der Entwickler und der Benutzer an. Wenn ein Entwickler eine Domäne nur schwer versteht und/oder die Programme einer Domäne nicht implementieren kann, dann sollte eine DSL nicht entwickelt werden. Wenn die Kunden und die Benutzer keine klaren Definitionen für eine Domäne liefern können und sie ständig erweitern wollen, so können die Kosten für die Entwicklung schnell sehr hoch werden. Eine DSL sollte also nur entwickelt werden, wenn die Benutzer sie eindeutig definieren kann und der Entwickler gute Kenntnisse über Softwaretechnik, Design, Testing und die Domäne besitzt [5, S.46f].

3.2 Interne und externe DSLs

Es gibt zwei Arten von DSLs. Interne DSLs und externe DSLs. Bei einer **internen DSL** handelt es sich um eine Sprache, die in eine andere Sprache eingebettet ist. Diese andere Sprache wird Hostsprache genannt. Die interne DSL verwendet also die gleiche Syntax der Hostsprache und ist mit ihren Mitteln beschrieben. Zur Verwendung von internen DSLs bieten sich vor allem Hostsprachen mit geeigneten Erweiterungsmöglichkeiten an. Eine **externe DSL** ist eine Sprache, die von Grund auf neu erstellt wurde. Sie ist demnach nicht in eine andere Sprache eingebettet. Zum Lesen der Informationen wird ein Parser erstellt. Dadurch lässt sich die externe DSL zur Interpretation und Generierung innerhalb einer anderen Programmiersprache verwenden [8, S.98]. Der Vorteil einer externen DSL ist ihre Fähigkeit, eine Domäne in einfachster Form zum Lesen und Modifizieren darzustellen. Zusätzlich kann eine externe DSL zur Laufzeit evaluiert werden. So werden zum Beispiel Parameter verändert, ohne das Programm neu übersetzen zu müssen. Der Nachteil einer externen DSL liegt darin, dass ein Übersetzer und Parser neu erstellt werden muss und das sie nicht mit einer Basissprache

verbunden ist. Im Gegensatz dazu kann eine interne DSL alle Möglichkeiten einer Basissprache (Hostsprache) ausschöpfen. Eine interne DSL ist aber nur auf die Syntax und Struktur der Basissprache beschränkt [7, S.29f].

4 Modulare DSL

Eine wichtige Frage bei dem Einsatz von DSL ist die Möglichkeit der Reduzierung und Einsparung von Kosten und Investitionen. Dazu sollten Sprachen von Beginn an so entwickelt werden, dass sie möglichst oft wiederverwendet werden können. Dabei muss nicht die gesamte Sprache wiederverwendet werden. Die Sprachelemente und Rumpfstrukturen einer Sprache eignen sich bereits zur Wiederverwendung. Um das zu ermöglichen, müssen die einzelnen Sprachdefinitionen möglichst getrennt voneinander spezifiziert und flexibel miteinander kombiniert werden können. Diese „modulare“ Entwicklung erlaubt es, neue Sprachen aus existierenden Komponenten zusammenzustellen. Eine solche DSL wird als „Modulare DSL“ bezeichnet [6]. „Die grundlegende Technik ist die modulare Entwicklung einer Sprache aus bereits existierenden Bausteinen und gegebenenfalls einer erweiterbaren Rumpfsprache“ [6]. Modulare Sprachen bestehen aus einem minimalen Sprachkern und einer Bibliothek von Sprachmodulen, die je nach Bedarf importiert und genutzt werden können. In Abbildung 6 wird eine solche modulare Sprache dargestellt. Der Kern ist eine kleine Sprache (**my L**) mit wenigen, aber starken Sprachkonzepten (α, β). Neue oder individuelle Sprachmodule (a, b, c...) können jederzeit importiert und verwendet werden. Ein Sprachmodul ist wie eine Bibliothek, besitzt aber seine eigene Syntax. Ein importiertes Sprachmodul verhält sich wie ein Teil der zusammengesetzten Sprache und kann mit anderen Modulen verbunden sein [5, S.35f]. Ein Sprachmodul kann Teil anderer Sprachen sein und selbst aus Untermodulen bestehen.

4.1 Modularität von Sprachen

Wie bereits erwähnt, macht die Modularisierung von DSLs die Softwareentwicklung effizienter und billiger. Das liegt daran, dass ähnliche und gleiche Funktionalitäten nicht

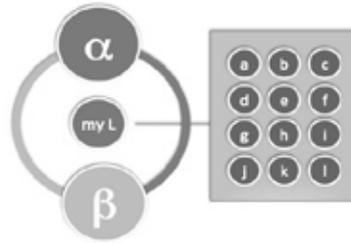


Abbildung 6: Modulare Sprache [5]

mehrfach entwickelt, sondern stattdessen wiederverwendet werden. Sprachen werden miteinander kombiniert und wiederverwendet. Welche Möglichkeiten gibt es, Modularität umzusetzen und was muss bei der Komposition beachtet werden? Bei der Komposition von Sprachen muss die Komposition der abstrakten Syntax, der konkreten Syntax, der Einschränkungen, Typsysteme und der Semantik durchgeführt werden. Bei der Komposition von Sprachen wird zwischen vier Arten der Modularisierung unterschieden: 1. Referenzierung (Referencing), 2. Wiederverwendung (Reuse), 3. Erweiterung (Extension) und 4. Einbettung (Embedding). Der Einsatz von modularen DSLs und die daraus resultierende Komposition und Wiederverwendung der Sprachen führt zur Reduzierung unnötiger halb fertiger DSLs (siehe Herausforderungen in Kapitel 3.1). Die Erweiterung von Sprachen erlaubt dem Benutzer das Hinzufügen neuer Sprachkonstrukte in die bestehende Sprache [5, S.116f].

Referenzierung (Referencing) erlaubt homogene Querverweise zwischen den Sprachen. Die referenzierende Sprache ist dann von der referenzierten Sprache abhängig. Abbildung 7 zeigt eine Sprache **I2**, die abhängig von der Sprache **I1** ist, da mindestens ein Konzept **B1** der Sprache **I2**, eine Referenz auf ein Konzept **A2** der Sprache **I1** setzt. Zum Referenzieren werden Kenntnisse über die Beziehungen und das Design der Sprachen benötigt. Die abhängigen Sprachen können, wegen ihrer Abhängigkeit zu anderen Sprachen, nicht wiederverwendet werden [5, S.118f].

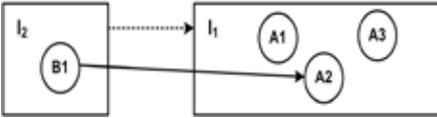


Abbildung 7: Referenzierung [5]

Bei der **Erweiterung (Extension)** erweitert eine Sprache **B** eine andere Sprache **A** um weitere Sprachkonzepte. Alle Programme, die in der Sprache **B** geschrieben werden, können alle Konzepte der Sprache **A** zusätzlich zu denen der Sprache **B** verwenden. Durch Erweiterung kann also eine Sprache um neue Funktionen und Methoden erweitert werden, um ein neues Problem innerhalb einer Domäne zu lösen, ohne dafür eine neue Sprache zu definieren. Es gibt auch die Möglichkeit eine Sprache in der Erweiterung zu beschränken (**Restriction**). Dadurch werden nicht gebrauchte Konzepte der Sprache **A** in der Sprache **B** nicht zugelassen [11]. Abbildung 8 zeigt die Erweiterung einer Sprache **I1** durch die Sprache **I2**. Das Konzept **B3** erweitert das Konzept **A3** so, dass es als Kind von **A2** verwendet werden kann. Die Spracherweiterung ist sehr nützlich bei DSLs, die einfach starten und mit der Zeit komplizierter werden und neue Fälle und Feinheiten bekommen. Die Erweiterungen können in separaten Sprachmodulen definiert werden, welche die Kern-DSL erweitern. Die Benutzung der Module kann zusätzlich auf bestimmte Benutzer eingeschränkt werden [5, S.120f].

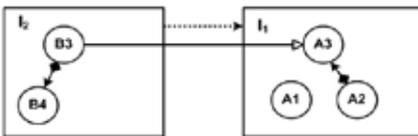


Abbildung 8: Erweiterung [5]

Die **Wiederverwendung (Reuse)** bezieht sich auf zwei voneinander unabhängige Sprachen. Die Sprachen wurden zu einem Zeitpunkt entwickelt, bei dem nicht klar war,

dass sie beide in einem gemeinsamen Kontext genutzt werden würden. Die Sprachen kennen sich nicht und können auch keine Konzepte untereinander referenzieren. Eine Möglichkeit die Sprachen wiederverwendbar zu machen, ist der Einsatz einer dritten Sprache. Die dritte Sprache dient als Adapter und ermöglicht die Referenzierung und Vererbung der zwischen den beiden unabhängigen Sprachen. Die Adaptersprache ist von beiden Sprachen abhängig. Abbildung 9 zeigt die unabhängigen Sprachen **I1** und **I2** [5, S.123f].

Mithilfe der Adaptersprache **IA** wird die Referenzierung ermöglicht. Das Konzept **B5** der Adaptersprache erbt von **B4** und referenziert auf **A3**. Während Sprachreferenzierung nur die Wiederverwendung der referenzierten Sprache unterstützt, bietet Reuse zusätzlich die Unterstützung der Wiederverwendung referenzierender Sprachen [5, S.123f].

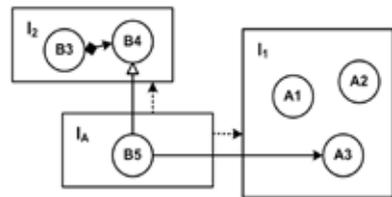


Abbildung 9: Wiederverwendung [5]

Einbettung (Embedding) ist der Wiederverwendung (Reuse) ähnlich. Es gibt wieder zwei unabhängige Sprachen. Im Gegensatz zu Reuse wird bei der Einbettung keine Referenz zwischen den Sprachen erstellt, sondern Instanzen von Konzepten der Sprache **I2** in Teile der Sprache **I1** eingebettet. Es gibt keine Abhängigkeiten zwischen der eingebetteten Sprache und der Hostsprache. Die Einbettung kann durch das Verwenden einer Adaptersprache realisiert werden. Abbildung 10 zeigt eine Adaptersprache **IA**, die von beiden Sprachen abhängig ist und das Konzept **A3** in sein Konzept **B5** einbettet. Einbettung unterstützt also die syntaktische Komposition von unabhängig entwickelten Sprachen. Eine besondere Art der Einbettung ist die Querschnitt-

Einbettung mit Metadaten (Cross-Cutting Embedding, Meta Data). Metadaten werden als Programmelemente definiert, die unabhängig von der Semantik sind. Die Daten müssen dennoch das Programm betreffen. Die Einbettung der Metadaten sollte in jeder Sprache möglich sein und dabei sollten keine Abhängigkeiten zu den Sprachen entstehen. Ein Beispiel für die Einbettung von Metadaten ist bei der Erstellung von Dokumentationen [5, 124ff].

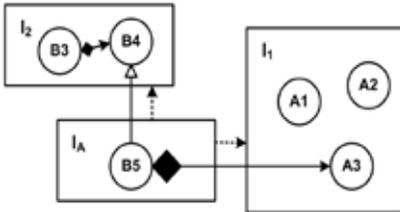


Abbildung 10: Einbettung [5]

4.2 Kriterien

In diesem Abschnitt werden die Kriterien und Herausforderungen aufgezählt, die eine Sprache erfüllen muss, um die vier oben genannten Arten der Modularisierung umsetzen zu können. Der herkömmliche DSL-Ansatz reicht hier nicht aus und muss um eine Modularisierungsmöglichkeit erweitert werden. Dies kann mit einer bestehenden DSL durchgeführt oder bereits bei der Definition einer neuen DSL berücksichtigt werden [12].

Syntax

Bei der *Referenzierung* und *Reuse* ist das Vermischen von konkreter Syntax nicht erforderlich, da eine Referenz zwischen Konzepten und Fragmenten der Sprache lediglich ein Bezeichner/Identifizierungszeichen ist und keine innere Struktur mit Grammatik enthält. Durch Namensauflösung werden die Querverweise in den abstrakten Syntax-Objekten gesetzt. Bei der *Erweiterung* und *Einbettung* werden Definitionen für die modulare konkrete Syntax benötigt, da zusätzliche Sprachelemente mit Programmen der Hostsprache oder Basissprache vermischt werden

müssen. Das Kombinieren von unabhängigen Sprachen kann Probleme verursachen. Die Parsertechnologien der Sprachen dürfen sich gegenseitig nicht ausschließen und die kombinierten Grammatiken verstehen können. Viele parserbasierte Sprachbaukästen (LWB) haben dieses Problem bisher nicht vollständig lösen können. Projektionale Sprachbaukästen brauchen keinen Parser und haben dieses Problem nicht [5, S.127f].

Typ-System

Bei der *Referenzierung* muss die referenzierende Sprache die Regeln und Einschränkungen der referenzierten Sprache berücksichtigen. Da die referenzierte Sprache bereits bei der Entwicklung der referenzierenden Sprache bekannt ist, können die Typ-Systeme unter Berücksichtigung der referenzierten Sprache implementiert werden. Bei der *Erweiterung* müssen die Typ-Systeme der Basissprache so entworfen sein, dass in der Spracherweiterung neue Typ-Regeln hinzugefügt werden dürfen. Bei der *Einbettung* und *Reuse* müssen sich die Typ-Regeln, die das Zusammenspiel der beiden Sprachen beeinflussen, in der Adaptersprache befinden. Die Typ-Systeme der beiden Sprachen müssen erweiterbar sein [5, S.128f].

Transformation und Generierung

Bei der *Referenzierung* werden drei Fälle unterschieden. Im ersten Fall wird die Struktur der Referenzierung in das Zielfragment (Zielcode) übernommen. Die referenzierte Sprache und die referenzierende Sprache benutzen ihre Generatoren. Der Generator der referenzierenden Sprache kann der Generator der referenzierten Sprache oder nach seinem Vorbild entworfen sein. Die beiden Sprachen benutzen ihre eigenen Konzepte und Fragmente als Quelle (single-sourced) und es entstehen zwei Zielfragmente mit derselben Referenzierung wie die Quellen. Im zweiten Fall wird von mehreren Quellen ausgegangen, die ein Zielfragment transformieren. Dazu muss eine neue Transformation geschrieben werden, die das referenzierende Fragment der Sprache bei der Generierung

des Zielcodes der referenzierten Sprache berücksichtigt. Im dritten Fall wird eine Transformation vor der Generierung durchgeführt. Diese Transformation verbindet die beiden Quellfragmente der beiden Sprachen in einer neuen einzelnen Quelle. Die bereits vorhandene Transformation der referenzierten Sprachen kann genutzt werden [5, S.129f].

Bei der *Erweiterung* werden die Spracherweiterungen aus sprachlichen Abstraktionen von Ausdrücken einer Domäne definiert. Der Generator für die neuen Sprachkonzepte kann diese Ausdrücke einfach durch Assimilierung nachbauen. Zum Beispiel wird Code in der Hostsprache vom Code generiert, der in der Spracherweiterung ausgedrückt wurde. Spracherweiterungen stellen eine Gefahr zu semantischen Interaktionen dar. Die Transformationen der unabhängig voneinander erstellten Spracherweiterungen könnten miteinander interagieren. Um dieses Problem zu vermeiden, müssen Transformationen so entworfen sein, dass sie keine knappen Ressourcen verbrauchen wie z.B. vererbte Links [5, 130f].

Beim *Reuse* besitzen die wiederverwendete Sprache und die Kontextsprache bereits ihre eigenen Generatoren. Diese Generatoren müssen den Code der beiden Quellsprachen in dieselbe Sprache oder in kompatible Sprachen transformieren. Falls die Generatoren dies nicht unterstützen, funktioniert die Wiederverwendung nicht. Bei der Wiederverwendung kann es drei Fälle geben. Im ersten Fall gibt es bereits existierende Transformationen für beide Sprachen und einen Generator für die Adaptersprache. In der Transformation der Kontextsprache wurden Löcher eingebaut. Der Generator der Adaptersprache wandelt den Code der Kontextsprache so um, dass er mit der wiederverwendeten Sprache kompatibel ist, und füllt dazu die zuvor eingebauten Löcher der Kontextsprache mit Elementen der wiederverwendeten Sprache auf. Im zweiten Fall wird der bereits existierende Generator der wiederverwendeten Sprache mit Transformationscode der Kontextsprache erweitert.

Dazu ist ein Mechanismus zur Komposition von Transformationen nötig. Im dritten Fall wird die Komposition der erzeugten Codefragmente von den Zielsprachen durchgeführt [5, S.131f].

Bei der *Einbettung* muss für die Transformation von eingebetteten Sprachen eine neue Transformation geschrieben werden, falls die bereits existierende Sprache nicht dieselbe Zielsprache erzeugt wie die Transformation der Hostsprache. Wenn die Transformationen dieselbe Zielsprache erzeugen, dann kann der Generator der Adaptersprache die Transformationen der Hostsprache und der eingebetteten Sprache koordinieren. In einem anderen Fall wird der Code der eingebetteten Sprache in die Hostsprache transformiert und danach zusammen mit der Hostsprache in die Zielsprache umgewandelt [5, S.132].

5 Language Workbenches

Die grundlegende Frage beim Language Oriented Programming ist nach Martin Fowler die Kosten-Nutzen-Abwägung. Welchen Gewinn bringt die Nutzung von DSLs und welche Kosten werden bei der Entwicklung von notwendigen Werkzeugen erbracht, um diese Sprache effizient zu unterstützen? Beim Nutzen von internen DSLs gibt es weniger Kosten für Werkzeuge, ist aber an die Syntax der Basissprache gebunden. Mit einer externen DSL kann der höchste Nutzen und Gewinn aus einer DSL gezogen werden. Jedoch steigen hier die Kosten bei der Entwicklung der Sprache, der Übersetzer und der Werkzeuge. Um diesen Kosten entgegen zu wirken, wurde eine neue Kategorie von Werkzeugen entwickelt. Diese Werkzeuge werden Sprachbaukästen (Language Workbenches) genannt. Diese bieten Flexibilität bei externen DSLs ohne eine semantische Barriere und machen es einfacher Werkzeuge für moderne Entwicklungsumgebungen zu bauen [2].

Martin Fowler beschreibt Language Workbenches mit den folgenden Charakteristika [2]:

1. Benutzer können frei neue Sprachen definieren, die vollständig miteinander integriert sind.
2. Die wichtigste Informationsquelle ist eine anhaltende abstrakte Darstellung.
3. Sprachentwickler definieren eine DSL in drei Hauptteilen: Schema (abstrakte Syntax), Editor (konkrete Syntax) und Generator(en).
4. Die Benutzer der Sprache manipulieren eine DSL über einen projektionalen Editor.
5. Eine Language Workbench kann aus unvollständigen oder widersprüchlichen Informationen in seiner abstrakten Darstellung bestehen.

Die Punkte 2-4 setzen die Benutzung eines projektionalen Systems voraus. Mittlerweile wird der Begriff Language Workbenches von Martin Fowler und anderen aber auch für Werkzeuge mit parserbasierten Systemen genutzt [11, S.5]. Laut Voelter spielt es keine Rolle, ob man anstatt eines projektionalen Editors einen parserbasierten Editor zum Erstellen von modularen Sprachen benutzt. Wichtig ist, dass eine Sprache modular ist. Welcher Ansatz zum Erreichen der Modularität benutzt wird, ist nicht wichtig, solange das Ziel erreicht wird [5, S.36].

5.1 Arten von Language Workbenches

Bevor auf die Arten von Language Workbenches eingegangen wird, müssen zunächst die Begriffe **Parser** und **Abstrakter Syntaxbaum (AST)** beschrieben werden. Ein Parser ist ein Stück Software, dessen Aufgabe es ist ein Modell aus seiner konkreten Syntax in seine abstrakte Syntax zu überführen [8, S.174]. Der Parser prüft die konkrete Syntax auf Korrektheit und erzeugt danach entsprechende Parsebäume aus den Ausdrücken. Diese Bäume werden abstrakte Syntaxbäume genannt und können von Interpretern oder Generatoren weiterverarbeitet werden [8, S.107].

```
var x: int; calc y: int = 1 + 2 * sqrt(x)
```

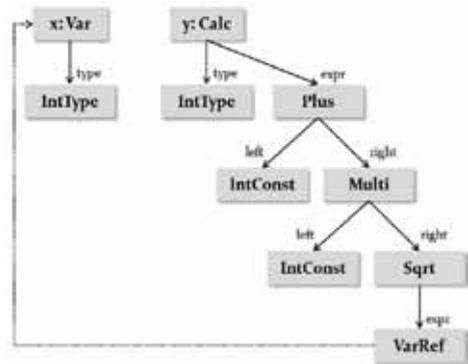


Abbildung 11: Abstrakter Syntaxbaum [5]

In Abbildung 11 ist im oberen Teil ein Programm in konkreter Syntax geschrieben und im unteren Teil der dazugehörige abstrakte Syntaxbaum. Das Programm hat eine hierarchische Struktur. Die Definitionen von **x** und **y** sind auf der obersten Ebene. Innerhalb von **y** gibt es einen geschachtelten Ausdruck, der sich in linke und rechte Schritte unterteilt. Die Boxen repräsentieren Instanzen der Sprachkonzepte.

Es gibt zwei Wege um die Beziehung zwischen konkreter Syntax (CS) und abstrakter Syntax (AS) in der Sprachentwicklung zu beschreiben. Beim Ansatz *CS-first* wird zuerst eine konkrete Syntax definiert und daraus eine abstrakte Syntax abgeleitet. Beim Ansatz *AS-first* wird zuerst eine abstrakte Syntax definiert. Danach wird eine konkrete Syntax geschrieben, die auf die abstrakte Syntax bezogen ist [5, S.178f].

Bei den Language Workbenches wird zwischen parserbasierten Systemen und projektionalen Systemen unterschieden. Im Folgenden werden beide Ansätze erläutert.

Parserbasierte Systeme:

Im parserbasierten Ansatz wird der abstrakte Syntaxbaum aus der konkreten Syntax des Programmes konstruiert. Ein Parser initiiert und füllt die abstrakte Syntax basierend auf den Informationen des Programmtextes. In diesem Ansatz kann der Benutzer den Text frei editieren, da die konkrete Syntax stark von der abstrakten Syntax getrennt ist. Beim parserbasierten Ansatz gibt es die Möglichkeit den Ansatz *CS-first* oder *AS-first* je nach LWB zu benutzen. Die Parser können selbst per Hand geschrieben werden oder von in den LWBs enthaltenen Parsergeneratoren automatisch erstellt werden. Das Parsen wird in den meisten Fällen in mehrere Prozesse aufgeteilt. Der Text wird zunächst in eine Sequenz von Zeichen/Symbolen aufgeteilt. Dieser Schritt wird von einem **Scanner** durchgeführt. Der Parser erstellt dann den AST aus der Sequenz. Der Scanner muss sich der Bedeutung der Eingaben bewusst sein. Zum Beispiel muss er alle Schlüsselwörter kennen und trotzdem dein Einsatz von Bezeichnern mit dem gleichen Namen (wie das Schlüsselwort) erlauben und durch den Kontext erkennen, dass es sich an dieser Stelle nicht um ein Schlüsselwort handelt. Viele Sprachen unterstützen diese Funktionalität nicht (z.B. Java). Deswegen gibt es neben dem normalen Parsen auch das Parsen ohne Scanner (scannerless parsing)[5, S.179ff].

In Abbildung 12 kann man auf der linken Seite das parserbasierte System erkennen. Der Benutzer sieht die konkrete Syntax und arbeitet auch mit dieser.

Projektionale Systeme:

Im projektionalen Ansatz wird der abstrakte Syntaxbaum direkt über Aktionen und Eingaben im Editor erstellt. Die konkrete Syntax wird vom AST mit Projektionsregeln erstellt. Während ein Benutzer das Programm editiert, wird der abstrakte Syntaxbaum direkt modifiziert. Eine Projektionsmaschine erstellt eine Repräsentation des AST. Mithilfe dieser Repräsentation kann der Benutzer interagieren und Änderungen sehen. Die In-

stantiierung eines Objektes des ASTs erfolgt über die Auswahl eines Sprachkonzeptes über ein Auswahlménú bei der automatischen Codevervollständigung. Die erstellte Instanz wird in den AST eingelagert und mit einer eindeutigen Identifikationsnummer (UID) versehen. Anhand dieser UID werden Referenzen zwischen Programmelementen über Pointer hergestellt [5, S.188f].

In Abbildung 12 ist auf der rechten Seite das Verfahren beim projektionalen System dargestellt. Der Benutzer sieht eine Repräsentation des AST in Form von konkreter Syntax, arbeitet aber direkt auf dem AST.

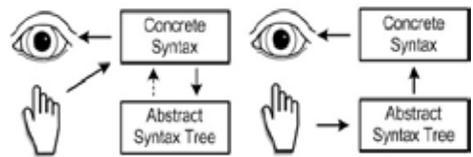


Abbildung 12: Parserbasiertes und projektionales System [5]

5.1.1 Modularität in MPS

Das Meta Programming System (MPS) von JetBrains ist ein Beispiel für eine projektionale Language Workbench. Es kommt also kein Parser zum Einsatz. MPS benutzt einen generativen Einsatz. Sprachen können erstellt und die Generatoren für sie geschrieben werden. MPS unterstützt dabei vor allem die Erzeugung von Java-Code, kann aber auch andere Sprachen generieren. Es gibt die Möglichkeit Text zu generieren, der selbst definiert wurde. Eigene Generatoren für weitere bisher noch nicht unterstützte Sprachen können geschrieben werden. MPS besitzt eine BaseLanguage. Diese ist eine Implementierung von Java. Bei der Erstellung neuer DSLs wird erst eine abstrakte Syntax definiert, dann der Editor spezifiziert und zuletzt der Generator definiert [13]. Bei der Modularisierung unterstützt MPS die *Referenzierung*, die *Wiederverwendung (Reuse)*, die *Erweiterung* und die *Einbettung*. Bei

der Wiederverwendung müssen die Sprachen jedoch bereits bei der Entwicklung so konzipiert werden, dass eine Wiederverwendung möglich ist. Die Erweiterung in MPS ist besonders einfach, da dafür zum Beispiel die BaseLanguage erweitert werden kann. MPS bietet bereits viele Erweiterungen ihrer BaseLanguage [5, S.394ff].

5.1.2 Modularität in Xtext

Xtext ist eine parserbasierte Language Workbench und ist eine Open Source Anwendung, die auf dem Eclipse Modeling Framework basiert. Die komplette Infrastruktur der Sprache, Parser, Linker, Compiler und Interpreter können umgesetzt werden [14]. Bezogen auf Modularität unterstützt Xtext die *Referenzierung*, die *Wiederverwendung (Reuse)* und die *Erweiterung*. Die *Einbettung* wird von Xtext nicht unterstützt, da die Adaptersprache von zwei BaseLanguages erben müsste. Xtext unterstützt aber nur die Erweiterung von einer Basisgrammatik [5, S.414ff].

5.1.3 Modularität in Spoofox

Die Spoofox Language Workbench ist eine Plattform für die Entwicklung von textuellen parserbasierten domänenspezifischen Sprachen und basiert auf dem Eclipse Framework. Die Spoofox-Umgebung unterstützt eine agile Entwicklung von Sprachen, indem sie die inkrementelle und iterative Entwicklung erlaubt und geeignete Editoren während der Entwicklung bereitstellt, mit denen unter anderem die abstrakte Syntax dargestellt werden kann. Spoofox benutzt die modulare Syntaxdefinition SDF, die stark modular sind [15]. Im Gegensatz zu Xtext benutzt Spoofox einen Parser ohne Scanner (scannerless parsing). Wie bereits erwähnt sind die Sprachdefinitionen in Spoofox typischerweise modularisiert. Bezogen auf Modularität unterstützt Spoofox die *Referenzierung*, die *Wiederverwendung (Reuse)*, die *Einbettung* und die *Erweiterung*. Die *Erweiterung* lässt sich in Spoofox wegen seiner modularen Struktur über Import der bestehenden Module realisieren [5, S.428ff].

6 Fazit

Domänenspezifische Sprachen können bei richtigem Einsatz und mit guten Grundkenntnissen deutliche Vorteile in der Softwareentwicklung bringen. Durch den Einsatz von Sprachbaukästen können die Kosten in der Entwicklung gering gehalten werden. Der Einsatz von DSLs lohnt sich aber erst ab einer bestimmten Projektgröße oder in Prozessen, die sich ständig wiederholen. Das Einführen der modularen Entwicklung ermöglicht zusätzlich die Wiederverwendung und Komposition der Sprachen und senkt die Kosten weiter. Die Entwicklung von Language Workbenches wie MPS oder Xtext wird meines Erachtens weiter gehen und stark an Bedeutung zunehmen. Vor allem die Möglichkeit der Komposition und Spracherweiterung wird den Einsatz dieser Sprachbaukästen in der Praxis fördern.

Literatur

- [1] M. P. Ward. Language oriented programming, 1994.
- [2] Martin Fowler. Language workbenches: The killer-app for domain specific languages? <http://martinfowler.com/articles/languageWorkbench.html>, letzter Aufruf am 06.02.2014, 2014.
- [3] JetBrains Sergey Dmitriev. Language oriented programming: The next programming paradigm, 2004.
- [4] Anneke G. Kleppe. *Software language engineering: Creating domain-specific languages using metamodels*. Addison-Wesley, Upper Saddle River and NJ, 2009.
- [5] Markus Voelter. *Dsl engineering. designing, implementing and using domain-specific languages*. <http://voelter.de/dslbook/markusvoelter-dslengineering-1.0.pdf>, letzter Aufruf am 21.02.2014, 2013.
- [6] Holger Krahn Steven Völkel Bernhard Rumpel. *Mit sprachbaukasten*

zur schnelleren softwareentwicklung:
Domaenenspezifische sprachen modular
entwickeln, 12.08.2013.

- [7] Turhan Özgür. *Domain-Specific Modeling: A practical approach: A comparison of Microsoft DSL Tools and Eclipse Modeling Framework in the context of Model-Driven Development*. Lambert Academic Publishing, Köln, 2009.
- [8] Thomas Stahl. *Modellgetriebene Softwareentwicklung: Techniken, Engineering, Management*. Dpunkt-Verl., Heidelberg, 2 edition, 2007.
- [9] Holger Krahn. *MontiCore: Agile Entwicklung von domänenspezifischen Sprachen im Software-Engineering*, volume 1 of *Aachener Informatik-Berichte, Software Engineering*. Shaker, Aachen, 2010.
- [10] Steven Kelly and Juha-Pekka Tolvanen. *Domain-specific modeling: Enabling full code generation*. Wiley-Interscience and IEEE Computer Society, Hoboken and N.J, 2008.
- [11] Markus Voelter. Language and ide modularization, extension and composition with mps. <http://voelter.de/data/pub/Voelter-GTTSE-MPS.pdf>, letzter Aufruf am 13.02.2014, 2011.
- [12] Markus Knecht Jürg Luthiger. Modulare domänenspezifische sprachen: 33 imvs fokus report 2012. <http://www.fhnw.ch/technik/imvs/publikationen/artikel-2012>, letzter Aufruf am 21.02.2014, 2012.
- [13] JetBrains, Inc, and <http://www.jetbrains.com>. JetBrains :: Meta programming system — language oriented programming environment and dsl creation tool. <http://www.jetbrains.com/mps/>, letzter Aufruf am 06.02.2014.
- [14] Sven Efftinge. Xtext - language development. <http://www.eclipse.org/Xtext/>, letzter Aufruf am 06.02.2014, 2014.
- [15] William R. Cook, Martin Rinard, Siobhán Clarke, Markus Völter, and Eelco Visser. Language extension and composition with language workbenches. page 301.

Strategy to Test Mobile Apps

Olaya de la Rosa Avitia
Reutlingen University
Olaya.de_la_Rosa_Avitia@Student.
Reutlingen-University.DE

Abstract

Nowadays the development of a mobile app implies challenges and difficulties, which have to be faced by mobile app developers. Innovations lead to a rapidly evolving mobile app market, therefore apps should be developed faster and offered in short release cycles to the market. Testing is a decisive activity within the development process that helps to improve the quality of the app. This research paper describes a strategy to test mobile apps that overcomes the challenges that mobile apps confront and permits to test the app in a structural test environment.

Keywords

Mobile Software, Test Strategy, Software Testing, Mobile App, Mobile Devices, Mobile Computing, Cloud Computing.

CR-Categories

D.2.5 [Software Engineering]: Testing and Debugging—*Testing tools*. K.6.3 [Software Management]: Software process

Betreuer Hochschule: Prof. Dr.- Ing. habil. Natividad
Martínez Madrid
Hochschule Reutlingen
Natividad.Martinez@Reutlingen-
University.de
Betreuer Firma: Carlos Orozco
Robert Bosch GmbH
Carlos.Orozco@de.bosch.com

Informatics Inside 2014
Wissenschaftliche Vertiefungskonferenz 2014
30. April 2014, Hochschule Reutlingen
Copyright 2014 Olaya de la Rosa Avitia

1 Introduction

As predicted by analysts, the number of users that access the Internet on mobile devices has exceeded the users on desktops. Users are changing desktops for mobile devices such as laptops, smartphones, tablets, glasses and smart watches [1]. As a consequence of this tendency companies are modifying the business models and creating a mobile strategy to react properly to those changes. This strategy includes often the adaptation of websites to be shown adequately on mobile devices, development of mobile apps, launching of mobile campaigns and location-based services.

This mobile business strategy is different to the traditional. Nguyen [2] mentions that because of the unique characteristics of mobile systems, mobile applications are usually more complicated to develop than non-mobile applications. This complexity represents a challenge for mobile developers, especially to those who develop for Android, which is highly fragmented: Hundreds of mobile devices exist, the hardware and screen resolutions are different and the components in the devices change constantly. Therefore as Nguyen mentioned, testing if a mobile app is going to work adequately in each device is a complex task.

Because of the different characteristics of mobile devices, specific testing methodologies and concepts are also necessary to evaluate the functionality of a mobile app [11].

But for this evaluation there are other options to test apps without the necessity of having each device. As Ridene [3] mentioned, acquiring hundreds of mobile devices to implement repetitive and large-scale test activities in an industrial way causes problems in terms of effort, quality assurance and cost.

This paper provides answers to the following questions:

- Which are the attributes that characterize mobile computing and differentiate it from traditional computing?
- How do the architecture and usage context of an app influence the test cases?
- What are the fundamental challenges on testing mobile apps?
- Which tools and concepts can be used to efficiently test an app?
- When is it necessary and reasonable to automate the tests?

This research paper is structured as follows: Section 2 introduces the mobile app testing topic taking in consideration the app background based on the context of use and the architecture. In section 3 the challenges that mobile app developers confront are described. The test execution process, with the testing types and automation testing are addressed in section 4. The strategy to test mobile apps is presented in section 5 and finally the conclusion of the research paper is shown in section 6.

2 Mobile apps testing

Haller describes in [1] that the process to develop mobile apps is not suitable to the classical waterfall development approach: Start of the project with the definition of specifications and requirements that the software has to satisfy, after that the implementation of these specifications by the developers. Then testing what was implemented in comparison with the specifications and finally if the tests are successful, the customer pays.

During the testing phase, the app has to be tested by developers and testers to check that it works correctly and fulfills the requirements. But according to Haller, from a business point of view the success of the app depends also on the answers to:

- Is it what the users expect?
- Are the offerings of the competition good?
- Does the app boost the reputation of a company? (Figure 1)



Figure 1: App quality and success in the mobile world [1]

These three questions help to recognize how the quality of an app is perceived by the users, therefore testing has to be more than the definition of test cases and their corresponding repetitive implementation [1]. Haller [1] presents a series of points that a tester has to know before the test cases are determined:

- **Context of use:** In which group is the app user? (repetitive now, bored now or urgent now)
- **Architecture:** Is it a native client/server app, native standalone app, mobile web app or a launcher app?
- **User base:** Is the app for an open market or only for a company?

After the information is available, the test type will be applied and the development stage in which the solutions will be used, can be defined. Other factors and influences are: Timeline of the project, complexity, cost/benefit analysis and the existing infrastructure. The following subsections

describe the first two aspects that Haller considers are important to define the test cases.

2.1 Context of use

The context of use is the environment and actual conditions in which an app is used and it provides a basis for the later design of the test cases. First, it is necessary to understand which mobile user group will use the app based in Leland Rechis' classification [4]:

- **Repetitive now:** Is the group of users that use a mobile device to do the same over and over. This could be checking for new emails, looking for the weather forecast or reading the news.
- **Urgent now:** Are the users who need information immediately, for example a train timetable.
- **Bored now:** These users are for example waiting for someone or sitting in the train and have time to spend.

Haller [1] states that once the usage group was identified, the context of use of an app can be defined answering three questions:

- What is the app type? The app can be interactive (for example a game) where the visuals and graphics are important or a no-interactive app like online shops or phone directories with a straight-forward design.
- How long is the app going to be used? It can be hours (playing a game), minutes (consulting a map) or just seconds (phone book).
- How frequent does the mobile device interact with back-end servers? An app can require data download and therefore needs a stable server connection.

For example an app that shows to the user when a hop-on hop-off tourist bus will arrive is within the user group urgent now. The answers to the three questions presented before are: The app is non-interactive, it is used only for a couple of minutes and does not have to interact with back-end servers since the

information is downloaded together with the app and stays static (delays in the tourist bus are not shown). This is the description of the environment or scenario in which the app is used.

2.2 Architecture

If the developers and testers understand the architecture of the app, it will be easy to identify which part can fail and hence where it is necessary to implement tests. In the mobile world, there are four main architectural patterns [1]:

- **Native standalone apps:** These are apps that are preinstalled on a mobile device and don't have to communicate with a server, for example the calculator or the alarm.
- **Mobile web apps:** These are web pages optimized to be shown and run code in a web browser on mobile devices.
- **Native client server (C/S) apps:** Are native apps that were downloaded and installed by users. The app retrieves data from a back-end server, for example a mobile banking app.
- **Launcher app:** Stand between native C/S apps and mobile web apps: The app is downloaded and installed on the device by the users and provides some features and/or can redirect to a mobile web app.

3 Challenges in testing mobile apps

In section one, the Android fragmentation was mentioned. This represents a problem for the developers and is at the same time an advantage for users [5] as users like to have a large pool of devices they can choose from. Only in 2013 about 12000 device types were registered in OpenSignal¹, a company specialized in mapping the mobile internet.

Despite of the slogan "write once, run everywhere" (W.O.R.E.) created by Sun Microsystems to define the benefits of Java,

¹<http://www.macerkopf.de/2013/07/31/android-fragmentation/>

the practice shows that adjustments are necessary to run an app on a different device that responds in the same way and has a consistent behavior [3].

Mobility is characterized and differentiated by the following constraints: Mobile elements are resource-poor, mobile devices are more vulnerable, and performance of connectivity is highly variable with a limited energy source. Since the mobile information systems are different to the traditional market, users need apps with an intuitive navigation, easy to use, with concise text, efficient and suitable for activities on the road [6].

The challenges that developers have to deal with taking in consideration the characteristics and differences of mobile apps arise according to Baride and Dutta [8] from four main areas:

1.- Diversity of the device environment:

Different devices make the reusability and maintenance of test cases more difficult [9]. One system targets multiple devices and all the devices have to be considered for the tests, which might be difficult to carry out and expensive. Tablets using the same app with the same environment may have different responses caused by sensors that are calibrated differently for each device.

2.- Hardware configuration and network-related changes:

The devices have to interact with a dynamic environment: Connectivity, bandwidth and networking considerations [10] as well as wireless signals [9]. Components are upgraded and the configuration changes constantly, how does a testing environment which can tackle these challenges have to look like?

3.- Rapid application development (RAD) methodologies:

The short innovation cycles for hardware and platforms generate the necessity to reduce the time taken for the development and testing cycle without compromising coverage and the quality of the app [1].

4.- Device limitations: Some restrictions are mentioned below, although many device

limitations could affect more a smartphone than a tablet:

- Processing ability
- Memory capacity
- Communication ability [9]

4 Testing methodology

For the development of high quality software products a well-defined software development process is necessary [5]. Testing helps to identify if the requirements defined at the beginning are fulfilled in the end product and it reduces the possibility that the software reacts undesirably. It saves costs and failure in the field.

Baride defines in [8] the factors that determine if a mobile testing program is successful. These are: Complexity of the app, testing for a mobile environment and the use of test automation, emulator and actual devices.

Mobile apps have to go through the following testing methodology: First the definition of the test strategy where the test plan with its approach, risks, contingencies, recommendations and resource requirements (schedule, tools, roles and responsibilities) are described. Then the specification of the test design contains features that have to be tested, scenarios in which the app is used and the acceptance and release criteria. After that the test cases are defined with the test actions, input data, execution conditions and expected results. Finally the summary report provides the information covered by the tests and the accomplished results after the test execution.

The following sections describe the different ways, in which a test can be executed, what the types of testing are and when it makes sense to implement the testing automation.

4.1 Test execution

There are different ways and solutions to carry out tests of mobile apps. These will be described in the following subsections.

4.1.1 Crowd-sourced mobile app testing

It consists of a community of common mobile users that get money for testing an app. The disadvantage here is that the testers are not professional, which represents difficulties and misunderstandings when the non-technical feedbacks have to be interpreted [3].

4.1.2 Emulators

An emulator simulates the functions of a mobile device on a laptop or PC, allowing to check the app during design and development². Emulators are designed for a specific platform and can even simulate a camera or incoming text messages [1]. Nevertheless emulators don't emulate all devices, network availability and speed, cache and stack size, memory limitations and the speed of content rendering on a specific device [8].

4.1.3 Local device

It consists of testing the app manually on the target mobile device [12] to check that at runtime in the user's hand the software has the expected behavior [3]. Usually the testers own the device and it might or might not be connected to the PC. The tester can also use tool support like eye tracking, recording software or automated tests e.g. Experitest³ [1]. Using adequate testing techniques maximizes the coverage of device diversity without the need of acquiring each device [13].

4.1.4 Private device cloud

It is a company-internal, centralized pool with devices connected to web servers and network operators to analyze how an app behaves with events like SMS, calls and low battery. The user experience and audio outputs can be captured [3]. The device can be selected and used as a local device; the keys are electrically wired so that a tester can realize inputs (screen

touch, unplug charger or USB) using a PC [14] [3]. The benefits of having a private device cloud are that a team takes care of device issues and has control over the infrastructure. Also if a device doesn't work, testers can switch to another one.

4.1.5 Public device clouds

The difference with the private device clouds is that for public device clouds the access is offered by service providers through the Internet. One of the disadvantages is the integration of the pool devices into an already existing test infrastructure [1]. For example Samsung offers a Remote Test Lab (RTL⁴), which allows testing apps on different Samsung devices for a certain number of hours per day for free.

4.2 Types of tests

Jayamalrao [15] presents in figure 2 a cycle to test mobile apps:



Figure 2: Mobile testing types [15]

- 1.- **Sanity test:** Proves the correct functionality of the app. It is the base for further tests.
- 2.- **Functional/UI:** Evaluates if the app is working as expected with the documented

² Using the android emulator:
<http://developer.android.com/tools/devices/emulator.html>

³ <http://experitest.com/>

⁴ Samsung RTL
<http://developer.samsung.com/remotetestlab/rtlDeviceList.action>

requirements. Test the text, logos, and usability aspects on the UI level.

3.- Interruption testing: Checks if the app reacts as expected during interruptions like incoming calls, low battery, zero network, etc.

4.- Regression testing: To ensure that changes and new features don't affect the basic functionality of the app.

5.- Stress/Performance: Analyses which is the response time and performance while navigating between different screens. Determines if the app is secure in terms of session handling, data transmission, storage, etc.

6.- Localization testing: Implementation of the same tests in a local language with translation of the interfaces, documentation and support.

7.- Field testing: Are explorative tests which have the purpose to test an app in the usage context and real world in which users will use the app. For example a hiking app must be carried to the top of mountains and valleys to test the correct functionality [1].

4.3 Test automation

There are different reasons why mobile test cases should be automated: Many test cases have to be executed many times and frequently using different devices to detect bugs early. In such cases, it is not possible for testers to keep focus and execute the test cases for a long period. Therefore automation plays a primordial role here [1]. Repeated testing can reduce testing effort, costs and workload of developers and testers [6], these also accelerate and facilitate the development and testing process. Test scripts cover the basic functionalities of the app and run on a predetermined time and collect useful data. With automated compatibility tests, testers can be sure that an app runs on relevant devices [1] and test scenarios can be adapted or reused for similar apps [3].

The use of automated tests makes more sense for non-interactive apps than for interactive apps since it is more difficult to put the physical interaction in a script, for example in sport simulation apps [1].

An open-source tool for Android apps testing is Robotium⁵ that permits to write and execute automated tests of activities, menus, dialogs and context menus for Android [16]. In Lesspainful⁶ it is also possible to implement automated app testing on its device cloud.

5 Strategy to test mobile apps

In the previous sections, the areas involved in mobile app testing were described. What has to be tested can be defined by a test strategy for mobile testing.

Figure 3 shows the strategy to follow when testing mobile applications. As an example the app Bosch Events⁷ will be used. This app is used to view information of an event at Bosch (schedule, location, speaker information, exhibitors), to take photos of the event and upload them, see who is going to attend the event, send feedback to the speakers and see likes, comments and trends of the attending people. This is a native mobile app which is used by the user group repetitive now. The platform where the app is used is iOS (5.1 and later) and Android (version 2.3.3 and up).

For the second phase (test strategy) the suitable test tool to use in the first phases of testing is an emulator since it facilitates and speeds up the early resolution of problems. Emulators also cover GUI aspects like the size of the screen. For testing physical interactions and network effects, local devices will be used. The selected devices represent those who are used more like the iPhone and Samsung Galaxy. In this case only the device and emulator approach will be used.

In the test design phase the test scenarios and test cases are defined. For this a table can be

⁵ <http://code.google.com/p/robotium/>

⁶ <https://www.lesspainful.com/>

⁷ <https://itunes.apple.com/app/id726491006>

used where all the information can be systematically tested (Table 1):

Table 1: Sample test scenario

Requirement	Test scenario	Test cases
Login	OK Cancel	1. Verify ok with valid data 2. Verify ok with invalid data 3. Verify cancel
Search	Search Cancel	1. Verify Search with valid data 2. Verify Search with invalid data 3. Verify Cancel
Add comment	OK Cancel	1. Verify OK after entering a string 2. Verify OK with invalid data 3. Verify Cancel

In the test execution, the different test cases are tested first with an emulator and then with local devices. If potential defects are detected, they are analyzed and prioritized. Then changes are made to fix the defects which are tested again to check that these were fixed. Finally a detailed report with the results of the test process is written.

6 Conclusion

The objective of this research paper was to describe a strategy to test mobile apps and to provide a basis knowledge that helps with the implementation of this test. It also offered answers to the questions in section 1.

This paper described first in which context scenarios an app can be used and the different architectures for apps. After this, the inconveniences imposed by the different challenges related to testing mobile apps and the peculiarities about mobile testing were discussed. Diverse device specific challenges were: The device fragmentation syndrome, rapid application development (RAD) methodologies, hardware configuration and network-related changes.

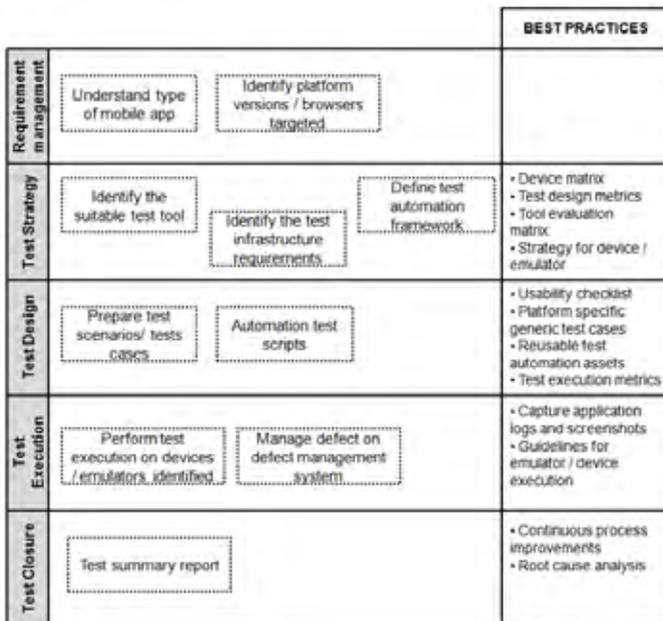


Figure 3: Mobile Testing – Process and Methodology [15]

The various solutions to test mobile apps were described and the types of tests as well. The topic test automation was approached and the advantages of automating test cases were mentioned.

Finally a strategy to test mobile apps was presented. This strategy aims to set a reference structure within the development process that identifies which solutions can be applied in each phase of the software process.

The testing process doesn't end with the release and publish of an app on the market. The app has to be continuously tested on new devices, new hardware conditions and OS versions. The market changes steadily and offers actually a great variety of testing tools like Robotium or Lesspainful. However testing still has many limitations, for example some tests cases depend on human sense, like color matching. These tests can't be implemented in automated tests and the results aren't easily quantized [6].

Therefore, mobile apps testing still needs improvements and solutions that enable cost-effective testing. But as Haller [1] writes: "*An app's market triumph requires more: Understanding the user, design and usability, and creativity and innovation*", not only in defining test cases.

7 References

- [1] K. Haller. Mobile Testing. Swisscom IT Services. ACM SIGSOFT Software Engineering Notes. November 2013 Volume 38. Online available in: <http://doi.acm.org/10.1145/2532780.2532813>. Accessed on: March 7th, 2014
- [2] M. D. Nguyen, H. Waeselync and N. Riviere. Testing Mobile Computing Applications: Toward a Scenario Language and Tools. Workshop on Dynamic Analysis, July 21th, 2008. ACM 978-1-60558-054-8/08/07
- [3] Y. Ridene and F. Barbier. A Model-Driven Approach for Automating Mobile Applications Testing. ECISA '11 Workshop SAVA '11. September 13, 2011. Essen, Germany. ISBN 978-1-4503-0618-8/11/09
- [4] S. Wellman. Google lays out its mobile user experience strategy. Information Week April 11 2007. Online available in: <http://www.uxmatters.com/mt/archives/2011/05/three-layers-of-mobile-user-experience.php#sthash.214qTSg3.dpuf> accessed on: March 4th, 2014
- [5] M. E. Delamaro, A. M. R. Vincenzi and J. C. Maldonado. A Strategy to Perform Coverage Testing of Mobile Applications. International Workshop on Automation of Software Test (AST '06) May 23th, 2006 Shanghai, China.
- [6] C. Wuthrich, G. Kalbfleisch, T. Griffin, and N. Passos. On-line Instructional Testing in a Mobile Environment. JCSC 18, 4. Midwestern State University. April 2003.
- [7] M. Satyanarayanan. Fundamental Challenges in Mobile Computing. Proceedings of the fifteenth annual ACM symposium on Principles of distributed computing, ser. PODC '96. New York, NY, USA: ACM, 1996, pp. 1–7. Online available in: <http://doi.acm.org/10.1145/248052.248053>, accessed on: March 3th, 2014
- [8] S. Baride and K. Dutta. A Cloud Based Software Testing Paradigm for Mobile Applications. ACM SIGSOFT Software Engineering Notes. May 2011 Vol. 36 Nr. 3
- [9] J. Bo, L. Xiang, G. Xiaopeng: MobileTest: A Tool Supporting Automatic Black Box Test for Software on Smart Mobile Devices. 2nd International Workshop on Automation of Software Test (AST'07), Minneapolis, MN, 20–26 May 2007
- [10] Y. Dubinsky and A. Abadi. Challenges and Research Question for Testing in Mobile Development. MobileDeli '13, October 28, 2013 Indianapolis, Indiana USA.

- [11] Q. H. Mahmoud. Testing wireless java applications. Sun Microsystems, on-line article, Nov. 2002. Online available at: <http://developers.sun.com/techttopics/mobility/midp/articles/test/>, accessed on: December 10th, 2005
- [12] I. Satoh. A testing framework for mobile computing software. IEEE Transactions on Software Engineering, 29(12):1112–1121, December 2003.
- [13] H. Muccini, A. Di Francesco, P. Esposito: *Software Testing of Mobile Applications: Challenges and Future Research Directions*, 7th International Workshop on Automation of Software Test (AST '12), June 2–3, 2012, Zurich, Switzerland
- [14] K. Heller. Mobile Testing. Swisscom IT Services. ACM SIGSOFT Software Engineering Notes November 2013 Volume 38. Online available in: <http://dl.acm.org/citation.cfm?doid=2532780.2532813>, accessed on: March 3th, 2014
- [15] G. Jayamalrao. Cognizant 2011. Online available in: <http://www.qaac.org/wp-content/uploads/2012/07/Mobile-Testing-QAAC.pdf>, accessed on: March 5th, 2014.
- [16] S. Diewald, L. Roalter, A. Möller and Matthias Kranz. Technische Universität - München. MUM '11, Dec 7-9 2011, Beijing China. ACM 978-1-4503-1096-3/11/12.
- [17] L. Zhifang, L. Bin, G. Xiaopeng: Test Automation on Mobile Device, AST'10, May 3-4 2010, Cape Town, South Africa
- [18] S. Waterson, J. A. Landay and Tara Matthews. In the lab and out in the wild: remote web usability testing for mobile devices. Conference on Human Factors in Computing Systems, 2002, pp. 296–297.

Optimierung der Usability von digitalen Fahrtenbüchern durch automatisches Erfassen von fahrzeugspezifischen Daten

Viktoria Hoffmann
Reutlingen University
Viktoria.Hoffmann@Student.
Reutlingen-University.DE

Abstract

Mit dem starken Wachstum des CarSharing-Angebots und der großen Menge an Flottenfahrzeugen in Unternehmen nimmt auch die Anzahl der Fahrtenbuch-Apps zu. Bei den meisten mobilen Fahrtenbuch-Anwendungen muss der Benutzer den Kilometerstand manuell eintragen. Dies wirkt sich negativ auf die Usability und die User Experience aus. Hinzu kommt, dass jede Minute kostbar ist, die der Fahrer im ausgeliehenen Auto verbringt. Aus diesen Gründen wird hier eine Lösung vorgestellt, bei der der Kilometerstand aus einer Mercedes Benz A-Klasse über den OBD-Anschluss mit Hilfe des CAN Interfaces „ISI b2air“ automatisch ausgelesen und per Bluetooth an die Fahrtenbuch-App der Berger Elektronik GmbH gesendet wird. Hierfür wird mittels der Software „ISI b2app“ die Kommunikation des Diagnostesters mit dem Fahrzeug aufgezeichnet. Anschließend werden die CAN-Botschaften analysiert und in Bezug auf den Kilometerstand gefiltert. Die

entsprechende Anfrage zum Erhalt des Kilometerstandes wird in den Programmcode des Berger Fahrtenbuches implementiert, so dass die App selbstständig den Kilometerstand auslesen kann.

Schlüsselwörter

Usability, User Experience, mobile App, Fahrtenbuch, Automobilindustrie, Fahrzeugtechnik, OBD, CAN, Bussystem, Diagnose, Steuergerät, Kilometerstand auslesen

CR-Kategorien

C.3 [Special-purpose and application-based systems]: *Real-time and embedded systems*;
J.7 [Computers in other systems]: *Real time*: H.5.2 [User Interfaces]: *User-centered design*

1 Einleitung

Zurzeit werden von vielen Firmen Fahrtenbücher für Android- und iOS-Smartphones angeboten. Beispiele hierfür sind die Fahrtenbücher von Stefan Meyer¹, Krämer IT² oder Ambertation³.

Betreuer Hochschule: Prof. Dr.-Ing. Natividad Martínez
Madrid
Hochschule Reutlingen
Natividad.martinez@Reutlingen-
University.de

Betreuer Firma:
Dipl.-Ing. Jochen Retter
Berger Elektronik GmbH
Jochen.Retter@bergerelektronik.de

Informatics Inside 2014
Wissenschaftliche Vertiefungskonferenz
30. April 2014, Hochschule Reutlingen
Copyright 2014 Viktoria Hoffmann

¹<https://itunes.apple.com/de/app/fahrtenbuch/id286070473?mt=8>

²<http://www.androidpit.de/de/android/market/apps/app/de.kraemerit.kfzmobile4/KFZ-Fahrtenbuch-Mobile>

³<https://itunes.apple.com/de/app/drivers-fahrtenbuch/id292648874?mt=8>

Mit digitalen Fahrtenbüchern lassen sich Fahrten mit detaillierten Angaben zum Fahrer, Fahrzeug, Fahrtzweck, Kilometerstand, Start und Ziel der Fahrt sowie Zeit und Datum erfassen. Außerdem verfügen viele Fahrtenbücher über eine Backup- und eine Synchronisationsfunktion, um die Fahrten bequem vom PC aus zu verwalten. Fahrtenbücher werden überwiegend in Flottenfahrzeugen und bei CarSharing-Unternehmen für die Dokumentation der Fahrten eingesetzt. Der aktuelle Standort des Fahrzeuges wird bei den meisten Fahrtenbüchern über das GPS des Smartphones automatisch erkannt.

Die Firma TomTom Business Solutions bietet in ihrer Flottenmanagementlösung WORKsmart™ unter anderem auch einen Fahrtenbuchmodus an⁴. Dieser kann bspw. über ein TomTom-Navigationsgerät genutzt werden. Mit Hilfe einer mit dem Fahrzeug verbundenen Blackbox können dann Echtzeitdaten wie Zeit, Ort, Geschwindigkeit und Kilometerstand direkt an das Fahrtenbuch übermittelt werden. Die Anschaffung aller dafür benötigten Soft- und Hardwarekomponenten liegt etwa zwischen 300,- und 400,- €⁵. Die mobile Anwendung eines Fahrtenbuches mit denselben Funktionen ist im Vergleich dazu mit maximal 25,- € deutlich günstiger. Jedoch existiert bislang kein Fahrtenbuch für Smartphones, das direkt mit dem Fahrzeug kommuniziert. Dabei bringt die direkte Kopplung an das Fahrzeug viele Vorteile mit sich. Angaben zur aktuellen Fahrgeschwindigkeit, zum Tachostand, Benzinverbrauch oder Tankfüllung können auf diese Weise automatisiert vom Smartphone ausgelesen werden und das Fahrtenbuch dadurch effektiver und

effizienter gestalten. Der Benutzer müsste diese Informationen dann nicht mehr manuell eintragen und würde sich dadurch Zeit – und Geld – beim Bedienen des Fahrtenbuches sparen.

Am Beispiel des Fahrtenbuches der Berger Elektronik GmbH soll gezeigt werden, wie der Kilometerstand eines Fahrzeuges automatisch ausgelesen werden kann. Hierbei werden zwei Komponenten analysiert: zum einen die standardisierte OBD2-Schnittstelle, die den Zugang zu den Steuergeräten gewährt, und zum anderen das Bussystem des Fahrzeuges, das den Datenaustausch zwischen den Steuergeräten regelt. Es werden verschiedene Soft- und Hardwarekomponenten vorgestellt, die dabei helfen, die Buskommunikation aufzuzeichnen, um in Erfahrung zu bringen, mit welcher Botschaft der Kilometerstand angefragt werden kann. Letztendlich soll das Ziel erreicht werden, dass das Berger Fahrtenbuch selbstständig den Kilometerstand ausliest, damit der Benutzer noch effizienter mit dem Fahrtenbuch arbeiten kann.

2 Das Berger Fahrtenbuch

Die Fahrtenbuch-App von der Berger Elektronik GmbH (im weiteren Verlauf Berger Fahrtenbuch genannt, siehe Abbildung 1) ist sowohl für Android- als auch für iOS-Smartphones geeignet.



Abbildung 1: Fahrtenbuch-App der Berger Elektronik GmbH

⁴ Vgl.

<https://de.support.business.tomtom.com/ci/fatch/get/386920/1386661492/redirect/1/session/L2F2LzEvdGltZS8xMzk1MzE2OTM5L3NpZC9sN1NQcElQbA==/filename/tb-exploreworksmart-int.pdf> S.36

⁵http://business.tomtom.com/de_de/products/ink/510/highlights/

In der App werden alle relevanten Daten zu den getätigten Fahrten mit einem Fahrzeug erfasst. Dabei handelt es sich um folgende Informationen:

- Fahrer
- Autokennzeichen
- Zeit und Datum
- Start und Ziel der Fahrt
- Kilometerstand
- Fahrtzweck (privat oder geschäftlich, Name der/des Person/Kunden)

3 Fahrtenbuch-Usability

Das Design von mobilen Apps sollte den ISO Normen und Standards für die Gestaltung von interaktiven Systemen entsprechen.

Beim Berger Fahrtenbuch werden einige Daten automatisch von der App erfasst und in die entsprechenden Felder eingetragen. Dabei handelt es sich zum einen um die Zeit und das Datum, die dem Smartphone bekannt sind und zum anderen um den genauen Stand- und Zielort der Fahrt, der jeweils über das GPS-Modul des Smartphones erkannt wird. Allerdings bleiben Daten übrig, die der Anwender selber eintragen muss:

- Fahrer
- Autokennzeichen
- Kilometerstand
- Fahrtzweck (privat oder geschäftlich, Name der/des Person/Kunden)

Den Namen des Fahrers kann der Benutzer entweder über eine hinterlegte Liste auswählen oder selber eintragen. Dasselbe gilt bislang auch für den Fahrtzweck und das Autokennzeichen. Nur den Kilometerstand zu Beginn und Ende einer Fahrt muss der Anwender bisher vollständig manuell eintragen.

Grundsätzlich ist das Eintragen oder Korrigieren vieler Daten von Hand, wie in diesem Fall, für den Benutzer sehr mühsam und zeitaufwendig. Nach Möglichkeit sollte dem Anwender soviel Arbeit wie möglich abgenommen werden. Hierfür muss der Dialog

bzw. das interaktive System nach EN ISO 9241-110 (Grundsätze der Dialoggestaltung) gestaltet sein. [1] Speziell das Kriterium der Aufgabenangemessenheit sollte erfüllt sein:

„Ein Dialog ist aufgabenangemessen, wenn er den Benutzer unterstützt, seine Arbeitsaufgabe effektiv und effizient zu erledigen.“ [2]

Bezogen auf das Berger Fahrtenbuch, sollten daher möglichst viele Informationen automatisch von der App eingetragen werden. Da es sich hierbei um fahrzeugspezifische Daten handelt und das Fahrtenbuch während einer Autofahrt verwendet wird, bietet es sich an, den aktuellen Kilometerstand über die Schnittstelle des Fahrzeugs auszulesen und diesen an die App zu senden.

4 Methodik

Um den Kilometerstand aus einem benzin- oder dieselpetriebenen Kraftfahrzeug mit einem Smartphone auslesen zu können, bedarf es einer genauen Analyse der Elektronik im Inneren des Fahrzeugs. Die Steuergeräte (SG) und Bussysteme eines Fahrzeugs kommunizieren durch unterschiedliche – teilweise herstellerspezifische – Protokolle miteinander. Der Kilometerstand ist üblicherweise in allen größeren Steuergeräten gespeichert, z.B. im Motorsteuergerät. Mit einem Diagnosetester, wie er in Werkstätten verwendet wird, kann eine fahrzeugspezifische Diagnose über den genormten On-Board-Diagnose-Stecker (OBD) durchgeführt werden, unter anderem kann dabei auch der Kilometerstand ausgelesen werden. Die Funktion existiert also bereits für Diagnosetester, jedoch nicht für Fahrtenbuch-Apps. Zudem ist der Programmcode des Diagnosetesters durch den Hersteller unzugänglich. Um seine Funktionalität in die Fahrtenbuch-App zu implementieren, gilt es herauszufinden, in welcher Nachricht der Kilometerstand enthalten ist, wie dieser Wert angefordert wird und wie die empfangene Nachricht interpretiert werden muss.

Die Rekonstruktion der Strukturen, Zustände und Verhaltensweisen eines fertigen Systems bzw. das Nachempfinden der Funktionalität dieses Systems wird als „Reverse Engineering“ bezeichnet [3]. Für diesen Zweck soll

ein Versuchsaufbau erfolgen, bei dem die App „ISI b2app“ [4] der Berger Elektronik GmbH als Mithörer zwischen der Kommunikation des Diagnostestesters „mega macs 42 SE“ der Firma Hella Gutmann Solutions GmbH⁶ und dem Bussystem einer Mercedes Benz A-Klasse eingesetzt wird.

Die aufgezeichneten Nachrichten werden daraufhin bzgl. des Kilometerstandes analysiert. Die entsprechenden Botschaften zum Anfragen und Erhalten des Kilometerstandes werden im nächsten Schritt in den Programmcode des Berger Fahrtenbuches implementiert. Abschließend wird diese Funktionalität auf verschiedenen Android-Smartphones und Tablets bei entsprechenden Testfahrten in der Mercedes Benz A-Klasse überprüft.

5 Technische Grundlagen der Fahrzeugtechnik

Dieses Kapitel behandelt die interne Kommunikation zwischen den Bussystemen und Steuergeräten in einem modernen Fahrzeug. Dabei wird die Durchführung einer Diagnose beschrieben, der Aufbau eines CAN-Buses und vor allem wie die Nachrichten auf einem Bus gesendet werden.

5.1 On-Board-Diagnose-Stecker

Der On-Board-Diagnose-Stecker (OBD) ist die einzige genormte Hardwareschnittstelle bei Kraftfahrzeugen für externe Geräte. Über diesen 16-poligen OBD-Stecker, der sich üblicherweise im vorderen Fahrer- oder Beifahrerbereich des Fahrzeugs befindet⁷ (z.B. im Fußraum oder in der Mittelkonsole), lassen sich verschiedene Daten auslesen, Steuergeräte umkonfigurieren, Fehler diagnostizieren oder neue Software aufspielen.

⁶ <http://www.hella-gutmann.com/diagnose/mega-macs-42-se/uebersicht/>

⁷ <http://www.obd-2.de/obd-2-stecker-einbauorte.html>

Das Gateway im Inneren eines Fahrzeugs dient als Kopplung der verschiedenen Steuergeräte. Wenn vom Diagnostestester eine Anfrage gesendet wird, wird sie über die OBD2-Schnittstelle an das Gateway geschickt und von dort an das entsprechende SG weitergeleitet (siehe Abbildung 2). Im Prinzip arbeitet das Gateway wie ein Router. [5,S.14]

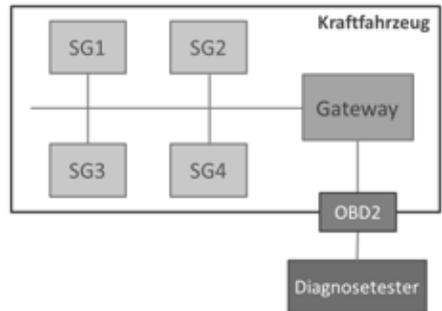


Abbildung 2: OBD2 als Schnittstelle zwischen Diagnostestester und Bussystemen im Kraftfahrzeug (eigene Darstellung i.A.a. [5,S.1])

5.2 Controller-Area-Network

Bussysteme werden als Verbindung zwischen den zahlreich verbauten Steuergeräten in einem Kraftfahrzeug eingesetzt. Das heute am meisten verwendete Kfz-Bussystem ist das Controller-Area-Network (CAN), das von Bosch entwickelt wurde und seit 1991 als internationaler Standard gilt. [5]

Bei der Datenkommunikation richten sich die Bussysteme an das standardisierte ISO/OSI Referenzmodell (siehe Tabelle 1). Der CAN-Bus spezifiziert die Layer 1 (Physical Layer) und 2 (Data Link Layer). Hierbei werden die Bitübertragung, der Botschaftsaufbau und der Buszugriff geregelt. CAN wird bspw. im Motorsteuergerät, im Getriebesteuergerät oder im ABS-Steuergerät verwendet. [5]

Für die gesamte Übertragung von CAN-Botschaften ist ein sog. CAN-Controller zuständig, der jeweils in den entsprechenden Steuergeräten verbaut ist. Eine Botschaft – auch „Frame“ genannt – wird dabei nicht über die Quell- oder Zieladresse identifiziert,

sondern anhand einer eindeutigen Kennzeichnung, dem *Message Identifier*. Beim Senden dieses Frames erhält jedes SG die Botschaft und entscheidet dann mit Hilfe des *Message Identifier*, ob es die CAN-Nachricht verarbeitet oder nicht.

Tabelle 1. ISO/OSI-Referenzmodell ⁸

Lay-er	Bezeich-nung	Aufgabe
7	Application	Dienste für den Anwender
6	Presentation	Darstellung der Daten
5	Session	Kommunikations-steuerung
4	Transport	Segmentierung
3	Network	Verbindungssteu-erung, Routing
2	Data Link	Botschaftsaufbau, Buszugriff, Feh-lersicherung
1	Physical	Bitübertragungs-schicht, Elektrische Signalpegel

Insgesamt existieren vier verschiedene Arten von Frames:

- Daten Frame (Übertragung von Nutzdaten bis zu 8 Byte lang)
- Remote Frame (Anfrage eines Daten-Frames)
- Error Frame (signifiziert einen Fehler bei der Übertragung)
- Overload Frame (Pause zwischen Daten und Remote Frame)

Um den Kilometerstand auszulesen, muss das Berger Fahrtenbuch in der Lage sein, CAN-Nachrichten zu senden und zu empfangen. Daher muss sowohl das Anfrage Frame als auch das Daten Frame von der App eingesetzt werden. Die Frames unterscheiden sich zum einen durch das Remote Transmission Bit, das angibt, ob es sich um ein Remote oder Daten Frame handelt und zum anderen durch das Data Field, das die Nutzdaten enthält. Dieses ist beim Remote-Frame leer. Der allgemeine Aufbau einer CAN-Botschaft ist in Abbildung 3 dargestellt. Der ID (Message Identifier) besagt, welcher Wert angefordert wird und der Data Length Code (DLC) gibt Auskunft über die Größe der Nutzdaten. Die anderen Kennfelder des Frames können hier außer Acht gelassen werden. [8]

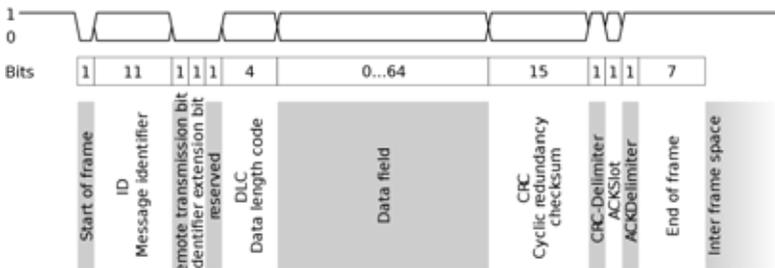


Abbildung 3: Aufbau einer Standard CAN-Nachricht [Quelle:

http://upload.wikimedia.org/wikipedia/commons/6/6f/CAN_telegramm_2.0A.svg]

⁸ Vgl. K. Reif: Automobilelektronik: Eine Einführung für Ingenieure. Vieweg + Teubner Verlag; 4. Auflage, 2012. S.3f.

6 Verwendete Soft- und Hardware

Neben den Bestandteilen der Fahrzeugtechnik sind noch weitere Soft- und Hardwarekomponenten für die Kommunikation mit dem Fahrzeug notwendig. In diesem Abschnitt werden der Diagnosetester vorgestellt so wie weitere Produkte von Berger Elektronik, die für die Verbindung zum Fahrzeug notwendig sind.

6.1 mega macs 42 SE

Mit dem Diagnosegerät „mega macs 42 SE“ können nicht nur Fehlercodes gelesen und gelöscht, sondern auch digitale Messwerte von Steuergeräten dargestellt werden. Unter anderem kann der aktuelle Kilometerstand des Fahrzeugs ausgelesen werden (siehe Abbildung 4).



Abbildung 4: mega macs 42 SE

Die Verbindung zum Fahrzeug erfolgt drahtlos über Bluetooth. Hierzu muss nur das Diagnosemodul (DT VCI) an den OBD-Stecker angeschlossen werden. [6]

6.2 ISI b2app und ISI b2air

Die Android-Anwendung „ISI b2app“ von Berger Elektronik GmbH ist eine Software für Smartphones und Tablets, mit der die Buskommunikation (CAN, LIN) angezeigt und aufgezeichnet werden kann [4]. Über „ISI b2air“ - ebenfalls von Berger Elektronik - können CAN und LIN Botschaften drahtlos übertragen werden [7]. Dieses Gateway kann

die Nachrichten per Bluetooth an einen Laptop, Smartphone oder Tablet senden. Dort können die Nachrichten weiter verarbeitet werden.

6.3 OBD-Breakout-Box

Für die Verbindung zwischen ISI b2air und OBD-Buchse des Fahrzeugs muss ein Kabel eingesetzt werden, das die richtige PIN-Belegung besitzt, um die Kompatibilität zwischen Fahrzeugschnittstelle und ISI b2air zu gewährleisten. Da sich die PIN-Belegung der OBD-Buchse je nach Automodell teilweise unterscheidet, wird eine OBD-Breakout-Box verwendet. Sie werden zum Beispiel in Laboren oder an Prüfständen eingesetzt, um Kabelverbindungen schnell herzustellen bzw. trennen zu können oder falls kein fertiges Kabel vorhanden ist. Die sogenannten Federstecker werden in die Federbuchsen der OBD-Breakout-Box eingesteckt (siehe Abbildung 5). Auf diese Weise können die 16-OBD-Pins beliebig gesetzt werden. [9]

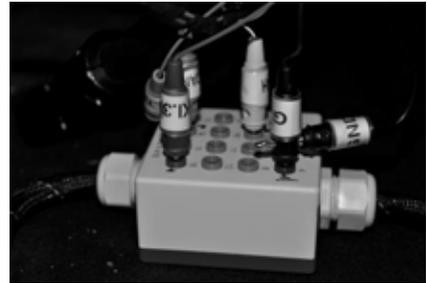


Abbildung 5: OBD-Breakout-Box mit Federsteckern

7 Umsetzung am Beispiel des Berger Fahrtenbuches

Anhand des Berger Fahrtenbuches soll der Kilometerstand aus einer Mercedes Benz A-Klasse automatisch ausgelesen und in das vorgesehene Textfeld eingetragen werden. Nachfolgend wird die Umsetzung dieses Vorhabens beschrieben, für die die Methodik aus Kapitel 4 zum Einsatz kommt.

7.1 Aufzeichnung der Daten und Versuchsaufbau

Um die Rohdaten der Buskommunikation im Fahrzeug zu erhalten, wird zunächst die OBD-Breakout-Box an die OBD-Buchse der Mercedes A-Klasse angeschlossen. Dann werden die Federstecker in die Federbuchsen für Masse, Versorgungsspannung, CAN-High und CAN-Low Datenleitung eingesteckt. Das ausgehende Kabel der Federstecker wird mit dem „ISI b2air“ verbunden. An die OBD-Buchse der OBD-Breakout-Box wird das Diagnosemodul „DT VCI“ angeschlossen. Über Bluetooth verbindet sich dann der Diagnosetester „mega macs 42 SE“ mit diesem Modul. Das letzte Verbindungsstück ist die Android-Software „ISI b2app“, die auch per Bluetooth mit dem „ISI b2air“ verbunden wird. Der gesamte Versuchsaufbau wird in Abbildung 6 skizziert.

Abbildung 7 stellt die Umsetzung dieses Versuchsaufbaus dar. Im Fußraum befindet sich die OBD-Breakout-Box mit dem „ISI b2air“ und dem Diagnosemodul.

Auf dem Fahrersitz ist der Diagnosetester und ein Tablet, auf dem „ISI b2app“ läuft, zu sehen.

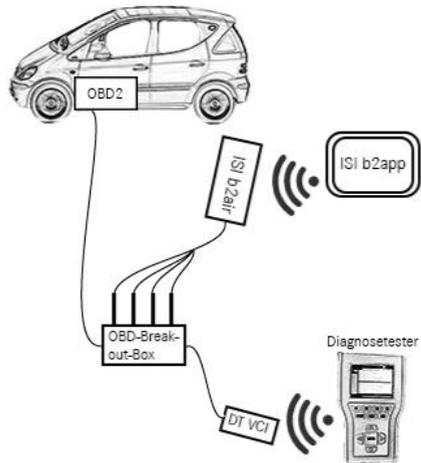


Abbildung 6: Versuchsaufbau als Skizze



Abbildung 7: Versuchsaufbau

Nachdem alle Komponenten miteinander verbunden sind, kann die Aufzeichnung der Daten beginnen. Als aller erstes muss die Zündung gestartet werden (siehe Abbildung

8), damit der Diagnosetester mit dem Bussystem kommunizieren kann. Dann gibt das „ISI b2app“ den Befehl zur Aufzeichnung der Buskommunikation.



Abbildung 8: Prozessablaufdiagramm zur Aufzeichnung der CAN-Nachrichten

Schließlich startet der Diagnosetester mit der Abfrage des Kilometerstandes auf dem CAN-Bus. Das „ISI b2app“ zeichnet währenddessen alle CAN-Botschaften auf, die zwischen dem Diagnosetester und dem Bussystem gesendet werden und speichert sie in einer Logdatei (siehe Abbildung 9).

Message ID	Frame ID	Type	Direction	Length	Data (hex)
11448217630	4F4	R	05	67	60 30 24 01 30 31
11448217630	5B4	R	02	21	60 FF FF FF FF FF
11448217631	4F4	R	05	67	60 30 24 01 30 31
11448217631	5B4	R	02	21	60 FF FF FF FF FF
11448217632	4F4	R	05	67	60 30 24 01 30 31
11448217632	5B4	R	02	21	60 FF FF FF FF FF
11448217633	4F4	R	05	67	60 30 24 01 30 31
11448217633	5B4	R	02	21	60 FF FF FF FF FF
11448217634	4F4	R	05	67	60 30 24 01 30 31
11448217634	5B4	R	02	21	60 FF FF FF FF FF
11448217635	4F4	R	05	67	60 30 24 01 30 31
11448217635	5B4	R	02	21	60 FF FF FF FF FF
11448217636	4F4	R	05	67	60 30 24 01 30 31
11448217636	5B4	R	02	21	60 FF FF FF FF FF
11448217637	4F4	R	05	67	60 30 24 01 30 31
11448217637	5B4	R	02	21	60 FF FF FF FF FF
11448217638	4F4	R	05	67	60 30 24 01 30 31
11448217638	5B4	R	02	21	60 FF FF FF FF FF
11448217639	4F4	R	05	67	60 30 24 01 30 31
11448217639	5B4	R	02	21	60 FF FF FF FF FF
11448217640	4F4	R	05	67	60 30 24 01 30 31
11448217640	5B4	R	02	21	60 FF FF FF FF FF
11448217641	4F4	R	05	67	60 30 24 01 30 31
11448217641	5B4	R	02	21	60 FF FF FF FF FF
11448217642	4F4	R	05	67	60 30 24 01 30 31
11448217642	5B4	R	02	21	60 FF FF FF FF FF
11448217643	4F4	R	05	67	60 30 24 01 30 31
11448217643	5B4	R	02	21	60 FF FF FF FF FF
11448217644	4F4	R	05	67	60 30 24 01 30 31
11448217644	5B4	R	02	21	60 FF FF FF FF FF
11448217645	4F4	R	05	67	60 30 24 01 30 31
11448217645	5B4	R	02	21	60 FF FF FF FF FF
11448217646	4F4	R	05	67	60 30 24 01 30 31
11448217646	5B4	R	02	21	60 FF FF FF FF FF
11448217647	4F4	R	05	67	60 30 24 01 30 31
11448217647	5B4	R	02	21	60 FF FF FF FF FF
11448217648	4F4	R	05	67	60 30 24 01 30 31
11448217648	5B4	R	02	21	60 FF FF FF FF FF
11448217649	4F4	R	05	67	60 30 24 01 30 31
11448217649	5B4	R	02	21	60 FF FF FF FF FF
11448217650	4F4	R	05	67	60 30 24 01 30 31
11448217650	5B4	R	02	21	60 FF FF FF FF FF

Abbildung 9: aufgezeichnete CAN-Botschaften von ISI b2app

7.2 Analyse der Daten

Aus der Logdatei muss nun das Anfrage Frame und das Daten Frame gefiltert werden, das den Wert des Kilometerstandes anfragt

bzw. enthält. Bei den CAN-Daten des Busses werden von Berger Elektronik, so wie allgemein von allen CAN-Tools, nur die relevanten Daten aus dem originalen CAN-Frame-Aufbau (vgl. Abbildung 3) herausgezogen und im Frame dargestellt. Tabelle 2 zeigt einen Ausschnitt solcher CAN-Frames. Zur Ermittlung des Kilometerstandwertes müssen die Datenbytes richtig interpretiert werden. Das Datenbyte 0 entspricht dem Transportprotokoll ISO-TP [vgl. 5, S.119f.]. Es wird zur Übertragung von Botschaften im CAN-Bus verwendet. Gleichzeitig wird mit dem Byte angegeben, wie groß die Nutzdaten sind. Gezählt wird ab dem Datenbyte 1. Das Datenbyte 1 beschreibt das Diagnoseprotokoll KWP 2000 [vgl. 5, S.26f.]. Es gilt als eines der Standardprotokolle zur Kommunikation zwischen einem Diagnosetester und einem SG. Zum Zeitpunkt der Anfrage betrug der Kilometerstand 74783 km. In hexadezimaler Form entspricht das dem Wert 01241F. Die Speicherung von Zahlenwerten folgt dabei einem bestimmten Format, der sog. Byte-Reihenfolge. Diese wird vom Steuergeräte-Hersteller selber definiert. Im SG der A-Klasse werden die Zahlenwerte nach „Little Endian“ gespeichert [vgl. 5, S.423]. Dies bedeutet, dass das niederwertigste Byte des Wertes an erster Stelle steht. Im Umkehrschluss wird also der HEX-Wert des Kilometerstandes in umgekehrter Reihenfolge abgebildet: an Stelle von 01241F steht in den Nutzdaten 1F2401. Dieser Wert entspricht einer Größe von 3 Bytes, d.h. die Nutzdaten des Frames, die den Kilometerstand enthalten, sind bei einer Kodierung ohne Multiplikations- oder Divisionsfaktor 5 Bytes groß. Als weiterer Parameter zur Analyse kann die PID (Parameter ID ~ Datenbyte 2) herangezogen werden [vgl. 5, S.174]. Sie gibt an, welcher Datenwert vom SG angefragt wird. Die PID im Anfrage-Frame wiederholt sich im Antwort-Frame. Daran wird erkannt, dass die beiden Frames zusammengehören. Da die PID-Liste i.d.R. nur der Steuergeräte-Hersteller kennt, kann anhand des Wertes der PID nicht bestimmt werden, ob sich die Anfrage auf den Kilometerstand bezieht.

Tabelle 2. Aufbau der CAN-Nachrichten des Busses der Mercedes A-Klasse

Messzeit	ID	DR	Typ	DLC	Datenbytes 1-8 (Data Field)							
					0	1	2	3	4	5	6	7
544.722864	5B4	Rx	d	8	30	08	28	FF	FF	FF	FF	FF
544.769981	4F4	Rx	d	8	21	5F	00	00	05	52	00	00
544.821274	4F4	Rx	d	8	22	32	34	34	35	30	30	31
544.941634	1C	Rx	d	8	02	3E	02	FF	FF	FF	FF	FF
544.968649	5B4	Rx	d	8	02	21	60	FF	FF	FF	FF	FF
544.979916	4F4	Rx	d	8	05	61	60	1F	24	01	30	31
545.087441	5B4	Rx	d	8	02	21	60	FF	FF	FF	FF	FF
545.099853	4F4	Rx	d	8	05	61	60	1F	24	01	30	31

Daher werden die Hinweise zur Größe der Nutzdaten und der HEX-Wert des Kilometerstandes als Hinweise zur Analyse der CAN-Botschaften verwendet. Es werden zunächst alle Frames gefiltert, bei denen im Datenbyte 0 der Wert 05 steht. Viele der Frames enthalten als Nutzdaten die Bytes 1F 24 01. Daraus lässt sich schließen, dass es sich dabei um den HEX-Wert des Kilometerstandes in umgekehrter Reihenfolge und damit um den gesuchten Datenframe handelt. Der dazugehörige Anfrage-Frame unterscheidet sich immer dadurch, dass der Wert des Datenbyte 1 um hexadezimal 40 kleiner ist. Zum Daten-Frame mit dem Datenbyte-1-Wert 61 gehört demzufolge der Anfrage Frame mit dem Datenbyte-1-Wert 21.

Der Daten Frame, der den Wert des Kilometerstandes enthält, lautet also wie folgt:

4F4 Rx d 8 05 61 60 **1F 24 01** 30 31

Der dazugehörige Anfrage Frame, der die Anfrage beinhaltet, hat folgenden Aufbau:

5B4 Rx d 8 02 21 60 FF FF FF FF FF

Das Senden des Anfrage Frames reicht aus, um die Antwort zu erhalten. Es muss vorher kein Verbindungsaufbau zum Bussystem hergestellt werden. [5]

Bevor aber die Anfrage in das Fahrtenbuch implementiert werden kann, muss sichergestellt werden, dass eine proprietäre Anwen-

dung nur mit Hilfe des „ISI b2air“ CAN-Nachrichten senden und empfangen kann. Um das zu Testen, wurde der Anfrage Frame zunächst von „ISI b2app“ über das „ISI b2air“ an den CAN-Bus derselben Mercedes A-Klasse gesendet. Dieser Test verlief erfolgreich, denn als Antwort kam der oben erwähnte Daten Frame.

7.3 Implementierung

In den Programmcode des Fahrtenbuchs muss die Funktionalität implementiert werden, die den CAN-Anfrage-Frame sendet, den CAN-Daten-Frame richtig interpretiert und den entsprechenden Kilometerstand in das Textfeld einträgt. Hierfür wird auf bereits vorhandene Softwarebibliotheken für das Senden und Empfangen von CAN-Daten über die Bluetooth-Schnittstelle zurückgegriffen.

Die App wird um weitere Funktionen ergänzt, die den Anfrage Frame an den CAN-Bus senden, die Antwort interpretieren und den Kilometerstand an die entsprechende Stelle schreiben.

Die Programmiersprache ist Java und genutzt wird das frei verfügbare Android SDK⁹.

⁹ <http://developer.android.com/sdk/index.html>

Als Entwicklungsumgebung wird Eclipse¹⁰ verwendet.

Folgende Klassen und Methoden wurden in das Berger Fahrtenbuch implementiert:

Klasse km_handler() :

Der Anfrage Frame wird in einen Byte-Array geschrieben:

```
byte[] message =
{0x02,0x21,0x60,(byte) 0xff,(byte)
0xff,(byte) 0xff,(byte) 0xff,(byte)
0xff};
```

Das Byte-Array message wird zusammen mit dem ID 5B4 an den CAN-Bus gesendet. Ein Timer sorgt dafür, dass diese Nachricht jede Sekunde verschickt wird. Sobald eine neue Fahrt angelegt wird, wird die Anfrage gesendet.

Methode gotCanMsg() :

Diese Methode wird immer dann aufgerufen, wenn das eingesetzte Smartphone eine CAN-Nachricht erhält. Die Botschaft wird von der Methode gotCanMsg() analysiert, indem es den ID überprüft. Wenn der Wert des ID 4F4 beträgt, so handelt es sich um das CAN-Frame mit dem Kilometerstand:

```
else if (id == 0x4F4) {
    long l = 0;
    l |= 0xff & data[3];
    l |= ((0xff & data[4]) << 8);
    l |= ((0xff & data[5]) << 16);
    gotKilometerstand((int) (l));
}
```

Der Wert des Kilometerstandes befindet sich im 3., 4. und 5. Datenbyte. Diese HEX-Bytes werden durch bitweise Operation in Dezimalzahlen umgewandelt. Das Ergebnis impliziert den Wert des Kilometerstandes. Dieser wird im letzten Schritt in das entsprechende Textfeld geschrieben und immer aktualisiert, sobald sich der Kilometerstand ändert.

7.4 Test und Ergebnisse

Die erste Testfahrt in der Mercedes Benz A-Klasse erfolgte mit dem „ISI b2air“, der OBD-Breakout-Box und dem Smartphone LG Nexus 5 mit Android-Version 4.4.2. Zunächst wurde die OBD-Breakout-Box mit den Federsteckern an die OBD-Buchse des Autos angeschlossen. An das andere Ende der OBD-Breakout-Box wurde das „ISI b2air“ angeschlossen. Nach dem Start der Zündung konnte sich das Berger Fahrtenbuch über Bluetooth zum „ISI b2air“ verbinden. Somit konnte im Fahrtenbuch eine neue Fahrt angelegt werden. Dabei müssen zunächst ein paar Informationen vom Benutzer eingetragen werden (vgl. Kapitel 2). Unmittelbar nach dem Anlegen einer neuen Fahrt wurde der Kilometerstand von der App richtig erkannt und in das entsprechende Textfeld automatisch eingetragen. Einzugeben war nur noch der Zweck der Fahrt. Auch während der Fahrt sollte der veränderte Kilometerstand angezeigt werden. Dieser Test verlief ebenfalls erfolgreich. Es wurde stets der richtige Wert dargestellt.

7.4.1 Optimierung

Da die OBD-Breakout-Box mit den Federsteckern und das „ISI b2air“ zusammen viel Platz im Fußraum des Fahrzeugs einnehmen, stört diese Vorrichtung den Fahrer beim Betätigen der Pedale und stellt damit eine gefährliche Situation beim Fahren dar. Unter diesem Risiko darf das Fahrtenbuch unter keinen Umständen benutzt werden. Daher gilt es, diese Vorrichtung so zu optimieren, dass für den Fahrer und eventuelle Beifahrer keine Gefahr im Straßenverkehr besteht.

Auf Grund dessen wurde die OBD-Breakout-Box mit den Federsteckern durch ein „OBD2 Standardkabel DSUB-9“ ersetzt (siehe Abbildung 10). Bei diesem Kabel musste zunächst die PIN-Belegung so angepasst werden, dass die Weiterleitung der CAN-Botschaften vom „ISI b2air“ zum Auto gewährleistet ist. Dafür wurde in der Entwicklungsabteilung bei Berger Elektronik die Verdrahtung der Masse, der Versorgungsspannung, der CAN-

¹⁰ <https://www.eclipse.org/downloads/>

High- und CAN-Low-Leitungen im OBD-Stecker des Kabels umgelötet.



Abbildung 10: OBD-Standard-Kabel DSUB-9

Mit dem Ergebnis, das in Abbildung 11 zu sehen ist, lässt sich wesentlich sicherer und ungestört fahren. Praktischerweise bietet sich das Ablagefach links neben dem Lenkrad gut an, um das „ISI b2air“ zu verstauen.



Abbildung 11: Das umgelötete OBD-ISIb2air-Kabel

7.4.2 Testfahrten

Zur Überprüfung des neuen Kabels wurde ein kurzer Test mit dem Google Nexus 7 durchgeführt, bei dem der Kilometerstand zu Beginn der Fahrt ausgelesen werden sollte. Nachdem dieser Test erfolgreich verlief, wurden drei Testfahrten unternommen, um den gesamten Aufgabenbereich der Fahrtenbuch-App zu prüfen. Hierbei wurden drei Android-Geräte verwendet:

- Google Nexus 5 (Android 4.4.2)
- Samsung Galaxy Nexus S (Android 4.1.2)
- Google Nexus 7 (Android 4.4.2)

Das Fahrtenbuch auf allen drei Geräten sollte den Kilometerstand von Anfang bis zum Ende der Fahrt durchgehend automatisch erkennen und in das vorgesehene Textfeld eintragen. Mit allen drei Android-Geräten wurde ein positives Ergebnis erzielt. Abbildung 12 zeigt denselben Wert (74813 km) am Tacho des Fahrzeugs und im Berger Fahrtenbuch mit dem Smartphone Samsung Galaxy Nexus S.



Abbildung 12: Anzeige des Kilometerstandes am Tacho (links) und im Berger Fahrtenbuch (rechts)

8 Fazit

Mit dem positiven Ergebnis des automatischen Auslesens des Kilometerstandes konnte gezeigt werden, dass auch die Schnittstelle eines Fahrzeugs genutzt werden kann, um die Usability und damit auch die User Experience eines Fahrtenbuches zu verbessern. Diese Methodik kann bei allen mobilen Apps, die in irgendeiner Weise im Straßenverkehr genutzt werden, angewendet werden.

Allerdings kann das Berger Fahrtenbuch nicht bei allen Fahrzeugen den Kilometerstand auslesen. Die entsprechende CAN-Anfrage ist bei jedem Fahrzeug anders. Sie variiert von Modell zu Modell und von Serie zu Serie. Um das Produkt erfolgreich auf den Markt zu bringen, müsste bei einer ganzen Flotte von Fahrzeugen die Buskommunikation analysiert und die entsprechende Botschaft in das Berger Fahrtenbuch implementiert werden.

9 Literaturverzeichnis

- [1] M. Herczeg: Software-Ergonomie: Theorien, Modelle und Kriterien für gebrauchstaugliche interaktive Computersysteme. Oldenbourg Wissenschaftsverlag, 2009. S.169ff.
- [2] HTW Chur: DIN EN ISO 9241-110 <http://www.cheval-lab.ch/was-ist-usability/usability-grundlagen/normen-und-richtlinien/iso-9241-110/> zuletzt geprüft: 08.03.2014.
- [3] Springer Gabler Verlag (Herausgeber), Gabler Wirtschaftslexikon, Stichwort: Reverse Engineering: <http://wirtschaftslexikon.gabler.de/Archiv/142096/reverse-engineering-v4.html> zuletzt geprüft: 24.02.2014.
- [4] Berger Elektronik GmbH: ISI b2app <http://www.bergerelektronik.com/de/produkte/automotive-smartphone-apps/isi-b2app.html> zuletzt geprüft: 08.03.2014
- [5] W. Zimmermann, R. Schmidgall: Bussysteme in der Fahrzeugtechnik: Protokolle, Standards und Softwarearchitektur. Vieweg + Teubner Verlag; 4. Auflage, 2011.
- [6] Hella Gutmann Solutions GmbH: megamacs 42 SE <http://www.hellagutmann.com/diagnose/mega-macs-42-se/uebersicht/> zuletzt geprüft: 08.03.2014
- [7] Berger Elektronik GmbH: ISI b2air <http://www.bergerelektronik.com/de/produkte/can/can-bluetooth-wireless-interface-isi-b2air.html> zuletzt geprüft: 08.03.2014
- [8] T. Dohmke: Bussysteme im Automobil CAN, FlexRay und MOST. Berlin, 2002, S. 4f.: <http://de.yu-yongxin.com/klausur/bussysteme.pdf> zuletzt geprüft: 14.03.2014
- [9] Steve V. Hatch: Computerized Engine Controls. Cengage Learning, 2011, S. 163f.

Vergleich der Single Sign On Verfahren SAML und OpenID Connect

Johannes Kartheiningner
Reutlingen University
Johannes.Kartheiningner@Student.
Reutlingen-University.de

Abstract

In dieser Arbeit werden die beiden Single Sign On-Protokolle SAML und OpenID Connect miteinander verglichen. Vorab wird die Aufgabe eines Single Sign On-Systemes erläutert und allgemeine Vorteile sowie Nachteile aufgezählt.

Schlüsselwörter

Authentifikation, Autorisation, Single Sign On, Identitätsmanagement

CR-Kategorien

D.4.6 [Security and Protection]: Authentication; C.2.m [Computer Systems Organization]: Computer-Communication Networks – Miscellaneous

1 Einleitung

In Unternehmen oder öffentlichen Institutionen ist es bereits der Fall, dass der Nutzer um Zugriff auf für ihn relevante Anwendungen und Ressourcen zu erhalten, sich nur einmal authentifizieren muss. Bei öffentlichen Anwendungen im Internet ist dieser Trend nun ebenso zu erkennen. So prangern auf vielen Seiten bereits verschiedene Loginverfahren, bei denen der

Nutzer sich mit einem bereits verbundenen Dienst gegenüber einem neuen Dienst identifizieren kann (siehe exemplarisch Abbildung 1).



Abbildung 1 Loginmöglichkeiten auf StackOverflow.com

Dies erspart ihm sowohl eine aufwendige Registrierung bei dem neu zu nutzenden Dienst, als auch das Verwalten mehrerer Passwörter. Man nennt dieses Verfahren auch Single Sign On.

2 Motivation

Durch die steigende Verbreitung vieler verteilter Einzelsysteme, die dennoch zusammenarbeiten, besteht der Bedarf die Authentifizierung des Nutzers nicht unnötig oft zu verlangen. Es soll dem Nutzer ermöglicht werden sich einmal gegenüber einem System zu authentifizieren und danach alle für ihn relevanten Systeme nutzen zu können. Single Sign On soll die Anzahl der Loginvorgänge reduzieren, sowie die Sicherheit und Effizienz erhöhen und die Usability zu verbessern (vgl. [6] & [7], S.10f).

Ebenso soll die Interoperabilität der IT-Landschaft gesteigert werden, indem die

Betreuer Hochschule: Prof. Dr.-Ing. habil. Natividad
Martínez Madrid
Hochschule Reutlingen
natividad.martinez@reutlingen-
university.de

Informatics Inside 2014
Wissenschaftliche Vertiefungskonferenz
30. April 2014, Hochschule Reutlingen
Convriecht 2014 Kartheiningner Johannes

Authentifizierung nicht mehr dezentral an jedem System ausgeführt sowie seitens der Administration verwaltet werden muss. Durch Single Sign On reduzieren sich ebenso Datenredundanzen (vgl. [8], S.1f).

Obwohl es Single Sign On bereits seit längerer Zeit gibt (bspw. Kerberos [1] seit 1980), so gibt es durch die wachsende Cloudlandschaft, sowie das Aufkommen mobiler Geräte wie Smartphones und Tablets neue Anforderungen an Single Sign On-Systeme. So wurde erst kürzlich ein neuer Standard im Bereich Single Sign On verabschiedet, welcher eben jene Anforderungen der mobilen Geräte erfüllt (siehe [15]).

3 Definitionen

Um die Aufgabe eines Single Sign On Verfahrens nachvollziehen zu können, ist es wichtig die Unterschiede der Begriffe Autorisation und Authentifizierung zu kennen.

Authentifizierung

Die Authentifizierung bezeichnet im Kontext von Single Sign On den Nachweis eines Nutzers, dass er derjenige ist, für den er sich ausgibt.

Autorisation

Die Autorisation bezeichnet die Erlaubnis eines Nutzers eine gewisse Aktion auszuführen, bspw. auf eine bestimmte Ressource zuzugreifen oder Zugang zu einem System zu erhalten.

Single Sign On

Single Sign On (SSO) bezeichnet die einmalige Authentifizierung eines Nutzers gegenüber einem einzelnen System, woraufhin er Zugriff auf alle Systeme erhält für die er autorisiert ist. Eine Kerneigenschaft von Single Sign On-Systemen ist es, dass die Authentifizierung nicht an jedem System nochmals stattfinden muss. Die Überprüfung der Autorisation übernimmt jedoch nicht zwingend das SSO-System, sondern das zu nutzende System kann selbst überprüfen ob der authentifizierte Nutzer dazu berechtigt ist.

Ebenso gibt es jedoch auch Systeme die beides kombiniert anbieten oder nur die Autorisierung übernehmen.

4 Single Sign On

SSO soll sowohl auf Seiten der Nutzer, als auch auf Seiten der Administration einiges vereinfachen. Aus Nutzersicht wären dies bspw. die Verwaltung weniger Konten sowie eines einzigen Logins und aus administrativer Sicht wird bspw. die Anzahl an Serviceanfragen bzgl. neu zu setzender Passwörter bei vergessenen Zugangsdaten geringer werden.

4.1 Vorteile

Theoretisch bedeutet für den Nutzer, dass wenn er nur einen Account zu verwalten hat, ein sicheres Passwort ausreicht. Untersuchungen ergaben, dass Nutzer dazu neigen Passwörter bei verschiedenen Diensten mehrmals zu verwenden, sowie einfach zu erratende Kombinationen nutzen (vgl. [11], S.26f, S.34-37). Ebenso entfällt für den Nutzer das Registrieren bei jeder einzelnen Anwendung.

Aus Sicht eines Unternehmens welches ein Single Sign On-Verfahren für ihre Anwendungen anbietet, verringert sich der Verwaltungsaufwand im Bezug auf Nutzerkonten. Es muss nur eine zentrale Identitätskomponente administriert werden, anstatt mehrere verschiedene. Ebenso benötigen neue Systeme, die nachträglich hinzugefügt werden, keine eigene Identitätskomponente, sondern können auf die bereits vorhandene zurückgreifen.

4.2 Nachteile

Es gibt jedoch auch einige negative Aspekte, die man sowohl bei der Entwicklung eines eigenen Single Sign On – Systems als auch bei der Auswahl eines fertigen Single Sign On – Systems bedenken muss. So kann der Ausfall der einzelnen Authentifikationskomponente bedeuten, dass die Nutzer ihre gesamten anderen Systeme während des Ausfalls nicht mehr verwenden können. Man könnte auch sagen, dass dies ein Single Point of Failure darstellen kann. Dies bedeutet, dass

die Authentifikations-komponente sowohl skalierbar sein muss, als auch ein Backupmechanismus/-system vorhanden sein muss, der die Verwendbarkeit der anderen Systeme ermöglicht.

Sollte das SSO-System Sicherheitslücken aufweisen, so sind alle Systeme betroffen, die auf dieses SSO-System als Identitäts-lieferanten zurückgreifen.

5 Single Sign On-Systeme

Single Sign On-Systeme werden unterschiedlich eingeordnet. In dieser Ausarbeitung wird die Unterscheidung von Rhada und Clerq angewandt (siehe [12], S.134f & [13], S. 44f). In Abbildung 2 ist diese Unterscheidung skizziert, sowie die hier betrachteten Web Single Sign On Systeme eingezeichnet.

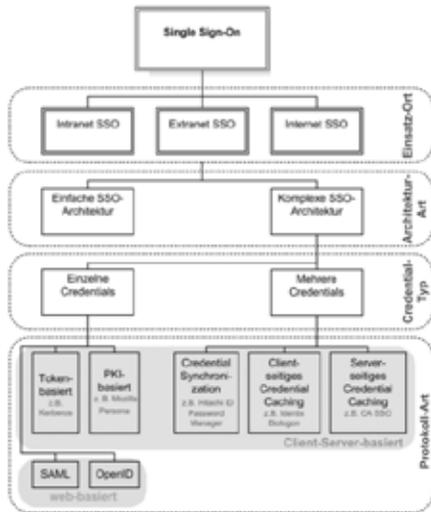


Abbildung 2 Kategorisierung von Single Sign On-Systemen ([11], S.58)

5.1 SAML 2.0

Security Assertion Markup Language 2.0 (SAML 2.0) ist ein auf XML basierendes Rahmenwerk für die Beschreibung von Autorisations- sowie Authentifikationsdaten und deren Austausch zwischen verschiedenen Systemen (siehe [2]).

In einem Single Sign On-System, welches SAML verwendet, trifft man die Rollen Identity Provider (IDP) sowie Service Provider (SP) an. Ein SP ist dabei eine Webanwendung, ein Webservice usw., welcher von einem Nutzer oder einem System genutzt werden möchte. Bevor ein SP genutzt werden kann, muss die Identität des Nutzers von einem IDP festgestellt werden. Ob dieser Nutzer nun den Service verwenden kann, obliegt dem SP, der IDP betrifft einzig und allein die Authentifizierung, nicht die Autorisierung. Die Kommunikation zwischen SP und IDP erfolgt üblicherweise über den Client, d.h. es findet keine direkte Kommunikation zwischen den beiden Parteien statt. Folgendes Schaubild stellt schemenhaft die verschiedenen Rollen in einem SSO-Szenario vor, sowie deren Beziehungen untereinander.

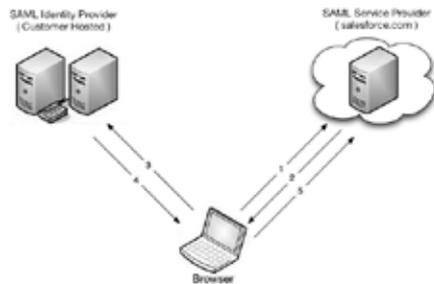


Abbildung 3 SAML Schaubild¹

5.2 OAuth 2.0

OAuth 2.0 ist ein offener Standard für SSO der nur die Autorisierung behandelt (siehe RFC6749²). So ist es möglich bei Anwendungen über OAuth 2.0 auf Ressourcen anderer Anwendungen zuzugreifen. Eine Authentifizierung des Nutzers findet jedoch nicht explizit statt, kann jedoch implizit erfolgen (siehe Abschnitt Authentifizierung & Autorisierung).

¹ http://wiki.developerforce.com/page/File:Saml_flow.png

² <http://tools.ietf.org/html/rfc6749>

OAuth 2.0 definiert vier Rollen³:

- resource owner
- resource server
- client
- authorization server

In Abbildung 4 sind die Beziehungen der Rollen untereinander dargestellt. Der User entspricht hierbei der Rolle des „resource owner“.

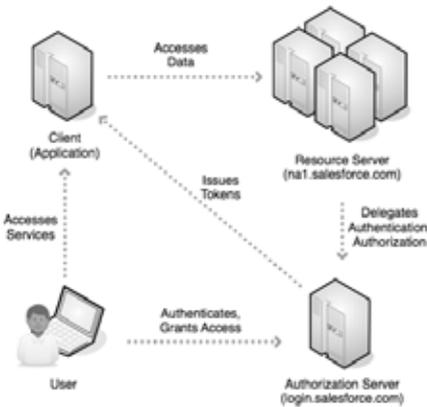


Abbildung 4 OAuth Schaubild⁴

5.2.1 OpenID Connect

Ein im Februar 2014⁵ erschienener Standard für Single Sign On basiert auf OAuth 2.0 und erweitert diesen um eine Authentifizierungskomponente. Bereits kurz nachdem der finale Standard verabschiedet wurde, gibt es OpenID Connect-Anbieter, die bereits während der Entwurfsphase des Standards zuerst hybride Mechanismen zur Verknüpfung von OAuth 2.0 und dem

ursprünglichen OpenID anbieten und nun bereits OpenID Connect implementiert haben (bspw. Google⁶ und Paypal⁷). Für einen Aufbau von OpenID Connect siehe Abbildung 4, da die Rollen, sowie Abfolgen exakt wie bei OAuth 2.0 definiert sind. Bei OpenID Connect ist es nur zusätzlich möglich Identitätsinformationen vom Authorization Server zu erhalten.

6 Detailbetrachtung

Auch wenn der Einsatzzweck von SAML 2.0 sowie OpenID Connect ein ähnlicher ist, unterscheiden sich beide Protokolle auf technischer Ebene.

6.1 Authentifizierung & Autorisierung

Ist SAML sowohl für Authentifizierung als auch Autorisierung verwendbar, so übernimmt OAuth 2.0 einzig die Autorisierung. Es kann implizit eine Autorisierung stattfinden, da nur der Besitzer eines Tokens auch Zugriff auf diesen hat. Es kann also bspw. der Token für den Zugriff auf irgendeine Komponente angefordert werden und beim nächsten Authentifizierungsvorgang überprüft werden ob immer noch der Nutzer der Zugriff auf diese Komponente hat. OAuth ist aber eigentlich eine reine Autorisierungskomponente und davon OAuth 2.0 für Authentifizierung zu verwenden wird abgeraten⁸. Es kann nicht davon ausgegangen werden, dass der Zugang zu einer Ressource gleichzeitig bedeutet, dass nur ein Nutzer eben Zugriff auf jene Ressource erhält. In einzelnen Fällen wie bspw. dem Zugriff auf Kontoinformationen von Facebook mag dies zutreffen. Jedoch sagt bspw. der Zugriff auf ein Dokument in GoogleDrive nichts über die

³ <http://tools.ietf.org/html/rfc6749#section-1.1>

⁴ <https://wiki.developerforce.com/page/File:OAuthRoles.png>

⁵ <http://openid.net/2014/02/26/the-openid-foundation-launches-the-openid-connect-standard/>

⁶ <https://developers.google.com/accounts/docs/OAuth2Login>

⁷ <https://developer.paypal.com/docs/integration/direct/identity/log-in-with-paypal/>

⁸ <http://www.threadsafe.com/2012/01/problem-with-oauth-forauthentication.html>

Identität des Nutzers aus, da mehrere Nutzer Zugriff darauf haben könnten.

Aufbauend auf dem Konzept, dass OAuth 2.0 einzig zur Autorisierung verwendet werden sollte, wurde OpenID Connect entwickelt um eine Möglichkeit zu haben Nutzer auch authentifizieren zu können.

6.2 Token/Nachrichtenformat

SAML sieht es vor, dass die Informationen in einem XML-Format ausgetauscht werden.

Bei OpenID Connect können die Informationen jedoch binär, per JavaScript Object Notation (JSON) oder gar per SAML-Format (Ein im SAML-Standard deklariertes XML-Format) übertragen werden.

6.3 Transport

Sieht OpenID Connect nur vor Tokens per Http(s) zu übertragen, so gibt es für SAML ebenso Bindings welche eine Übertragung per Http(s) vorsehen, jedoch ist auch bspw. SOAP möglich. Ebenso steht es dem Entwickler offen auch bspw. Java Message Service zu verwenden, da bei SAML eben keine festen Vorgaben gemacht wurden.

6.4 Single Logout

Single Logout bezeichnet den umgekehrten Vorgang zu Single Sign On. Wird eine Single Sign On-Session nicht geschlossen, ist der Zugang zu allen anderen Services die auf diesen SSO-Provider zugreifen möglich. Wird also bspw. an einem öffentlichen PC die Session nicht geschlossen oder die Session per Session Hijacking⁹ von einem Angreifer übernommen, so ist sowohl der Zugang zu dem System gegenüber dessen man sich identifiziert hat möglich, als auch alle anderen Systeme, die auf diesen SSO-Provider zurückgreifen.

Ist es bspw. möglich in SSO-Systemen die auf SAML setzen sich zentral abzumelden, so sieht der OpenID Connect Standard nur vor, dass nachdem auf einem einzelnen System

ein Logout getätigt wurde, der Nutzer abermals gefragt wird, ob er die gesamte Session beenden möchte¹⁰. Dies ist jedoch nur durch Verwendung eines Inlineframes (Einbindung einer anderen Webseite/-inhalte in einem vordefinierten Bereich) möglich und dadurch ebenso nur in Webbrowsern und nicht in nativen Anwendungen auf mobilen Geräten. Ebenso ist die Spezifikation zu Session-Management immer noch in einem Entwurfsstadium, obwohl der OpenID Connect-Standard bereits verabschiedet wurde¹¹.

6.5 Anwendungsbereiche

Beide SSO-Varianten können sowohl in Unternehmen, als auch im öffentlichen Internet verwendet werden. Dennoch kann festgehalten werden, dass SAML eher innerhalb bzw. zwischen Unternehmen Anklang findet und im Web eher OAuth 2.0 oder OpenID Connect zu finden ist (vgl. [14], S.5f).

6.5.1 Mobile

OpenID Connect wurde mit dem Gedanken entworfen ein einfacher und auf allen Geräten funktionierende Single Sign On-Standard zu sein. Erreicht wird dies dadurch, dass das Tokenformat per JSON ausgegeben werden kann um den Overhead bei XML zu sparen, aber auch dadurch, dass die Authentifizierung nicht zwingend per Webbrowser stattfinden muss. Im Gegensatz dazu ist bei SAML vorgesehen, dass die Authentifizierung des Nutzers über einen Webbrowser erfolgt. Auf mobilen Geräten jedoch können Anwendungen auch ausserhalb des Webbrowsers ausgeführt werden.

6.6 Sicherheit

Ist Verschlüsselung bei SAML bereits auf Nachrichtenebene geläufig und auf Transportebene vorgeschrieben, so wird bei OpenID Connect und OAuth 2.0 die

⁹ [http://projects.webappsec.org/w/page/13246944/Insufficient Session Expiration](http://projects.webappsec.org/w/page/13246944/Insufficient%20Session%20Expiration)

¹⁰ http://openid.net/specs/openid-connect-session-1_0.html#RPLLogout

¹¹ <http://openid.net/developers/specs/>

Verschlüsselung auf Nachrichtenebene selbst optional deklariert. Andererseits ist bei OpenID Connect die Verschlüsselung auf Transportebene ebenso zwingend vorgesehen (siehe ¹² S.10f).

7 Zusammenfassung

Ist SSO zwar ein Thema für das es bereits alte und bewährte Lösungen gibt, ist es jedoch nötig aufgrund neuartiger Geräte, sowie bspw. dem Wechsel zwischen Internetverbindungen (Mobilfunk und Wlan), die bewährten Techniken anzupassen oder nochmals neu zu erarbeiten (siehe neu erarbeitetes OpenID Connect ^{Fehler! Textmarke nicht definiert.}). Ebenso muss bei Einsatz eines Single Sign On-Systemes je nach Kontext nicht nur die technische Seite betrachtet werden, sondern auch die psychologische (siehe [9], S.65f & [10]). Von dieser Betrachtungsweise aus kann eine Ablehnung einer zentralisierten Authentifizierungsstelle psychologische Gründe haben, so möchten Nutzer explizit von-einander unabhängige Pseudoidentitäten um so bspw. privat und geschäftlich strikt voneinander zu trennen. Auch gilt es je nachdem ob SSO innerhalb eines Unternehmens oder im öffentlichen Internet benötigt wird, abzuwägen welche SSO-Technik verwendet wird. Selbst Lösungen die einst für SSO in beiden Bereichen geeignet waren, sollten nach einer Überarbeitung genauer analysiert werden. So ist bspw. OAuth 2.0 im Gegensatz zu OAuth 1.0 in der Standardumsetzung weniger sicher, da die Verschlüsselung der Authentifizierungsinformationen entfernt wurde und das ganze Verfahren nur noch auf Transportebene über TLS verschlüsselt wird (siehe Austrittsstatement einer der Mitbegründer von OAuth 1.0 sowie Anfangs auch OAuth 2.0¹³). Einer der Hauptentscheidungsgründe für eine der beiden Varianten wird die Integration in die

bestehende IT-Landschaft sein, sowie die zukünftige Erweiterung der selbigen. Ist es beispielsweise angedacht Webanwendungen auch per mobilen Endgeräten wie Tablets und Smartphones erreichbar zu machen, so könnte die Entscheidung eher zu OpenID Connect tendieren, da hierbei direkt das vom SSO-Server generierte Zugangstoken als JSON vorliegt und dadurch eine einfachere Verarbeitung auf den Geräten stattfinden kann. SAML benötigt anfangs mehr Konfigurationsaufwand, gilt jedoch als altbewährtes Verfahren als sicher. OpenID Connect muss erst noch ausreichender betrachtet werden um Aussagen bezüglich Sicherheit treffen zu können.

8 Literaturverzeichnis

- [1] Kerberos: The Network Authentication Protocol. Website, 2014. Online verfügbar unter <http://web.mit.edu/kerberos/>; Besucht am 10.01.2014.
- [2] OASIS Security Services (SAML) TC. Website, 2014. Online verfügbar unter https://www.oasis-open.org/committees/tc_home.php?wg_aabbrev=security; Besucht am 10.01.2014
- [3] Shibboleth. Website, 2014. Online verfügbar unter <https://shibboleth.net/>; Besucht am 10.01.2014
- [4] OAuth 2.0. Website, 2014. Online verfügbar unter <http://oauth.net/2/>; Besucht am 10.01.2014
- [5] OpenID Connect. Website, 2014. Online verfügbar unter <http://openid.net/connect/>; Besucht am 10.01.2014
- [6] Revar, A.G.; Bhavsar, M.D., "Securing user authentication using single sign-on in Cloud Computing," Engineering (NUiCONE), 2011 Nirma University International Conference on , vol., no., pp.1,4, 8-10 Dec. 2011
- [7] S. Xiong. "Web Single Sign-On System For WRL Company", Royal Institute of Technology (KTH) Stockholm, Sweden, 2005

¹² <https://blog.surfnet.nl/wp-content/uploads/2013/04/SURFnet-OpenID-Connect-1.1-.pdf>

¹³ <http://hueniverse.com/2012/07/oauth-2-0-and-the-road-to-hell/>

- [8] D. Stojceski. Konzeption einer Kerberos-basierten Single Sign-On Lösung für ein ausgewähltes Szenario im Hochschulbereich. Fachhochschule Bonn-Rhein-Sieg, Fachbereich Informatik, Bachelorthesis, Bonn-Rhein-Sieg, 2006
- [9] San-Tsai Sun, Yazan Boshmaf, Kirstie Hawkey, and Konstantin Beznosov. 2010. A billion keys, but few locks: the crisis of web single sign-on. In Proceedings of the 2010 workshop on New security paradigms (NSPW '10). ACM, New York, NY, USA, 61-72. 2010
- [10] San-Tsai Sun, Eric Pospisil, Ildar Muslukhov, Nuray Dindar, Kirstie Hawkey, and Konstantin Beznosov. What makes users refuse web single sign-on? an empirical investigation of OpenID. In Proceedings of Symposium on Usable Privacy and Security, July 2011
- [11] S. Langer. Sicherheit von passwortbasierten Authentifizierungssystemen. Hochschule für Angewandte Wissenschaften Hamburg, Fakultät Technik & Informatik, Bachelorthesis, Hamburg. 2013
- [12] V. Radha, D.H. Reddy. A Survey on Single Sign-On Techniques. In: Procedia Technology 4 (2012), 134–139. Hyderabad, Indien. 2012
- [13] J. Clerq. Single Sign-On Architectures. In Infrastructure Security: Lecture Notes in Computer Science Volume 2437, 2002, pp 40-58. 2
- [14] Ping Identity. A Standards-based Mobile Application IdM Architecture. Whitepaper, 2012
- [15] OpenID Connect Pressemitteilung. Website, 2014. Online verfügbar unter <http://openid.net/2014/02/26/the-openid-foundation-launches-the-openid-connect-standard/>; Besucht am 10.01.2014

Umgebungswahrnehmung von agentenbasierten simulierten Menschmodellen in virtuellen Welten im Kontext C3D*

André Antakli
Reutlingen University
Andre.Antakli@Student.
Reutlingen-University.DE

Abstract

Das Ziel dieser Arbeit war die Umsetzung eines Wahrnehmungssensors für Softwareagenten, die über ein virtuelles Menschmodell in einer dreidimensionalen Umgebung agieren. Hierbei sollen die Agenten über den Sensor in der Lage sein, semantische Informationen zu geometrischen Objekten in der Umgebung zu erhalten. Hierfür wurden zwei Verfahren umgesetzt, die das menschliche Sehen simulieren, indem Objekten erkannt werden wenn diese innerhalb eines Sichtfeld liegen. Ein Problem das dabei gelöst werden muss, ist die Identifizierung möglicher Verdeckungen der Objekte. Ein Ansatz dieses Problem zu lösen, ist der Ray-Tracing Ansatz welcher für das erste Verfahren umgesetzt wurde. Das zweite Verfahren verwendet den Occlusion-Culling Ansatz. Auswertungen beider Verfahren haben gezeigt, dass der Ray-Tracing Ansatz eine schnellere Laufzeit aufweist, der Occlusion-Culling Ansatz jedoch mehr unverdeckte Objekte im Sichtfeld erkennt.

Schlüsselwörter

Softwareagent, Wahrnehmung, Wahrnehmungssensor, virtuelle Welt, Sichtbarkeitsproblem

CR-Kategorien

I.2.10 [Vision and Scene Understanding]:
Perceptual reasoning

1 Einleitung

Die vorliegende Arbeit wurde im Rahmen des Projektes Collaborate3D (C3D) am deutschen Forschungszentrum für künstliche Intelligenz (DFKI) in Saarbrücken durchgeführt. Ziel dieses Projektes ist die Entwicklung eines Systems, in dem die kollaborative Bearbeitung von Problemstellungen in einer immersiven dreidimensionalen Umgebung möglich ist. Diese Umgebung wird mit XML3D¹ umgesetzt und über den Webbrowser dargestellt. Um ein Szenario simulieren zu können, werden in C3D intelligente Systeme verwendet um beispielsweise menschliches Verhalten nachzuahmen. [19] Ein Szenario welches in C3D betrachtet wird und zentraler Bestandteil dieser Arbeit ist, ist die Wegfindungssimulation eines Einkäufers in einem Supermarkt. Der simulierte Einkäufer erhält dafür eine Art Einkaufsliste mit Objekten die dieser selbstständig in der 3D-Welt ausfindig machen muss. Dabei findet die Wegfindung durch zuvor erlerntem Wissen und der Auswertung der Umgebung statt. Der Einkäufer wird mit Hilfe

¹XML3D: <http://xml3d.org>

Betreuer Hochschule: Prof. Dr. Uwe Kloos
Hochschule Reutlingen
Uwe.Kloos@Reutlingen-
University.de
Betreuer Firma: Ingo Zinnikus
DFKI GmbH
Ingo.Zinnikus@DFKI.DE

Informatics Inside 2014
Wissenschaftliche Vertiefungskonferenz 2014
30. April 2014, Hochschule Reutlingen
Copyright 2014 André Antakli

eines Softwareagenten, der in der dreidimensionalen Umgebung als virtuelles Menschmodell repräsentiert wird, simuliert. Zum Thema Softwareagent, siehe [7, 5].

Ziel der vorliegenden Arbeit war es, einem Softwareagenten im Zuge von C3D und im speziellen für das Supermarktszenario, die Möglichkeit zu geben, die dreidimensionale Umgebung wahrzunehmen, damit dieser anhand dieser Informationen Schlussfolgerungen ziehen und Aktionen planen kann. Für die Wahrnehmung der dreidimensionalen Welt wurden zwei Verfahren umgesetzt, die in diesem Paper erläutert werden. Nachfolgend wird von einem virtuellen Agenten gesprochen, wenn ein Softwareagent in einer dreidimensionalen Umgebung als virtuelles Menschmodell repräsentiert wird und mit diesem in dieser Umwelt agiert.

Im nachfolgenden Abschnitt werden drei verschiedene Methoden aus der Literatur vorgestellt, wie die Wahrnehmung eines virtuellen Agenten in einer dreidimensionalen Welt simuliert werden kann. In den Abschnitten 3 und 4 werden die für C3D umgesetzten Methoden vorgestellt und miteinander verglichen. Anschließend werden in Abschnitt 5 die Ergebnisse zusammengefasst und ein Ausblick auf die zukünftige Arbeit gegeben.

2 State of the Art

In [12, 6, 11] wird meist unter zwei Wahrnehmungsmethoden unterschieden: künstliche und synthetische Wahrnehmung. Wird eine reale Umgebung durch ein Computersystem wahrgenommen, spricht man von einer künstlichen Wahrnehmung. Diese wird meist in Robotersysteme eingesetzt, die ihre Umwelt über Sensoren auswerten müssen um in ihr agieren zu können. Hierdurch erhält das Robotersystem die semantischen Informationen und somit die Bedeutung von Objekten in dessen Umwelt, im Falle eines eingesetzten Kamerasystems, nur über die Auswertung der Bildinformationen. Die synthetische Wahrnehmung hingegen, ist die simulierte Wahrnehmung in einer virtuellen dreidimensionalen Welt. [6, 11] Die semantischen Informationen der virtuellen Welt können dabei dem virtuellen Agenten zum Beispiel über den Zugriff auf Daten-

bankinformationen bereitgestellt werden. [15] Im Gegensatz zu einem Robotersystem kann ein virtueller Agent somit Zugriff auf alle Informationen seiner virtuellen Umwelt haben. Nachfolgend werden drei Methoden vorgestellt, die bei der Simulation der Wahrnehmung von Agenten in virtuellen Welten eingesetzt werden.

2.1 Filterbasierte Methoden

Eine Methode wie ein virtueller Agent seine Umgebung wahrnimmt, ist die Filterung von Informationen beispielsweise anhand seiner Position in der virtuellen Welt. Die semantische Bedeutung der virtuellen Welt und deren interne Objekt-Beziehungen werden zum Beispiel in einer Datenbank gespeichert und je nach Zustand des Agenten für diesen freigegeben. [4, 2] In [4] wird die dreidimensionale Welt aus zwei Objekttypen aufgebaut: geometrische Objekte die visuell in der Szene vertreten sind und Objekte die in der Szene vorhanden aber nicht visuell repräsentiert werden. Die virtuelle Umgebung wird in Bereiche unterteilt die in semantischer Relation zueinander stehen. Diese Bereiche werden als nicht sichtbare Objekte in der Szene repräsentiert. Der Agent erhält hierdurch die semantischen Informationen eines Bereichs, über die Kennung des Objektes in der sich dessen virtuelles Menschmodell befindet. In [2] besitzt ein virtueller Agent mehrere Filter um seine Umwelt wahrzunehmen. So besitzt dieser beispielsweise einen Sichtfilter. Über diesen werden nur semantische Informationen von dreidimensionalen Objekten an den Agenten freigegeben, die sich in einem simulierten Sichtfeld befinden. Zudem werden Filter eingesetzt die nur für kontextspezifische Objekte gelten. Wie in [4] wird auch in [2] die virtuelle Welt hierarchisch in Bereiche unterteilt.

2.2 Falschfarben-Methode

Ein Vorgehen, bei dem das „Erkennen“ von Objekten simuliert wird, ist die sogenannte Falschfarben-Methode (eng.: False coloring) Dieses Verfahren wird zum Beispiel in [12, 6, 8, 10] eingesetzt. Bei dieser Methode lokalisiert der virtuelle Agent über ein

für ihn generiertes 2D-Bild Objekte in seiner Umgebung. Das Bild wird dabei für jeden virtuellen Agenten, abhängig von dessen Position und Ausrichtung gerendert. Die Objekte im gerenderten Bild besitzen dabei einen eindeutigen Farbwert, siehe Abbildung 1. Der Farbwert wird hierfür anhand der Objekt-Identifikationsnummer generiert. Wird bei der Bildanalyse ein Farbwert ermittelt, kann dieser wieder in eine Objekt-Identifikationsnummer übersetzt werden. Mit dieser Identifikationsnummer können im Anschluss die semantischen Objektinformationen aus zum Beispiel der Datenbank des Simulationssystems ausgelesen werden. Wird das Bild ausgewertet, erhält das Wahrnehmungsmodul des Softwareagenten Informationen über „sichtbare“ Objekte durch die in den Pixeln gespeicherten Farbwerte. Dieses Verfahren ähnelt somit dem künstlichen Wahrnehmung. Bei dem Renderingprozess werden Texturen oder andere Effekte wie Schattwurf nicht beachtet. [15, 6, 7, 11]

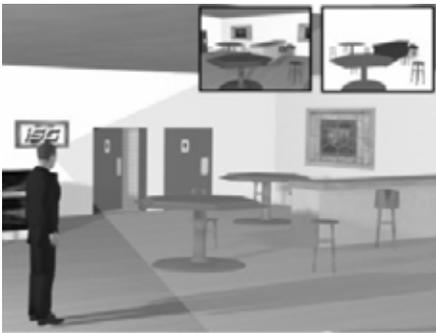


Abbildung 1: Falschfarbenmethode (oben links: Sichtfeld virtueller Agent; oben rechts: Falschfarbenversion) [12]

2.3 Ray-Tracing Methode

Die Raytracing-Methode ist ein weiteres Verfahren, das vor allem in der Spieleindustrie eingesetzt wird, um „Sehen“ virtueller Spielfiguren zu simulieren [11]. Bei der Raytracing-Methode werden mit Hilfe sogenannter Strahlen (engl.: Rays) Objekte in einem Sichtfeld lokalisiert. Ein Strahl besteht dabei aus

zwei Elementen: Richtungvektor und Ausgangspunkt. Der Ausgangspunkt des Strahls ist zugleich der Standort des virtuellen Agenten in der Szene. Hat ein Strahl Schnittpunkte mit einem Objekt in der Szene, gilt es als sichtbar. Schneidet der Strahl mehrere Objekte gleichzeitig, gilt nur jenes Objekt als sichtbar, dessen Distanz zum virtuellen Menschmodell, am geringsten ist. Das Sichtfeld des Agenten entsteht dadurch, dass mehrere verschiedene ausgerichtete Strahlen gleichzeitig eingesetzt werden, siehe hierzu Abbildung 2. [11, 14, 3]

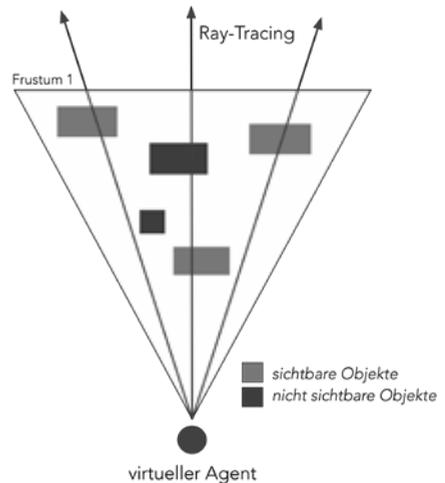


Abbildung 2: Ray-Tracing

3 Umsetzung

Für die Umsetzung der Wahrnehmung des virtuellen Agenten, wird ein Sensor umgesetzt, der wie in [4, 2] die geometrischen Informationen der dreidimensionalen Szene auswertet. Dabei identifiziert der Sensor Objekte in einem Sichtfeld abhängig der Position des virtuellen Menschmodells und dessen Ausrichtung. Wurden die Objekte beziehungsweise deren eindeutige Kennung vom Sensor identifiziert, liest dieser anhand der Kennungen die semantischen Informationen der Objekte aus einer Datenbank aus und leitet diese Informationen an den Softwareagenten weiter.

Die nachfolgend vorgestellten Verfahren verwenden nicht die Geometrie der Objekte in der Szene, sie benutzen stattdessen deren Hüllkörper beziehungsweise Boundingboxen, um mit Hilfe derer geringeren geometrischen Komplexität die Verfahren, zu beschleunigen. Die verwendeten Boundingboxen sind achsenorientiert und werden durch zwei Punkte definiert, einem n - und p -Vertex. Der n -Vertex ist dabei der Eckpunkt der Boundingbox, der die niedrigsten Koordinatenwerte aufweist. Der p -Vertex beschreibt den Punkt mit den größten Koordinatenwerten. [1]

3.1 View-Frustum Culling

Da das menschliche Verhalten bei der Wegfindung simuliert wird, muss der Informationszugriff über die Umgebung des virtuellen Agenten eingeschränkt werden. Dies wird damit begründet, da ein Mensch nur ein eingeschränktes Sichtfeld besitzt und somit beispielsweise Objekte die hinter ihm liegen, nicht wahrnehmen kann. Aus diesem Grund wird ein Sichtfeld wie in [2] simuliert, welches aus der Position und der Orientierung des virtuellen Menschmodells resultiert. Um dies zu realisieren, wird ein Verfahren aus der Computergrafik verwendet, das View-Frustum Culling [1]. Bei dieser Methode wird ein Sichtvolumen beziehungsweise Frustum erzeugt, mit dessen Hilfe Objekte die außerhalb des Sichtvolumens liegen, ausgeschlossen werden. Ein Frustum besteht aus sechs Flächen, die aus der Position und der Ausrichtung des virtuellen Menschmodells erzeugt werden. [1] Wie in Abbildung 3 zu sehen ist werden drei Vektoren f , u und r generiert. Der erste Vektor ist der Richtungsvektor f welcher die selbe Ausrichtung besitzt, wie das virtuelle Menschmodell C . Zusätzlich zu diesem, werden zwei Normalvektoren r und u erzeugt die orthogonal zum Richtungsvektor liegen. Anhand des Richtungsvektors und einer definierten Distanz d wird ein Punkt F errechnet. Ausgehend von diesem und der definierten Breite ($width$) und Höhe ($height$) der Flächen die orthogonal zum Richtungsvektor liegen, werden die Eckpunkte P des Sichtvolumens errechnet.

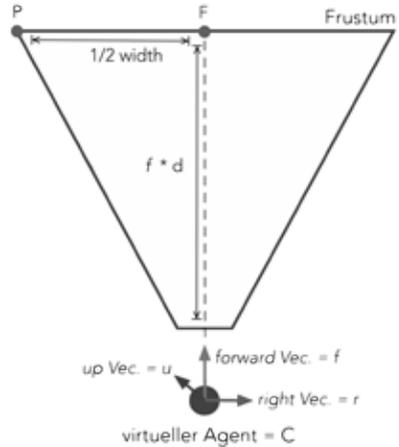


Abbildung 3: Ermittlung der Eckpunkte eines View-Frustums

$$F = C + f * d$$

$$P = F + (1/2 * height) * u - (1/2 * width) * r$$

Mithilfe zweier Eckpunkte A und B und dem Ausgangspunkt D wird im Anschluss die Normale n der Fläche aus A , B und D errechnet und anhand dieser getestet, ob ein Objekt innerhalb oder außerhalb des Sichtvolumens liegt.

$$n = \vec{DA} * \vec{DB}$$

Bei dem Test ob ein Objekt innerhalb des Sichtvolumens liegt, wird der orthogonale Abstand d der Eckpunkte M der Boundingbox des Objektes zu den sechs Flächen des Frustums errechnet. Zeigen die Normalen der Flächen des Sichtvolumens nach außen, liegt ein Punkt M außerhalb des Volumens, wenn dessen orthogonale Distanz d zu einer Fläche einen positiven Wert besitzt.

$$|n| = \sqrt{n^2}$$

$$d = \frac{n * M - 1}{|n|}$$

3.2 Ray-Tracing

Durch das View-Frustum Culling kann überprüft werden, ob ein Objekt innerhalb des Sichtfeldes liegt. Verdeckungen der Objekte untereinander werden hierbei nicht erkannt. Da nur „sichtbare“ Objekte für den virtuellen Agenten relevant sind, um ein möglichst menschliches Verhalten zu simulieren, müssen die vollständig verdeckten Objekte identifiziert werden. Ein Verfahren, das das Sichtbarkeitsproblem löst, ist das aus Kapitel 2.2 erwähnte Ray-Tracing, welches im Zuge dieser Arbeit für den Sichtsensor des Softwareagenten realisiert wurde.

Ein Strahl besteht aus einem Ausgangspunkt und einer Ausrichtung. Um zu untersuchen, ob eine Boundingbox von einem Strahl mit Ausrichtung d und Richtungsvektor o geschnitten wird, werden der n -Vertex und der p -Vertex ($p1$ und $p2$) der Boundingbox mit dem Strahl verrechnet, wodurch sich Schnittpunkte max und min ergeben:

$$min = \frac{p1 - o}{d}$$

$$max = \frac{p2 - o}{d}$$

Um ein Sichtfeld zu erzeugen, sind mehrere Strahlen mit verschiedenen Ausrichtungen notwendig. Hierfür werden maximale Höhe und Breite des Sichtfeldes definiert und die Anzahl der Strahlen in der Höhe und Breite festgelegt. Anhand dieser Angaben und der Position des virtuellen Menschmodells und dessen Ausrichtung, können die einzelnen Punkte für das Sichtfeld und die Ausrichtung der Strahlen errechnet werden, siehe Abbildung 4.

3.3 Occlusion-Culling

Das Rendern eines Bildes und das anschließende Auswerten dieses, gilt als zeitintensiv [vgl. 3, S. 11]. Da bei dem betrachteten Szenario mehrere Agenten gleichzeitig agieren und dabei ihre dreidimensionale Umwelt verzögerungsfrei wahrnehmen müssen, wurde entschieden den Occlusion-Culling Ansatz zu testen, um das Sichtbarkeitsproblem zu lösen [3, 1, 9]. Hierbei werden potenziell sichtbare

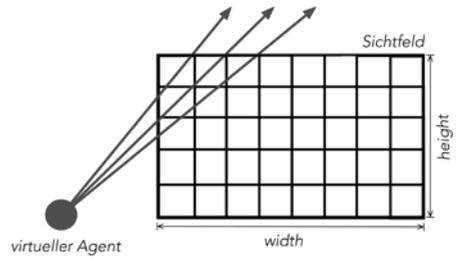


Abbildung 4: Erstellen eines Ray-Sichtfeldes

Objekte (eng.: potentially visible objects, kurz PVO) mit Hilfe eines View-Frustums ermittelt. Anschließend müssen Occluder (dt.: Verdeckter) identifiziert werden. Anhand dieser wird ein sogenanntes Occludee beziehungsweise Verdeckungsvolumen erzeugt, mit dessen Hilfe verdeckte PVOs, also Objekte die sich vollständig innerhalb des Verdeckungsvolumens befinden, als „sichtbar“ ausgeschlossen werden können. [3, 1, 9, 18] Siehe hierzu Abbildung 5. Um das Occlusion-Culling

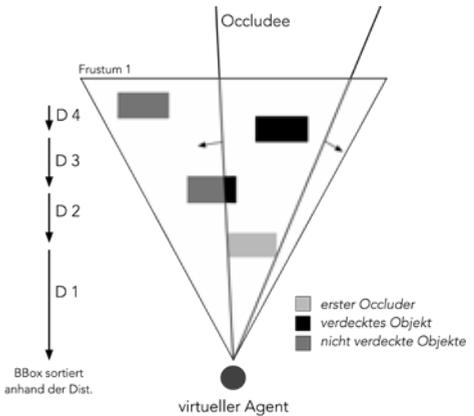


Abbildung 5: Ermittlung des erst zu betrachtenden Occluder und Erzeugung eines Verdeckungsvolumens

zu realisieren, werden mit dem in Kapitel 3.1 vorgestellten View-Frustum, die POVs ermittelt. Die Liste der PVOs wird anschließend

anhand derer Distanz zur Position des virtuellen Menschmodells hin sortiert. Im nächsten Schritt wird das PVO, welches den geringsten Abstand zum Menschmodell besitzt, aus der Liste ausgewählt. Das Verdeckungsvolumen wird anschließend aus den Extrempunkten der Boundingbox (BBox) des ausgewählten POVs aus Sicht des Menschmodells erstellt. Hierfür wurden die Boundingboxen um weitere Eckpunkte und einem Zentrum erweitert, wie in Abbildung 5 zu sehen ist. Für die Ermittlung der Extrempunkte, werden jeweils zwei Vektoren erstellt und anschließend deren Winkel zueinander berechnet. Für die Vektoren wird die Position des Menschmodells D , das Zentrum C und ein Eckpunkt P der Boundingbox verwendet. Der Vektor DC gilt als Vergleichsvektor, mit dessen Hilfe der Winkel zum Vektor DP errechnet wird. Wurden für alle Eckpunkte der Boundingbox diese Winkel ermittelt, werden je nach Orientierung des Objektes in Bezug zur Position des Menschmodells, vier beziehungsweise sechs Extrempunkte (Eckpunkte mit dem größten ermittelten Winkel) ausgewählt. Siehe hierzu Abbildung 6. Mithilfe dieser Ex-

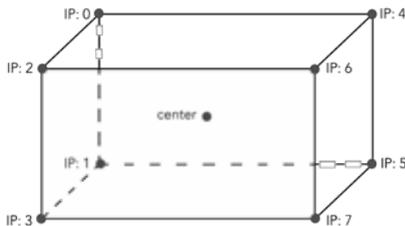


Abbildung 6: Erweiterte Boundingbox mit eindeutiger Kennung für Eckpunkte

trepunkte kann ein Frustum erstellt und alle übrigen PVOs auf eine Verdeckung getestet werden. Hierbei werden die einzelnen Eckpunkte der Boundingboxen dieser PVOs betrachtet und als verdeckt gekennzeichnet, falls diese innerhalb des Verdeckungsvolumens liegen. Ist ein Objekt vollständig verdeckt, sind also alle Eckpunkte der Boundingbox als verdeckt markiert, wird das betrachtete Objekt aus der Liste der PVOs entfernt. Wurden alle

PVOs betrachtet, sind nur noch Objekte in der Liste der PVOs enthalten, die von der Position des virtuellen Menschmodells „sichtbar“ sind.

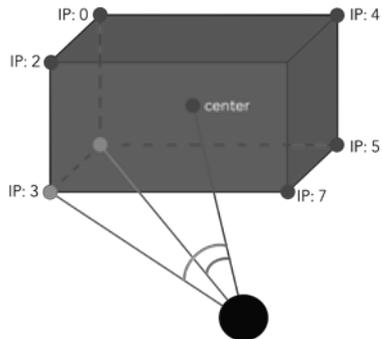


Abbildung 7: Ermittlung der Extrempunkte aus der Sicht des virtuellen Menschmodells. Der Punkt mit der IP: 3 gilt hier als Extrempunkt

4 Vergleich beider Verfahren

4.1 Laufzeitanalyse

Ray-Tracing Ansatz

Die Laufzeit des Ray-Tracing Ansatzes ist Abhängig von den erzeugten Strahlen und den zu testenden Objekten. In Abbildung 8 ist der Pseudocode des Ansatzes dargestellt.

Um einen Strahl zu erzeugen, ist die Position und die Ausrichtung des Strahls notwendig. Hierfür werden zwei *for*-Schleifen in den Zeilen 12 und 14 genutzt und anschließend die Ausrichtung des Strahls in Zeile 16 bis 20 errechnet. Diese Schleifen werden mit der Anzahl der definierten Zeilen und Reihen durchlaufen. In Zeile 22 wird der Funktion *findRayIntersection()* der zuvor definierte Strahl und die Liste der potenziell sichtbaren Objekte n aus dem ersten Frustumtest übergeben. Diese Funktion überprüft dabei, welche Objekte sich mit dem Strahl schneiden und gibt als Resultat das geschnittene Objekt zurück, das die geringste Distanz zum virtuellen Menschmodell aufweist. Der Algorithmus wird immer $columns \times rows \times n$ -mal durch-

```

1
2
3   RAY-TRACING(agentOrigin, agentDirection, objectsInViewfrustum, width, height, rows, columns)
4
5   visibleObjects[] = 0
6
7   right = generateRightvector(agentDirection)
8   up = generateUpvector(agentDirection)
9
10  center = agentOrigin + (agentDirection * distance)
11
12  for i = 0 to columns
13      for j = 0 to rows
14
15          point = center + (up * (height/2 - ((height/columns) * i))) - (right * (width/2 - (width/rows) * j))
16
17          direction = point - origin
18
19          ray[direction, origin]
20
21          hitObject = findRayIntersection(objectsInViewfrustum, ray)
22
23          visibleObjects.push(hitObject)
24
25
26  return visibleObjects

```

Abbildung 8: Pseudocode: Ray-Tracing Algorithmus

laufen. *columns* und *rows* sind Konstante, wodurch die Laufzeit des Algorithmus von der Anzahl der zu prüfenden Objekte abhängt. Für den Ray-Tracing-Algorithmus gilt die Laufzeitkomplexität $O(n)$.

Occlusion-Culling Ansatz

Der Occlusion-Culling Ansatz setzt voraus, dass die potenziell sichtbaren Objekte im Voraus nach deren Distanz zum Menschmodell sortiert sind. Der Sortieralgorithmus wird in dieser Ausarbeitung nicht näher betrachtet, für diesen gilt für n zu sortierende Objekte das Laufzeitverhalten $O(n(n-1)/2)$. Der Algorithmus, der die sortierten Objekte auf Verdeckungen überprüft, ist als Pseudocode in Abbildung 9 dargestellt.

In Zeile 7 wird eine *while*-Schleife maximal so oft durchlaufen, bis alle potenziell sichtbaren Objekte n betrachtet wurden. Hierfür wird bei jedem Durchgang, siehe Zeile 5 beziehungsweise 30, ein Objekt, beziehungsweise dessen Boundingbox aus der sortierten Liste entfernt und mit diesem in Zeile 9 ein neues Verdeckungsvolumen generiert. Ab Zeile 11 werden anschließend alle in der Liste verbliebenen Objekte darauf getestet, ob diese vollständig im Verdeckungsvolumen liegen. Hierfür werden zwei *for*-Schleifen benötigt. Die Schleife in Zeile 13 wird j -mal durchlaufen, wobei für j die Anzahl der zu prüfenden Ex-

trepunkte des zu testenden Objektes gilt. Die *for*-Schleife in Zeile 16 wird k -mal durchlaufen, wobei für k die Anzahl der Flächen des Verdeckungsvolumen gilt. Ab Zeile 17 wird überprüft, ob der ausgewählte Extrempunkt innerhalb des Verdeckungsvolumens liegt. Ist dies der Fall, wird dieser als „nicht sichtbar“ gekennzeichnet. Erst wenn alle Punkte als nicht sichtbar gekennzeichnet sind, wird das getestete Objekt in Zeile 27 aus der Liste der potenziell sichtbaren Objekte entfernt. Wenn kein Objekt ein anderes verdeckt, wird die *for*-Schleife in Zeile 11 maximal $(n-1)/2$ mal durchlaufen. Die Anzahl der Durchläufe der *for*-Schleifen in Zeile 13 und 16 sind konstant, da die Anzahl der zu prüfenden Extrempunkte der Boundingboxen und der generierten Flächen des Verdeckungsvolumen konstant sind. Für die Laufzeitanalyse sind somit nur die *while*-Schleife und die *for*-Schleife in Zeile 11 relevant, wodurch für die Laufzeitkomplexität $O(n(n-1)/2)$ gilt.

4.2 Laufzeittest

Die in Kapitel 3 vorgestellten Verfahren wurden in JavaScript umgesetzt, da die virtuelle Welt in XML3D realisiert ist und mit JavaScript direkt auf das virtuelle Menschmodell, die Objekte und deren Boundingboxen in der Szene zugegriffen werden kann. Nachfolgend werden die Ergebnisse eines Laufzeittests der

```

1  OCCLUSION-CULLING(sortedObjectsInViewFrustum, agentOrigin|
2
3  visibleObjects[] = 0
4  objectTest = 0
5  nearestObject = sortedObjectsInViewFrustum.pop
6
7  while nearestObject != 0
8
9      newFrustum = generateFrustum(nearestObject, agentOrigin)
10
11     for i = 0 to sortedObjectsInViewFrustum.länge
12
13         for j = 0 to sortedObjectsInViewFrustum[i].anzahlPunkte
14             testPoint = sortedObjectsInViewFrustum[i].point[j]
15
16             for k = 0 to newFrustum.anzahlFlächen
17                 distancePoint = getDistanceToPlane(newFrustum.plane[k], testPoint)
18
19                 if distancePoint > 0 then
20                     testPoint.visibility = 1
21                     objectTest = 1
22
23                 else
24                     testPoint.visibility = 0
25
26             if objectTest != 1
27                 sortedObjectsInViewFrustum.remove(sortedObjectsInViewFrustum[i])
28
29     visibleObjects = visibleObjects + nearestObject
30     nearestObject = sortedObjectsInViewFrustum.pop
31

```

Abbildung 9: Pseudocode: Occlusion-Culling Algorithmus

umgesetzten Verfahren vorgestellt. Der Laufzeittest wurde auf einem Windows 7 System mit einem Intel Core i7 bei einer Taktung von 2,80 GHz und 8 GB Arbeitsspeicher durchgeführt. Als Browser wurde Chrome in Version 33.0.1750.146 m verwendet. Bei einem Testlauf läuft der virtuelle Agent eine definierte Wegstrecke ab und untersucht dabei Objekte in der Szene auf deren Sichtbarkeit. Siehe hierzu Abbildung 9. Bei jedem Zeitintervall beziehungsweise bei jeder Positionsänderung des virtuellen Menschmodells wird das zu testende Verfahren eingesetzt. Bei einem Zeitintervall wird untersucht, wie viele Objekte als sichtbar markiert werden und wie viel Zeit dafür benötigt wird. Es wurden zwei Szenarien untersucht. Beim ersten Szenario waren 47 Objekte im Szenario enthalten. Diese wurden so platziert, dass es keine Verdeckungen gab, damit der Occlusion-Ansatz alle Objekte als Occluder ansieht. Im zweiten Szenario, wurden 310 Objekte in der Szene verwendet die 1/7 der Größe der Objekte im ersten Szenario entsprachen. Bei einem Testlauf wurden 12 x 60 Zeitintervalle untersucht. Für jedes Szenario wurden 12 Testläufe durchgeführt. Die Auswertung der Tests ist in Tabelle 1 und 2 im Appendix dargestellt. Werden die Ergeb-

nisse der Laufzeittests betrachtet, so ist zu erkennen, dass der Laufzeitunterschied der Ansätze bei einer erhöhten Anzahl an Objekte erkennbar ist. Der Anstieg der Gesamtzeit entspricht jedoch nicht dem Laufzeitverhalten der Verfahren. So stieg die Gesamtzeit bei dem Ray-Tracing Ansatz nicht linear mit der Anzahl der Objekte an. Dies resultiert aus Konstanten die ebenfalls in die Zeitermittlung einfließen. So werden immer 140 Strahlen unabhängig der Objektanzahl n erzeugt. Der Laufzeittest hat zudem gezeigt, dass die Genauigkeit und die Laufzeit von der Objektgröße abhängt. So wurden bei dem Ray-Tracing Ansatz 40 % der zu sehenden Objekte als sichtbar markiert, wenn diese um das siebenfache verkleinert wurden, die Anzahl der Strahlen jedoch konstant blieb. Siehe hierzu Abbildung 10 im Appendix. Dies resultiert aus der Anzahl und Verteilung der Strahlen im Sichtfeld. Die Größe eines Objektes beeinflusst auch den Occlusion-Culling Ansatz. Verdeckt beispielsweise das erste zu untersuchende Objekt alle anderen Objekte, werden diese aus der Liste der PVOs entfernt und somit nicht weiter betrachtet.



Abbildung 10: Szenario: Laufweg (Linie) des virtuellen Agenten und dessen gesehene Objekte (hell)

5 Fazit und Ausblick

Der Vergleich beider Verfahren hat gezeigt, dass das Ray-Tracing Verfahren prinzipiell schneller als das Occlusion-Culling ist, dessen Schnelligkeit beziehungsweise Genauigkeit von der Größe der Objekte in der Szene abhängt. Eine genauere Beurteilung der Verfahren, kann erst vollzogen werden, wenn die dreidimensionale Umgebung für das Szenario erstellt wurde. Denkbar ist der Einsatz beider Verfahren in verschiedenen Situationen. Der Ray-Tracing Ansatz kann verwendet werden, wenn der virtuelle Agent lediglich große Objekte, wie beispielsweise Regale oder markante Punkte im Raum wahrnehmen muss, um sich im Raum zu orientieren. Der Occlusion-Culling Ansatz, kann eingesetzt werden, wenn der virtuelle Einkäufer zum Beispiel kleine Objekte in einem Regal Wahrnehmen muss, da er eine bestimmte Ware sucht. Um beide Verfahren in deren Laufzeit zu optimieren, kann der Ansatz der hierarchischen Unterteilung des Raumes wie in [4, 2] verfolgt werden. Hierdurch müssen nur die Objekte auf deren Sichtbarkeit getestet werden, wenn deren Elternobjekt als sichtbar gelten. Weitere Ansätze zur Einteilung des Raumes, sind das BSP-, das Octree- oder das K-D-Baum-Verfahren [16, 13, 17].

Literatur

- [1] Ulf Assarsson and Tomas Möller. Optimized view frustum culling algorithms for bounding boxes. *Journal of Graphics Tools*, 5:9–22, 2000.
- [2] Christophe Bordeux, Ronan Boulic, and Daniel Thalmann. An efficient and flexible perception pipeline for autonomous agents, 1999.
- [3] Daniel Cohen-Or, Yiorgos L. Chrysanthou, Cláudio T. Silva, and Frédo Durand. A survey of visibility for walkthrough applications. *IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER*, 9, 2003.
- [4] Nathalie Farenc, Ronan Boulic, and Daniel Thalmann. An informed environment dedicated to the simulation of virtual humans in urban context. In *PROCEEDINGS OF EUROGRAPHICS'99*, pages 309–318, 1999.
- [5] Martin Griss, Edited George, T. Heineman, Ph. D, Ph. D, and William Council. Software agents as next generation software components. In *In "Component-Based Software Engineering," Edited by*, pages 641–657. Addison-Wesley, 2001.
- [6] Hansrudi Noser and Daniel Thalmann. Synthetic vision and audition for digital actors. In *Proc. Eurographics '95*, pages 325–336, 1995.
- [7] Hyacinth S. Nwana. Software agents: An overview. *Knowledge Engineering Review*, 11:205–244, 1996.
- [8] Sejin Oh, Woonhyuk Baek, and Woon-tack Woo. Synthetic vision-based perceptual attention for augmented reality agents, 2010.
- [9] Ioannis Pantazopoulos and Spyros Tzafestas. Occlusion culling algorithms: A comprehensive survey. *Journal of Intelligent and Robotic Systems*, 35:123–156, 2002.

- [10] Christopher Peters. A perceptually-based theory of mind for agent interaction initiation., 2006.
- [11] Christopher Peters, Ginvera Castellano, Matthias Rehm, Elisabeth Andre©, Amaryllis Raouzaïou, Kostas Rapantzikos, Kostas Karpouzis, Gaultiero Volpe, Antonio Camurri, and Asimina Vassalou. Fundamentals of agent perception and attention modelling. In *Emotion-Oriented Systems*, pages 293–319. Springer Berlin Heidelberg, 2011.
- [12] Christopher Peters and Carol O’Sullivan. Synthetic vision and memory for autonomous virtual humans, 2002.
- [13] Hanan Samet. Implementing ray tracing with octrees and neighbor finding. *Computers And Graphics*, 13:445–460, 1989.
- [14] T. Steel, Dane Kuiper, and Rym Z. Wenkstern. Virtual agent perception in multi-agent based simulation systems. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, volume 2, 2010.
- [15] Daniel Thalmann, Nathalie Farenc, and Ronan Boulic. Virtual human life simulation and database: why and how. pages 471–479, 1999.
- [16] Csaba D. Tóth. Binary space partitions: Recent developments, 2005.
- [17] Ingo Wald and Vlastimil Havran. On building fast kd-trees for ray tracing, and on doing that in $o(n \log n)$. In *IN PROCEEDINGS OF THE 2006 IEEE SYMPOSIUM ON INTERACTIVE RAY TRACING*, pages 61–70, 2006.
- [18] Hansong Zhang. Effective occlusion culling for the interactive display of arbitrary models, 1998.
- [19] Ingo Zinnikus, Xiaoqi Cao, Matthias Klusch, Christopher Krauss, Andreas Nonnengart, Torsten Spieldenner, and Philipp Slusallek. A collaborative virtual workspace for factory configuration and evaluation, 2013.

APPENDIX

Tabelle 1: Testszenario 1: 47 Objekte in der Szene ohne Verdeckungen

	Zeit in ms	Größe Objekt	Identifikation sichtbarer Objekte	n Objekte	Tests pro n	Komplexität	Tests gesamt pro Zeitintervall
View-Frustum	Ø 0,2485	1/1		47	1	$O(n)$	47
Ray-Tracing	Ø 1,637	1/1	100 %	ca. 8	140	$O(n)$	1120
Occlusion-Culling	Ø 0,656	1/1	100 %	ca. 8	48	$O((n(n-1)/2))$	1344

Tabelle 2: Testszenario 1: 310 Objekte bei 1/7 Größe ohne Verdeckungen

	Zeit in ms	Größe Objekt	Identifikation sichtbarer Objekte	n Objekte	Tests pro n	Komplexität	Tests gesamt pro Zeitintervall
View-Frustum	Ø 0,3235	1/7		310	1	$O(n)$	47
Ray-Tracing	Ø 1,957	1/7	ca. 40%	ca. 43	140	$O(n)$	1120
Ray-Tracing	Ø 19,524	1/7	ca. 100%	ca. 43	2125	$O(n)$	91375
Occlusion-Culling	Ø 2,850	1/7	ca. 100%	ca. 43	48	$O((n(n-1)/2))$	43344

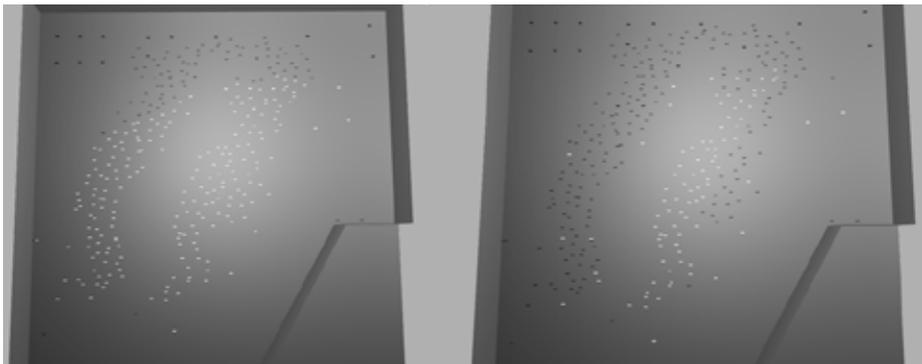


Abbildung 11: Genauigkeitsvergleich: 310 Objekte bei 1/7 Größe. Occlusion-Culling (links), Ray-Tracing mit 140 Strahlen (rechts)

Steigerung von Immersion durch die Integrierung von virtuellen Menschen in die virtuelle Welt NeueWelt9*

Lars Schneider
Reutlingen University
Lars.Rainer.Schneider@Student.
Reutlingen-University.DE

Abstract

Faktoren, die die Intensivität von Immersion in virtuellen Welten steigern, sind unter anderem der Detailreichtum der Umgebung, die Vorahnung und Wahrnehmung von Ereignissen, der Realismus der Welt und der Umfang von Interaktionen [4]. Zur Steigerung dieser Faktoren wurde eine Projektarbeit im Rahmen des Masterprojektes „NeueWelt9“ an der Hochschule Reutlingen durchgeführt. Erreicht wurde dies über die Integrierung von virtuellen Menschen in die virtuelle Welt NeueWelt9.

Es erfolgte eine Evaluation der Grafiken-Engine (CryEngine FreeSDK v3.4.5) und der Vorgehensweisen zur Entwicklung von dynamischen Verhaltensweisen virtueller Menschen. Es fand eine konkrete prototypische Umsetzung statt, wobei die Navigation und die Interaktionen von virtuellen Menschen implementiert wurden.

Schlüsselwörter

Virtuelle Welt, Künstliche Intelligenz, virtuelle Navigation, Verhaltensweisen, multiagenten Systeme, virtuelle Menschen, Grafiken-Engine, CryEngine

*

Betreuer Hochschule: Prof. Dr. Gabriela Tullius
Hochschule Reutlingen
Gabriela.Tullius@Reutlingen-
University.de

Informatics Inside 2014
Wissenschaftliche Vertiefungskonferenz
30. April 2014, Hochschule Reutlingen
Copyright 2014 Lars Schneider

CR-Kategorien

H.5.1 [INFORMATION INTERFACES AND PRESENTATION]: Multimedia Information Systems; I.2.11 [ARTIFICIAL INTELLIGENCE]: Distributed Artificial Intelligence

1 Das Projekt NeueWelt9

NeueWelt9 ist als Masterprojekt ein Bestandteil des Masterstudiums Medien- und Kommunikationsinformatik. Entwickelt wird eine interaktive multimediale virtuelle Welt, in der das Gebäude 9 der Hochschule Reutlingen spielerisch entdeckt werden kann. Derzeitig wird die Architektur und die Inneneinrichtung des Gebäudes mittels statischen 3D Modellen digital nachgebaut. Das betrifft Vorlesungsräume, Laborräume, Büroräume und Flure. Die Immersion wird derzeitig hauptsächlich durch eine möglichst realistische 3D Modellierung der virtuellen Welt gefördert.

Für die Entwicklung wird auf die CryEngine FreeSDK v3.5¹ zurückgegriffen, die als Grafiken-Engine dient. Deren Zweck ist es über integrierte Berechnungen, Funktionen und Werkzeuge die virtuelle Umgebung zu realisieren. Das betrifft Licht- und Schattenberechnungen, die Texturierung, die Darstellung von 3D Modellen, Animationen und Weiteres. Für die Integrierung von 3D Objekten in die Grafiken-Engine wird 3ds Max 2012 mit entsprechenden Schnittstellenwerkzeugen verwendet.

¹Zum Beginn der Projektarbeit war es v3.4.5



Abbildung 1: Virtuelle Studentinnen in NeueWelt9

2 Zielsetzung

In der virtuellen Welt NeueWelt9 fehlt es an Detailreichtum, Ereignissen und Interaktion. Darunter leidet besonders die Immersion. Deren Steigerung ist das Ziel dieser Projektarbeit. Die Umsetzung sollte über die Einbindung von virtuellen Menschen in die Umgebung erfolgen. Dadurch wird der Detailreichtum der Umgebung, die Menge an Ereignissen und die Interaktion gefördert. NeueWelt9 würde somit lebendiger wirken, mehr der Realität entsprechen und Vorgänge abbilden, die regelmäßig im Gebäude stattfinden.

Die virtuellen Menschen sollten wie Studenten agieren und deren Verhaltensweisen imitieren. Dies entspricht dem Kontext von NeueWelt9, dass „virtuelle Studenten“ das virtuelle Gebäude 9 beleben. Die virtuellen Studenten bekommen Vorlesungspläne zugewiesen. Entsprechend dem Vorlesungsplan sollten bestimmte Räume betreten und verlassen werden. Diese Prozesse dauern solange an, wie Vorlesungen anstehen.

Als weiterer Schritt sollte ein erster Ansatz zur Interaktion mit den virtuellen Studenten ermöglicht werden. Dazu sollte ein statischer virtueller Student integriert werden, der den Anwender wahrnehmen kann. Das ist das Fundament zur späteren Integrierung von

Dialogen zwischen Anwender und virtuellen Menschen.

Das Endprodukt der Projektarbeit soll eine prototypische Version von NeueWelt9 sein, bei der eine Integrierung von virtuellen Studentinnen erfolgen soll (Abb. 1). Dabei sollen zwei Typen von virtuellen Menschen entwickelt werden, die jeweils verschiedene Funktionen und Ereignisse in der virtuellen Welt abdecken sollen:

- **Statische virtuelle Menschen** reagieren und interagieren direkt mit der virtuellen Repräsentation des Anwenders. Hierbei soll eine virtuelle Studentin implementiert werden, die sich zum Anwender umdreht, wenn er in ihre Nähe kommt.
- **Autonome virtuelle Menschen** haben definierte Ereignisse und Prozesse, die sie durchlaufen. Dabei soll vorerst keine Einbindung des Anwenders stattfinden. Dem Kontext von NeueWelt9 entsprechend, sollen die virtuellen Menschen bzw. virtuelle Studenten, die für Studenten typischen Verhaltensweisen imitieren. Die Funktionalität soll sich hauptsächlich auf die Navigation durch die virtuelle Welt und die Interaktion mit Türen beschränken.

3 Betrachtung der CryEngine FreeSDK v3.4

Zur Bestimmung eines Ansatzpunktes für die Projektarbeit war eine Evaluation der Grafikkengine erforderlich. Das umfasst die Funktionalitäten, die von Grund auf verfügbar sind (Kap. 3.1, 3.2) und Vorgehensweisen zur Realisierung der Zielsetzung (Kap. 3.3).

3.1 Grundfunktionalitäten

Die CryEngine bietet bereits umfassende Funktionen und Werkzeuge zur Integrierung von virtuellen Menschen bzw. Agenten. Das umfasst einen exemplarischen 3D Charakter mit Animationen, ein Navigationssystem (Kap. 3.2), Audiospuren und definierte Ereignisse, die Animationen auslösen z.B. Kopfkratzen wenn der Agent im „Idle Status“ ist. Per Drag-and-Drop lässt sich ein Agent in das Level ziehen und positionieren, allerdings endet dort die Funktionalität. Jede weitere Funktion muss manuell in Form von Programmcode oder Flowgraphs (Kap. 3.3) implementiert werden.

3.2 Navigation

Seit der Version 3.4 ist in der CryEngine standardmäßig festgelegt, dass die Navigation von Agenten über die neuartige Multilayer Navigation (MNM) erfolgt. Dabei wird einmalig das Level beim Start komplett automatisiert abgetastet. Auf dieser Grundlage werden Bereiche im Level definiert, die ein Agent betreten darf. In Abb. 2 ist die MNM als dunkelgrauer rechteckiger Bereich auf dem Boden und der Treppe sichtbar. Die Navigationsfläche ist statisch und ändert sich nach dem Levelstart zu keinem Zeitpunkt.

Des Weiteren wird über die MNM die Wegfindung im Level koordiniert. Implementiert ist der A*-Algorithmus, [2] um den kürzesten Weg vom aktuellen Standort zum nächsten zu ermitteln. Die betretbaren Navigationsflächen sind Netze aus Trigonon, die als Grundlage für die Wegfindung dienen [2].



Abbildung 2: MNM in der CryEngine

3.3 Möglichkeiten Verhalten zu integrieren

Es gibt zwei Ansätze um eine Künstliche Intelligenz in der CryEngine zu realisieren. Es ist möglich über Lua-Skripte das Verhalten explizit zu programmieren. Dabei bekommt ein Agent ein Lua-Skript zugewiesen, das Klassen enthält, die mittels logischen Kontrollstrukturen sequenzielle Aktionen ausführen lassen. Jede dieser Klassen wird über ein oder mehrere Ereignisse angesprochen. Die Verknüpfung von den Ereignissen mit dem jeweiligen auszuführenden Code erfolgt in den Behavior Selection Trees (XML).

Die andere Möglichkeit ist das Verhalten der Künstlichen Intelligenz mittels Flowgraphs zu implementieren. Flowgraphs sind in der CryEngine das typische Werkzeug zur Integrierung von Logik, Ereignissen und Levelfunktion. Insbesondere die Interaktion in einem Level erfolgt ausschließlich über Flowgraphs. Sie sind eine Form der visuellen Programmierung. Sie bestehen aus Knoten, die verbunden sind und Informationen austauschen (Abb. 3).

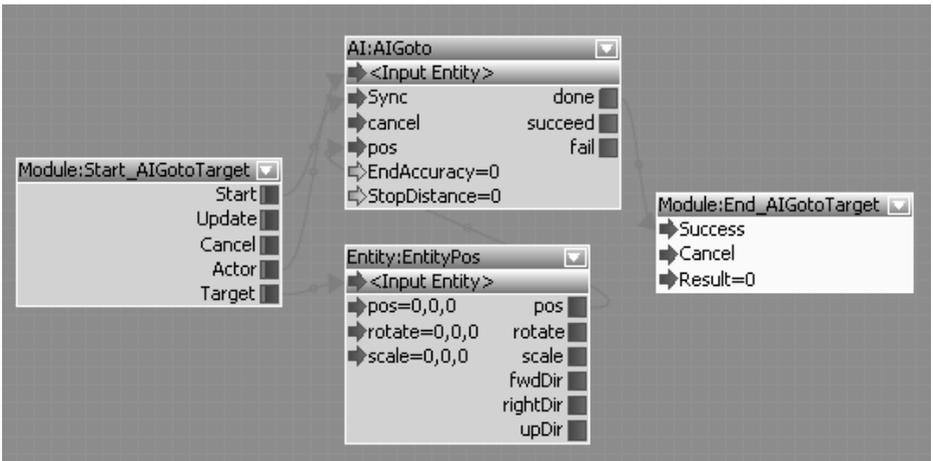


Abbildung 3: Flowgraphs

Für das Projekt fiel die Entscheidung auf Flowgraphs. Die Gründe sind, dass Flowgraphs leicht portierbar und nachbaubar sind. Sie sind leicht verständlich und lassen sich problemlos erweitern. Dadurch können Projektmitarbeiter von NeueWelt9 auf den Flowgraphs aufbauend weiter entwickeln.

4 Implementierung des Verhaltens

Der Kern der Projektarbeit ist die Entwicklung von Verhaltensmustern, die den virtuellen Menschen zugewiesen werden. Für die Umsetzung sind verschiedene kleinere Projekte abzuschließen - neben der eigentlichen Entwicklung der Verhaltensmuster (Kap. 4.2, 4.3). Zu Beginn war es notwendig 3D Charaktere in die Grafikkengine zu integrieren und sie im Level zu generieren und zu löschen (Kap. 4.1). Des Weiteren bestand ein Konflikt zwischen den Verhaltensmustern, die einen eigenen Zeitraum haben und der realen Geschwindigkeit der Zeit (Kap. 4.4). Optional fand eine Optimierung des Projektergebnisses statt, indem die Verhaltensmuster und Navigationswege für jeden virtuellen Menschen individualisiert wurden (Kap. 4.5).

4.1 Population des Levels

Autonome virtuelle Menschen (Kap. 2) sind für gewöhnlich in weiten und detaillierten virtuellen Welten vorhanden. Diese werden zum Levelstart generiert und erst beim Schließen des Levels gelöscht. Für NeueWelt9 sollte ein anderer Ansatz gewählt werden, der mehr der Realität entspricht: Studenten betreten kurz vor Vorlesungsbeginn das Gebäude und die Mehrheit verlässt es nach der letzten Vorlesung. Für jeden virtuellen Menschen gibt es einen individuellen Zeitpunkt t , zu dem er generiert wird und einen Zeitpunkt $t+1$, zu dem er entfernt wird. In NeueWelt9 gibt es einen dynamischen Tagesablauf, bei dem es Sonne- und Mondphasen, eine wandernde Sonne und sich veränderte Schatten gibt. Somit herrscht vormittags reger Betrieb im Gebäude, während es abends verlassen ist.

Es wurde auch untersucht, wie man virtuellen Menschen Eigenschaften zuweist, die sich im Laufe des Geschehens verändern lassen. Dazu wurde exemplarisch jedem Agenten die Eigenschaft „müde“ zugewiesen (Kap. 4.5). Des Weiteren gibt es Möglichkeiten die Population des Levels zu steuern:

- Die Anzahl der Agenten
- Die Orte an denen sie generiert werden
- Zeitpunkt der Generierung und die individuelle Verzögerung zwischen den Agenten
- Die „Kategorie“ (Student, Studentin, Professorin...) lässt sich festlegen und per Kontrollstrukturen auch steuern
- Das Verhalten bzw. der Vorlesungsplan wird bei der Generierung festgelegt

4.2 Interaktionen

Bei NeueWelt9 handelt es sich um eine virtuelle Welt, bei der der Entdeckungsdrang und der Informationsaustausch im Vordergrund stehen. Nach der Kategorisierung von [1] ist der Anwender von NeueWelt9 ein Entdecker (Abb. 4).

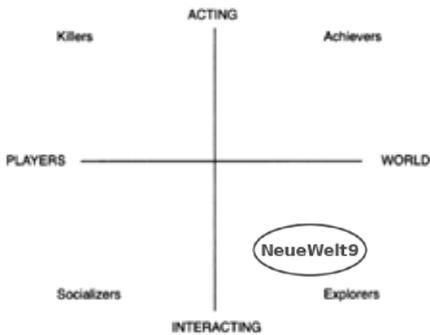


Abbildung 4: NeueWelt9 Einordnung [1]

Der Entdecker bevorzugt die Interaktion mit der virtuellen Umgebung, wobei die Belohnungen das Entdecken und der Informationsgewinn sind [1]. Für NeueWelt9 bedeutete das, dass die Interaktion mit Objekten und virtuellen Menschen maßgeblich das Erlebnis beeinflusst.

Die direkte Interaktion zwischen virtuellen Menschen und der virtuellen Repräsentation des Anwenders wurde in diesem Projekt nur in einer Form umgesetzt: Es gibt Agenten (statische virtuelle Menschen), die dynamisch auf den Anwender reagieren, sollte dieser in

ihre unmittelbare Nähe kommen. Vorerst drehen sich diese Agenten nur zu dem Anwender um (Abb. 5).



Abbildung 5: Reaktion auf den Anwender

Dieses Verhalten ließe sich mit dem Abspielen von Audiospuren und dem Erscheinen von Pop-Up Fenstern bzw. Dialogen leicht erweitern. Die Interaktion erfolgt durch Ereignisse [3], die initialisierend einen Interaktionsprozess anstoßen.

Die Interaktion lässt sich nicht nur auf die Beziehung zwischen statischen virtuellen Menschen und Anwendern beschränken. Es finden auch Interaktionen zwischen mehreren virtuellen Menschen oder einem virtuellen Menschen und der Umgebung (z.B. Tür) statt. Dabei rückt der Anwender in eine Beobachterrolle und nimmt nicht direkt an der Interaktion teil.

4.3 Sequenzieller Ablauf

Ein virtueller Mensch hat eine Menge an Prozessen zugewiesen bekommen. Jeder Prozess wird durch ein Signal angestoßen, das aufgrund von einem Ereignis übertragen wird.

Ein Signal wird durch die Interaktion mit der Umgebung, einen Agenten oder der virtuellen Repräsentation des Anwenders übertragen.

In der Projektarbeit sind die Ereignisse spezielle Uhrzeiten. Je nachdem wieviel Uhr es ist, wird ein Prozess in dem Agenten gestartet. Ein Prozess kann eine einfache Anweisung sein wie: „Gehe zu den Koordinaten x/y/z“. Beispielsweise: Ein Agent der Kategorie „MKIB1“ erhält um 11:30 Uhr von der Umgebung ein Signal mitgeteilt, woraufhin der Agent nach einer individuellen Verzögerung die Vorlesung verlässt.

Auf diese Weise wird der komplette Tagesablauf eines Agenten kontrolliert. Die Uhrzeit definiert die Handlungen der Agenten. Die Uhrzeit verläuft linear und entsprechend sind die Ereignisse sequenziell angeordnet. Tritt ein neues Signal auf, dann überlagert es das ältere Signal. Für das Projekt wurde ein exemplarischer Vorlesungsplan „MKIB1“ erstellt und Agenten zugewiesen (Abb. 6). Dieser wird bei der Generierung eines virtuellen Menschen diesem zugewiesen.

4.4 Stauchung des Zeitraumes

Eine virtuelle Welt lebt von Ereignissen und Interaktionen, deshalb war es für das Projekt notwendig den Zeitraum zu stauchen. Ansonsten hätte der Anwender 1,5 Stunden² warten müssen, um eine Änderung im Verhalten der Agenten zu bemerken. Deshalb lässt sich die Geschwindigkeit der Zeit aus der wirklichen Welt nicht auf die virtuelle Welt 1:1 übertragen. In der Projektarbeit vergeht die Zeit deshalb schneller, sodass das Gesamtbild der Prozesse nicht verloren geht.

Für andere Anwendungsszenarien wäre eine Abbildung der wirklichen Geschwindigkeit der Zeit auf die virtuelle Welt durchaus möglich. Es ließe sich per Flowgraph (node:realtime) die aktuelle Zeit auslesen und die Geschwindigkeit der Zeit auf die Wirkliche setzen.

²Das ist die Dauer einer Vorlesung

4.5 Verhalten und Navigation individualisieren

Der in Abb. 6 dargestellte Ablauf verzweigt sich bei 11:15 Uhr. Damit ist der sequenzielle Ablauf dynamisch. Für das Projekt sollte ein Verhalten implementiert werden, das über einen sequenziellen Ablauf hinausgeht. Dazu wurde bei der Generierung den Agenten eine Eigenschaft zugewiesen:

```
var bool:muede
```

Diese Variable wird zum Levelstart deklariert und zufällig mit wahr oder falsch initialisiert. Kommt der Agent im Ablauf an einen bestimmten Punkt, erfolgt eine Abfrage der Variable muede. Auf der Basis des Ergebnisses erfolgt eine Abzweigung des sequenziellen Ablaufes. Ist der Agent müde,

```
if (muede == true)
do take_break ;
```

dann führt er den Prozess "Pause" durch und entsprechend wird die Variable geändert. Sollte der Agent nicht müde sein,

```
if (muede == false)
do continue_class ;
```

dann fährt er direkt mit dem nächsten Prozess fort.

Ein weiterer Schritt, um das Verhalten der Agent zu individualisieren, war es, den Laufwegen der Agenten mehr Varianz zu geben. Die Konfiguration und der Navigationsalgorithmus der Multilayer Navigation lässt sich nicht bearbeiten. Üblicherweise wird ein Agent mit dem Befehl „gehe zu x/y/z“ durch das Level navigiert. Anstatt dessen wird ein zweidimensionaler Rahmen (Zahlenraum) definiert. In diesem Zahlenraum wird zufällig eine Navigationskoordinate ausgewählt und als Ziel angegeben (Abb. 7). Dem Agenten wird angegeben: „Gehe zu einem zufälligen Punkt im Zahlenraum x,y“. Somit ist jeder Laufweg in NeueWelt9 individuell.

5 Ausblick

Das Projekt war der erste Schritt zur festen Integrierung von virtuellen Menschen in NeueWelt9. Es wurde ein Prototyp entwickelt,

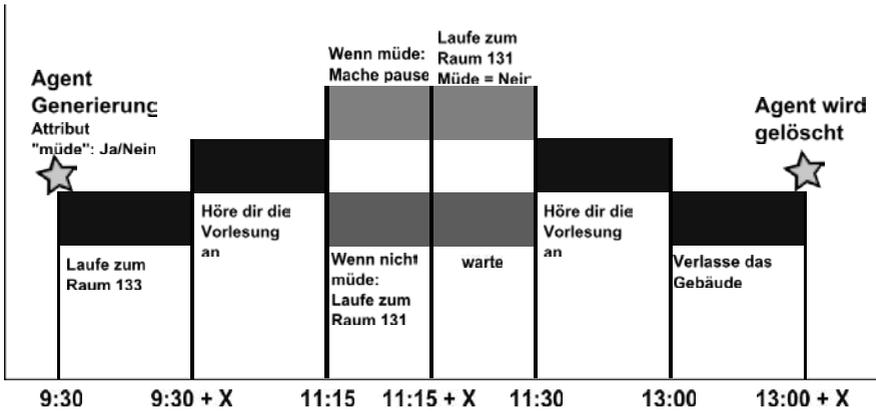


Abbildung 6: Sequenzieller Ablauf eines Agenten

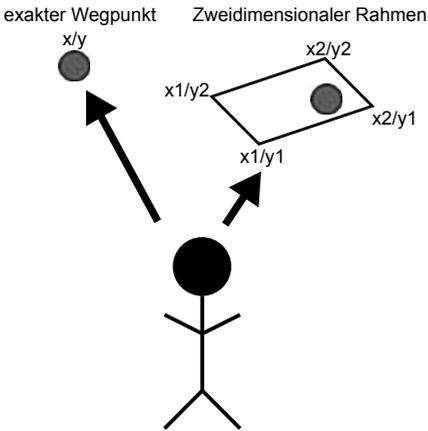


Abbildung 7: Varianz bei den Laufwegen

Es sind alle Funktionen implementiert, die in der Zielsetzung beschrieben sind: Implementiert ist ein statischer virtueller Mensch, der auf den Anwender reagiert. Auf der Grundlage eines sequenziellen Ablaufes navigieren die autonomen virtuellen Menschen durch die virtuelle Umgebung.

Für die Zukunft haben sich Punkte aufgetan, an denen - im Rahmen des Projektes NeueWelt9 - weiter geforscht und entwickelt werden sollte:

Mit der neuen Version 3.5 der Engine werden Charakteranimationen in neue Dateiformate gespeichert. Dadurch sind die standardmäßigen Animationen, die in der CryEngine verfügbar sind, stark beschränkt. Bei einem Wechsel auf Version 3.5 muss darauf geachtet werden, dass „zivile Animationen“ wie auch in der Version 3.4.5 verfügbar sind. Hierzu zählt auch die Entwicklung von vollständig eigenen Charakter Modellen. Das heißt, dass es eine Vielfalt an Charakteren geben soll wie Student/in, Professor/in und Mitarbeiter/in. Dabei wäre es auch möglich weitere Verhaltensmuster für die jeweiligen Charaktere zu integrieren - Bisher gibt es nur ein Verhaltensmuster: MKIB1 für Studenten.

Des Weiteren sollten die Interaktion zwischen Charakteren und dem Anwender erweitert werden. In Kap. 4.2 ist bereits der Ansatz zu sehen, wie es umgesetzt werden kann. Dabei ließen sich Dialoge und Audiospuren zum Informationsaustausch einbinden, wodurch der Entdeckungsdrang des Anwenders befriedigt wird. Zusätzlich ließe sich untersuchen, wie die Funktionalität von statischen und autonomen virtuellen Menschen kombinieren lässt.

In Kap. 4.5 wurde gezeigt, wie über Eigenschaften und Entscheidungswege die Dynamik und Individualität der Agenten gesteigert werden kann. Das müsste ausgebaut werden, wobei zu untersuchen wäre, wie die Individualisierung vom Anwender wahrgenommen wird und ob es weitere Vorgehen gibt, diese zu steigern.

Es sollte das Smart-Objects Systems genutzt werden um verschiedene Multilayer

Navigationsbereiche z.B. bei Türen zu überbrücken.

Literatur

- [1] R. Bartle. *Designing Virtual Worlds*. New Riders Games, 2003.
- [2] Crytek. Multi-layer navigation. Website, 2012. Online verfügbar unter <http://freesdk.crydev.net/display/SDKDOC2/Multi-layer+Navigation>; Besucht am 24.02.2014.
- [3] B. Ulicny and D. Thalmann. Crowd simulation for interactive virtual environments and vr training systems. In *Proceedings of the Eurographic Workshop on Computer Animation and Simulation*, pages 163–170, New York, NY, USA, 2001. Springer-Verlag New York, Inc.
- [4] B. G. Witmer and M. J. Singer. Measuring presence in virtual environments: A presence questionnaire. *Presence: Teleoper. Virtual Environ.*, 7(3):225–240, June 1998.

Erkennung eigener Körperbewegungen in animierten Charakteren

Anna Wellerdiek
Reutlingen University
Anna.Wellerdiek@Student.
Reutlingen-University.de

Abstract

Wenn es um die eigenen Körperbewegungen geht, sind Menschen sehr feinfühlig. Das ist eigentlich verwunderlich, denn im Alltag haben sie viel mehr Erfahrung damit die Bewegungen Anderer zu beobachten als die Eigenen. Deshalb wurde eine psychophysische Studie durchgeführt um die Erkennung eigener Körperbewegungen zu erforschen. Von mehreren Versuchspersonen wurden eine Vielzahl von Bewegungen (gehen, rennen, Tischtennis spielen und tanzen) mit einem Motion-Capture System aufgenommen und anschließend auf verschiedenen animierten Figuren dargestellt (Point-Light-Figur, männlicher und weiblicher Charakter). Die Versuchspersonen bekamen die Aufgabe ihre eigenen Bewegungen auf den Figuren wiederzuerkennen. Das Ergebnis zeigte, dass die Selbsterkennungs-Rate auf allen drei Figuren sehr hoch war, allerdings war sie höher wenn die Bewegungen auf einer Point-Light-Figur dargestellt wurden als auf einem animierten Charakter. Trotzdem haben die Versuchspersonen überraschenderweise angegeben, dass die Selbsterkennung auf der

Point-Light-Figur schwieriger gewesen sei als bei den animierten Charakteren. Außerdem hat die statistische Analyse der Daten ergeben, dass das Geschlecht des animierten Charakters keinen Einfluss auf die Selbsterkennung hatte.

Schlüsselwörter

Animation, Selbsterkennung, Motion Capture, virtuelle Charaktere, Geschlecht

CR-Kategorien

I.3.7 [Computer Graphics] Three-Dimensional Graphics and Realism, Animation

1 Einführung

Menschen haben viel Erfahrung damit, die Bewegungen von Verwandten, Freunden oder Fremden zu beobachten, aber die eigenen Bewegungen sehen sie Vergleichsweise selten. Trotzdem besitzen Menschen die Fähigkeit eigene Bewegungen aus der dritten Perspektive zu erkennen. 1973 hat Johansson eine Methode entwickelt mit der die biologischen Bewegungen eines Menschen dargestellt werden können: die Point-Light-Figur [1]. Er hat herausgefunden, dass bereits 10-12 kreisförmige Punkte, die an den Gelenken eines Menschen befestigt werden, für den Betrachter ausreichend Informationen liefern um verschiedene Bewegungen zu erkennen. Außerdem wurde gezeigt, dass diese Point-Light-Figuren es ermöglichen, einen bestimmten Darsteller [2] sowie sein Geschlecht zu erkennen [3]. Interessanterweise sind Menschen trotzdem besser darin, die eigenen Bewegungen zu

Betreuer Hochschule: Prof. Dr. rer. nat. Uwe Kloos
Hochschule Reutlingen
Uwe.Kloos@Reutlingen-
University.de
Betreuer Institut: Markus Leyrer
Max-Planck-Institut für biologische
Kybernetik
Markus.Leyrer@Tuebingen.Mpg.de
Informatics Inside 2014
Wissenschaftliche Vertiefungskonferenz
30. April 2014, Hochschule Reutlingen
Copyright 2014 Anna Wellerdiek

erkennen, als die von Freunden oder Fremden [4].

Welche Informationen werden benötigt um die eigenen Bewegungen aus der dritten Perspektive erkennen zu können? Zum einen gibt es Forminformationen. Diese beinhalten die genaue Position der Gelenke in Relation zur räumlichen Statur des Körpers (z.B. Schulter- oder Hüftbreite des Darstellers). Die Informationen zur Statur des Darstellers werden bei Charakteranimationen oft verändert oder komplett entfernt werden. Bewegungen, die mit einem Motion-Capture System aufgenommen wurden, können auf einem durchschnittlichen Charakterskelett dargestellt werden, wodurch alle Forminformationen des ursprünglichen Darstellers verloren gehen oder bewusst entfernt werden können. Weitere wichtige Informationen zur Selbsterkennung können die Bewegungen selbst enthalten, wie beispielsweise die dynamische und temporale Veränderung von Position und Orientierung der Gelenke [5]. Der dritte wichtige Punkt zur Selbsterkennung ist die visuelle Erscheinung der verwendeten Figur, auf der die Bewegungen dargestellt werden. Versuchspersonen haben die Tendenz dazu Bewegungen als natürlich zu bewerten, wenn sie auf einer Point-Light-Figur dargestellt werden. Wenn jedoch die gleichen Bewegungen auf einem menschenähnlichen Charakter dargestellt werden, bewerten die Versuchspersonen die Bewegungen eher als künstlich [6]. Hodgins und Kollegen haben mit einer Bewegungs-Unterscheidungs-Aufgabe gezeigt, dass Bewegungen besser unterschieden werden können wenn sie auf einem Charakter dargestellt werden als auf einer Point-Light-Figur [7]. Diese verschiedenen Erkenntnisse lassen zum einen vermuten, dass bei der Erkennung von eigenen Bewegungen die Verwendung von animierten Charakteren die Ergebnisse einschränken könnte, weil die Bewegungen eine künstliche Wirkung auf den Betrachter haben. Andererseits könnte es auch sein, dass die animierten Charaktere den Betrachter dabei unterstützen eigene Bewegungen von Fremden zu unterscheiden. Animierte

Charaktere geben mehr Informationen an den Betrachter weiter, als Point-Light-Figuren.

Bei der Verwendung von animierten Charakteren können die winkelförmigen Bewegungen von mehreren Personen unverändert beibehalten und auf einem durchschnittlich großen Charakter dargestellt werden. Allerdings kann dieser Ansatz dazu führen, dass einige Bewegungen leichte Fehler in der Darstellung aufweisen. Diese zeigen sich darin, dass beispielsweise die Hände während einer Bewegung durch den Oberkörper des Charakters hindurchgehen. Diese Fehler erscheinen vor allem dann, wenn die Statur des Darstellers stark von der Statur des Charakters abweicht.

Wenn Bewegungen auf animierten Charakteren dargestellt werden, kann die Animation zwei Geschlechter beinhalten: das Geschlecht der bewegungsausführenden Person sowie das Geschlecht des animierten Charakters. Wenn diese zwei Geschlechter nicht übereinstimmen, kommen bei dem Betrachter zweideutige Informationen bezüglich des Geschlechts an. Zibrek und Kollegen haben Emotionen auf virtuellen Charakteren hinsichtlich der Geschlechtererkennung untersucht. Dabei haben sie herausgefunden, dass das Geschlecht des Charakters die Erkennung der Emotionen beeinflusst. Angst und Traurigkeit wurden besser erkannt, wenn sie auf einem männlichen Charakter gezeigt wurden, als auf einem Weiblichen [8]. Weitere Studien haben gezeigt, dass die visuelle Erscheinung von animierten Charakteren das wahrgenommene Geschlecht der Bewegung [9] sowie die wahrgenommene Akzeptanz und Vertrauenswürdigkeit entgegen einem Charakter beeinflussen kann [10]. Diese Ergebnisse lassen vermuten, dass animierte Charaktere auch die Erkennung von eigenen Bewegungen beeinflussen könnten.

In der vorliegenden Studie wurde die Methodik von einer Darsteller-Erkennungsaufgabe von Loula und Kollegen repliziert. Darin wurde mit Point-Light-Figuren untersucht wie gut Personen darin sind sich selbst, Freunde und fremde Personen zu

erkennen [4]. Für die Studie in dem vorliegenden Artikel wurde eine Teilmenge der von Loula und Kollegen ausgewählten Bewegungen verwendet und ihre Studie wurde erweitert indem nicht nur Point-Light-Figures verwendet wurden, sondern auch ein männlicher und ein weiblicher Charakter. Außerdem wurde untersucht welchen Einfluss das Geschlecht der bewegungsausführenden Person, des Charakters und des Betrachters auf die Selbsterkennung hat.

2 Methodik

Die in diesem Artikel präsentierte Studie besteht aus zwei Teilen. Im ersten Teil wurden die Bewegungen der Teilnehmer aufgenommen und auf drei unterschiedlichen Figuren abgebildet (Point-Light-Figur, männlicher und weiblicher Charakter), siehe Abbildung 1.

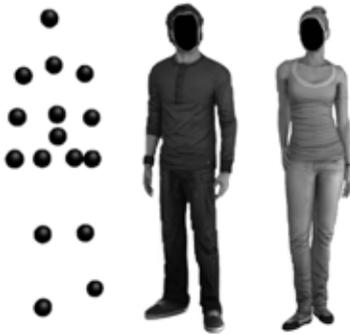


Abbildung 1: Die drei Figuren auf denen die Bewegungen dargestellt wurden.

Im zweiten Teil der Studie wurden alle Teilnehmer aus dem ersten Teil erneut eingeladen, um an der Selbsterkennungsaufgabe teilzunehmen. Hierbei mussten sie die Bewegungen aller Teilnehmer auf allen drei Figuren anschauen und daraus die eigenen Bewegungen erkennen. Dafür wurde eine Studie von Loula und Kollegen teilweise repliziert und erweitert [4].

2.1 Teil 1: Motion Capture und die Generierung der Animationssequenzen

In den folgenden Abschnitten wird die Vorgehensweise zum ersten Teil des Versuchs erläutert.

2.1.1 Versuchsteilnehmer

Es wurden Bewegungen von zwölf Teilnehmern (sechs davon weiblich) aufgenommen um daraus kurze Animationssequenzen zu generieren. Die Teilnehmer waren zwischen 19 und 30 Jahre alt ($\bar{x} = 24,1$). Sie waren alle Rechtshänder, hatten keine professionelle Erfahrung im Tanzen oder Tischtennis spielen und konnten alle Bewegungen problemlos ausführen. Die Fragestellung der Studie war ihnen nicht bekannt.

2.1.2 Verwendete Systeme

Für die Motion-Capture Aufnahmen wurde der MVN Anzug von Xsens verwendet [11]. Das System des MVN Anzug besteht aus 17 einzelnen Sensoren, die in einem Lycra-Anzug befestigt und mit Kabeln miteinander verbunden sind. Der Anzug erleichtert die korrekte Positionierung der Sensoren am Körper und ermöglicht es dem Nutzer zugleich alle Bewegungen uneingeschränkt durchzuführen. Jeder Sensor besteht aus einem Magnetometer, Gyrometer und einem Beschleunigungsmesser. Die aufgezeichneten Daten werden kabellos an den Computer gesendet, auf dem die Software MVN Studio sie verarbeitet und in Echtzeit auf einem menschlichen skelettähnlichen 3D-Modell dargestellt.

2.1.3 Bewegungskategorien

Von der replizierten Studie wurden sechs Bewegungen ausgewählt [4] und mit dem zuvor beschriebenen Motion-Capture System aufgenommen: (a) begrüßen: der Teilnehmer läuft auf den Versuchsleiter zu und schüttelt ihm die Hand, fünfmal nacheinander; (b) gehen: der Teilnehmer geht viermal im Aufnahmeraum hin und her; (c) rennen: der Teilnehmer rennt sechs Runden im Aufnahmeraum, drei im Uhrzeigersinn, drei

gegen den Uhrzeigersinn; (d) der Teilnehmer spiel Tischtennis gegen eine Wand des Tischtennis-Tisches; (e) spontanes tanzen (spon.) zu einem Popmusiktitel; (f) eine erlernte Choreographie tanzen (chor.).

Die Bewegungskategorie begrüßen diente nur zur Übung des Versuchs und hatte für die Datenanalyse keine Verwendung. Tischtennis spielen und eine Choreographie nachtanzen sind ungewöhnliche Bewegungen. Deshalb gab es vor der Aufnahme vom Tischtennis spielen für jeden Teilnehmer eine fünfminütige Übungsphase, gefolgt von einer zwei- bis vierminütigen Aufnahmezeit, je nach Können des Teilnehmers. Für das choreographische Tanzen wurde eine einfache Choreographie ausgewählt („Macarena“ / Los del rio), dessen Refrain für die Aufnahme siebenmal hintereinander abgespielt wurde. Vor der Aufnahme wurde die Choreographie mit jedem Versuchsteilnehmer geübt. Für das spontane Tanzen wurde ein populärer Musiktitel ausgewählt („Lady (hear me tonight)“ / Mojo) und die Aufnahme ging ca. zwei Minuten. Während der Aufnahme wurde der Raum abgedunkelt um den Teilnehmern die Scheu vor dem Tanzen zu nehmen. Unabhängig von der durchzuführenden Bewegung hatten die Teilnehmer die Anweisung sich so natürlich wie möglich zu bewegen.

2.1.4 Aufnahmeverfahren

Während den Aufnahmen waren immer zwei Versuchsleiter anwesend. Die Teilnehmer bekamen zu Beginn des Versuchs eine schriftliche Anweisung, mussten die Einverständniserklärung unterschreiben und den Motion-Capture Anzug anziehen. Nach einer erfolgreichen Kalibrierung begannen die Aufnahmen. Jede Bewegung wurde den Teilnehmern vorher erklärt und sie wurden in einer bestimmten Reihenfolge aufgenommen (begrüßen, gehen, rennen, Tischtennis spielen, tanzen (spon.), tanzen (chor.)).

2.1.5 Generierung der Animationssequenzen

Es wurden insgesamt sechs verschiedene Bewegungen von den Teilnehmern aufgezeichnet. Aus diesen langen Bewegungsaufnahmen konnten kurze Animationssequenzen, mit jeweils einer Länge von fünf Sekunden, herausgeschnitten werden. Pro Bewegungskategorie entstanden für jeden Teilnehmer fünf Sequenzen. Insgesamt wurden somit 300 verschiedene Animationssequenzen in der Selbsterkennungs-Aufgabe verwendet.

Alle erstellten Sequenzen wurden für die Selbsterkennungs-Aufgabe auf drei verschiedenen Figuren dargestellt (Point-Light-Figur, weiblicher und männlicher Charakter). Dabei wurden die Forminformationen der Darsteller entfernt, damit die Figuren für jeden Darsteller die exakt gleiche visuelle Erscheinung hatten. Die Gesichter der zwei Charaktere bekamen eine schwarze Textur um alle nicht animierten Merkmale zu verbergen und eventuell auftretende Seiteneffekte durch Gesichtsinformationen zu eliminieren.

2.2 Teil 2: Die Selbsterkennungs-Aufgabe

Im Folgenden wird die Methodik der Selbsterkennungs-Aufgabe erläutert.

2.2.1 Versuchsteilnehmer

Sechs bis elf Wochen ($\bar{\sigma} = 9,01$) nach den Bewegungsaufnahmen wurden alle Teilnehmer erneut eingeladen um an dem zweiten Teil der Studie teilzunehmen: der Selbsterkennungs-Aufgabe. Die große Zeitspanne zwischen den zwei Teilen wurde gewählt, um den Versuchspersonen die Erinnerung an bestimmte Bewegungen während der Aufnahme zu erschweren.



Abbildung 2: Beispiel für einen Versuchsdurchlauf

2.2.2 Vorbereitung und Durchführung des Versuchs

Durch die Implementierung mithilfe von Unity 4.0 [12], konnten den Teilnehmern die entsprechenden Animationssequenzen präsentiert werden. Die Darin eingebundene Versuchsanleitung und anschließende Übungsphase machten die Teilnehmer mit ihrer Aufgabe vertraut und bereiteten sie auf die Versuchsdurchläufe vor. Zu Beginn jedes Durchlaufes wurde für 500 ms ein Fixierungskreuz eingeblendet, gefolgt von einer fünf Sekunden langen Animationssequenz. Danach erschienen zwei Schaltflächen mit der Aufschrift „Me“ und „Not me“, siehe Abbildung 2. Die Aufgabe der Teilnehmer war es auszuwählen ob die gezeigte Bewegung von ihnen stammte oder nicht. Die Auswahl musste mit der Eingabetaste bestätigt werden. In der Übungsphase wurden zwölf Animationssequenzen der Kategorie begrüßen gezeigt, eine Sequenz von jedem Teilnehmer. Es wurde zufällig ausgewählt welche Bewegung auf welcher Figur erschien. Jede Figur kam vier Mal in der Übungsphase vor. In den folgenden Versuchsdurchläufen wurden die übrigen fünf Bewegungskategorien gezeigt (gehen, rennen, Tischtennis spielen, tanzen (spon.), tanzen (chor.)). Von jedem Teilnehmer wurden fünf Animationssequenzen pro Bewegung gezeigt, daraus ergeben sich 25 Sequenzen pro Teilnehmer. Diese wurden wiederum auf allen drei Figuren dargestellt, somit wurden von jedem Teilnehmer 75 Animationssequenzen in der Selbsterkennungs-Aufgabe gezeigt. Insgesamt ergeben sich daraus 900 gezeigte Sequenzen (12 Teilnehmer). Die Reihenfolge des gezeigten Darstellers, die Animations-

sequenz und die dargestellte Figur wurden zufällig ausgewählt. Die Durchführung der Selbsterkennungs-Aufgabe dauerte zwei bis drei Stunden, deshalb wurden die Teilnehmer immer nach 60 Versuchsdurchläufen (ca. 10 min) aufgefordert eine kurze Pause zu machen um Ermüdung zu vermeiden. Bei Unklarheiten konnten sich die Teilnehmer während dem Versuch jederzeit an den Versuchsleiter wenden. In einem anschließenden Fragebogen wurden die Teilnehmer gebeten die Schwierigkeit der Aufgabe für jede Figur auf einer 10-Punkt Likert-Skala zu bewerten (1 = sehr einfach, 10 = sehr schwer).

3 Ergebnisse

Allgemein soll in Untersuchungen meistens gezeigt werden, dass es abhängig vom Design des Versuchs einen bestimmten Unterschied gibt. Deshalb wird in der Statistik grundsätzlich zuerst angenommen, dass alle Sachverhalte gleich sind und es somit keine Unterschiede gibt und keine Zusammenhänge bestehen (Nullhypothese). In der Regel werden bei der statistischen Analyse unterschiedliche Gruppen miteinander verglichen, dabei wird der arithmetische Mittelwert (M) berechnet sowie teilweise die Standardabweichung (s), der t-Wert (Ergebnis des T-Tests) und der p-Wert (Überschreitungswahrscheinlichkeit). Die Standardabweichung gibt an wie groß die Streuung um den Mittelwert ist und wird in den Abbildungen als Fehlerbalken dargestellt. Der p-Wert zeigt wie wahrscheinlich es ist ein solches Ergebnis zu bekommen, wenn die Nullhypothese wahr ist. Der p-Wert liegt immer zwischen null und eins. Je kleiner der Wert ist, desto unwahrscheinlicher ist die

Nullhypothese. Bei einem T-Test werden die Werte von mehreren Gruppen miteinander verglichen. Der t-Wert ist somit abhängig von der Größe der Differenz zwischen den Mittelwerten. Je größer der Wert ist, desto größer ist die Differenz.

Zusätzlich dazu müssen noch die Sensitivitätswerte (d' , sprich: d-Prime) berechnet werden. Dieser Wert trennt die Selbsterkennung-Rate von dem sogenannten falschen Alarm, den Durchläufen bei denen der Teilnehmer mit „Me“ geantwortet hat, obwohl die dargestellte Bewegung zu einem anderen Teilnehmer gehörte. Ein hoher Sensitivitätswert zeigt, dass der Beobachter seine eigenen Bewegungen sicher erkennen konnte. Ein negativer Wert hingegen zeigt, dass es eine Tendenz gab sehr oft die Schaltfläche „Me“ zu drücken.

Für die Analyse der Ergebnisse wurden die Daten einer männlichen Versuchsperson ausgeschlossen, weil er quer durch alle Bewegungskategorien und dargestellten Figuren negative Sensitivitätswerte hatte.

3.1 Selbsterkennung auf verschiedenen Figuren

Die Treffsicherheit der Erkennung von eigenen Bewegungen wird bewertet durch die Prozentzahl der Durchläufe, welche die Teilnehmer korrekt als ihre eigenen identifiziert haben. Somit entspricht eine Rate von 100% einer perfekten Selbsterkennungs-Rate. In dieser Studie wurden von allen zwölf Teilnehmern die gleiche Anzahl von Animationssequenzen pro Bewegung verwendet. Angenommen, ein Teilnehmer hätte bei jedem Durchlauf des Versuchs geraten, läge seine Selbsterkennungs-Rate bei 8,3% (in englischer Fachliteratur wird dies als „chance performance“ bezeichnet, im folgenden Text wird hierfür der Begriff „zufällige Leistung“ verwendet). Entscheidet sich ein Teilnehmer durchweg bei allen Durchläufen dafür die Schaltfläche mit der Aufschrift „Me“ zu drücken, führt dies auch zu einer perfekten Selbsterkennungs-Rate. Deshalb müssen bei der Statistik außerdem

noch die Sensitivitätswerte d' betrachtet werden.

Quer durch alle Bewegungen und dargestellte Figuren wurde von den Versuchspersonen eine durchschnittliche Selbsterkennungs-Rate von 65,3% erzielt, welche deutlich über der zufälligen Leistung liegt. Die Ergebnisse haben gezeigt, dass die Figur auf der die Bewegungen dargestellt wurden eine geringe Auswirkung auf die Erkennung eigener Bewegungen hatte. Die Teilnehmer waren besser darin eigene Bewegungen auf der Point-Light-Figur (71,3%, $s = 0,04$) zu erkennen, als auf einem animierten Charakter (62,4%, $s = 0,05$), $t(10) = 2,93$, $p = 0,015$, siehe Abbildung 3. Es machte keinen Unterschied, ob der weibliche oder männliche Charakter verwendet wurde ($p = 0,686$). Außerdem gab es keinen signifikanten Zusammenhang zwischen der verwendeten Figur und dem Geschlecht des Betrachters, $p = 0,313$. Daraus lässt sich schließen, dass es für die Selbsterkennung keine Rolle spielt, ob die eigenen Bewegungen auf einem animierten Charakter des eigenen oder gegenteiligen Geschlechts dargestellt werden.

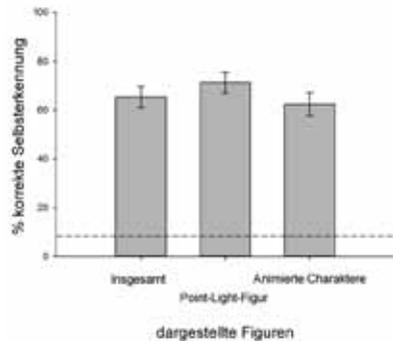


Abbildung 3: Selbsterkennungs-Rate auf den verschiedenen Figuren. Die gestrichelte Linie stellt die zufällige Leistung dar.

Da das Skelett (Forminformationen) der animierten Charaktere nicht an die Statur der Teilnehmer angepasst wurde, hatten 43% der Animationssequenzen kleine Fehler in der Darstellungsweise. Diese zeigen sich vor allem darin, dass die Hände, während einer

Bewegung, manchmal in dem Oberkörper des Charakters verschwinden. Am meisten kam die fehlerhafte Darstellung bei den Tanzbewegungen vor, insgesamt 70,5% dieser Sequenzen waren betroffen (insgesamt 60% der Tanzbewegungen (spon.) und 91,7% (chor.)). Da die Selbsterkennungs-Rate trotzdem durchgehend sehr hoch war (auch bei den Tanzbewegungen) und der Ausschluss dieser fehlerhaften Sequenzen das Design des Versuchs beeinflusst hätte, wurden diese Fehler der Darstellungsweise in der Datenanalyse nicht berücksichtigt.

Es wurden auch die Sensitivitätswerte (d') betrachtet, welche zeigen wie sicher die Teilnehmer ihre eigenen Bewegungen erkannt haben. Bei der Betrachtung der unterschiedlichen Figuren wurden keine signifikanten Unterschiede in den Sensitivitätswerten gefunden, $p = 0,387$, siehe Abbildung 4. Dies lässt vermuten, dass die Betrachter sensitiv gegenüber ihren eigenen Bewegungen waren, unabhängig davon ob sie auf der Point-Light-Figur ($M = 1,0$; $s = 0,13$) oder auf einem animierten Charakter ($M = 1,22$; $s = 0,34$) dargestellt wurden.

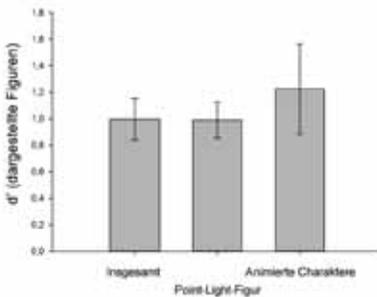


Abbildung 4: Durchschnittliche Sensitivitätswerte.

3.2 Selbsterkennung in verschiedenen Bewegungen

Die Teilnehmer haben ihre eigenen Bewegungen am besten bei spontanem Tanzen erkannt ($M = 86\%$, $s = 0,07$), siehe Abbildung 5. Gefolgt war dies von dem Tanzen einer zuvor gelernten Choreographie ($M = 73\%$, $s = 0,06$) und Tischtennis spielen

($M = 64\%$, $s = 0,08$). Diese Unterschiede in der Selbsterkennung in verschiedenen Bewegungen wurden auch in einer anderen Studie [4] beobachtet. Die Selbsterkennung hat bei allen Bewegungskategorien eindeutig die zufällige Leistung überschritten, unabhängig davon auf welcher Figur die Bewegungen gezeigt wurden. Am schlechtesten war die Selbsterkennung bei den Bewegungen rennen ($M = 55\%$, $s = 0,09$) und gehen ($M = 49\%$, $s = 0,07$). Trotzdem waren diese Selbsterkennungs-Raten höher als bei anderen Studien [4].

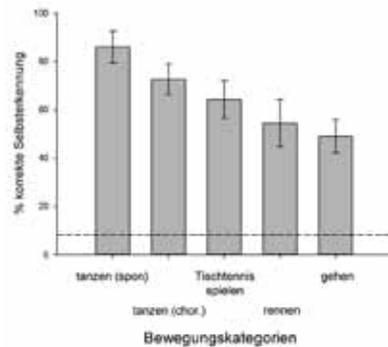


Abbildung 5: Selbsterkennungs-Rate heruntergebrochen auf die verschiedenen Bewegungskategorien. Die gestrichelte Linie stellt die zufällige Leistung dar.

Wie sensibel die Betrachter ihre eigenen Bewegungen erkennen konnten ist abhängig von der gezeigten Bewegung. Spontanes tanzen war die ausdrucksstärkste und persönlichste Bewegung die verwendet wurde, deshalb ist die Sensitivität zur Selbsterkennung hierbei sehr hoch ($M = 4,20$; $s = 0,96$), siehe Abbildung 6. Für das Tanzen einer erlernten Choreographie ($M = 2,18$; $s = 0,76$) und Tischtennis spielen ($M = 1,89$; $s = 0,78$) konnten die Teilnehmer ihre Bewegungen immer noch gut erkennen. Bei den Bewegungen rennen ($M = 0,69$; $s = 0,93$) und gehen ($M = 0,56$; $s = 0,13$) war die Sensitivität zur Erkennung niedriger. Dies spiegelt frühere Ergebnisse wieder welche zeigten, dass die Sensitivität gegenüber Geh- und Rennbewegungen niedriger ist als bei anderen Bewegungen [4].

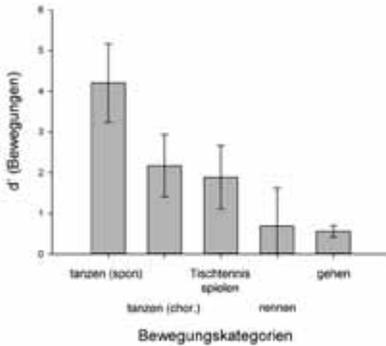


Abbildung 6: Durchschnittliche Sensitivitätswerte heruntergebrochen auf die verschiedenen Bewegungskategorien.

Die Ergebnisse zeigen, dass es in dem Versuch einen signifikanten Zusammenhang zwischen der Art der Bewegung und dem Geschlecht des Beobachters gab, siehe Abbildung 7. Dies lässt vermuten, dass die männlichen und weiblichen Beobachter ihre eigenen Bewegungen auf eine unterschiedlich Art und Weise erkannt haben. Dieser Unterschied war unabhängig davon welche Figur zur Darstellung verwendet wurde. Männer sind signifikant besser als Frauen darin, ihre Gehbewegung zu erkennen, $t(9) = 2,42$; $p = 0,039$. Dahingegen sind Frauen besser darin sich selbst beim Tischtennis spielen zu erkennen, $t(9) = -3,5$; $p = 0,007$.

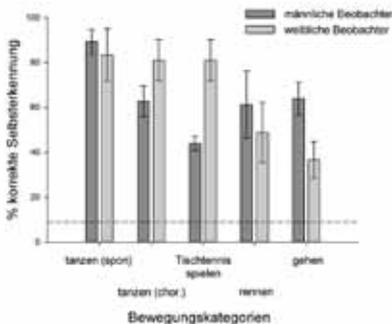


Abbildung 7: Selbsterkennungs-Rate für die verschiedenen Bewegungskategorien, heruntergebrochen auf das Geschlecht der Beobachter. Die gestrichelte Linie stellt die zufällige Leistung dar.

3.3 Subjektive Bewertungen

Auch die erhobenen Daten aus dem Fragebogen wurden ausgewertet. Darin mussten die Teilnehmer beurteilen wie schwer die Selbsterkennungs-Aufgabe auf den verschiedenen Figuren war. Insgesamt haben die Teilnehmer es am schwierigsten empfunden ihre Bewegungen auf der Point-Light-Figur zu erkennen ($M = 7,82$; $s = 0,30$), wohingegen es als einfacher empfunden wurde die eigenen Bewegungen auf dem männlichen ($M = 5,45$; $s = 0,31$) und weiblichen Charakter ($M = 5,45$; $s = 0,31$) zu erkennen, $t(10) = 6,99$; $p < 0,001$ und $t(10) = 3,42$; $p = 0,007$, siehe Abbildung 8.

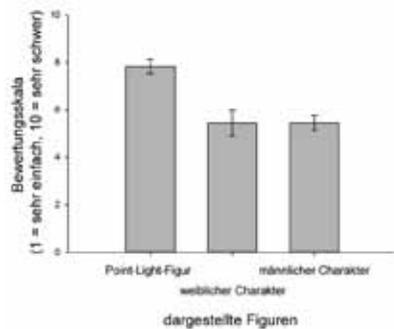


Abbildung 8: Subjektive Bewertung aus dem Fragebogen.

Interessanterweise haben die Ergebnisse der Selbsterkennungs-Aufgabe das exakte Gegenteil gezeigt als das subjektive Gefühl der Versuchspersonen. Während die visuellen Informationen die Selbsterkennungs-Aufgabe für den Teilnehmer scheinbar einfacher machen, lassen die Ergebnisse des Versuchs vermuten, dass die visuellen Informationen der animierten Charaktere den Beobachter bei der Erkennung von eigenen Bewegungen eher ablenken als unterstützen.

4 Diskussion

Insgesamt waren die Teilnehmer besser darin ihre eigenen Bewegungen zu erkennen, wenn eine Point-Light-Figur zur Darstellung verwendet wurde, anstelle eines animierten Charakters. Allerdings zeigen die subjektiven Ergebnisse des Fragebogens das gegenteilige

Ergebnis. Überraschenderweise haben die Versuchsteilnehmer angegeben, dass es schwieriger sei die eigenen Bewegungen auf einer Point-Light-Figur zu erkennen. Dieses Ergebnis deutet darauf hin, dass Charaktere als einfachere Figur zur Selbsterkennung bewertet werden, weil sie mehr Informationen beinhalten, als eine Point-Light-Figur [7]. Darüber hinaus haben Chaminade und Kollegen herausgefunden, dass Animationen auf Charakteren für die Betrachter künstlicher aussehen, als wenn die gleichen Bewegungen auf einer Point-Light-Figur dargestellt werden [6]. Dies könnte ein Grund dafür sein, warum die Versuchspersonen dazu tendiert haben „Not me“ zu drücken, wenn sie einen animierten Charakter mit ihren eigenen Bewegungen gesehen haben, folglich eine niedrigere Selbsterkennungs-Rate haben und zudem berichten, dass es einfacher sei.

Normalerweise enthalten Point-Light-Figuren Forminformationen zur Statur des Darstellers. In dieser Studie wurden alle darstellerspezifischen Informationen aus den Animationssequenzen entfernt. Die Sequenzen, die für die Selbsterkennungs-Aufgabe verwendet wurden enthalten somit nur Bewegungsinformationen des Darstellers und keine Forminformationen. Ungeachtet davon war die Selbsterkennungs-Rate der eigenen Bewegungen sehr hoch. Dies lässt vermuten, dass die Forminformationen nicht notwendig sind um eigene Bewegungen zu erkennen. Es wäre jedoch eine weitere Studie notwendig um diesen Verdacht zu bestätigen.

Des Weiteren besteht die Vermutung, dass die Teilnehmer erst das Geschlecht der Bewegung betrachten bevor sie versuchen sich selbst zu erkennen, insbesondere in dem Fall der Verwendung eines animierten Charakters als visuelle Figur. Wenn das Geschlecht der Bewegung und das des Charakters nicht übereinstimmen, hilft dies dem Beobachter das Geschlecht der Bewegung zu erkennen. Dies könnte als Ausschlussverfahren verwendet werden zur Entscheidung ob es die eigene Bewegung ist oder nicht. Diese Theorie stimmt mit den Ergebnissen überein, dass die Sensitivitätsrate

höher ist wenn die eigenen Bewegungen auf einem animierten Charakter dargestellt werden, als bei der Verwendung von einer Point-Light-Figur. Um diese Theorie zu bestätigen könnten weitere Versuche mit den verwendeten Sequenzen durchgeführt werden, um die Geschlechterkennung zu untersuchen.

Darüber hinaus wurde entdeckt, dass das Geschlecht des Beobachters die Erkennung von eigenen Bewegungen beeinflusst. Die Ergebnisse zeigen, dass bestimmte Bewegungen (gehen) von Männern besser erkannt werden und andere Bewegungen (Tischtennis spielen) von Frauen.

5 Fazit

In der vorliegenden Studie wurde untersucht wie wichtig die visuelle Darstellung von Bewegungen ist sowie welche Rolle das Geschlecht der Versuchsteilnehmer und das der animierten Charaktere bei der Erkennung von eigenen Bewegungen spielt. Dafür wurde die Studie von Loula und Kollegen teilweise repliziert und erweitert [4]. Es wurde eine Vielzahl von aufgenommenen Bewegungen auf einer Point-Light-Figur und animierten Charakteren (weiblich und männlich) dargestellt. Anschließend mussten die Versuchspersonen ihre eigenen Bewegungen darauf erkennen.

Die vorliegende Studie erweitert was bisher über die Erkennung von eigenen Bewegungen auf animierten Charakteren bekannt war. Mit dieser Studie wurde gezeigt, dass animierte Charaktere die Selbsterkennung nicht verbessern, unabhängig von dem Geschlecht des Charakters.

Die Ergebnisse der Selbsterkennung auf den dargestellten Figuren stehen im Gegensatz zu den subjektiven Antworten der Versuchsteilnehmer. Die Erkennung eigener Bewegungen auf einer Point-Light-Figur wurde von den Teilnehmern als schwieriger bewertet, als auf den animierten Charakteren, obwohl die Selbsterkennungs-Rate bei den Point-Light-Figuren höher war.

6 Danksagung

Mein Dank gilt Professor Uwe Kloos der mich während dieser Arbeit betreut hat. Des Weiteren möchte ich mich bei Jeanine Stefanucci und Joachim Tesch für die Gespräche und ihre Unterstützung bei den Versuchen bedanken. Mein besonderer Dank geht an Markus Leyrer, Ekaterina Volkova, Dong-Seon Chang und Betty Mohler für ihre Unterstützung und Mitarbeit bei der Planung und Durchführung dieser Studie.

7 Literaturverzeichnis

- [1] Johansson, G. (1973). Visual perception of biological motion and a model for its analysis. *Perception & psychophysics*, 14(2), 201-211.
- [2] Cutting, J. E., & Kozlowski, L. T. (1977). Recognizing friends by their walk: Gait perception without familiarity cues. *Bulletin of the psychonomic society*, 9(5), 353-356.
- [3] Kozlowski, L. T., & Cutting, J. E. (1977). Recognizing the sex of a walker from a dynamic point-light display. *Perception & Psychophysics*, 21(6), 575-580.
- [4] Loula, F., Prasad, S., Harber, K., & Shiffrar, M. (2005). Recognizing people from their movement. *Journal of Experimental Psychology: Human Perception and Performance*, 31(1), 210-220.
- [5] Cook, R., Johnston, A., & Heyes, C. (2012). Self-recognition of avatar motion: how do I know it's me?. *Proceedings of the Royal Society B: Biological Sciences*, 279(1729), 669-674.
- [6] Chaminade, T., Hodgins, J., & Kawato, M. (2007). Anthropomorphism influences perception of computer-animated characters' actions. *Social cognitive and affective neuroscience*, 2(3), 206-216.
- [7] Hodgins, J. K., O'Brien, J. F., & Tumblin, J. (1998). Perception of human motion with different geometric models. *Visualization and Computer Graphics, IEEE Transactions on*, 4(4), 307-316.
- [8] Zibrek, K., Hoyet, L., Ruhland, K., & McDonnell, R. (2013, August). Evaluating the effect of emotion on gender recognition in virtual humans. In *Proceedings of the ACM Symposium on Applied Perception* (pp. 45-49). ACM.
- [9] McDonnell, R. (2012). Appealing Virtual Humans. In *Motion in Games* (pp. 102-111). Springer Berlin Heidelberg.
- [10] Vinayagamoorthy, V., Gillies, M., Steed, A., Tanguy, E., Pan, X., Loscos, C., & Slater, M. (2006) Building Expression into Virtual Characters. The Eurographics Association.
- [11] Xsens MVN, <http://www.xsens.com/>
- [12] Unity 3D, <http://unity3d.com/>

Bestimmung von 3D-Messpositionen durch genetischen Algorithmus

Frank Bertagnolli

ESB, Reutlingen University

Frank.Bertagnolli@Reutlingen-University.DE

Abstract

Um in der optischen 3D-Messtechnik den Messsensor zu positionieren, werden in der Regel Messrobotersysteme eingesetzt. Die Bestimmung der Messpositionen erfolgt manuell und wird durch „teach-in“ einge­lernt.

Mittels einer Messstrategie können optimale Messpositionen automatisch bestimmt werden. Das vollautomatische Digitalisieren von Werkstücken mit der Losgröße 1 wird hierbei möglich. Ziel der Messstrategie ist es, eine fast vollständige Oberflächenabdeckung mit möglichst wenigen Messansichten zu erreichen. Durch verschiedenste, meist komplexe, Werkstückgeometrien ergibt sich daraus ein multimodales, NP-vollständiges Optimierungsproblem. Für die Lösung bieten sich genetische Algorithmen an.

Schlüsselwörter

Optische 3D-Messtechnik, Industrieroboter, Messroboter, Messpositionen, Optimierungsproblem, genetische Algorithmen

CR-Kategorie

I.2.8 [Computing Methodologies]: Problem Solving, Control Methods, and Search—*Heuristic methods*

1 Problemstellung

Das digitale Erfassen von Geometrien ist in verschiedensten Bereichen notwendig. So ist die optische Digitalisierung in der Produktentwicklung, dem Reverse-Engineering, der

Qualitätssicherung sowie für die virtuelle Welt nicht mehr wegzudenken.

Das Digitalisieren von Geometrien aus verschiedenen Perspektiven kann bei hohen Stückzahlen automatisiert durch Roboter mit einge­lernten Programmen erfolgen. Sobald jedoch eine neue Geometrie erfasst werden soll, muss die Messung manuell und die Sensorpositionierung iterativ durchgeführt werden. Dabei entstehen für eine vollständige Digitalisierung oft mehr Ansichten als nötig. Ein Optimum wird auf diese Art nie erreicht.

Es ergibt sich die Fragestellung, wie mit möglichst wenigen Messansichten die vollständige Geometrie bedienerunabhängig erfasst werden kann. Das Problem stellt sich als multimodales Problem, also mit vielen lokalen Nebenoptima, dar. Es handelt sich um ein NP-vollständiges Optimierungsproblem. Die vollständige Enumeration des Lösungsraumes bietet sich aufgrund der Suchraumgröße nicht an.

2 Genetischer Algorithmus

Um das vorliegende Optimierungsproblem mit seiner hohen Komplexität effizient und robust zu lösen, setzt man als universelles Optimierungswerkzeug einen genetischen Algorithmus ein [1]. Der prinzipielle Ablauf ist in Abbildung 1 dargestellt.

Mit dem Einsatz von genetischen Algorithmen ist es durch die geschickte Kombination einzelner guter Messpositionen möglich, einen geeigneten Messplan zu generieren. Dabei wird aus einer Messsimulation die Qualität und Vollständigkeit der zu vermessenden Oberfläche bestimmt. Mehrfache Messungen werden vermieden.

```

Population  $Pop_0$  mit Individuen initialisieren
 $t := 0$ 
wiederholen
    Erzeugung neuer Individuen durch:
    - Kreuzung von Individuen aus  $Pop_t$ 
    - Mutation von Individuen aus  $Pop_t$ 
    Bewertung neuer Individuen: Zielfunktion
    Selektion einer neuen Population  $Pop_{t+1}$ 
     $t := t + 1$ 
bis Terminierungskriterium erfüllt ist

```

Abbildung 1: Prinzipieller Ablauf eines genetischen Algorithmus im Pseudocode

3 Bewertung von Messplänen

Bei genetischen Algorithmen kommt der Zielfunktion F eine zentrale Bedeutung zu. Ziel ist einen akzeptablen und guten Messplan zu finden. Er zeichnet sich durch möglichst wenige Messansichten M und eine möglichst vollständige Abdeckung der Objektoberfläche O aus. Die Faktoren „Anzahl Messansichten“ und „Oberflächenabdeckung“ werden zusätzlich gewichtet (w).

$$F_i = \left(1 - \frac{M_i}{q \cdot n'_{\text{Vorbelegung}}} \right)^{w_M} \cdot \left(\frac{O_i}{\sum O} \right)^{w_O}$$

Die Messpositionsanzahl wird in das Verhältnis mit einer Vorberechnung ($q \cdot n$) gesetzt. Hierin geht auch die Komplexität des Messobjektes ein [2], um eine gute Ausgangsposition für die Iterationen zu haben.

4 Ergebnis in der Anwendung

Der Algorithmus generiert wesentlich bessere Messpläne, als ein Bediener. Der quantitative Gewinn durch die automatische Messplanung ist direkt von der geringeren Anzahl an Messpositionen abhängig.

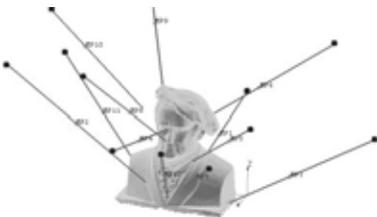


Abbildung 2: Messplanungsergebnis an Wagner-Skulptur mit 12 Messpositionen

Die Skulptur von Wagner (Abbildung 2) (374 mm × 275 mm × 390 mm) wurde mit 12 Messansichten automatisiert digitalisiert. Die Rechenzeit für den Algorithmus betrug 215 Sekunden. Dabei wurde eine Oberflächenabdeckung von 98,1% erreicht. Ein manueller Bediener hätte hierfür rund 22 Messungen benötigt, also 83% mehr.

5 Praxisanwendung

Der Algorithmus kommt im Zusammenhang mit einem optischen Messsystem auf einem Industrieroboter zum Einsatz (Abbildung 3).



Abbildung 3: Optischer Messroboter [3]

Im Reverse-Engineering-Prozess werden die Werkstücke vollautomatisch digitalisiert. Ein Objekt (500 mm × 300 mm × 150 mm) wurde in 8 Minuten mit ca. 2 Millionen dreidimensionalen Einzelpunkten vollständig erfasst.

6 Literaturverzeichnis

- [1] E. Schöneburg, F. Heinzmann und S. Feddersen. Genetische Algorithmen und Evolutionsstrategien: Eine Einführung in Theorie und Praxis der simulierten Evolution. Addison Wesley. Bonn, 1994.
- [2] F. Bertagnolli. Verfahren und Vorrichtung zur Erstellung eines Messplans zur Vermessung eines 3D-Messobjekts. Patent DE 10 2005 037 837 A1. Deutsches Patent- und Markenamt. Berlin, 2007.
- [3] DaimlerChrysler AG. Messtechnik: Schnell und sehr genau. In: Hightech Report, Faszination Forschung & Technik, Ausgabe 2/2003, S. 6. Prisma Direkt. Bad Homburg, 2003.

Neue Welt 9: Vorteile und Probleme bei der Migration zur CryEngine Version 3.5.4

Christian Henssler
Reutlingen University
Christian.Henssler@Student.
Reutlingen-University.de

Peter Einberger
Reutlingen University
Peter.Einberger@Student.
Reutlingen-University.de

Abstract

Dieses Paper befasst sich mit den letzten Änderungen, dem aktuellen Stand und der Zukunft des Projekts „neue Welt 9“ im Studiengang MKI der Hochschule Reutlingen. Eingegangen wird zudem auf die Migration von der CryEngine 3.4.5 zu 3.5.4 und die dabei auftretenden Probleme sowie Änderungen in der neuen Version.

Schlüsselwörter

Informatics Inside, CryEngine 3.5, neue Welt 9

CR-Kategorien

I.3.7 [Three-Dimensional Graphics and Realism]: Virtual reality

1 Einleitung

Das Masterprojekt „neue Welt 9“ ist ein Multimediaprojekt, dessen Ziel es ist, die Geschichte des Gebäudes 9 der Hochschule Reutlingen interaktiv darzustellen. Der Benutzer kann das Gebäude erforschen und stößt dabei auf Elemente, die von dem Werdegang des Gebäudes erzählen. Die geschichtlichen Ereignisse kann der Benutzer mittels Video-, Bild- und Tonmaterial nachvollziehen.

Zur Darstellung wird aktuell eine Spiele-Engine verwendet. Dies ist die CryEngine mit

der Version 3.5 [1]. Sie dient als Framework zur visuellen und interaktiven Darstellung des Gebäudes. Im mitgelieferten „Software Development Kit“ (SDK) werden die Medien über Skripte in die Welt eingebunden.

2 Fortschritt

Bisher wurde das Projekt hauptsächlich mit der Modellierung von 3D Objekten und dem Ausbau des Gebäudes vorangetrieben. Die Arbeiten am Gerüst des Gebäudes sind zu 70% abgeschlossen. Die Arbeiten am Erdgeschoss sowie dem Keller und dem Eingangsbereich wurden zuletzt aufgenommen.

Aktuell werden Grafik- und Performanzoptimierungen durchgeführt. Dafür werden Texturen so angepasst, dass sie sich physikalisch korrekt verhalten. Das heißt, dass sie Licht und Schatten realistisch annehmen. Oberflächeneigenschaften wie Rauheit oder Ebenheit sollen ebenfalls simuliert werden.

Des Weiteren wurden bisher Objekte mit unnötig vielen Materialien und Teilobjekten erstellt. Wird ein Objekt auf dem Bildschirm dargestellt, muss die Grafikkarte das Objekt berechnen. Dies geschieht für jedes zusätzliche Material immer wieder von neuem. Dieser Effekt summiert sich für jedes weitere Objekt auf. Deshalb werden neben der Migration auch Materialien überarbeitet.

3 Aktualisierung

„neue Welt 9“ wechselte im Wintersemester 2012 von Quest3D auf die CryEngine 3.4 und wurde seitdem nicht mehr aktualisiert. Anlass

Informatics Inside 2014
Wissenschaftliche Vertiefungskonferenz
30. April 2014, Hochschule Reutlingen
Copyright 2014 Christian Henssler, Peter Einberger

für eine Aktualisierung auf eine aktuelle Version bot Ende 2013 die Vorstellung der CryEngine 3.5, welche umfangreiche Änderungen enthält. Im *Kapitel 3.1* werden neue Features innerhalb der CryEngine 3.5 zusammengefasst.

3.1 CryEngine 3.4 zu 3.5

Mit der Veröffentlichung der CryEngine 3.5 fand ein Versionssprung von 3.4.5 auf 3.5.4 statt. Weshalb die Änderungen zwischen diesen Versionen behandelt werden [1], die sich in vier Bereiche aufteilen lassen: Renderer, Editor, Künstliche Intelligenz und Spielelemente.

Im Bereich Render wurde das Silhouette Parallax Occlusion Mapping (SPOM) [2] eingeführt. SPOM ist eine Erweiterung des Parallax Occlusion Mapping. Damit können Objekte detailreicher gestaltet werden, ohne dass die Polygonzahl erhöht wird.

Weiterhin wurde das Cloth-Simulation-Gerüst überarbeitet und bestehende Implementationen vereinheitlicht und optimiert, um auf Konsolenhardware performanter zu reagieren. So vereint das Cloth-System nun Textilien- als auch Vegetationssimulationen.

Änderungen am Editor umfassen einen einfachen Objekt-Editor, der im SDK genutzt werden kann, ein neues Lens Flare-Tool und ein Falten-System für Charaktermodelle. Eine Vielzahl von Änderungen gab es im Bereich der Künstlichen Intelligenz. So enthält die CryEngine 3.5 unter anderem modulare Verhaltensweisen, einen Bewegungsplaner, Sprechblasen, einen vereinfachten KI-Editor und verbesserte Wegfindung für fliegende Einheiten. Im Bereich der Spielelemente wurde das Werkzeug Telemetry eingeführt. Dieses sammelt große Mengen an Spielablaufdaten für die spätere Auswertung.

3.2 Probleme

Bei der Migration sind einige Probleme aufgetreten. Der Großteil der Fehlermeldungen, die bei der Migration zustande gekommen sind, ist allerdings auf Änderungen in der Ordnerstruktur zurückzuführen. Das gilt sowohl für Optimierungen des Projekts, als

auch für die verwendeten Ressourcen, die mit dem SDK geliefert werden.

Weitere Fehlermeldungen entstanden durch das falsche Einbinden oder Benennen von Dateien. Die CryEngine besitzt einen Compiler für Ressourcen, der dazu verwendet wird, Texturen vom Crytiff-Format des SDK in das zur Laufzeit verwendete DDS Format zu kompilieren. Das Crytiff-Format wird für die Rohdaten verwendet und besitzt Metadaten, die bei der Kompilierung in das DDS-Format als Parameter verwendet werden. [3]

Zusätzlich zu den Metadaten müssen auch bestimmte Namenskonventionen eingehalten werden. So sollen beispielsweise Normalmaps nach dem Muster "name_ddn.tif" benannt werden. Ist dies nicht der Fall, werden die Dateien nicht einwandfrei als Normalmaps erkannt und enthalten fehlerhafte Informationen als kompilierte DDS-Datei.

4 Aussicht

Zunächst soll nun die Migration zur Version 3.5 vollständig abgeschlossen werden. Hierfür müssen noch viele Texturen aktualisiert und angepasst werden.

Soll das Gebäude mit möglichst allen Objekten eingerichtet werden, muss die Planung und Entstehung von Objekten in Zukunft optimiert werden.

5 Literaturverzeichnis

- [1] M. Pinard. Cryengine Documentation; Webseite, 2014. Online verfügbar unter: <http://docs.cryengine.com/display/SDK/DOC1/CryENGINE+3.5+Features>; Besucht am 16.03.2014
- [2] A. Johnson. Cryengine Documentation; Webseite, 2014. Online verfügbar unter: <http://docs.cryengine.com/display/SDK/DOC2/Silhouette+POM>; Besucht am 27.03.2014
- [3] E. Liang. Cryengine Documentation; Webseite, 2014. Online verfügbar unter: <http://docs.cryengine.com/display/SDK/DOC3/Using+CryTif>; Besucht am 16.03.2014

QUEST 3C - Kooperation, Kommunikation, Präsentation ohne Grenzen

Hazel Grünewald
Reutlingen University
hazel.gruenewald@reutlingen-
university.de

Michael Moser
Reutlingen University
michael.moser@reutlingen-
university.de

Abstract

Das Planspiel QUEST 3C wird im Rahmen der Initiative Exzellenz in der Lehre von der Baden-Württemberg Stiftung gefördert. Wirtschaftsingenieure müssen vernetzt, ganzheitlich, interkulturell und interdisziplinär denken und handeln. Im Zuge der zunehmenden Globalisierung, des Fortschritts in den Kommunikationstechnologien und der steigenden kulturellen Diversität in den Teams werden qualifizierte Wirtschaftsingenieure weiteren Herausforderungen in der internationalen Teamarbeit ausgesetzt. Die Studierenden erhalten im Planspiel eine für Wirtschaftsingenieure relevante Aufgabenstellung, die sie in funktionsübergreifenden Projektteams in einer simulierten Matrix-Organisation praxisorientiert lösen. Dafür setzen sie modernste Technologien ein und müssen Wege finden, verbindliche Teamregeln standortübergreifend einzuführen, Fachwissen zu teilen und sich trotz Distanz oder zeitlicher Asynchronität gemeinsam für den Projekterfolg einzusetzen. Durch den berufsbezogenen Umgang mit neuen Medien, Tools und Kommunikati-

onsmitteln trägt Quest 3C zur Erhöhung der IT-Literacy von Studierenden bei.

Schlüsselwörter

Mobile Office; Kollaboration; Kompetenzen; e-Portfolio; Lehre; SAAS.

CR-Kategorien

H.3.5 [Information Systems]: Online Information Services—*Web-based services*; H.5.3 [Information Systems]: Group and Organization Interfaces—*Collaborative computing*

1 Aufbau und Ablauf des Planspiels

Das Planspiel besteht aus 4 Sequenzen:

- **Briefing** Das Briefing erfolgt im Plenum bzw. über eine webbasierte Plattform.
- **Simulation** Die Aufgabenstellung wird in organisations- und hierarchieübergreifenden Projektteams gelöst. Gemeinsame Absprachen, Verhandlungen und Präsentationen erfolgen entweder face-to-face oder über Kollaborationstools.
- **Debriefing** Erfolgt mittels 360-Grad-Feedback zwischen den TeilnehmernInnen. Ideen zur Gestaltung und Optimierung von Teamprozessen werden mit Hilfe der Moderation der Spielleiter diskutiert.
- **Reflexion** des Lernprozesses erfolgt über ein e-Portfolio, um Erkenntnisse auszuwerten und Schwachstellen systematisch zu verbessern. Kompetenzen, Stärken und

Betreuer Hochschule: Prof. Dr. Hazel Grünewald
Hochschule Reutlingen
ESB Business School
hazel.gruenewald@reutlingen-
university.de

Informatics Inside 2014
Wissenschaftliche Vertiefungskonferenz
30. April 2014, Hochschule Reutlingen
Copyright 2014 Michael Moser / Prof. Dr. Hazel Grünewald

neugewonnene Expertise werden herausgestellt.

1.1 *Mobile Office*

Mittels Cloud Computing (SAAS) können umfangreiche, nahezu barrierefreien Kollaborationsmöglichkeiten zur Verfügung gestellt werden. Dies sorgt für eine bessere Zusammenarbeit über Grenzen hinweg, da alle Nutzer – unabhängig vom Standort – sich abstimmen und organisieren, an Dateien und freigegebenen Anwendungen synchron arbeiten können[1]. Ebenso kann der Wissenstransfer zentral gesteuert werden. Dies reduziert auftretende Wartezeiten, zudem wird eine größtmögliche Transparenz über das unternehmerische Wissen geschaffen. Dies eröffnet in der Arbeitswelt, in welcher nationale oder kontinentale Grenzen verschwimmen, integrierte Arbeitsweisen innerhalb einer Supply Chain ohne Brüche über IT-Systeme. Eine weitere Stärke ist die nahtlose Verzahnung der Dienste und ihre Verwendbarkeit auf allen gängigen Client-Systemen ohne lokale Installationen[3]. Dies erleichtert die geplante länderübergreifende Umsetzung des Planspiels in Zusammenarbeit mit Partnerhochschulen.

1.2 *e-Portfolio*

In einem e-Portfolio können Studierende ihre Lernergebnisse und ihre Kompetenzentwicklung dokumentieren, reflektieren und präsentieren. Die einfache Zugänglichkeit des e-Portfolios bietet Potenzial zur intensiveren Rückmeldung: Studierende erhalten ein regelmäßiges Feedback und haben die Chance, ihre Leistungen laufend zu verbessern [2]. Darüber hinaus können sie multimediale Elemente in das Portfolio aufnehmen und entscheiden, wer später welche Inhalte online ansehen darf. Somit ist das e-Portfolio für Bewerbungszwecke sehr geeignet.

2 **Kompetenzvermittlung**

- Bei der Problemlösung lernen die TeilnehmerInnen Zusammenhänge zu erken-

nen und erworbenes Fachwissen aus diversen Disziplinen vernetzt anzuwenden. Begleitseminare vermitteln das notwendige Handwerkszeug, zudem eignen sich die Studierenden selbstständig oder kooperativ das erforderliche Know-how an.

- Handlungsorientierte Sozialkompetenz wird durch interdisziplinäre und internationale Teamarbeit hinweg trainiert. Durch wechselnde Rollen, Aufgaben und Komplexitätsstufen werden sich ändernde Perspektiv-, Positions- und Beziehungsstrukturen in das interaktive Lernszenario eingebaut, die weit über eine fachbezogene Vorlesung hinausgehen.

- Das Erlernen und das Evaluieren geeigneter Methoden, Instrumente und Werkzeuge sind unabdingbare Voraussetzungen für lösungsorientiertes Lernen. Die Studierenden lernen situativ effiziente Strategien, Methoden und einen sicheren und korrekten Umgang mit modernen Medien und Techniken.

3 **Literaturverzeichnis**

[1] D. Banks, J.S. Erickson, M. Rhodes. Toward Cloud-based Collaboration Services, Hewlett-Packard Labs, 2009. Online verfügbar unter http://static.usenix.org/events/hotcloud09/t ech/full_papers/banks.pdf; Besucht am 06.02.2014

[2] G. Bisovsky, S. Schaffert. Lehren und Lernen mit dem E-Portfolio – eine Herausforderung für die Professionalisierung der Erwachsenenbildner/innen, Deutsches Institut für Erwachsenenbildung Leibniz-Zentrum für Lebenslanges Lernen, 2009. Online verfügbar unter: www.die-bonn.de/doks/bisovsky0901.pdf; Besucht am 06.02.2014.

[3] M. Miller. Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate Online, Que Publishing, 2008.

Technologien verteilter Systeme zur mobilen Kommunikation

Sebastian Hirth
Reutlingen University
Sebastian.Hirth @
Student.Reutlingen-
University.DE

Susanne Hönig
Reutlingen University
Susanne.Hönig @
Student.Reutlingen-
University.DE

Alexander Kunz
Reutlingen University
Alexander.Kunz @
Student.Reutlingen-
University.DE

Dominic Lyons
Reutlingen University
Dominic.Lyons @
Student.Reutlingen-
University.DE

Jens Mahler
Reutlingen University
Jens_Mathias.Mahler@
Student.Reutlingen-
University.DE

Abstract

In diesem Paper werden Themen des Masterprojektes Web Mobile Content Collaboration (wmc²) der Hochschule Reutlingen vorgestellt. Thema des Projekts ist die zunehmenden Vernetzung von Gegenständen im Alltag, welche den Menschen bei seinen Aufgaben unterstützt - auch "Internet of Things" genannt. Das Projekt beschäftigt sich mit Technologien rund um verteilte Systeme für mobile Kommunikation, unter anderem in den Bereichen Telemedizin, Automotive und Mobile Computing.

Schlüsselwörter

Mobile Kommunikation, Vitalüberwachung, biometrische Sensoren, anpassungsfähige Usability, automatisches Notrufsystem, EKG-Kompressionsverfahren, Car2Car

Betreuer Hochschule: Prof. Dr. Wolfgang Keller
Hochschule Reutlingen
Wolfgang.Keller@Reutlingen-
University.de
Prof. Dr. Natividad Martínez
Hochschule Reutlingen
Natividad.Martinez@Reutlingen-
University.de

Wissenschaftliche Vertiefungskonferenz 2014, 30. April
2014, Hochschule Reutlingen
Copyright 2014 Sebastian Hirth, Susanne Hönig, Alexan-
der Kunz, Dominic Lyons, Jens Mahler

CR-Kategorien

H.1.2 [Information Systems]: User/Machine Systems—*Human factors, Human information processing*; J.3 [Computer Applications]: Life and Medical Sciences—*Health, Medical information systems*

1 Einleitung

Das Forschungsprojekt Web Mobile Content Collaboration (wmc²) findet im Masterstudiengang Medien- und Kommunikationsinformatik statt. Dieses Projekt beschäftigt sich mit Technologien rund um verteilte Systeme für mobile Kommunikation in den Bereichen Telemedizin, Automotive und Mobile Computing. Diese Themen gewinnen aufgrund der fortschreitenden Digitalisierung in unserem Informationszeitalter zunehmend an Bedeutung. Das Ziel des Masterprojekts ist, sich im Rahmen einer wissenschaftlichen Arbeit mit einem Fachthema auseinanderzusetzen und ein wissenschaftliches Ergebnis zu liefern. In diesem Artikel werden aktuelle Themen des Projekts vorgestellt.

2 Dynamic Convoy

Die Car-2-Car-Kommunikation stellt seit einigen Jahren ein großes Forschungsgebiet im Automobilbereich dar. Anhand von Forschungserkenntnissen eines vorhergehenden Masterprojektes (Car2Convoy) [1] soll basie-

rend auf der Car-2-Car-Kommunikation ein Konzept für eine Erweiterung des Projektes entworfen werden, welche eine dynamische Konvoibildung unter Verkehrsteilnehmern im Straßenverkehr ermöglicht. Durch die Erweiterung soll es einem Fahrzeug ermöglicht werden, weitere Verkehrsteilnehmer in einem bestimmten Umkreis zu erkennen und direkt im Straßenverkehr zu einem Fahrzeugverbund einzuladen. Innerhalb dieses Nahbereichs-Ad-Hoc-Netzwerks muss eine dynamische Hostmigration nach bestimmten Kriterien erfolgen, welche eine Schnittstelle zum Car2Convoy-System darstellt.

3 EKG-Datenkompression mithilfe verlustfreier Audio-Kompressionsverfahren

Bei der Diagnose von Herzkrankheiten spielt das Elektrokardiogramm (EKG) eine wichtige Rolle. EKG-Aufzeichnungen bedeuten meistens ein hohes Datenvolumen, womit die Kompression eine große Bedeutung bekommt. Bei Telemedizinlösungen können komprimierte EKG-Daten schneller übertragen werden. Dies ist insbesondere an Standorten von Vorteil, an denen nur geringe Übertragungsgeschwindigkeiten möglich sind. Hierbei sollen bekannte Algorithmen zur Kompression von Audiodaten untersucht und auf EKG-Daten übertragen werden.

4 Automatische Erkennung und Meldung von Notfällen mithilfe von Smartphones

Smartphones haben sich zum ständigen Begleiter des modernen Menschen entwickelt. Eine Funkverbindung vorausgesetzt, ermöglichen sie ortsunabhängig die Meldung von Notfällen, was jedoch in der Regel vom Anwender initiiert werden muss. Der Forschungsansatz besteht daher in einem Notrufsystem, das die technischen Möglichkeiten eines Smartphones nutzt, um Notsituationen automatisch zu erkennen – ein Beispiel ist die Detektion eines Sturzes durch in das Smartphone integrierte Beschleunigungssensoren. Bei der Benachrichtigung eines Notrufzentrums können Informationen wie der

Aufenthaltsort, Art des Notfalls und Informationen zum Patienten übermittelt werden. Ein Rückkanal ermöglicht es Rettungskräften, weitere Details zu erfragen und Hilfestellungen für den Patienten/Ersthelfer zu liefern.

5 Anpassungsfähige Usability

In der Regel sind die kognitiven Fähigkeiten im mobilen Umfeld meist stärker beeinträchtigt. Äußere Einflüsse erschweren die Bedienung der Geräte. Sonnenstrahlen führen zu Spiegelungen auf dem Display. Im Zuge des Masterprojektes soll eine Lösung gefunden werden, die es Internetanwendungen ermöglicht, auf das Problem wechselnder Nutzungssituationen und Umwelteinflüsse zu reagieren. Genauer bedeutet dies, dass sich das Design, der Inhalt oder die Interaktionsform der Anwendung als Folge der Analyse und Verwertung ortsbezogener sowie nutzungsabhängiger Daten anpasst. Diese Anpassungen sollen die äußeren störenden Einflüsse kompensieren und dem Nutzer somit einen Mehrwert generieren.

6 Biometrische Sensoren für mehr Sicherheit in Autos

Das Projekt beschäftigt sich mit der Verwendung biometrischer Sensoren in Autos zur Sicherheitssteigerung, Unfallprävention und Erhöhung des Fahrkomforts. Durch biometrische Sensoren wird der Vitalzustand des Autofahrers überwacht. Wenn die Sensoren Werte messen, die im Gefahrenbereich liegen, kann das verbundene System frühzeitig reagieren, um gefährlichen Situationen entgegenzuwirken. Wenn z.B. Daten gemessen werden, die auf einen Herzinfarkt hinweisen, kann das Fahrzeug die Warnblinkanlage anschalten, automatisch zur Seite fahren und einen Notruf auslösen.

7 Literaturverzeichnis

[1] R. Mangold. Car2Convoy – Vernetzung von Fahrzeugen zur gegenseitigen Kommunikation. Masterprojekt 2013
<https://bscw.serv.reutlingenuniversity.de/bscw/bscw.cgi/d6217509/Car2Convoy.pdf>
14.03.2014

Bottom-up and Top-down Face Analysis based on 3D Face Models*

Patrik Huber
University of Surrey
p.huber@surrey.ac.uk

Josef Kittler
University of Surrey
j.kittler@surrey.ac.uk

Matthias Räscht
Reutlingen University
matthias.raetsch@reutlingen-university.de

Abstract

Many tasks in human-computer-interaction, especially in humanoid robots, require the detection of faces and localization of facial landmarks in images and videos. While the task is considered to be solved for controlled scenarios, most existing approaches are limited to a pose-range of up to half-profile view. Only recently, more holistic approaches [5] have been published able to cope with uncontrolled scenarios. In our approach, we adapt a fast cascaded Wavelet Vector Machine detector [3] from face to facial landmark detection and improve its accuracy by combining it with a 3D face model [4]. As a first application, we use the obtained landmarks with a tree (ERT, [2]) for head pose estimation up to full profile view and show how inaccuracies in the detected landmarks influence performance.

Keywords

Face detection, tracking, face models, 3D morphable model, face analysis, human-robot-interaction (HRI)

CR-Categories

I.4.8 [Image Processing and Computer Vision]: Scene Analysis; I.2.9 [Artificial Intelligence]: Robotics; I.5 [Pattern Recognition]: General

*

Supervisor Reutlingen University: Prof. M. Räscht
Hochschule Reutlingen
matthias.raetsch@reutlingen-university.de

Supervisors ext. University: Prof. J. Kittler, W. Christmas
University of Surrey
j.kittler, w.christmas@surrey.ac.uk

Informatics Inside 2014, Wissenschaftliche Vertiefungskonferenz, 30. April 2014, Hochschule Reutlingen, Copyright 2014 Patrik Huber

1 Bottom-up / Top-down Face- and Landmark Detection

Given an input image, we find the most probable face location by running a series of cascaded detectors [3], each able to detect a certain subset of faces. We use their quasi-probabilistic output to combine their result to a single probability map that shows the face-region probability. At each high-probability location, we then search for the location of twelve facial landmark points (eyes, eye corners, nose, nose corners, mouth corners, upper lip and the ears). This allows robust detection, but produces many false alarms. However, most combinations of landmark points are impossible when considering they must fulfill the constraints of a projection of the 3D face model.

To find the correct face location, we combine the detectors with a model-based approach. To validate face candidates, we use a RANSAC-based approach similar to [4]. Combinations of landmark points are randomly chosen and the model is then projected using the POSIT algorithm (pose from orthography and scaling with iterations, [1]). In the end, we are left only with landmark point candidates that are validated by the 3D model. Fig. 1 gives a graphical overview of the algorithm.

2 Experiments

Our primary goal is to use a 3D model for face recognition, in particular to do a pose-normalization for non-frontal images. As a first step, we conducted preliminary head pose estimation (HPE) experiments on the CasPeal and AFLW face databases. We show that the false acceptance rate can be significantly reduced by combining landmark detectors with

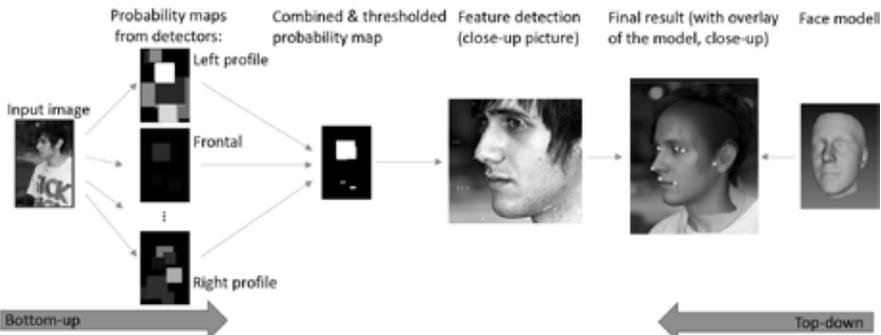


Figure 1: Algorithm overview. *from the left:* Face- and landmark detection on an input image. *from the right:* A 3DMM is used to validate landmark combinations.

a face model. The impact of landmark inaccuracies is shown for the POSIT algorithm. As an example for one landmark (right mouth corner), Fig. 2 shows the head pose estimation error on AFLW for displacements of 0 to 35 pixels. The results also suggest that it is beneficial to use more than just the 5 to 10 landmarks that are usually used.

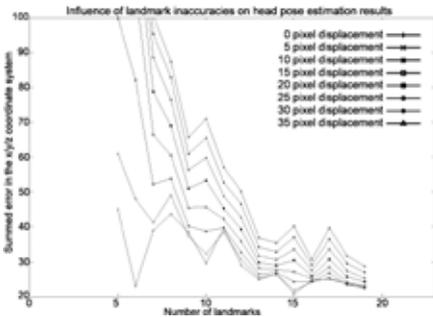


Figure 2: Head pose estimation error on AFLW for different number of landmarks.

Based on the detected landmarks, we compared HPE using POSIT, complete 3D Morphable Model fitting and a support vector regression (SVR) approach. Best results were obtained with the ERT-based SVR with an average error of 2.17° on the CasPeal database.

3 Conclusion

Our experiments have shown promising results for the model-based approach. The next steps will be to run a thorough evaluation on the LFW and AFLW databases and to use our system for pose-normalized face recognition.

Preliminary experiments have been done using a head detector for full 360° tracking, as well as using fitting of the 3DMM. The cascaded wavelet reduced classifiers are efficient enough to open the possibility to run our system in real-time on a consumer desktop PC.

Acknowledgements

The authors would like to thank Tiphaine Rousseau, EURECOM, for conducting the POSIT experiments. This research project is partially funded by Cognitec Systems GmbH (www.cognitec.com).

References

- [1] D. DeMenthon and L. Davis. Model-based object pose in 25 lines of code. *IJCV*, 1995.
- [2] M. Rätzsch, P. Quick, P. Huber, T. Frank, and T. Vetter. Wavelet reduced support vector regression for efficient and robust head pose estimation. *CRV*, 2012.
- [3] M. Rätzsch, G. Teschke, S. Romdhani, and T. Vetter. Wavelet frame accelerated reduced support vector machines. *TIP*, 2008.
- [4] S. Romdhani and T. Vetter. 3d probabilistic feature point model for object detection and recognition. *CVPR*, 2007.
- [5] X. Zhu and D. Ramanan. Face detection, pose estimation and landmark localization in the wild. *CVPR*, 2012.

Is Virtual Reality The Future Of The Video Games Industry?

Markus Rapp
Stuttgart Media University
mr113@hdm-stuttgart.de

Stefan Radicke
Stuttgart Media University
radicke@hdm-stuttgart.de

Abstract

In the past, Virtual Reality (VR) failed several times to be commercially successful in the games industry [6]. Today, Oculus VR tries to conquer the video games market place with their Head-Mounted Displays (HMD). Is the time right for VR games or is another failure of HMDs unavoidable?

The Oculus Rift Development Kit was utilized in order to answer this question. Two versions of a racing game prototype were developed to be able to compare a classical racing game with an Oculus Rift version, respectively. Since the degree of immersion in video games is a highly subjective and largely varies for each individual, a user test was conducted involving 40 people in order to find out if the gaming experience with VR is indeed more enjoyable. Furthermore, the user test was aimed to identify weaknesses of the Oculus VR hardware, which negatively affect immersion.

Keywords

Oculus Rift, Head-Mounted Displays, Virtual Reality, Games Development

CR-Categories

I.3.7 [Computer Graphics]:

Three-Dimensional Graphics and Realism---
Virtual reality

Betreuer Hochschule: Stefan Radicke
Hochschule der Medien
radicke@hdm-stuttgart.de

Informatics Inside 2014
Wissenschaftliche Vertiefungskonferenz
30. April 2014, Hochschule Reutlingen
Copyright 2014 Markus Rapp, Stefan Radicke

1 Introduction

The Oculus Rift Development Kit [3] seems to be the starting point for the success of VR in video games. The Oculus Rift was designed from ground up with computer games in mind. The main features of this HMD are optical lenses, which allow a field of view of 110 degrees vertical and 90 degrees horizontal, low latency rotational head tracking and the use of affordable smartphone displays. This allows Oculus VR to have a low price tag, which is one crucial factor to make VR mainstream.

In the following paragraphs the racing game prototype, the execution and the results of the user test are described.

2 Racing Game Prototype

The two versions of a racing game prototype were developed in Unity 3D [8]. The classical display version has an option to change the camera perspective between cockpit, bonnet, bumper and outside-view. The Oculus Rift version provides only the cockpit-view but with the added advantage of an enhanced field of view and rotational head tracking. Handling of the car and game assets are identical for both versions. As input device an Xbox 360 Wireless Racing Wheel is used. The steering wheel setup provides a fixed, forward directed and seated experience with the possibility to look around. It has no additional side effects and it is possible to identify cockpit drifts.

Both prototypes were created with the car tutorial [4] of Unity Technologies. The car model was replaced with a free available car model [5] of the website Archive 3D.

3 User Test

Each participant of the user test played both prototype versions. Half of the participants started the Oculus Rift version first and the other half began with the classical display version. Each participant played each version for up to 9 minutes. Lap times and the overall playing time were recorded to capture player performance. A webcam was used to record the entire user test. Afterwards the captured video data was analysed for cockpit drifts, unexpected body movements and the excitement of the participants. At the end the participants filled out a survey for statistical evaluation.

4 Results

One of our key questions was whether or not VR gaming actually leads to a higher immersion. This was answered very clearly. On a scale of 1 to 10, the average immersion for the Oculus Rift was rated with 7.775. In contrast the average immersion of the classical display version was rated with 4.2.



Figure 1: Possible hardware improvements

In terms of possible hardware improvements (Figure 1) our participants reported several issues. The relatively low resolution of the device was considered to be the most irritating aspect. Motion blur and judder [1] were recognized by most participants as an uncomfortable factor. Positional head tracking was identified as an important missing feature for more precise head tracking. Another problem was motion sickness. 75% of the participants suffered from motion sickness or dizziness.

Regardless of the technical issues and motion sickness, 70% had a better experience with the VR version and 60% would prefer to play a commercial racing game with a HMD.

5 Conclusion

VR has made huge steps forward. However, there are still technical problems and motion sickness that need to be solved before VR will be a commercial success. The biggest technical problem seems to be the display resolution. This is not surprising because Michael Abrash [2] and Tim Sweeney [7] claimed that for a field of view of 90 degrees a resolution of 8000-by-4000 for each eye is needed to reach the limit of human visual perception.

Despite all the technical issues the majority of the participants prefer VR. In taking to account that Oculus VR and their competitors are working on improved HMDs, it is likely that VR gaming experience will improve in the future.

6 References

- [1] Abrash, M. 2012. *Why virtual isn't real to your brain: judder.* <http://blogs.valvesoftware.com/abrash/>. Accessed 25 March 2014.
- [2] Abrash, M. 2012. *When it comes to resolution, it's all relative.* <http://blogs.valvesoftware.com/abrash/>. Accessed 6 March 2014.
- [3] Bolas, M. et al. 2013. *Open Virtual Reality.*
- [4] Echterdorf, J. 2014. *Car Tutorial by Unity Technologies.* <https://www.assetstore.unity3d.com/>. Accessed 7 March 2014.
- [5] Kvitelashvili, T. 2014. *Car McLaren MP4-12C N090211.* <http://archive3d.net>. Accessed 7 March 2014.
- [6] McFerran, D. 2014. *Reality Crumbles: Whatever happened to VR?* <http://www.eurogamer.net>. Accessed 25 March 2014.
- [7] Sweeney, T. *The Future of Gaming - Tim Sweeney (Epic) DICE 2012.*
- [8] Unity Technologies. 2014. *Unity - Game Engine.* <http://unity3d.com/>. Accessed 24 March 2014.

Oculus Rift mit externem Motion Tracking

Natascha Stumpp
Reutlingen University
Natascha.Stumpp@Student.R
eutlingen-University.de

Greg Rauhöft
Reutlingen University
Greg.Rauhoeft@Student.
Reutlingen-university.de

Florian Strieg
Reutlingen University
Florian.Strieg@Student.
Reutlingen-university.de

Abstract

In diesem Paper wird zunächst kurz das Virtual Reality Labor (VRLab) der Hochschule Reutlingen vorgestellt. Zu Beginn wird auf die Forschungsgebiete eingegangen, um daraufhin eines der aktuellen Projekte mit der Oculus Rift genauer zu erläutern. In diesem geht es um die Erweiterung des Head Mounted Displays um ein externes Motion Tracking Verfahren im Low-Cost Bereich.

Schlüsselwörter

Virtual Reality, Head-Mounted Display, Motion Tracking.

CR-Kategorien

I.4.6 [Image Processing and Computervision]: Tracking; I.3.7 [Three-Dimensional Graphics and Realism]: Virtual reality

1 Einleitung

Das VRLab der Hochschule Reutlingen ist seit 2006 darauf spezialisiert virtuelle Welten mit der Realität zu verbinden. Im Mittelpunkt des zweisemestrigen Masterprojektes steht

die Forschung im Bereich Mensch-Maschine-Interaktion in Bezug auf virtuelle Realitäten. Es geht um die Frage, wie sich neue Technologien verwenden lassen um die Immersion und Handhabung noch weiter zu verbessern. In den vergangenen Semestern wurden unter anderen Anwendungen für den Multi-Touch Cube des Labors und für Tracking Geräte, wie die Kinect und Leap Motion, entwickelt.

2 Oculus Rift

Eine der neuesten Errungenschaften des VRLabs ist die Entwicklerversion 1.1 der Oculus Rift. Dieses Head-Mounted Display versetzt den Benutzer direkt in eine virtuelle Umgebung und bietet neben dem Bildschirm, der das komplette Sichtfeld abdeckt, auch Bewegungssensoren, die Kopfbewegungen direkt in die Anwendung übertragen. Bei diesem Gerät erhält jedes Auge des Trägers ein eigenes, horizontal verschobenes Bild. So wird das menschliche Sehverhalten nachgeahmt, mit dem Unterschied, dass die Welt, die man wahrnimmt, über einen PC frei wählbar ist. [3]

In einem aktuellen Projekt soll die Oculus Rift um ein externes, low-cost Motion Tracking erweitert werden.

3 Warum Motion Tracking?

Die Oculus Rift enthält in der aktuellen Entwicklerversion zwar ein Gyroskop um die Roll-, Nick- und Gierwinkel des Tragenden zu messen, bietet aber keine Möglichkeit sonstige Bewegungen, wie die Position des

Betreuer Hochschule: Hochschule Reutlingen
Prof. Dr. Uwe Kloos
Uwe.Kloos@Reutlingen-
University.de
Prof. Dr. Gabriela Tullius
Gabriela.Tullius@Reutlingen-
University.de

Informatics Inside 2014
Wissenschaftliche Vertiefungskonferenz
30. April 2014, Hochschule Reutlingen
Copyright 2014 N. Stumpp, G. Rauhöft, F. Strieg

Benutzers im Raum, zu erfassen. [5] Es fällt zum Beispiel störend auf, dass das eingebaute Tracking der Oculus Rift keine Scherbewegungen erkennt und man daher nicht die Möglichkeit hat um Ecken, über Geländer oder Klippen zu schauen. [4]

Die nächste Entwicklerversion der Oculus Rift soll bereits mit einem Trackingsystem ausgeliefert werden. Dieses ist jedoch fest an die Brille gebunden. [4] Das Tracking System, das im VRLab entwickelt wird, soll zwar in Kombination mit der Oculus Rift verwendet werden, es soll jedoch auch unabhängig von ihr funktionieren, damit verschiedene Objekte erkannt und verfolgt werden können.

4 Warum Low-Cost?

Es gibt bereits mehrere Trackingsysteme auf dem Markt. Diese verwenden jedoch meist teure, proprietäre Kameras und Software, die für den Durchschnittsverbraucher nicht erschwinglich sind. Beispiele hierfür sind Precision Position Tracking von Worldviz [2], welches auch im VRLab installiert ist, sowie verschiedene Systeme von Vicon [1]. Es scheint bisher noch kein freies oder quelloffenes Motion Tracking System zu geben, eine Lücke, die im VRLab geschlossen werden soll.

Ein selbst eranntes Ziel des Oculus Rift Projektes ist es, eine möglichst günstige VR-Brille auf den Markt zu bringen. Der Hintergedanke ist eine breite Masse mit dem Produkt zu erreichen. [3] Dieser Philosophie folgend, wird im VRLab versucht eine möglichst kostengünstige Trackinglösung für die Oculus Rift zu finden. Zudem ist das angestrebte System durch den Einsatz von Webcams sehr handlich und portabel und kann schnell auf- und abgebaut werden. Dies ist wiederum praktisch für den Hausgebrauch, Spieleentwickler und für Amateurbenutzer.

5 Ansätze für Low-Cost Tracking

Im VRLab gab es bereits einige Ansätze für solche Systeme mit stereoskopischen

Kameras. Aus den stereoskopischen Bildern wird eine Tiefenkarte errechnet, aus der sich die Positionsdaten eines Punktes in den Bildern errechnen lassen. Dies wird jedoch unmöglich, wenn der Punkt nicht von beiden Kameras erfasst werden kann. Dieses Problem tritt beispielsweise auf, wenn ein Trackingpunkt hinter einem Gegenstand verschwindet. Aus diesem Grund wird nun ein neuer Ansatz mit mehreren Kameras verfolgt, die an beliebigen Positionen im Raum mit einem gemeinsamen Referenzpunkt angebracht werden können. Die genaue Position des Punktes wird dann aus der Übereinstimmung des Punktes aus vielen verschiedenen Perspektiven errechnet. Je mehr Kameras verwendet werden, desto genauer wird die Position errechnet und desto höher sind die Chancen, dass mehrere Kameras gleichzeitig einen Punkt erfassen. Des Weiteren wird auch an einer Trackinglösung unter Verwendung von Microsoft's Kinect gearbeitet. Da eine einzelne Kinect aber bei der Generierung von Tiefeninformationen auf ähnliche Probleme stoßen müsste wie ein stereoskopisches System, bleibt zu prüfen, ob es in dem Fall nicht sinnvoll wäre mehrere Kinects zum Tracking zu verwenden.

6 Literaturverzeichnis

- [1] Vicon website:
<http://www.vicon.com/System/TSeries>
(Stand: 16.03.2014)
- [2] Worldviz website:
<http://www.worldviz.com/products/pptr>
(Stand: 16.03.2014)
- [3] Oculus Rift Website:
<http://www.oculusvr.com/rift/> (Stand: 17.03.2014)
- [4] Oculus Rift Crystal Cove Website:
<http://www.oculusvr.com/blog/crystal-cove-debut-ces-recap-and-steam-dev-days/> (Stand: 17.03.2014)
- [5] Oculus Rift Sensoren Website:
<http://www.oculusvr.com/blog/sensor-fusion-keeping-it-simple> (Stand: 17.03.2014)

Autoren der Hauptvorträge



André Antakli

*andre.antakli@student.
reutlingen-university.de*



Thomas Bauer

*thomas.bauer@student.
reutlingen-university.de*



Olaya De la Rosa Avitia

*olaya.de_la_rosa_avitia@
student.reutlingen-univer-
sity.de*



Matthias Gutekunst

*matthias_sebastian.gute-
kunst@student.reutlingen-
university.de*



Viktoria Hoffmann

*viktoria.hoffmann@student.
reutlingen-university.de*



Johannes Kartheininger

*johannes.kartheininger@
student.reutlingen-univer-
sity.de*



René Mangold

*rene.mangold@student.
reutlingen-university.de*



Stanislas Mauser

*stanislas_christophe_yves.
mauser@student.reutlin-
gen-university.de*



Lars Schneider

*lars_rainer.schneider@stu-
dent.reutlingen-university.
de*



Arkadius Weister

*arkadius.weister@student.
reutlingen-university.de*



Anna Wellerdiek

*anna.wellerdiek@student.
reutlingen-university.de*



/DIE SPEICHEREXPERTEN



Advanced UniByte (AU) ist der Experte für Datenspeicher-Lösungen. Mit über 20 Jahren Erfahrung zählen wir zu den führenden Systemhäusern in Deutschland, wenn es um die Planung und Umsetzung komplexer Datenspeicher- und Virtualisierungslösungen geht. In unserem unternehmenseigenen Labor werden alle Konzepte auf „Herz und Nieren“ geprüft, denn begeisterte Kunden und langfristige Kundenbeziehungen sind uns wichtig. Zu unseren Kunden gehören namhafte Industrieunternehmen, Banken und Sparkassen, Behörden, Hochschulen und große Forschungseinrichtungen in Baden-Württemberg, Deutschland und weltweit. Unsere Kunden schätzen unsere Expertise, die sich in zahlreichen Auszeichnungen (z.B. Bestes Systemhaus 2013) widerspiegelt.

Für diesen spannenden Zukunftsmarkt suchen wir Menschen, die sich für das Thema Storage interessieren, neugierig auf neue Entwicklungen sind, Verantwortung übernehmen, gerne die Zukunft mitgestalten wollen und sich mit einer Firmenphilosophie identifizieren, die **GUT. ECHT. ANDERS.** ist.

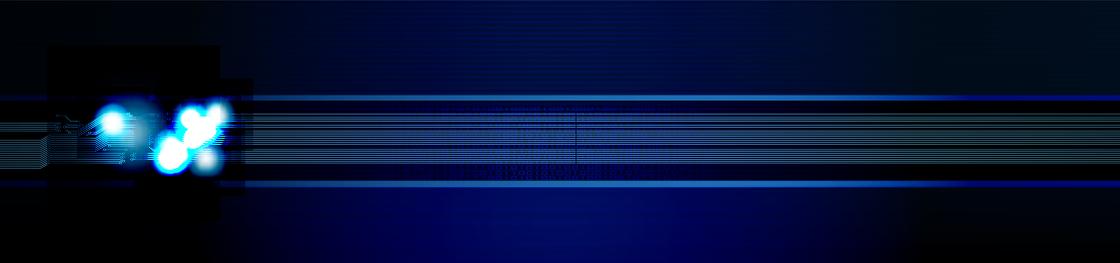
Wir vergeben

im Bereich Wirtschaftsinformatik sowie Medien- und Kommunikationsinformatik

- Traineeprogramme
- Bachelor- & Masterabschlussarbeiten
- Werkstudententätigkeiten
- Festanstellung
- Praktika

Kontakt

Advanced UniByte GmbH
Abteilung Personal & Ausbildung
Teresa Monopoli, Tel: 07121 483-258
Birnenweg 15, 72766 Reutlingen
E-Mail: bewerbung@au.de
www.au.de/jobs



E-Mail: infoinside@reutlingen-university.de
Internet: <http://www.infoinside.reutlingen-university.de>

Hochschule Reutlingen
Fakultät Informatik
Medien- und Kommunikationsinformatik
Alteburgstraße 150
D-72762 Reutlingen
<http://www.mki.reutlingen-university.de>

Telefon: +49 7121 271 -4002
Telefax: +49 7121 271 -4042



ISBN 978-3-00-045427-1



Umschlagsbanner:

Background Vector Abstract Futuristic by Vecart

(<http://vecteezy.com/backgrounds-wallpaper/29033-background-vector-abstract-futuristic>)
steht unter der Creative-Commons-Lizenz Namensnennung 3.0 Unported.

Um eine Kopie dieser Lizenz zu sehen, besuchen Sie <http://creativecommons.org/licenses/by/3.0/deed.de>

