



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Masterarbeit

Sobin Ghose

**Integrationsplattform zur Untersuchung von Gestensteuerung
in Smart Environments**

*Fakultät Technik und Informatik
Studiendepartment Informatik*

*Faculty of Engineering and Computer Science
Department of Computer Science*

Sobin Ghose

**Integrationsplattform zur Untersuchung von Gestensteuerung
in Smart Environments**

Masterarbeit eingereicht im Rahmen der Masterprüfung

im Studiengang Master of Science Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Kai von Luck
Zweitgutachter: Prof. Dr. Philipp Jenke

Eingereicht am: 12. Juli 2019

Sobin Ghose

Thema der Arbeit

Integrationsplattform zur Untersuchung von Gestensteuerung in Smart Environments

Stichworte

3D-Gestensteuerung, Gestenerkennung, Smart Environments, Smart Home, Mensch-Maschinen-Interaktion (HMI), Integrationsplattform, HCI Studien, Multimodale Systeme

Kurzzusammenfassung

In dieser Arbeit konnte der aktuelle Forschungsstand der 3D-Gestenerkennung analysiert und ein Grundproblem von Interaktionssystemen identifiziert werden. Die Vergleichbarkeit von verschiedenen Interaktionskonzepten, Algorithmen, Sensoren, Gesten und Versuchsaufbauten ist durch die unterschiedlichen Variationen verschiedener Parameter schwer herzustellen. Die Entwicklung einer Integrationsplattform zur Untersuchung von Gestensteuerung in Smart Environments wird als mögliche Lösung, um eine Vergleichbarkeit herzustellen, vorgestellt. Diese Integrationsplattform wurde im Rahmen der vorliegenden Arbeit erfolgreich konzipiert, entwickelt und evaluiert.

Sobin Ghose

Title of the paper

Integration platform for HCI Studies in Smart Environments

Keywords

3D-Gesture Control, Gesture Recognition, Smart Environments, Smart Home, Human-Computer-Interaction (HCI), Integration Platform, HCI Studies, Multimodal Systems

Abstract

In this thesis the current state of research of 3D gesture recognition has been analyzed and a major problem of interaction systems could be identified. The comparability of different interaction concepts, algorithms, sensors, gestures and experimental set-ups is hardly achievable by varying different parameters. The development of an integration platform for the investigation of gesture control in Smart Environments is presented as a possible solution to establish comparability. This integration platform was successfully designed, developed and evaluated within the scope of this work.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Ziele	2
1.3	Gliederung der Ausarbeitung	2
2	Analyse	4
2.1	Grundsätzliche Betrachtung der Fragestellung	4
2.1.1	Vorstudie: Interaktiver Badezimmerspiegel	4
2.1.2	Vision zur gestenbasierten Interaktion in Smart Homes	6
2.1.3	Fragestellung	7
2.2	Gestik	8
2.2.1	Kulturelle Interpretation	10
2.2.2	Midas Touch Problem	11
2.3	Gestenerkennung	12
2.3.1	Grundlegende Verfahren der Gestenerkennung	12
2.3.2	Sensorik	13
2.4	Kontexterkenkung zur Unterstützung von Gestenerkennung	15
2.4.1	Anwendungskontext	15
2.4.2	Kontextsensitivität (Context-aware Computing)	16
2.4.3	Emotion-Aware-Computing	16
2.4.4	Weitere Techniken zur Kontexterkenkung	18
2.4.5	Kontext nach Dourish	19
2.4.6	Schlussfolgerungen für die kontextabhängige Gestensteuerung	19
2.5	Vergleichbare Arbeiten Gestenerkennung	20
2.5.1	MISO: A Context-sensitive Multimodal Interface for Smart Objects Based on Hand Gestures and Finger Snaps	20
2.5.2	Imaginary Interfaces: Spatial Interaction with Empty Hands and Without Visual Feedback	22
2.5.3	Exploring the Usefulness of Finger-Based 3D Gesture Menu Selection	24
2.5.4	Bremen Ambient Assisted Living Lab	27
2.5.5	Fazit der vergleichbaren Arbeiten	31
2.6	Verfahren zur Echtzeit Gestenerkennung	32
2.6.1	Maschinelles Lernen	32
2.6.2	Heuristik basierte Verfahren	34
2.6.3	Rückschlüsse zur Verfahrenswahl	36

2.7	Companion-Systeme	36
2.7.1	Anwendungsgebiete und Vorläufer	37
2.7.2	Forschungsbereiche von Companion-Systemen	40
2.8	Vergleichbare Arbeiten Multimodale Plattformen	42
2.8.1	Companion Technology for Multimodal Interaction	42
2.8.2	Human Gesture Analysis using Multimodal Features	46
2.8.3	Combining Speech, Gaze, and Micro-Gestures for the Multimodal Control of In-Car Functions	46
2.8.4	Rapid Development of multimodal Interactive Systems: A Demonstration of Platform for Situated Intelligence	47
2.9	Rückschlüsse auf Basis des Forschungsstands	47
2.9.1	Interaktionsrepertoire und Bedienungskonzept	47
2.9.2	Identifizierte Herausforderungen und Fragestellungen der Gestenerkennung	48
2.10	Integrationsplattform als Lösungsansatz	50
2.11	Laborumgebung Living Place Hamburg	52
2.12	HCI Vorstudie zu Gesten im Smart Home	53
2.13	Zusammenfassung der Analyse	55
3	Design	57
3.1	Konzeption	57
3.1.1	Plattformarchitektur	57
3.1.2	Systemarchitektur des Interaktion-Management-Systems	58
3.2	Vorversuch: Evaluation of an Omnidirectional Walking-in-Place User Interface	61
3.3	Zusammenfassung des Designs	63
4	Realisierung	65
4.1	Framework und Middleware	65
4.1.1	Framework	65
4.1.2	Middleware	66
4.2	Systemkomponenten der Integrationsplattform	67
4.2.1	Kinect v2 Agent	68
4.2.2	Omnidirectional Fusion Agent	70
4.2.3	Multimodal Fusion Agent	73
4.2.4	Gesture Recognition Toolkit und Agenten	75
4.2.5	Interaction Management Agent	76
4.2.6	Vibration Agent	76
4.3	Integration in die Living Place Architektur	79
4.4	Zusammenfassung der Realisierung	81
5	Evaluation	82
5.1	Alpha-Test	82
5.2	Beta-Test: Nutzbarkeit für Entwickler	83

5.3	Diskussion der Erfüllung der funktionalen Anforderungen	84
5.4	Gestenbasierte HCI Studie auf Basis der Integrationsplattform	85
5.4.1	Aufbau der Studie	85
5.4.2	Explorative Vorstudie	86
5.4.3	Implementation des Interaktions Repertoire	86
5.4.4	Validierung der prototypischen Umsetzung	87
5.4.5	Rückschlüsse der Studie für die Integrationsplattform	89
5.5	Fazit der Evaluierung	89
6	Schlussbetrachtung	91
6.1	Zusammenfassung	91
6.2	Fazit und Ausblick	92

Abbildungsverzeichnis

1.1	Smart Board und Tablets als neue Bedienungskonzepte	1
2.1	Veröffentlichung zum Thema Gesten bei der ACM	7
2.2	Klassifizierung von Gesten nach Pavlovic	9
2.3	Beispiele für kulturelle Unterschiede von Gesten	11
2.4	Körpermodell der Kinect v2 von Microsoft	13
2.5	Leap Motion	14
2.6	Kontext von Gesichtsmimiken	16
2.7	Finger Snap	21
2.8	Imaginary Interfaces - Auswahl	23
2.9	Imaginary Interfaces Koordinatensystem	24
2.10	Finger-Based 3D Gesture Menu Selection	26
2.11	Bremen Ambient Assisted Living Lab	27
2.12	Rolland, der autonome Rollstuhl des BAAL	31
2.13	Support Vector Machines - Beispiel einer optimalen Hyperebene	33
2.14	RealEdge Prototyp auf Basis heuristischer Verfahren	35
2.15	Roboter Robbe Paro	38
2.16	Entertainment-Companion Furby	39
2.17	Hauptfunktionseinheiten eines Companion-Systems	42
2.18	Schematische Darstellung der Architektur eines Companion-Systems	43
2.19	Prototyp des <i>Multimodal-Interaction-Companion</i>	44
2.20	3D-Modell des Living Place Hamburg	53
2.21	Wizard-of-Oz-Experiment im Living Place Hamburg	54
2.22	Tabelle zur präferierten Steuerung des Lichts beim Wizard-of-Oz-Experiment	55
3.1	Systemarchitektur des Interaktion-Management-Systems	58
3.2	Komponenten des Interaktion-Management-Systems	60
3.3	Versuchsaufbau mit vier Kinects v2	62
3.4	Virtuelle Hammaburg	62
4.1	Systemkomponenten der Integrationsplattform	67
4.2	Kinect v2 von Microsoft	68
4.3	Laboraufbau des Omnidirectional Fusion Agents	70
4.4	Spiegelung des Skelett-Modells	72
4.5	Sender und Empfänger auf XBee und Arduino Basis	77
4.6	Zweiter Prototyp des Vibrationsarmbands	78
4.7	Web Interface der RESTful API für Licht, Fenster, Gardinen und Rollos	80

5.1	Tabelle zur präferierten Steuerung des Lichts im Living Place	87
5.2	Probanden steuern die Rollos im Wohnzimmer des Living Place mittels Gesten auf Basis der Integrationsplattform.	88
6.1	Agile User-Centered-Development-Methode auf Basis der Integrationsplattform	93

1 Einleitung

1.1 Motivation

Bereits vor mehr als 25 Jahren hat Mark Weiser durch seine Vision die Art und Weise, wie wir aktuell und zukünftig mit Computern umgehen, maßgeblich geprägt. In "The computer for the 21st century" beschreibt Weiser (1991) unter anderem die heutige Rechnerallgegenwärtigkeit (ubiquitous computing) und neue Bedienungskonzepte (siehe Abb.: 1.1), welche es uns ermöglichen, durch natürliche Schnittstellen wie 2D- und 3D-Gesten oder Sprachsteuerung intuitiv und nahtlos mit unserer Umgebung zu interagieren.

So wird beispielsweise bereits Sprache zur Steuerung des Autonavigationsgeräts, des Smartphones oder für Bestellungen im Internet genutzt sowie 2D-Gesten zur Steuerung von Touchscreens oder 3D-Gesten als Alternative zum Controller bei Spielkonsolen oder Virtual Reality Anwendungen.

Auch wenn die Systeme stetig verbessert werden, können diese Schnittstellen aktuell nur für beschränkte Aufgaben genutzt werden, da komplexere Anwendungen meist den Kontext des Nutzer mit einbeziehen müssten, um eine akkurate Interpretation der Situation herstellen zu können. David McNeill zeigte bereits 1992 in McNeill (1992), dass beispielsweise die Interpretation ei-

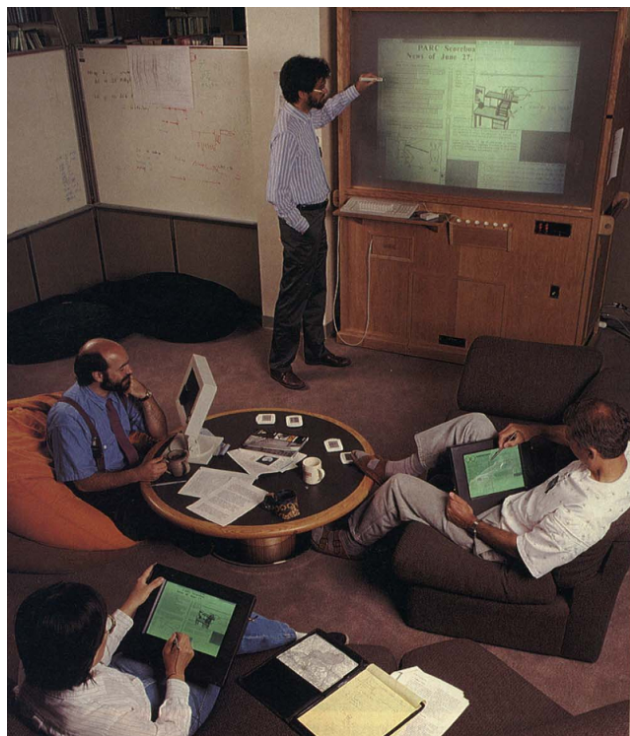


Abbildung 1.1: Smart Board und Tablets als neue Bedienungskonzepte [Weiser (1991)]

nes Satzes häufig nur unter Betrachtung der begleitenden Gestik interpretiert werden kann. Dieser Zusammenhang spielt insbesondere bei der Interaktion mit Smart Environments (Intelligente Umgebungen) eine wichtige Rolle. Die vielfältigen Anwendungsgebiete, in denen Smart Environments genutzt werden können, gehen über Intelligente Arbeitsplätze, Smart Homes, Wearables, Intelligente Flugkabinen und Smartphones hinaus.

Auch wenn in den letzten Jahren große Fortschritte bei der Ausstattung von Smart Environments wie beispielsweise bei Smart Homes mit intelligenter Haustechnik und deren Ansteuerung per Smartphone und Spracherkennung errungen worden sind, ist die Interaktion mit diesen häufig noch nicht nahtlos und intuitiv.

Daher soll in dieser Arbeit untersucht werden, wie 3D-Gesten zur einer natürlichen Interaktion mit Smart Environments am Beispiel von Smart Homes eingesetzt werden können und welche Vorgehensweisen sich zur Entwicklung dieser Mensch-Computer-Interaktion (*Human-computer interaction* abgekürzt HCI) eignen.

1.2 Ziele

Ziel dieser Arbeit ist, den aktuellen Forschungsstand zur Interaktion mit 3D-Gesten zu untersuchen. Dazu soll überprüft werden, wie multimodale Ansätze mit dem Schwerpunkt auf Gestensteuerung für eine natürliche und nahtlose Interaktion mit Smart Environments eingesetzt und die Entwicklung solcher Systeme leichtgewichtig und effektiv umgesetzt werden können.

Um dies konkret zu realisieren, wird die Entwicklung einer Integrationsplattform für eine Smart Home als Unterstützung für HCI Entwickler angestrebt. Diese soll die Vergleichbarkeit verschiedener Ansätze und damit die Untersuchung von natürlichen und nahtlosen Interaktionen ermöglichen.

1.3 Gliederung der Ausarbeitung

Am Anfang der Analyse (Kapitel 2) wird die grundsätzliche Fragestellung (Abschnitt 2.1) genauer erläutert und auf die verschiedenen Arten und Kategorien von Gesten (Abschnitt 2.2), so wie deren Nutzen für eine Interaktion mit Smart Environments eingegangen.

Aufbauend darauf werden die grundlegenden Verfahren der 3D-Gestenerkennung diskutiert (Abschnitt 2.3.1) und auf weitere Modalitäten bei der Interaktion durch Gesten und den Kontext dieser eingegangen (Abschnitt 2.4), um den Rahmen der Herausforderungen bei der

3D-Gestenerkennung herauszustellen.

Im Anschluss werden aktuelle Forschungsarbeiten zur 3D-Gestenerkennung vorgestellt (Abschnitt 2.5) und die grundsätzlichen Verfahren und Algorithmen für Gestenerkennung diskutiert (Abschnitt 2.6).

Um die Architektur von Multimodalen Systemen näher zu betrachten, werden anschließend Companion Systeme (Abschnitt 2.7) und die Architektur von Multimodalen Plattformen im Allgemeinen vorgestellt und diskutiert (Abschnitt 2.8).

Abschließend werden Rückschlüsse der Analyse gezogen (Abschnitt 2.9) und ein Lösungsansatz für aktuelle Herausforderungen sowie die Entwicklung von natürlicher Interaktion mit Smart Environments durch 3D-Gesten vorgestellt (Abschnitt 2.10).

Im Design wird die Konzipierung und die Architektur des Systems auf Basis der in der Analyse gewonnen Erkenntnisse, entwickelt (Kapitel 3) und in der Realisierung auf die Umsetzung dieses eingegangen (Kapitel 4).

Aufbauend auf den in der Realisierung implementierten Prototypen, wird in der Evaluierung das System durch einen Alpha-Test, einen Beta-Test sowie eine HCI-Studie evaluiert und die Ergebnisse diskutiert (Kapitel 5).

Abschließend wird eine Schlussbetrachtung der Fragestellung sowie der vorgestellten Lösung durch ein Fazit und einen Ausblick vorgenommen (Kapitel 6).

2 Analyse

In der Analyse werden die aktuellen Herausforderungen der Gestenerkennung in Smart Environments identifiziert. Anschließend wird der Forschungsstand zur Gestik und der 3D-Gestenerkennung diskutiert sowie eine Übersicht von vergleichbaren Arbeiten der Gestenerkennung vorgestellt. Nachfolgend wird das Thema dieser Arbeit konkretisiert und die Rahmenbedingungen für Systeme zur Gestenerkennung diskutiert sowie die Laborumgebung erläutert.

2.1 Grundsätzliche Betrachtung der Fragestellung

Für eine genauere Betrachtung der zugrundeliegenden Fragestellung dieser Arbeit wird an dieser Stelle kurze auf eine Vorarbeit eingegangen, bevor die Fragestellung weitergehend diskutiert wird.

2.1.1 Vorstudie: Interaktiver Badezimmerspiegel

In der Arbeit [Ghose \(2014\)](#) wurde ein interaktiver Badezimmerspiegel für die nahtlose Interaktion mit einem Smart Object im Badezimmer entwickelt. Dabei kamen bereits verschiedene Modalitäten wie 3D-Gesten und Sprachsteuerung zum Einsatz, da eine Interaktion per Touch oder Maus und Tastatur für einen Spiegel im Badezimmer ungeeignet sind. In diesem Versuch konnte die Bedienung eines Objektes mit eingeschränktem Interaktionsraum beispielhaft veranschaulicht werden.

Dieser Spiegel verfügt über verschiedene Modi, sodass der User zwischen einem normalen Spiegel, einer Mischform, in welcher der User sich selbst im Spiegel sowie Teile der Benutzeroberfläche sieht und einem dritten Modus, dem Display Modus, wählen kann. Im Display Modus kann der User den Spiegel wie ein normales Display verwenden. Der Spiegel ermöglicht dem Benutzer Zugriff auf Anwendungen, welche beispielsweise den Zugriff auf E-Mails oder den digitalen Kalender ermöglichen. Zudem kann der Nutzer allgemeine Informationen wie das aktuelle Wetter, Wettervorhersage sowie die Uhrzeit und das Datum abrufen.

Das Interaktionsrepertoire beruhte auf drei verschiedenen Interaktionsarten. So ermöglichte das System durch Markieren, Aktivieren und Scrolling die Bedienung der Oberfläche. Diese konnte jeweils durch Sprachbefehle, 3D-Gesten oder eine Kombination aus beidem verwendet werden.

Rückschlüsse aus der Vorstudie

Durch die Vorstudie konnten erste Erfahrungen bei der semantischen Vermischung von Sprach- und Gestensteuerung sowie der Umgang mit entsprechenden Sensoren und der Verarbeitung der Daten gesammelt werden. Nutzertests zeigten, dass einige Nutzer eine andere Art der Gestensteuerung erwartet hätten. Nach kurzer Erläuterung der Funktionsweise der Gestenerkennung konnten sie jedoch den Spiegel problemlos bedienen. Die semantische Vermischung von Sprach- und Gestensteuerung wurde gut angenommen, da dies dem Nutzer die Wahl ließ, welche Steuerelemente durch Sprache oder Gestik angesteuert werden sollten. So wurde beispielsweise das Herunterscrollen scrollen einer Liste durch eine kontinuierliche Geste als effektiver wahrgenommen als die Wiederholung des entsprechenden Sprachbefehls.

Auf Basis dieser Erfahrungen der Vorstudie soll nun versucht werden, die Anforderungen an ein System für Gestensteuerung in Smart Environments zu erarbeiten, welche gegenüber der Steuerung von Smart Objects, wie dem Badezimmerspiegel neue Herausforderungen bietet. Der Fokus soll dabei auf der 3D-Gestensteuerung gegenüber der Sprachsteuerung liegen, da die Entwicklung der Sprachsteuerung durch Siri¹, Google Now², Cortana³ und Alexa⁴ aktuell stark voranschreitet.

Dabei gibt es bei der Betrachtung von Smart Environments gegenüber des interaktiven Badezimmerspiegels einen freien Interaktionsraum sowie einen offenen Kontext der Umgebung und des Nutzers. Allerdings besteht die große Herausforderung darin, eine optimale Kombination von der Auswahl des Gestenrepertoires, der Sensorik, der Algorithmen sowie der zugrundeliegenden Plattform zu finden.

Zudem zeigten sich unterschiedliche Wahrnehmungen bei der intuitiven Nutzung der Gestensteuerung zwischen dem Entwickler und den Nutzern, sodass die Steuerung kurz erläutert werden musste. Daher ist ebenso eine genauere Betrachtung der Wahl der Gestik für eine

¹<https://www.apple.com/de/siri/>

²<https://www.google.com/intl/de/landing/now/>

³<https://www.microsoft.com/de-de/windows/cortana>

⁴<https://www.amazon.de/b?node=12775495031>

nahtlose intuitive Interaktion nötig.

Um das Spektrum an möglichen Anforderungen zu verdeutlichen, wurde die folgende Vision entwickelt.

2.1.2 Vision zur gestenbasierten Interaktion in Smart Homes

In der folgenden Vision soll gezeigt werden, wie sich das zukünftige Leben in Smart Homes gestalten kann. Dabei liegt der Fokus auf der Steuerung der Haustechnik durch Gesten, Sprache, Mimik und Emotionen. Ferner ist das Szenario an die bereits vor 25 Jahren entwickelte Vision von Mark Weiser in "The computer for the 21st century" [vgl. [Weiser \(1991\)](#)] angelehnt und für eine gestenbasierte Interaktion angepasst.

Der morgendliche Aufenthalt in einer intelligenten Wohnung

Behutsam wird im Schlafzimmer das Licht, einem Sonnenaufgang nachempfunden, verstärkt, und die Vorhänge langsam hochgefahren. Durch das natürlich nachempfundene Licht erwacht Tony auf sanfte und angenehme Weise. Auf dem Weg ins Badezimmer bemerkt Tony, dass ihm das Licht doch noch etwas zu hell ist und befiehlt der Wohnung per Sprachbefehl, das Licht leicht zu dimmen. Beim Zähneputzen vor seinem intelligenten Badezimmerspiegel kann Tony bereits sehen, dass sein erster Termin heute morgen verschoben wurde.

Durch die dazu gewonnene Zeit kann Tony noch in Ruhe auf dem Sofa seinen Kaffee genießen. Um schneller wach zu werden öffnet Tony durch Zeigen und den Sprachbefehl "öffnen" zwei Fenster, damit frische Luft in den Raum kommt. Beim Lesen seiner Mails reflektiert die aufgehende Sonne stark auf dem Fernseher. Daher signalisiert er der Wohnung durch das Herunterheben beider Hände, alle Rollos im Zimmer herunterzufahren. Beim verlassen des Wohnzimmers möchte Tony die Fenster schließen und zeigt dafür auf die offenen Fenster. Dies missinterpretiert die Wohnung und lässt das Rollo des Fensters wieder nach oben fahren. Durch die Irritation in Tonys Gesicht hebt die Wohnung selbstständig den Befehl auf und fragt Tony, ob er stattdessen das Fenster schließen wollte. Tony bejaht diese Frage und macht sich für die Arbeit fertig. Auf dem Weg zur Tür informiert die Wohnung Tony, dass es auf dem Weg zur Arbeit durch einen Verkehrsbehinderung aktuell zu einem Stau kommen könnte und fragt Tony, ob er eine alternative Route benutzen möchte. Tony lässt sich die alternative Route anzeigen und ändert diese, um vorher noch eine Kleinigkeit auf dem Weg zu erledigen. Die Wohnung verabschiedet sich von Tony so, dass Tony gutgelaunt in einen neuen Tag starten kann.

Erläuterung und Schlussfolgerungen aus der Vision

In der eben beschriebenen Vision (2.1.2) wurde als Beispiel für Smart Environments ein Smart Home herangezogen. Gegenüber anderen Smart Environments wie beispielsweise einer Cockpit-Situation oder eines Sitzplatzes in Autos oder Flugzeugen, besteht die Herausforderung, dass es keinen eingeschränkten Interaktionsraum sowie einen extrem breiten Kontext und eine Vielzahl von möglichen Anwendungen gibt. Die Interaktion zwischen Tony und der Wohnung ist bewusst wie die Interaktion zwischen zwei Freunden bzw. wie mit einem Companion dargestellt.

Der Zweck dieser entwickelten Vision liegt darin, dass durch die Untersuchung verschiedene Arten von Gestik, Sensorik sowie vergleichbare Arbeiten nun in einen konkreten Kontext gebracht werden können.

2.1.3 Fragestellung

Auch wenn konzeptionelle Untersuchungen der Gestenerkennung für die Mensch-Computer-Interaktion bereits vor fast 30 Jahren von Forschern wie u.a. Weiser (1991) untersucht worden sind, zeigt ein Blick auf die Zahlen der Veröffentlichungen zu diesem Thema bei der ACM, dass diese in den letzten Jahren stark an Bedeutung zugenommen hat. (siehe Abb.: 2.1)

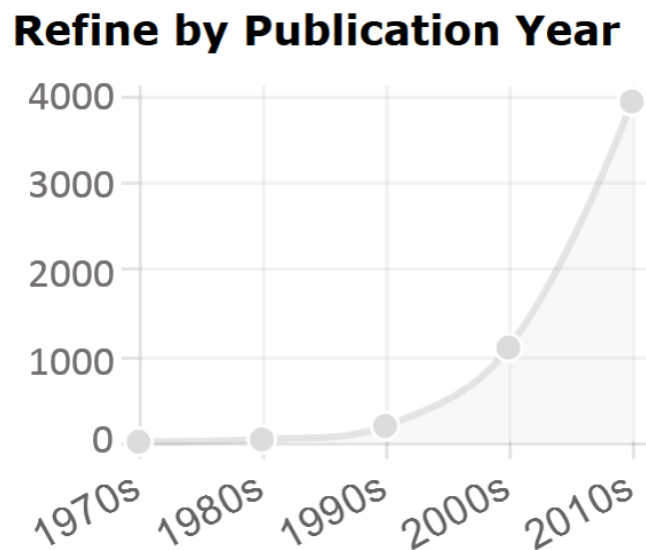


Abbildung 2.1: Veröffentlichung zum Thema Gesten bei der ACM [ACM (2019)] Anzahl der Veröffentlichungen pro Jahrzehnt

Dies lässt sich durch die technologische Entwicklung in den letzten Jahren und der damit zusammenhängenden, steigenden Verbreitung von 3D-Gestenerkennung erklären. Kontinuierlich wurde ebenso die Forschung fortgeführt bis hin zu Anwendungen für Smart Environments oder Mixed Reality Anwendungen.

Daher beschäftigt sich die Analyse dieser Arbeit damit, wie effektiv diese Systeme zur Gestenerkennung funktionieren, welche Forschungsansätze sich etabliert haben und welche aktuellen Forschungsansätze vielversprechend für die Weiterentwicklung sind. Des Weiteren soll sowohl analysiert werden, wie ein System zur Gestenerkennung in ein Smart Environment integriert werden kann, als auch wie gut bestehende Ansätze in diesem Bereich bereits funktionieren. Um diese Fragestellung zu beantworten, werden im Folgenden diese vier Teilbereiche analysiert:

- Gestik
- Sensorik
- Algorithmen
- Multimodale Systeme

Daher werden die konkreten Anforderungen an ein System zur Interaktion über Gesten in Smart Environments im Anschluss der Betrachtung des aktuellen Forschungsstands dieser Bereiche aufgestellt (siehe Abschnitt 2.10).

2.2 Gestik

Im Folgenden sollen Grundlagen von Gesten und der Gestenerkennung in Bezug auf Systeme im Kontext für Smart Environments analysiert werden. Dies soll zur Klärung der Rahmenbedingungen sowie der Anforderungen dienen.

Klassifizierungen von Gesten

Zur Klärung, was unter einer Geste verstanden wird und welche Arten von Gesten für die Ansteuerung und Interaktion mit Smart Environments geeignet sind, wird im Folgenden auf verschiedene Klassifizierungen und Kategorisierungen von Gesten eingegangen.

Eine der am meisten verbreiteten allgemeinen Kategorisierung wurde von McNeill (1992) erarbeitet. Diese unterteilt Gesten in die Kategorien: ikonische, metaphorische, didaktische und rhythmische (sprachbegleitende) Gesten. Im Folgenden wird kurz auf drei für die HCI wichtigen Gestentypen eingegangen.

- Zeigegesten: Bei Zeigegesten oder auch didaktischen Gesten wird durch das Zeigen mit der ausgestreckten Hand oder einem Finger auf ein Objekt oder einen Ort hingewiesen. Da Zeigegesten zu den als ersten erlernten Gesten von Kindern gehören, sind diese besonders intuitiv.
- Ikonische Gesten: Ikonische Gesten werden häufig sprachbegleitend genutzt. Dabei wird das Verhalten oder die Art realer Objekte nachempfunden, wie es typischerweise bei der Pantomime getan wird. Es können sowohl Orte als auch Objekte oder Handlungen durch ikonische Gesten beschrieben werden.
- Metaphorische Gesten: Durch metaphorische Gesten werden abstrakte Bilder dargestellt. Beispielsweise wünscht man jemandem Erfolg, wenn man ihm den Daumen nach oben zeigt. Auch bekannt ist die Geste des Zurechtrückens einer imaginären Fliege als Metapher für "sich schick machen". Sie können auch zur Beschreibung der Manipulierung eines imaginären Objekts verwendet werden.

In [Pavlovic u. a. \(1997\)](#) wurde eine weitere Klassifizierung von Gesten für die Mensch-Computer-Interaktion vorgenommen. Die erste Unterscheidung nach [Pavlovic u. a. \(1997\)](#) besteht darin, die Bewegungen in Absichtslose bzw. Ziellose und Gesten, also Bewegungen mit einer Intention, zu unterscheiden (siehe [Abb.: 2.2](#)).

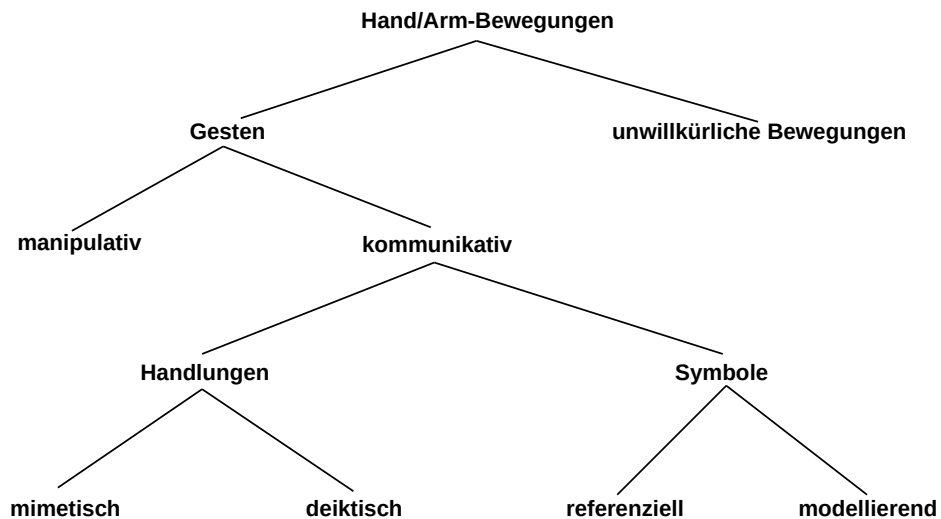


Abbildung 2.2: Klassifizierung von Gesten nach [Pavlovic u. a. \(1997\)](#). Gesten werden in manipulative und kommunikative Gesten unterteilt.

Eine Weiterentwicklung der auf [Pavlovic u. a. \(1997\)](#) basierenden Klassifizierung wurde bei den Wismarer Wirtschaftsinformatiktagen 2008 durch eine Arbeitsgruppe erarbeitet [vgl. [Boetzer u. a. \(2008\)](#)]. Dabei werden die von Pavlovic beschriebenen manipulativen Gesten in drei weitere Unterkategorien aufgeteilt.

- **Bewegungsverfolgende Gesten:** Intuitive Gesten, welche meist einer Manipulation von Objekten in der realen Welt nachempfunden wurden.
- **Kontinuierliche Gesten:** Diese Geste haben keinen direkten Zusammenhang mit der realen Welt, dafür sind sie jedoch stark metaphorisch an die reale Welt angelehnt und dadurch einfach erlernbar.
- **Symbolisch-manipulative Gesten:** Da in der virtuellen Welt grundsätzlich mehr Interaktionsmöglichkeiten mit der Umgebung als in der realen Welt denkbar sind, können für diese Interaktion neue abstrakte Gesten entwickelt werden. Diese Symbolisch-manipulativen Gesten müssen jedoch vom Nutzer erlernt werden.

Grundsätzlich werden in der Mensch-Computer-Interaktion aktuell größtenteils die lautsprachersetzenden Gesten in Abgrenzung zu lautsprachbegleitenden Gesten zur Bedienung und Steuerung genutzt. Jedoch sollte eine intuitive Steuerung auch lautsprachbegleitende Gesten interpretieren können, um eine nahtlose Interaktion zu erreichen. Weitere Erläuterungen dazu finden sich in [Abschnitt 2.4](#).

2.2.1 Kulturelle Interpretation

Durch unterschiedliche Kulturen kann die Gestik, bzw. die Interpretation von Gestik in unterschiedlichen Ländern stark variieren. Dies ist besonders in HCI-Anwendungen, welche weltweit genutzt werden, zu berücksichtigen. In [Cassell \(1998\)](#) werden einige Gesten, die stark abweichende Bedeutung in unterschiedlichen Ländern haben, aufgezeigt. Diese sind in [Abbildung 2.3](#) aufgeführt. Eine nähere Betrachtung von kulturellen Unterschieden liefert unter anderem [Cassell \(1998\)](#).

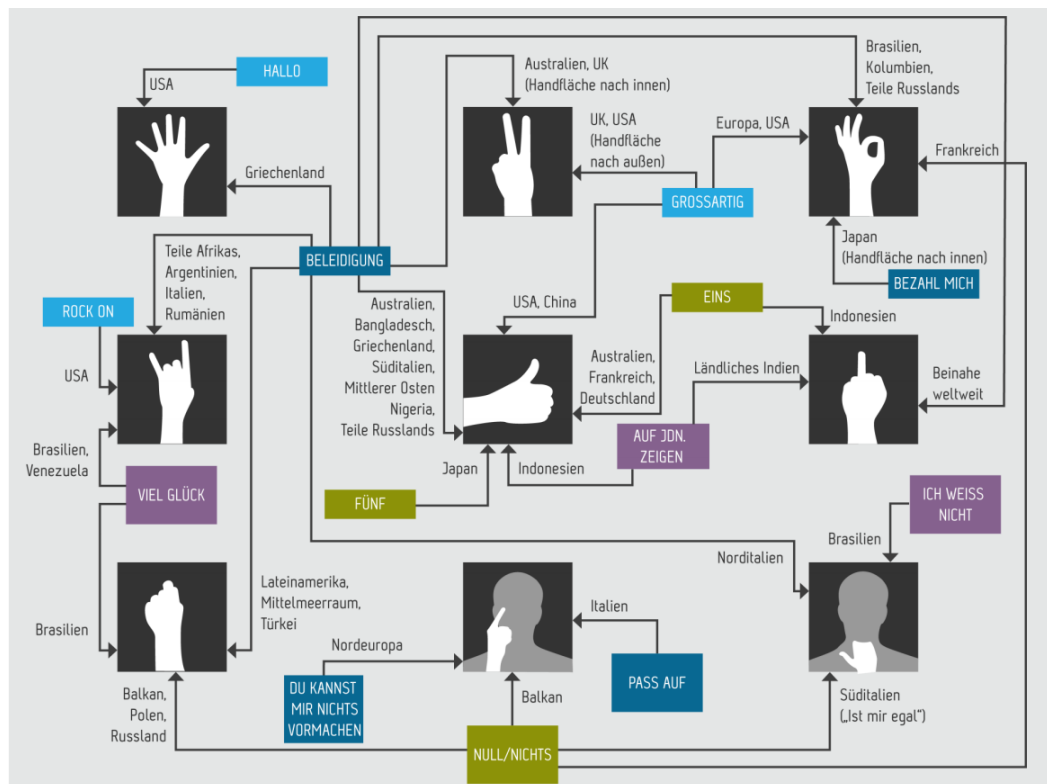


Abbildung 2.3: Beispiele für kulturelle Unterschiede von Gesten [Sylvia Le Hong (2010)]

Außerdem wird auf weitere Unterschiede bei der Verwendung von Gesten zwischen Deutschen und Chinesen im Abschnitt 2.5.4 eingegangen.

2.2.2 Midas Touch Problem

Eines der grundlegenden Problemen bei der Gestensteuerung (insbesondere bei 3D-Gesten) sowie bei Sprachsteuerungen ist das *Midas Touch Problem*⁵ oder Start-End-Problem. Dieses Problem, beschreibt die Schwierigkeit, unbewusste Bewegungen von denen mit einer Intention zu unterscheiden (siehe Abschnitt 2.2) [vgl. Ghose (2014)]. Während bei 2D-Gesten der Start der Geste meist durch das Berühren des Touchscreens und das Ende durch das Loslassen umgesetzt werden kann, ist dies bei 3D-Gesten nicht möglich [vgl. Potratz (2011)]. Dieses Problem wird

⁵Es gibt eine Reihe von mythischen Anekdoten welche von der Gier und Dummheit des Königs Midas, dem Herrscher von Phrygien um 645 v. Chr, handeln. In einer dieser Anekdoten wünscht sich König Midas, dass alles, was er berührt, zu Gold wird. Nach Erfüllung dieses Wunsches zeigte sich schnell, dass dies eher ein Fluch als ein Segen war, da auch geliebte Menschen oder seine Nahrung bei Berührung zu Gold wurde.

noch verstärkt durch die Erwartung, dass Gesten in einem großen Bereich wie einer Wohnung möglich sein sollen. Denn hier findet eine Vielzahl an Bewegungen statt, welche nicht als Gesten interpretiert werden sollen. Lösungsansätze wie bei einem Smart Mirror, bei dem es nur einen kleinen definierten Bereich gibt vor dem die Gestik interpretiert wird [vgl. Ghose (2014)] oder wie bei dem WiiMote Controller bei dem man einen Knopf drücken kann, um den Erkennungszeitraum für die Gestenerkennung zu aktivieren, scheiden daher aus. Genauere Details dazu liefert u.a. Boetzer (2008).

Im ideal Fall würde auf Basis des semantischen Zusammenhangs der Bewegung zum aktuellen Kontext des Nutzers und der Umgebung, eine Interpretation der eventuellen Intention des Nutzers ermöglicht. Daher sollte ein System zur Gestenerkennung das Zusammenspielen von verschiedenen Sensoren zur Kontakterfassung des Nutzers zulassen.

2.3 Gestenerkennung

In diesem Abschnitt sollen verschiedene, etablierte Systeme zur Gestenerkennung vorgestellt werden. Darüber hinaus wird auf erste Möglichkeiten der Kontexterfassung eingegangen.

2.3.1 Grundlegende Verfahren der Gestenerkennung

Die Grundvoraussetzungen für eine zuverlässige 3D-Gestenerkennung sind stabile Motion-Tracking-Verfahren. Diese lassen sich in markerbasierte Verfahren, wie beispielsweise gerätebasierte Verfahren, bei denen der Nutzer z.B. einen Controller nutzt [vgl. Potratz (2011)] oder markerlose Verfahren unterscheiden. Die markerlosen Verfahren lassen sich weiter in optische (kamerabasierte) und nicht optische, wie magnetische oder akustische Tracking-Verfahren differenzieren. Der Fokus dieser Arbeit liegt auf den optischen markerlosen Verfahren, da diese gegenüber den gerätebasierten Verfahren eine höhere Akzeptanz bei den Nutzern erreichen und die Usability des Systems erhöhen, da der Nutzer keine zusätzlichen Gegenstände tragen muss.

Grundsätzlich wird auf Basis dieser meist visuellen Daten, wie beispielsweise Punktwolken, die Position und der Veränderung der Bewegungen durch Algorithmen der Mustererkennung erkannt. Im Anschluss wird eine Gestenerkennung durch heuristisch basierte Verfahren oder Maschine Learning Verfahren umgesetzt. Eine genauere Betrachtung der verschiedenen Verfahren wird im Abschnitt 2.6 aufgeführt.

2.3.2 Sensorik

Die Auswahl der analysierten Sensoren erfolgt anhand folgender Kriterien: Geringe Latenz, hohe Genauigkeit, Zuverlässigkeit, Verfügbarkeit und gute Konnektivität z.B. durch umfangreiche SDKs sowie die Größe und Aktivität der Entwickler-Community.

Kinect v2 für Windows Die Kinect v2 für Windows ist ein vielseitiger Sensor, welcher unter anderem eine auf Time of Light basierende Tiefenkamera besitzt. Das SDK liefert bis zu sechs Skelette von Nutzern die sich in einer Entfernung von 0,5 bis zu 4,5 m in einem 120° Kamerawinkel vor der Kamera bewegen. Die 'sweet area' der Kinect v2 liegt zwischen 1 m und 4 m. Eigene Alphatests haben gezeigt, dass Personen außerhalb dieses Bereichs nur unzuverlässig erkannt werden. Jedes Skelette wird mit bis zu 25 Punkten (Joints), welche meist Gelenke des Körpers darstellen, erkannt (siehe Abb.: 2.4).

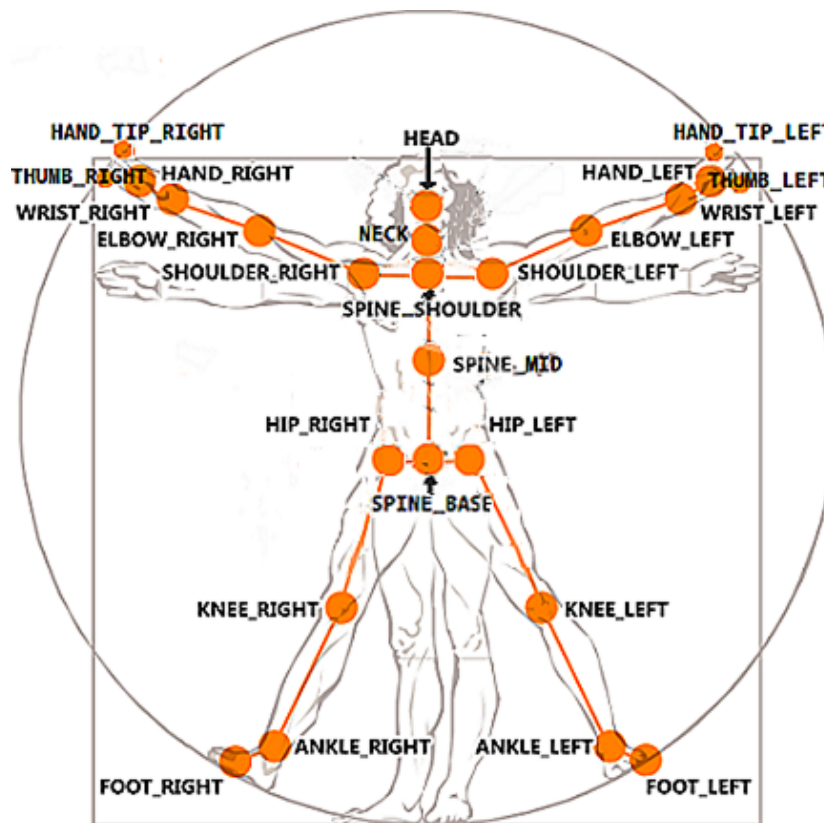


Abbildung 2.4: Körpermodell der Kinect v2 von Microsoft [Microsoft (2017a)]

Zudem werden einfache Gesten, wie zum Beispiel ob eine Hand auf oder geschlossen ist, erkannt. Die Latenz beträgt zwischen 50-80ms, und ist damit für die Verwendung im HCI-Bereich zur Gestensteuerung gerade noch ausreichend [vgl. [Sheridan und Ferrell \(1963\)](#)]. Weitere Streams wie ein Audio Stream mit Winkelangabe der Quelle des Signals, ein Video Stream und ein Face Tracking Stream machen die Kinect zu einem vielseitigen Sensor, welcher sinnvoll zur Kontexterkenkung des Nutzers verwendet werden kann [vgl. [Microsoft \(2016\)](#)].

Leap Motion Controller Die Leap Motion (siehe Abb.: [2.5a](#)) stellt ein genaues, zuverlässiges und detailliertes Modell der Hand (siehe Abb.: [2.5b](#)) zur Verfügung. Dabei liefert der Sensor bis zu 200 fps aller erkannten Hände und “Pointable Objects” (z.B. Stifte). Der Sensor lässt sich daher gut für Anwendungen, in denen der Nutzer feingranulare Bewegungen mit den Händen durchführt, nutzen [vgl. [Leap Motion \(2016\)](#)]. Die Latenz liegt lediglich bei ca. 30 ms [vgl. [Motion \(2013\)](#)]. Der Interaktionsbereich ist jedoch auf ca. 60 cm nach oben und zu den Seiten begrenzt.



(a) Leap Motion Controller



(b) Handmodell des Leap Motion Controllers

Abbildung 2.5: Leap Motion

Weitere Sensoren Einige gerätebasierte Produkte zur Gestenerkennung, wie das Myo Armband von Labs (2014) und die Meta 2 von META (2013) sowie die der Hololens von Microsoft (2017b), könnten zukünftig in Betracht gezogen werden. Vor allem bieten Sie gegenüber statischen Kamerasystemen den Vorteil, dass nicht die gesamte Wohnung mit einer Vielzahl von fest verbauten Sensoren erfasst werden muss, da diese auf einem Koordinatensystem basiert, bei dem das Gerät selber den Nullpunkt darstellt. Dafür erweist sich das Mapping mit der Umgebung als aufwendiger. Zudem könnten die Meta 2 oder die Hololens durch Augmented Reality Elemente dem Nutzer ein visuelles Feedback geben und somit die Steuerung der Haustechnik erleichtern.

Zu Beginn der Entwicklung waren einige der hier vorgestellten Sensoren noch nicht frei auf dem Markt verfügbar. Daher sollte das System zur Gestenerkennung erweiterbar sein, damit dieses leicht mit weiteren Sensoren ausgebaut werden können.

2.4 Kontexterkenkung zur Unterstützung von Gestenerkennung

Wie in Abschnitt 2.2.2 beschrieben, kann die Erfassung des Kontextes einer Geste eine wichtige Vorbedingung für eine intuitive, stabile Gestenerkennung darstellen. Im Folgenden soll auf verschiedene Bereiche und Techniken eingegangen werden, welche die Gestensteuerung unterstützen können.

2.4.1 Anwendungskontext

Besonders bei der Lösung des *Midas Touch Problem* spielt der Anwendungskontext eine zentrale Rolle. In vielen Anwendungen wird aktuell bereits der softwareseitige Anwendungskontext einer Geste zur Bedienung einbezogen, beispielsweise durch Hervorheben der möglichen Gesten im aktuellen Anwendungskontext [vgl. Dorau (2011)]. Diese Strategie lässt sich besonders bei einem kleinen Repertoire an Gesten im jeweiligen Anwendungskontext einsetzen.

In einer offenen Umgebung wie der eines Smart Homes kann jedoch das Problem bestehen, dass dem Nutzer nicht bewusst ist, welche Gesten ihm aktuell zur Bedienung zur Verfügung stehen, da es bei der Bedienung von Haustechnik nicht unbedingt ein visuelles Feedback wie bei Bildschirmen oder Touchscreens gibt, bei denen einige Optionen inaktiv (ausgegraut/ausgeblendet) sind.

2.4.2 Kontextsensitivität (Context-aware Computing)

Neben dem Anwendungskontext spielt der reale Kontext der Person und der Geste ebenfalls eine wichtige Rolle zur Erkennung und Interpretation von Gesten. Dazu gehört der Zustand der Umgebung, der Person sowie die aktuelle Tätigkeit der Person. Durch Einbeziehung dieser Faktoren lassen sich robustere Rückschlüsse auf die Intention einer Person schließen. Dies ist nötig, da nach [Pavlovic u. a. \(1997\)](#) die Intention den zentralen Unterschied zwischen einer einfachen Bewegung und einer Geste definiert (siehe Abschnitt 2.2). Das Ziel der Einbindung von Context-Aware-Computing ist daher, herkömmliche Gestenerkennungssysteme durch Systeme zu ergänzen, welche Mimik, Sprache sowie die Blickrichtung des Nutzers erfassen, um eine bessere Schlussfolgerung über die Intention des Nutzers zu ziehen.

2.4.3 Emotion-Aware-Computing

Zu dem Kontext einer Bewegung, und daher einer potentiellen Geste, gehören auch die Emotionen und die Mimik des Nutzers. So lassen sich durch Affective Computing die, auch für HCI-Anwendungen wichtigen Emotionen, wie Verwirrtheit, Frustration, Ärger sowie Freude durch kamerabasierte Emotionserkennung feststellen [vgl. [Ekman \(2005\)](#)].

Aus diesem Grund wird die Pose- und Gestenerkennung auch im Forschungsfeld um Companion-Systeme untersucht. Zudem kann häufig nur über den Kontext erkannt werden, welche Emotion konkret vorliegt. Ein Beispiel hierfür liefert Abbildung 2.6.



Abbildung 2.6: Ekel und Ärger lassen sich oft nur durch den Kontext der Gesichtsmimiken voneinander unterscheiden [Aviezer u. a. \(2008\)](#).

Emotionen bezeichnen eine Gemütsbewegung, welche durch eine Situation oder ein Ereignis ausgelöst werden kann. In der Emotionspsychologie bzw. Emotionstheorie gibt es eine Vielzahl

an Kategorisierungen für Basisemotionen und Emotionsmodelle. Im Folgenden wird auf die Emotionstheorie von Paul Ekman eingegangen, da diese eine der verbreitetsten Theorien mit diskretem Emotionsmodell ist und sich daher gut zur Verwendung einer Emotionserkennung eignet. Nach Ekman gibt es sieben Basisemotionen: Freude, Wut, Traurigkeit, Überraschung, Furcht, Ekel und Verachtung [vgl. Ekman (1988)]. Des Weiteren muss zwischen Stimmungen, Gefühlen und Emotionen unterschieden werden.

Emotionserkennung Die Möglichkeit der Emotionserfassung durch Biosensoren wie z.B. ein EEG wird aufgrund der schlechten Usability für die Anwendung im Smart Home ausgeschlossen. Wobei diese gegebenenfalls zukünftig durch die Integration von Sensoren zur Messung von beispielsweise Elektrodermaler Aktivität (EDA) in Smart Watches oder Fitness Trackern eine höhere Bedeutung gewinnen können.

Grundsätzlich lassen sich daher im HCI-Bereich Emotionen durch die Gesichtsmimik oder durch die Stimme erkennen. Da zur Steuerung von Systemen durch Gesten meist lautspracheretzende Gesten genutzt werden (siehe Abschnitt 2.2) und emotionale Reaktionen auf Aktionen des Computers typischerweise nicht ausgesprochen werden, wird lediglich die Emotionserkennung durch kamerabasierte Verfahren betrachtet. Bei einem System welches über Sprachbefehle gesteuert wird, wie beispielsweise Alexa von Amazon, würde eine Auswertung der Emotionen durch die Stimmen sinnvoll sein.

Ein Beispiel für erste Untersuchungen zur emotionsbasierten Interaktion liefert unter anderem das Emotionbike Projekt ⁶ an der HAW Hamburg.

Da, wie beschrieben, Emotionen mit in die Betrachtung zur Erkennung von Gesten einbezogen werden sollten und ebenfalls aus der Forschung um Companion Systeme erprobte Konzepte für Multimodale Systeme entwickelt worden sind, wird auf Companion Systeme im Abschnitt 2.7 weitgehend eingegangen.

Software Die wichtigsten Softwarelösungen zur Echtzeit Emotionserkennungen, welche für wissenschaftliche Untersuchungen frei verfügbar sind, sind Affectiva von Affectiva (2017) und die SHORE Libery vom Fraunhofer Institute for Integrated Circuits (2011). Die Shore Libery bietet zwar umfangreiche Features, jedoch zur Emotionserkennung lediglich die Emotionen: Freude, Überraschung, Wut und Traurigkeit.

Affectiva nutzt das von Paul Ekman entwickelte Facial Action Coding System (FACS). FACS ist ein Verfahren zur Beschreibung von Gesichtsausdrücken [Ekman und Friesen (1978)]. Durch das FACS lassen sich verschiedene Gesichtsausdrücke zu Bewegungseinheiten (Action Units)

⁶<https://www.emotionbike.org>

zuweisen. Durch Verfahren des Maschinens lernen konnte das System eine schnelle Erkennung von Emotionen erreichen (weitere Informationen dazu finden sich in [Bernin u. a. \(2017\)](#)). Affectiva kann u.a. folgende Informationen aus einem Gesicht auslesen: grundsätzlich positive, negative sowie neutrale Gesichtsausdrücke, die sieben Basisemotionen (siehe Abschnitt 2.4.3) sowie Frustration und Verwirrtheit. Zudem werden noch Orientierungspunkte (Facial Landmarks) zu den Augen, der Nase, dem Mund sowie der Winkel und die Position des Kopfes erfasst.

Einen ausführlichen Vergleich der verschiedenen Systeme zur Emotionerkennung findet man in [Bernin u. a. \(2017\)](#).

Anwendungsmöglichkeiten Leider lassen sich aktuell weder aus der Shore Libery noch aus Affectiva direkt Informationen dazu gewinnen, ob eine Person konzentriert ist. Dies könnte ein Indiz dafür sein, dass die Bewegung des Nutzers eine Steuerungsgeste sein soll. Da Affectiva bereits Facial Landmarks und die Kopfposition erfasst, könnten daraus diese Informationen abgeleitet werden [vgl. [Lee u. a. \(2013\)](#)]. Zudem muss untersucht werden, inwieweit Emotionen wie z.B. Frustration und Verwirrtheit, Rückschlüsse auf die Korrektheit der Ergebnisse einer Gestenerkennung liefern können.

Ebenfalls sollte untersucht werden, ob die Stimmung oder Emotion einer Person Einfluss auf die Art der Ausführung von Gesten bzw. die Steuerung von Smart Environments hat. Daher sollte das Design eines solchen Systems zur Gestenerkennung darauf ausgelegt sein, die erfassten Daten, wie beispielsweise ein Video Stream, mehreren System wie einer Gestenerkennung und einer Emotionserkennung gleichzeitig zur Verfügung zu stellen.

2.4.4 Weitere Techniken zur Kontexterkenkung

Gaze-aware Eine weitere Möglichkeit zur Kontexterkenkung bietet Gaze Awareness oder auch Eye-Tracking. Dabei wird der Blickwinkel einer Person festgestellt. Diese Information wurde unter anderem bei dem HoloDesk [vgl. [Hilliges u. a. \(2012\)](#)] genutzt, um zu erfassen, auf welches Steuerelement in einem 3D-Interface sich die aktuell ausgeführte Geste bezieht. Auch weniger genaue Systeme wie eine zusätzliche Erkennung der Kopfausrichtung bei der Skeletterkennung wie bei der Kinect v2 können zur Kontexterkenkung beitragen.

Location-aware Ortsabhängige (Location awareness) Ansätze wurden z.B. beim ASGARD Projekt [[Maurizio u. a. \(2011\)](#)] untersucht, bei dem über 3D-Gesten Haustechnik wie Lampen und Ventilatoren bedient werden können. In der Studie [Anastasiou u. a. \(2013\)](#) konnte zudem gezeigt werden, dass deutsche Probanden im Gegensatz zu chinesischen Probanden von Intelli-

genten Systemen erwarten, dass diese ihre Gestik abhängig vom Ort und ihrer Blickrichtung interpretieren können.

Sprachkontext Sprachbegleitende Gesten sind, wie bereits in Abschnitt 2.2 erwähnt, meist ikonische Gesten. Die Interpretationen von diesen können häufig nur im Kontext des Gesprochenen erfolgen [vgl. McNeill (1992)]. Dies gilt auch andersherum. So kann ein erläuternder Satz mit dem Wort “da” häufig nur durch das Unterstützen einer Geste interpretiert werden.

Weitere Kontexte Weitere Daten, deren Nutzen für die Interpretation einer Geste betrachtet werden können, sind Terminkalender, Uhrzeit, Wetter, vorhergehende Tätigkeiten, das Umfeld sowie weitere.

2.4.5 Kontext nach Dourish

Ein anderes Verständnis des Kontextes, über die Gestik hinaus, in HCI Systemen wird in “What We Talk About when We Talk About Context” von Paul Dourish (2004) Dourish (2004) vorgeschlagen.

By turning our attention from “context” (as a set of descriptive features of settings) to “practice” (forms of engagement with those settings), we assigned a central role to the meanings that people find in the world and the meanings of their actions there in terms of the consequences and interpretations of those actions for themselves and for others.

...

Dourish (2004)

Dourish zeigt die enorme Wichtigkeit des Kontextes auf. Zudem weist er darauf hin, dass eine Betrachtung des Kontextes rein auf Basis einer konkreten Applikation, weder dem Kontextbegriff an sich noch dem Nutzer gerecht werden. Ein Informationssystem sollte es den Nutzern ermöglichen, im Laufe der Interaktion mit diesem zu “verhandeln” und es dadurch weiterzuentwickeln. Daher sollte ein System zur Gestenerkennung leicht veränderbar, erweiterbar und im Idealfall personalisierbar sein, da der Nutzer selbst am besten die Intention seines eigenen Handelns versteht.

2.4.6 Schlussfolgerungen für die kontextabhängige Gestensteuerung

Dieser Abschnitt (2.4) verdeutlicht, dass eine Gestenerkennung rein basierend auf Motion-Tracking-Verfahren nicht ausreichend ist, um Rückschlüsse auf die mögliche Intention einer

Person in einem offenen Kontext wie einem Smart Home zu ziehen. Das bedeutet, dass eine semantische Interpretation bezüglich des Kontextes der Gestik und der Person unabdingbar ist, um diese Unterscheidung zu treffen. Daraus ergibt sich eine zentrale Anforderung für Systeme zur Gestenerkennung. Diese müssen leicht um weitere Sensoren und Aktoren erweitert werden können, um eine natürliche, nahtlose und intuitive Interaktion mit der Umgebung zu ermöglichen. Dies wird deutlich bei der gestenbasierten Steuerung von Smart Homes, da hier zusätzliche Informationen wie die Blickrichtung des Nutzers (Gaze-Awareness) oder der Raum, in dem sich der Nutzer befindet (Location-Awareness), zur zutreffenden Interpretation der Gestik benötigt werden können. So lässt sich beispielsweise eine ironisch gemeinte Geste lediglich an der Betonung der Sprache oder der Mimik erkennen [vgl. McNeill (1992)].

2.5 Vergleichbare Arbeiten Gestenerkennung

Im Folgenden werden aktuelle Themen und relevante Abhandlungen zur Gestenerkennung vorgestellt sowie die konkreten Konzepte auf Mächtigkeit, Usability sowie Lösung des *Midas Touch Problem* und ihrer Tauglichkeit für Smart Environments analysiert.

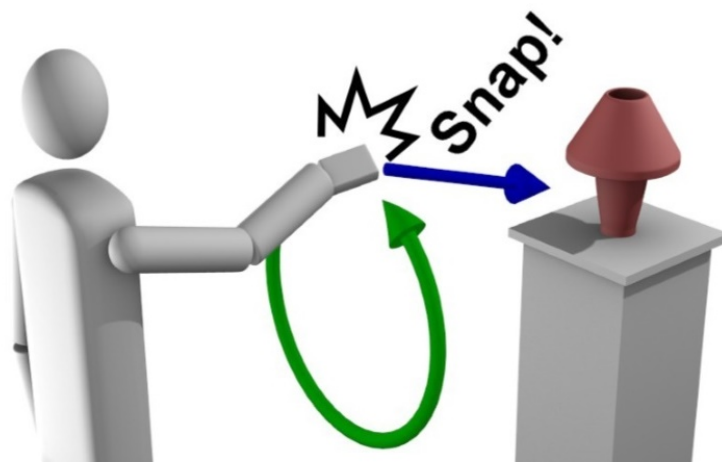
2.5.1 MISO: A Context-sensitive Multimodal Interface for Smart Objects Based on Hand Gestures and Finger Snaps

Ein Ansatz zur Lösung des *Midas Touch Problem* wurde an der Universität Bielefeld entwickelt [vgl. Fleer und Leichsenring (2012)]. Konkret werden Geräte wie eine Lampe oder eine Musikanlage durch Gesten angesteuert. Der Nutzer selektiert durch eine Zeigegeste ein Objekt und bestätigt die Selektion durch ein Fingerschnippen. Dieses Fingerschnippen leitet gleichzeitig die Geste zur Manipulation des Zustands des Objekts ein. Beispiele dafür sind:

1. Hand bewegt sich nach oben/unten steht für An/Aus
2. Hand bewegt sich nach rechts/links steht für Weiter/Zurück
3. Hand bewegt sich im Kreis nach rechts/links zur Veränderung kontinuierlicher Parameter wie beispielsweise Lauter/Leiser (siehe Abb.: 2.7)



(a) Proband wählt die Musikanlage durch eine Zeigegeste aus



(b) Nach der Zeigegeste signalisiert der Proband durch ein Fingerschnippen den Start einer Manipulation und dreht die Lautstärke auf, indem er die Hand in Richtung gegen den Uhrzeigersinn dreht.

Abbildung 2.7: MISO - Finger Snap **Fleer und Leichsenring (2012)**

Rückschlüsse und Bewertung des Konzepts Da hier bekannte und einfache Gesten genutzt werden, ist die Steuerung ohne großen Aufwand erlernbar. Jedoch können aktuell nur eine geringe Anzahl verschiedener Manipulationen für ein Gerät umgesetzt werden. Durch das Nutzen der intuitiven Zeigegeste können schnell Objekte in der näheren Umgebung selektiert werden. Das Auswählen von weiter entfernten Objekten wird jedoch durch die Ungenauigkeit der Zeigerichtung und weiteren anwählbaren Objekten in Zeigerichtung schwieriger. Zudem können grundsätzlich nur von der aktuellen Position sichtbare Objekte ausgewählt werden. Des Weiteren muss bedacht werden, dass nicht alle Menschen mit den Fingern Schnipsen

können.

Durch das Schnipsen mit den Fingern wird der Start einer Geste bzw. das Bestätigen einer Selektion signalisiert. Durch dieses Merkmal wird die Rate der False/Positive Erkennungen⁷ bei einer kontinuierlichen Verwendung der Gestenerkennung gering gehalten. Durch die Kombination mit Sprachbefehlen oder anderen Arten der Selektion, könnten die oben beschriebenen Nachteile ausgeglichen werden. Daher sollte bei dem Entwurf des Systems zur Gestenerkennung darauf geachtet werden, dass die Kombinationen verschiedener Modalitäten einfach integrierbar sind.

2.5.2 Imaginary Interfaces: Spatial Interaction with Empty Hands and Without Visual Feedback

Am Hasso-Plattner-Institut wurde in dem *Human Computer Interaction Lab* das Konzept der Imaginary Interfaces [vgl. [Gustafson u. a. \(2010\)](#)] entwickelt und erprobt. Diese ermöglichen dem Nutzer die Steuerung eines Computersystems durch Gesten auf einem lediglich vorgestellten visuellen Interface.

Wie in [Abbildung 2.8](#) dargestellt, verwendet der Nutzer die linke Hand um den Nullpunkt eines Koordinatensystems zu definieren. Der Abstand zwischen den Werten auf den Achsen wird durch die Länge des Daumens für die x-Achse sowie die Länge des Zeigefingers für die y-Achse vorgegeben (Siehe [Abbildung: 2.8](#)). Basierend auf diesem Koordinatensystem kann der Nutzer mit der rechten Hand Koordinaten auswählen. Dies tut er durch die Berührung des Zeigefingers mit dem Daumen (Start der Geste). Durch das Öffnen der Finger wird das Ende der Geste bzw. das Auswählen eines Punktes bestätigt.

⁷False/Positive sind Fälle, in denen ein System eine Geste erkennt, obwohl keine durchgeführt worden ist.

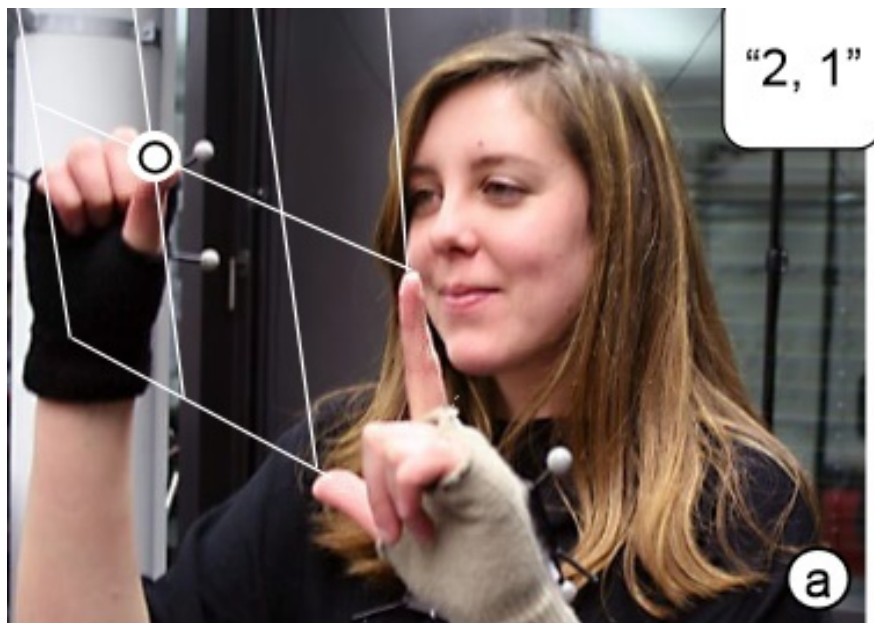


Abbildung 2.8: Probandin wählt den Punkt 2,1 aus [Gustafson u. a. (2010)].

Erste Untersuchungen des Hasso-Plattner-Instituts konnten zeigen, dass die Verwendung des imaginären Koordinatensystems maßgeblichen Einfluss auf das visuelle Gedächtnis einer Person hat. Beispielsweise konnte nachgewiesen werden, dass Nutzer gemalte Linien und Ecken wesentlich genauer bei einer zweiten Verwendung mit dem imaginären Interface wiederfinden konnten (siehe Abb.: 2.9b).

Rückschlüsse Ähnliche Umsetzungen finden sich inzwischen auch für VR und AR Inhalte, beispielsweise mit der Verbindung eines Leap Motion Sensors⁸. Diese bieten den Vorteil, dass der Nutzer sich das Interface nicht nur vorstellen muss, sondern dieses auch sehen kann. Ideal könnte eine Verbindung mit einer AR Brille wie der Hololens sein, mit der ein großer Funktionsumfang sowie ein gute Usability gegeben sind.

Die Nutzung von metaphorischen Gesten bietet grundsätzlich eine große Vielzahl an Möglichkeiten der Bedienung. Besonders bei einer großen Anzahl von Aktoren muss der Nutzer diese erst aufwendig erlernen. Eine Erleichterung beim Lernen der metaphorischen Gesten könnte durch eine Individualisierung der Gesten erreicht werden. Die Lösung des *Midas Touch Problem* wird hier elegant durch eine gesonderte Start- bzw. End-Geste gelöst. Jedoch benötigt

⁸<https://developer.leapmotion.com/orion>

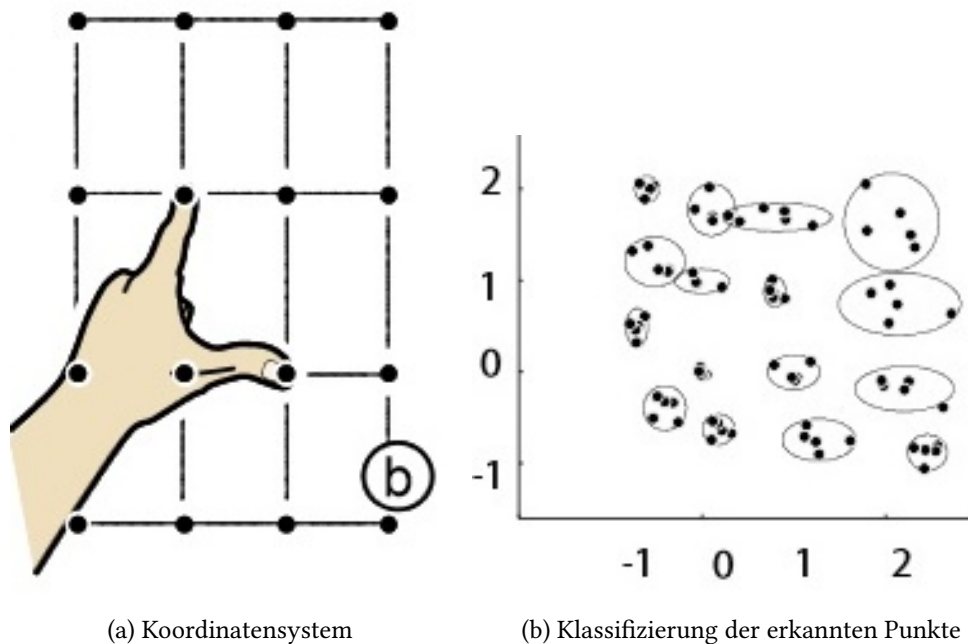


Abbildung 2.9: Imaginary Interfaces [Gustafson u. a. (2010)]

das System eine hohe Genauigkeit, um die filigranen Unterschiede innerhalb des Koordinatensystems sowie das Öffnen/Schließen der Finger richtig zu interpretieren. Dies wird in diesem Projekt durch eine an der Brust montierte Kamera umgesetzt. Das hat einerseits Vorteile, da das Interface auch mobil genutzt werden kann. Andererseits ist die feste Positionierung einer Kamera an der Brust eine hohe Einschränkung für die Usability des Systems.

Zudem wäre innerhalb eines Smart Environments eine direkte Interaktion mit einem Objekt erstrebenswert, da dies unserer natürlichen Interaktion mit anderen Menschen oder Lebewesen entspricht. Systeme zur Gestenerkennung für Smart Enviroments sollten grundsätzlich die Umsetzung von Ansätzen wie Imaginary Interfaces ermöglichen. Durch den konkreten Anwendungsfall in einem Smart Home wird der Fokus dieser Arbeit jedoch auf kamerabasierten Systemen liegen und nicht auf Wearables.

2.5.3 Exploring the Usefulness of Finger-Based 3D Gesture Menu Selection

Auf der ACM CHI 2014 wurde von [Kulshreshth und LaViola \(2014\)](#) eine weitere Möglichkeit der Umsetzung von metaphorische Gesten veröffentlicht. Der Schwerpunkt der Studie wurde auf

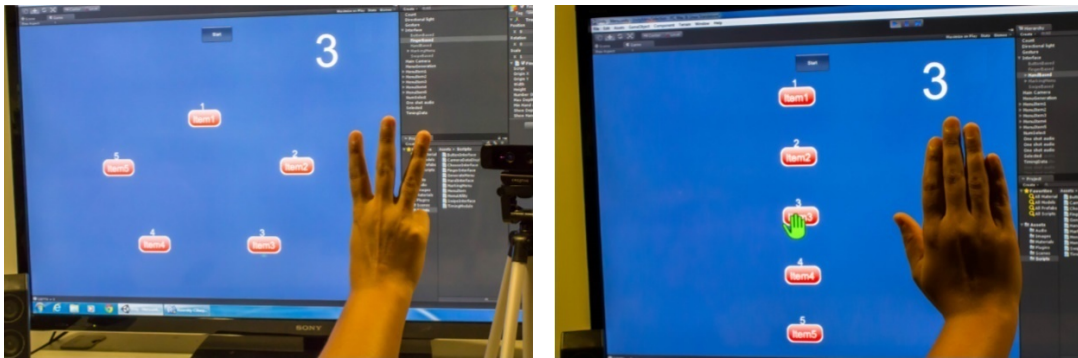
die Entwicklung und den Vergleich von metaphorischen Gesten zur Auswahl von Menüpunkten mittels der Finger gelegt. Dabei wurden vier verschiedene Methoden konzipiert und verglichen.

- **Finger-Count:** Der Proband hält die Anzahl an Fingern hoch, welche der Nummer des Menüpunktes entspricht (siehe Abb.: 2.10a).
- **Hand-n-Hold:** Der Proband wählte durch vertikale Bewegungen einen Menüpunkt aus. Durch das Halten auf einer Höhe wurde der markierte Menüpunkt ausgewählt (siehe Abb.: 2.10b).
- **Thumbs-Up:** Durch horizontale Bewegung der Faust markierte der Proband die Menüpunkte. Durch die Geste "Daumen hoch" wurde der aktuell markierte Menüpunkt ausgewählt (siehe Abb.: 2.10c).
- **3D-Marking:** Durch eine Bewegung der Hand in Richtung eines der sternförmig angeordneten Menüpunkte wurde der entsprechende Menüpunkt ausgewählt.

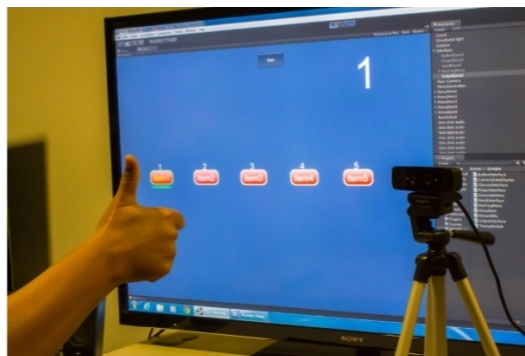
Das *Midas Touch Problem* wird bei diesem Ansatz durch einen Start- und End-Zustand der Hand umgesetzt. Start der Geste wird durch das Bilden einer Faust signalisiert.

Die Studie hat gezeigt dass die stabilste und effektivste Methode die Finger-Count Methode ist. Bei dieser werden Selektion und Manipulation durch nur eine einfache schnelle Geste umgesetzt.

Rückschlüsse Gegenüber des zuvor erläuterten Ansatzes der Imaginary Interfaces (siehe Abschnitt 2.5.2) ist diese Art der Gestik leicht zu erlernen, unkompliziert durchzuführen und kann mit lediglich einer Hand durchgeführt werden. Jedoch kann durch die Einschränkungen auf fünf Menüpunkte eine weite Verschachtelung von verschiedenen Menüs nötig sein. Dies führt bei großen Systemen unweigerlich zu komplexen und unübersichtlichen Strukturen, welche ohne gutes visuelles Interface kaum zu bedienen sind. Daher ist die Anwendung im Kontext Smart Environments ohne visuelles Interface nur sehr schwer umzusetzen.



- (a) Finger-Count: Proband wählt durch die Anzahl der Finger den entsprechenden Menüpunkt aus.
- (b) Hand-n-Hold: Proband wählt durch Bewegungen der Hand in der Vertikalen einen Menüpunkt aus und bestätigt diesen durch Halten.



- (c) Thumbs-Up: Proband wählt durch Bewegungen der Hand in der Horizontalen einen Menüpunkt aus und bestätigt diesen durch das Heben des Daumens.

Abbildung 2.10: Drei verschiedene Arten der Auswahl eines Menüpunkts durch fingerbasierte Gesten [Kulshreshth und LaViola (2014)]

2.5.4 Bremen Ambient Assisted Living Lab

An der Bremer Universität wurden verschiedene Studien im Bereich der multimedialen Interaktion durchgeführt. Im Folgenden soll auf zwei Wizard-of-Oz-Experimente⁹, welche im Bremer Ambient Assisted Living Lab (BAAL) (siehe Abb.: 2.11) durchgeführt wurden eingegangen werden.

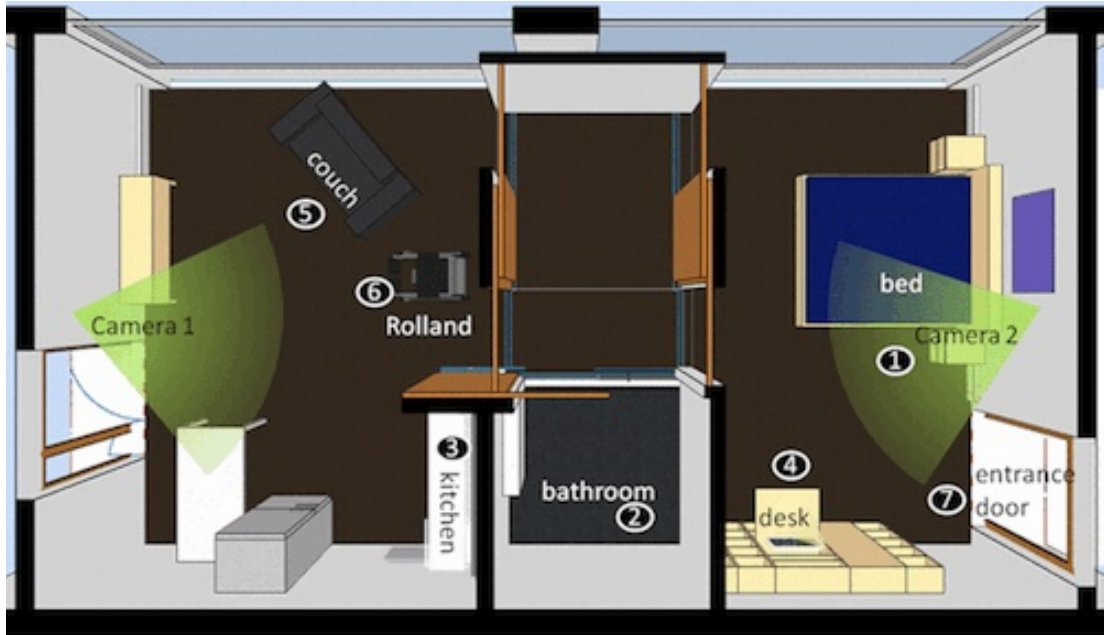


Abbildung 2.11: Bremen Ambient Assisted Living Lab [[Anastasiou u. a. \(2012\)](#)]

BAAL Studie 1: A German-Chinese Speech-gesture Behavioural Corpus of Device Control in a Smart Home

In der Studie [Anastasiou u. a. \(2013\)](#) wurden Elektronikgeräte sowie Haustechnik durch Gesten und/oder Sprache angesteuert. Durch die Verwendung multimodaler Interaktion mit einem Smart Environment ist die thematische Nähe zu dieser Arbeit ungemein hoch. In der Studie wurden zusätzlich kulturelle Aspekte durch Vergleich von deutschen (20) und chinesischen (6) Probanden untersucht. Die Probanden mussten vier verschiedene Aufgaben durchführen:

1. Eine automatische Schiebetür schließen und öffnen (diskreter Parameter: Auf/Zu).

⁹Wizard-of-Oz-Experimente ist ein HCI-Experiment, bei dem der Proband annimmt mit einem autonomen/intelligenten System zu interagieren, jedoch wird das System verdeckt von einem Menschen gesteuert.

2. Eine Deckenlampe an- und ausschalten (diskreter Parameter: An/Aus).
3. Kopf- und Fußteil eines Bettes hoch und runter fahren (kontinuierlicher Parameter: Höher/Tiefer).
4. Eine Bettlampe an- und ausschalten (diskreter Parameter: An/Aus).

Dabei durchlief jeder Proband drei verschiedene Tests:

1. Nur Gesten durften zur Steuerung verwendet werden.
2. Nur Sprache durften zur Steuerung verwendet werden.
3. Der Proband durfte frei zwischen Gesten und Sprache wählen und diese kombinieren.

Durchlauf 1: Gestensteuerung Das Öffnen/Schließen von Türen haben 20 von 26 Probanden durch die Wischgeste umgesetzt. Fünf der 20 deutschen Probanden nutzten die Zeigegeste, um das Licht anzusteuern. Von den chinesischen Probanden nutzte keiner die Zeigegeste. Während die chinesischen Probanden die meisten Gesten schnell und abrupt durchgeführt haben, sind die deutschen Probanden durch den Raum gegangen, um näher bei den Objekten zu sein welche sie steuern wollten.

Die Probanden haben grundsätzlich ein schnelles Feedback von dem System erwartet. Dies zeigte sich dadurch, dass die Probanden verunsichert waren, ob das System die Geste erkannt hat, wenn das System nicht umgehend reagiert hat. Dies führte dazu, dass Probanden eine zweite Geste ausführten und sich im Anschluss nicht sicher waren, welche der beiden Gesten korrekt war. Größtenteils waren die Probanden erstaunt, dass im Rahmen der Studie die Wohnung alle Gesten richtig erkennen konnte.

Einige Nutzer waren bei der Ausführung der Gesten verunsichert, da sie nicht wussten wo sich der Sensor zur Erkennung der Gesten befindet.

Durchlauf 2: Sprachsteuerung Ein bemerkenswerter Unterschied bei der Spracherkennung war, dass 113 von 244 Sprachbefehle der deutschen Probanden kontextsensitiv waren, während die chinesischen Probanden keine kontextsensitiven Sprachbefehlen verwendeten. Stattdessen gaben die chinesischen Probanden in dem Sprachbefehl direkt an, welches Gerät sie steuern wollten.

Durchlauf 3: Multimodale Steuerung 20 von 26 Probanden gaben an, dass sie die Wahl zwischen Gesten- oder Sprachsteuerung von der konkreten Aufgabe abhängig machen würden. Dabei unterschied sich jedoch stark, je nach Person die Wahl, ob Gesten- oder Sprachsteuerung verwendet werden sollte. Daher lassen sich keine Aufgaben identifizieren, für die eine der beiden Steuerungsarten präferiert wurde.

Einige Probanden empfanden die Sprachsteuerung für die kontinuierlichen Parameter als zu ungenau, daher würden sie Gesten für diese Aufgaben bevorzugen. Außerdem nutzten mehrere Probanden die Zeigegesten, um Objekte für die Manipulation durch Sprachbefehle zu selektieren.

Kulturelle Unterschiede Zwei Thesen wurden im Rahmen der Abhandlung gebildet:

- Es gibt bei der Ansteuerung von Haustechnik und Geräten keine großen Unterschiede der Gestenwahl zwischen Chinesen und Deutschen
- Chinesen nutzen gegenüber den Deutschen keine kontextsensitiven Sprachbefehle

Jedoch wird darauf hingewiesen, dass umfangreichere Studien nötig sind um diese Thesen zu verifizieren.

Rückschlüsse für Systeme zur Gestenerkennung Diese Studie liefert eine Reihe von wichtigen Informationen, aus denen sich Anforderungen für Systeme zur Gestenerkennung in Smart Homes ergeben.

Unverzögliches Feedback Einige Nutzer gaben an, dass sie verunsichert waren, wenn das System nicht direkt reagiert hat. Dies führte häufig dazu, dass Sie weitere Gesten probiert haben. Das bedeutet, dass ein System zur Gesten- oder Sprachinterpretation den Nutzern ein direktes Feedback geben muss. Dies gilt besonders dann, wenn das System Verzögerungen durch die Ansteuerung der Haustechnik hat, welche unabhängig von der Reaktionszeit der Sensorik und der Gestenerkennung sind oder eine Veränderung keinen direkt sichtbaren Effekt hat, wie beispielsweise das Ändern der Raumtemperatur. Außerdem sollte das System ein Feedback geben, wenn ein Befehl nicht vollständig verstanden werden konnte.

Selektion Wie bereits im Abschnitt 2.2.1 beschrieben, müssen kulturelle Unterschiede bei der Wahl der Gestik beachtet werden. Dies zeigte sich in dieser Studie besonders bei der Selektion eines Objekts. In China gilt es als sehr unhöflich, direkt auf Personen mit einem Finger zu zeigen. Stattdessen wird eine offene Hand verwendet [vgl. Knigge (o.D)]. Daher lässt

sich die Hypothese bilden, dass aus diesem Grund die chinesischen Probanden in der Studie wahrscheinlich nicht die Zeigegeste zur Selektion genutzt haben. Während die deutschen Probanden von der Möglichkeit einer kontextsensitiven Selektion ausgegangen sind, haben die chinesischen Probanden genau ausgewählt, welches Objekt sie ansteuern wollen. Daher sollten Systeme zur Steuerung in Smart Environments verschiedene Arten der Selektion und Ansteuerung anbieten, sodass der Nutzer nach seinem Vorlieben und seinen kulturellen Hintergrund wählen kann.

Ferner müssen Blickrichtung (Gaze Tracking), Körperausrichtung und die Position in der Wohnung zur Interpretation des Befehls mit einbezogen werden, um eine kontextsensitive Auswahl zu ermöglichen. Dies bedeutet, dass das System multimodal ausgelegt sein muss. Zudem muss die Integration von weiteren Sensoren schnell und einfach ermöglicht werden.

BAAL Studie 2: Speech and Gesture Interaction in an Ambient Assisted Living Lab

Bei dem zweiten Wizard-of-Oz-Experiment [Anastasiou u. a. \(2012\)](#) wurden Untersuchungen zur Steuerung eines intelligenten motorisierten Rollstuhls durch Sprache und Gestik durchgeführt. Die Probanden sollten mit dem Rollstuhl "Rolland" (Abbildung 2.12) verschiedene Alltagsaufgaben bewältigen. Dabei wurden keine konkreten Bedienungskonzepte der Sprach- oder Gestensteuerung untersucht, jedoch konnten verschiedene relevante Zusammenhänge zwischen den beiden gezeigt werden.

Die Studie zeigte u.a., dass für eine Aufgabe viele verschiedene Sprachbefehle verwendet worden sind. Dabei erwarteten einige Nutzer bereits beim Ansprechen des Rollstuhls ein Feedback. Zudem wurden kontextsensitive Befehle wie "hierher" und Landmarken wie z.B. Bett, Spüle etc. häufig für die Auswahl eines Ortes genutzt. Zudem haben männliche Probanden häufiger eine direkte Befehlsform gewählt als Frauen. In 6 von 20 Versuchen nutzten Probanden Zeigegesten neben der Sprache, um eine Richtung, ein Ziel oder ein Objekt zu selektieren.

Grundsätzlich nutzten die Probanden vorrangig Sprachbefehle, um den Rollstuhl anzusteuern. Erwartungsgemäß war die meist genutzte Geste die Zeigegeste. Es wurde lediglich eine ikonische Geste von einem einzigen Probanden genutzt (Hände reiben als Geste zum Händewaschen). Dies kann evtl. damit zusammenhängen, dass die Nutzer davon ausgegangen sind, dass ein Computersystem ihre Gesten verstehen muss und sie deshalb ausschließlich die intuitiven Zeigegesten nutzten. Ein Großteil der Gesten wurde eingesetzt, wenn der Rollstuhl sich nicht den Vorstellungen des Nutzers entsprechend verhielt.

Rückschlüsse für Systeme zur Gestenerkennung Diese Studie zeigt neben einigen Hinweisen, welche schon in [Anastasiou u. a. \(2013\)](#) aufgezeigt wurden, noch einen weiteren



Abbildung 2.12: Rolland der autonome Rollstuhl des BAAL [[Anastasiou u. a. \(2012\)](#)]

wichtigen Punkt: Es ist unklar, wie der Nutzer mit dem System im Falle einer False/Positive Erkennung umgeht bzw. wie ein System dies erkennen kann. Eine Möglichkeit wäre, dass intelligente Systeme ebenfalls “Abbruch/Stop” bzw. “Befehl zurück” -Befehle/Gesten erkennen. Da in diesem, wie auch im vorigen Experiment, ein Wizard-of-Oz-Experiment durchgeführt wurde und daher Menschen die Gesten interpretiert haben, können aus diesem Experimenten keine Rückschlüsse bzgl. des *Midas Touch Problem* gezogen werden.

2.5.5 Fazit der vergleichbaren Arbeiten

Die vorangegangenen, vorgestellten Arbeiten zeigen nur einen kleinen Ausschnitt von einer Vielzahl an unterschiedlichen Möglichkeiten der Steuerung von Smart Environments durch Gesten auf. Die Auswahl zeigt Lösungsansätze für das *Midas Touch Problem*, welche für die

konkrete Entwicklung von Gesten hilfreich sein könnten. Jedoch wird angesichts der unterschiedlichen Anwendungsfälle auch ein grundsätzliches Problem bei der Gegenüberstellung verschiedener Konzepte zur Gestenerkennung deutlich. Durch die Verwendung unterschiedlicher Algorithmen, Sensoren, Studiendesigns und Umgebungen ist ein konkreter Vergleich der verschiedenen Konzepte nicht möglich. Selbst wenn Studien Messungen, wie den Anteil an erfolgreich identifizierter Gesten angeben, lassen sich diese meist aufgrund unterschiedlicher Studiendesigns, wie das Verwenden bzw. nicht Verwenden von kontinuierlichen Datenströmen, bezüglich des *Midas Touch Problem* nicht vergleichen. Auch die Gegenüberstellung verschiedener konzeptioneller Ansätze und Verfahren zur Gestenerkennung lassen sich häufig aufgrund der Verwendung verschiedener Sensoren oder Algorithmen nur schwer vergleichen.

2.6 Verfahren zur Echtzeit Gestenerkennung

Grundsätzlich werden zur Analyse und Erkennung von Gesten Verfahren angewendet, welche sich in zwei verschiedene Kategorien einteilen lassen. Diese sind maschinelles Lernen und heuristikbasierte Verfahren. Im Folgenden soll kurz auf einige dieser Verfahren eingegangen werden.

2.6.1 Maschinelles Lernen

Für die Klassifizierung von 3D-Gesten werden überwiegend Verfahren des maschinellen Lernens eingesetzt. Im ersten Schritt müssen grundsätzlich wichtige Charakteristiken (Features) der aufgenommenen Geste extrahiert werden, welche dann zur Einordnung an den Klassifizierungsalgorithmus übergeben werden [vgl. [Huang u. a. \(2009\)](#)]. Die Identifizierung und Gewichtung dieser Features ist eine der zentralen Herausforderungen bei der 3D-Gestenerkennung von auf Maschinellem Lernen basierten Verfahren. Zudem muss eine Vielzahl an hochwertigen Trainings- und Validierungsdaten vorliegen, um eine hohe Erkennungsrate zu erreichen.

Im Folgenden sollen einige wichtige Verfahren zu Maschinellem Lernen basierten Verfahren vorgestellt werden:

- Support Vector Machines (SVMs): Die mit an weitesten verbreitetste Methoden zur Maschinellem Lernen basierten Gestenerkennung sind die SVMs. Sie gehören zu den Verfahren des Überwachten-Lernens und benötigen daher eine große Menge an Trainingsdaten. Dabei wird für jede Geste ein Objekt in einem mehrdimensionalen Vektorraum definiert. Jedes Feature beschreibt eine Dimension dieses Vektorraums. Ziel der SVMs ist es eine Hyperebene in dem Raum zu definieren, welche den maximalen Abstand zwi-

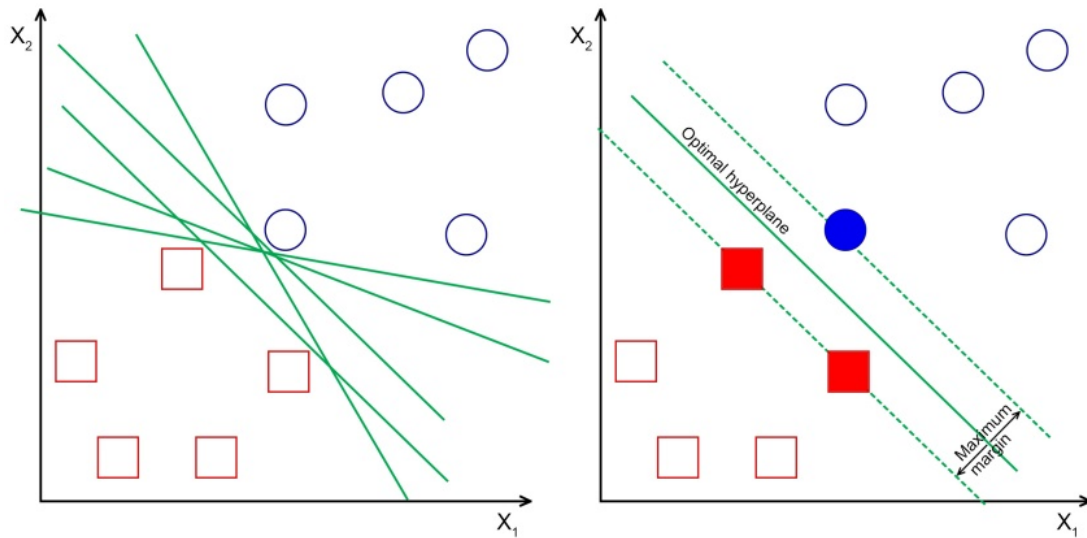


Abbildung 2.13: SVM: Beispiel einer optimalen Hyperebene in einem zweidimensionalen Vektorraum [LaViola und J. \(2014\)](#)

schen den verschiedenen Objekten (Gesten) hat. Diese Hyperebenen beschreiben bei der Einordnung neuer Daten/Gesten, wie diese klassifiziert werden (siehe Abb.: [2.13](#)).

- **Decision Trees (DTs):** DTs sind ein graphentheoretischer Ansatz zur Klassifizierung. Dabei bildet in den geordneten und gerichteten Graphen jedes Blatt eine konkrete Geste ab. Alle weiteren Knoten stehen für eine Entscheidungsregel. Diese Entscheidungsbäume werden mittels rekursiver Verfahren wie den “top-down induction of decision trees” (TDIDT) erlernt bzw. aufgebaut. Durch das Abarbeiten der Regeln von oben nach unten wird die Klassifizierung durchgeführt.
- **Decision Forests (DFs):** DFs oder auch Random Forests bestehen jeweils aus einer Vielzahl von unterschiedlich trainierten Entscheidungsbäumen (DTs). Die unterschiedlichen Ergebnisse der DTs werden gesammelt, wobei die Geste gewinnt, welche von den meisten DTs erkannt wurde. Weitere Informationen zu DFs findet man u.a. in [Criminisi u. a. \(2012\)](#).
- **Hidden Markov Models (HMMs):** Eine Markov-Kette basiert auf endlichen Zuständen, für die gilt, dass ein Zustand gedächtnislos ist, also die Historie der Zustände vor dem aktuellen keinen Einfluss auf den aktuellen Zustand hat (Markov-Annahme) und einer Menge zufälliger Übergangsfunktionen. Dieses stochastische Modell kann vielfältig für

die Klassifizierung von Gesten eingesetzt werden. Einige Beispiele dazu finden sich u.a. in [Chen u. a. \(2003\)](#).

- Conditional Random Fields (CRFs): CRFs sind ebenfalls ein stochastisches Verfahren zur Klassifizierung und können als Generalisierung von HMMs aufgefasst werden. Gegenüber den HMMs besteht nicht die Gefahr eines “labeling bias”, da HMMs eine geringe Entropie der Übergangswahrscheinlichkeiten zwischen den Zuständen haben kann. Beispiele von CRFs finden sich u.a. in [Lafferty u. a. \(2002\)](#).
- Deep Learning (DL) oder Deep Neural Networks: Aktuell werden auch vermehrt DL-Ansätze zur Klassifizierung von Gesten genutzt. Dabei wird jeder Input (z.B. Punkt der Punktwolke oder Pixel eines Bildes) durch ein Neuron dargestellt. Darauf folgen weitere Layers mit Neuronen, welche mit den Neuronen des vorigen Layers verbunden sind. Jedes Neuron hat eine Aktivierungsfunktion. Das Ergebnis, bzw. die Signale der Aktivierungsfunktion werden mit den antrainierten Gewichtung gewichtet und an Neuronen des nächsten Layers weitergegeben. Die Neuronen am Output Layer stehen für die entsprechenden Klassen von Gesten. Bei neuronalen Netzen, insbesondere beim Deep Learning (neuronalen Netze mit vielen Layern), wird jedoch eine große Menge an Trainingsdaten benötigt. Dennoch konnten unlängst hervorragende Ergebnisse bei der Gestenerkennung mittels DL erzielt werden [vgl. [Maghoumi und LaViola \(2018\)](#), [Devineau u. a. \(2018\)](#)].

Rückschlüsse Es gibt eine Vielzahl von weiteren lernenden Verfahren zur Erkennung von Gesten. Auch wenn Zusammenfassungen wie die von [LaViola und J. \(2014\)](#) einen sehr guten Überblick verschiedener Verfahren und deren erzielten Ergebnisse geben, ist eine direkte Vergleichbarkeit aufgrund unterschiedlicher Beispieldaten zum Training und Testen, unterschiedlicher Filterung der Daten, unterschiedlicher Wahl der Features sowie stark variierender Versuchsaufbauten kaum möglich. Daher muss ein System zur Gestenerkennung unbedingt die Möglichkeit bieten, verschiedene Verfahren durch gleichzeitige Auswertung auf Basis der gleichen Ausgangsdaten durchzuführen.

2.6.2 Heuristik basierte Verfahren

Heuristisch basierte Verfahren eignen sich besonders gut zur Erkennung von einer kleinen Anzahl an einfachen Gesten. Sie werden eher selten genutzt, obwohl Arbeiten wie beispielsweise [Williamson u. a. \(2011\)](#) sehr gute Ergebnisse erzielen konnten. Ein großer Vorteil ist, dass keine Trainingsdaten zur Erstellung der Regeln vorliegen müssen.

In [Williamson u. a. \(2011\)](#) konnte beispielsweise das Springen aufgrund einer einfachen Regel J implementiert werden. Dabei wurde auf Basis des Skelettmodells der Kinect von Microsoft aufgebaut:

$$J = H_y - \bar{H}_y < C$$

H_y steht für die aktuelle Höhe des Kopfes (y-Wert der Kopfposition), \bar{H}_y steht für die Höhe des Kopfes und wird für jedes Skelett einzeln automatisch kalibriert. Die Konstante C gibt den Schwellwert zwischen Stehen und Springen an. J ist das Ergebnis, welches entweder *wahr* oder *falsch* ist, je nachdem, ob die Differenz zwischen H_y und \bar{H}_y den Schwellwert C überschreitet.



Abbildung 2.14: RealEdge Prototyp auf Basis heuristischer Verfahren [[Williamson u. a. \(2011\)](#)]

Ein Nachteil von den heuristischen Verfahren besteht darin, dass das Hinzufügen von neuen Gesten zum Interaktionsrepertoire ggf. Anpassungen an andere Gesten benötigt, um eine Verwechslung auszuschließen. Zudem kann das Finden von optimalen Parametern wie beispielsweise den Schwellwert in [Williamson u. a. \(2011\)](#) beim Springen oder die Bestimmung der Intervall Längen, aufwendig sein.

2.6.3 Rückschlüsse zur Verfahrenswahl

Lernende Verfahren und heuristische Verfahren haben beide Vorzüge. Heuristische Verfahren eignen sich besonders dann, wenn keine umfangreichen Trainingsdaten für eine konkrete Geste vorliegen. Demgegenüber sind lernende Verfahren grundsätzlich robuster gegenüber False Positiv-Erkennungen.

Eine eindeutige Wahl der Algorithmen kann auf Basis des Forschungsstandes für den allgemeinen Anwendungsfall eines Smart Environments nicht getroffen werden, da diese die Vergleichbarkeit verschiedene Verfahren voraussetzen würde, welche wie u.a. in [LaViola und J. \(2014\)](#) beschrieben, aktuell nicht möglich ist.

2.7 Companion-Systeme

Companion-Systeme könnten zukünftig als virtuelle Gefährten, welche die persönlichen Vorlieben und Fähigkeiten ihres Nutzers kennen, und somit als vertrauenswürdige und partnerschaftliche Dienstleister auftreten, gesehen werden[vgl. [Yorick \(2010a\)](#)].

In diesem Kapitel sollen die Grundlagen von Companion-Systemen erläutert werden, um die Verständlichkeit der in Abschnitt multimodalen Plattformen (Abschnitt 2.8) vorgestellten Forschungsarbeiten, zu verbessern. Während im Folgenden eine Analyse und Erläuterung von Companion Systemen sowie des Forschungsbereichs gegeben wird, wird in Abschnitt 2.8 der Schwerpunkt auf der Architektur von Companion System liegen, da diese, wie auch Smart Environments, als multimodale Systeme betrachtet werden können.

Allgemeines

Computer und Softwaresysteme nehmen einen immer größer werdenden Raum in unserem Alltag ein. Dabei können Menschen emotionale Bindungen mit diesen Systemen eingehen. So gab es bereits verschiedene Produkte wie *Furby* [[Benyon und Mival \(2010\)](#)] oder *Paro* [[Ick-Dietl \(2012\)](#)], die genau dies erreichen wollten. Zudem zeigen viele Studien [vgl. [Yorick \(2010b\)](#)], dass bei der Mensch-Maschinen-Interaktion das Einbeziehen von Emotionen außerordentlich wichtig ist. Daher sind Companion-Technologien bereits ein zentrales Forschungsthema des *Artificial Intelligence* Bereichs.

Einen guten Überblick über Companion-Systeme liefert u.a. die Arbeit von [Biundo und Wendemuth \(2010b\)](#). Dabei werden *kognitive technische Systeme* als Systeme, welche selbstständig ihre Umgebung wahrnehmen und über ein internes Modell ihrer Funktionsweise sowie der Nutzer verfügen, beschrieben. Dadurch sollen sie in der Lage sein, Schlussfolgerungen zu

ziehen, Handlungen zu planen und mit Menschen und anderen Systemen zu interagieren und zu kommunizieren [vgl. [Biundo und Wendemuth \(2010a\)](#)].

2.7.1 Anwendungsgebiete und Vorläufer

Companion-Systeme sind für eine Vielzahl an Anwendungen von Bedeutung. So werden u.a. in der Abhandlung [Benyon und Mival \(2010\)](#) einige Systeme, zu denen Menschen eine Beziehung aufgebaut haben sowie Anwendungsgebiete für Companion-Systeme aufgezeigt.

Educational-Companion

Für für erzieherische oder pädagogische Zwecke können Educational-Companion verwendet werden. So können kleine Roboter Kinder mit pädagogisch wertvollen Geschichten unterhalten oder als pädagogisches Spielzeug dienen. Ein gutes Beispiel für einen solchen Companion liefert eine Studie "A Storytelling Robot for Pediatric Rehabilitation" von [Plaisant u. a. \(2000\)](#) des *Computer Interaction Laboratory* der Universität von Maryland. Dort wurde ein großer pelziger Roboter entwickelt, den die Kinder fernsteuern und ihm neue Emotionen beibringen konnten. Zudem war es möglich, Geschichten mit einer Geschichtenerzählungs-Software zu schreiben und die erlernten Emotionen des Roboters in die Geschichte einzubringen.

Caring-Companion

Ein weiteres wichtiges Anwendungsgebiet decken die Caring-Companions ab. Sie können in verschiedenen Bereichen der Medizin oder unter *Ambient-Assisted-Living*-Gesichtspunkten eingesetzt werden. Für den AAL-Bereich wurde z.B. die Roboter *Robbe Paro* entwickelt, welche therapeutisch für Senioren in einem Altenheim eingesetzt wird. Durch die emotionale Bindung mit Paro sollten bei den Senioren Depressionen und Stress gelindert und die kognitiven Fähigkeiten von Demenzkranken verbessert werden. Dabei wurde absichtlich das Aussehen eines Tieres gewählt, mit dem die Nutzer keine Erfahrungen haben. Somit können die nachteiligen Auswirkungen einer gewissen Erwartungshaltung gegenüber dem Roboter vermieden werden [vgl. [Gelderblom u. a. \(2010\)](#), [Müller \(2013\)](#)].

Informative-Companions

Der wohl unbeliebteste Informative-Companion ist die Büroklammer Clippy von Microsoft. Entwickelt, um eine einfache Hilfschnittstelle in Microsoft Word zu liefern, wurde Clippy bereits kurze Zeit später entfernt. Ein aktuelles Beispiel ist der intelligente persönliche Assistent Google Now, welcher den Zugriff auf relevante Informationen vereinfachen soll und z.B.



Abbildung 2.15: Paro im Caritas-Altenheim St. Michael - München [Ick-Dietl (2012)]

automatisch die eigenen E-Mails auf wichtige neue Nachrichten durchsucht. Dabei verknüpft Google Now bereits verschiedene Informationen. So werden zum Beispiel automatisch die Routen zum nächsten Termin berechnet und angeboten. Dabei sollen individualisierte, persönliche Assistenzsystem pro-aktiv den Nutzer unterstützen und Empfehlungen geben, ohne dabei zu stören [vgl. Biundo und Wendemuth (2010b)].

Entertainment-Companion

Zu den ersten erfolgreichen Entertainment-Companions kann man die bereits 1996 erschienenen Tamagotchis zählen. Der Nutzer musste sich um sein Tamagotchi 10-18 Tage wie um ein Haustier kümmern. Die Interaktion fand über drei Knöpfe und ein Display statt. Auch wenn die frühen Entertainment-Companions in ihrer Wahrnehmung und Interaktionsmöglichkeit stark eingeschränkt waren, so konnte man bereits viele interessante soziale Beobachtungen durchführen [vgl. Benyon und Mival (2010)].

Der Furby war bereits mit ersten Sensoren an Bauch, Rücken und Mund ausgestattet. Zudem hatte er Licht-, Geräusch- und Bewegungssensoren. Daher konnte er auf vielfältige Weise seine Umgebung wahrnehmen und der Nutzer mit ihm interagieren. So konnte er z.B. bemerken, wenn er gestreichelt wurde, ob mit ihm geredet wurde oder ob er in einen dunklen Schrank gesteckt oder geworfen wurde. Durch Motoren am Mund, den Augenlidern, den Beinen und den Ohren sowie durch einen Lautsprecher, konnte er singen, reden, tanzen und Gesichtsmimik bzw. einfache Emotionen zeigen [vgl. [Benyon und Mival \(2010\)](#)].



Abbildung 2.16: Furby [[Benyon und Mival \(2010\)](#)]

2.7.2 Forschungsbereiche von Companion-Systemen

In [Biundo und Wendemuth \(2010b\)](#) wird die methodische Grundlagenforschung für Companion-Systeme in drei Bereiche aufgeteilt und liefert damit einen ersten Eindruck über die verschiedenen Forschungsfragen, welche im Umfeld von Companion-Systeme untersucht werden. Diese werden im Folgenden vorgestellt.

Planung und Entscheidung Ziel ist es, ein Companion-System zu entwickeln, welches in der Lage ist, selbstständig oder in Zusammenarbeit mit dem Nutzer Lösungen für komplexe Aufgabenstellungen zu entwickeln. Dabei soll es dem Nutzer Empfehlungen für Handlungen geben oder im Dialog dem Nutzer bei der Entscheidungsfindung helfen. Zudem soll das System die Fähigkeiten, Interessen und die emotionale Verfassung des Nutzers sowie die Umgebungssituation berücksichtigen. Dazu ist es nötig, dass das Companion-System eine Wissensmodellierung vornimmt, auf die sich die Planungs-, Interaktions- und Dialogprozesse stützen können.

Interaktion und Verfügbarkeit Menschen interagieren mit ihren motorischen und kognitiven Fähigkeiten über verschiedene Kanäle mit ihrer Umwelt. Daher muss das Companion-System in der Lage sein, sich je nach Aufgabenstellung und Situation auf den Menschen und die Umgebung einzustellen. So muss sich z.B. ein Companion-System - gegenüber aktueller Sprachdialog-Technologie oder durch maschinelles Lernen trainierte Chat Bots - nicht nur in einer klar definierten Domäne bewegen. Um eine Adaptivität und Flexibilität, wie sie in der natürlichen Sprache vorkommt, zu erreichen, stehen Methodenentwicklung und Modellbildung für die Interaktion im Zentrum dieses Forschungsbereichs. Dabei muss eine multimodale Interaktion als Input über Sprache, Gesten, Touchscreen etc. sowie als Output über verschiedene Geräte und Kanäle entwickelt werden. Zudem muss gewährleistet werden, dass das System dauerhaft verfügbar ist und selbstständig die Modalität bzw. den Kommunikationskanal ändert, sollte dies nötig sein.

Situation und Emotion Eine wesentliche Eigenschaft von Companion-Systemen soll die Fähigkeit sein, genau wie Menschen, Situationen kontextabhängig und in Beachtung der Emotionen und Intentionen der agierenden Personen zu beurteilen. Daher soll durch Multimodalität und Fusion eine zuverlässige dynamische Erfassung der Situation (Ort, Bewegungsorientierung, Aufmerksamkeit, Emotionen, etc.) ermöglicht werden. Dabei spielt der emotionale Zustand des Nutzers eine maßgebliche Rolle. Dies ergibt sich aus aktuellen neurologischen Erkenntnissen, welche zeigen, dass in der Mensch-Computer-Interaktion eine Beschränkung auf kognitive

Analysen lediglich unbefriedigende Ergebnisse liefert. Besagtes trifft besonders häufig auf technische Systeme zu, welche an sozialen Interaktionen beteiligt sind, bei denen Emotionen eine wichtige Informationsquelle sind. Daher müssen neben den linguistischen, humanen Verhaltensweisen auch die eher bewusstseinsfernen, menschlichen Interaktionen berücksichtigt werden.

Sichten

Grundsätzlich lassen sich die relevanten Forschungsfragen aus zwei Sichten zwischen dem Zusammenspielen von Nutzer und System betrachten: Der Systemsicht, welche die Konstruktion des kognitiven Systems sowie weitere Companion-Eigenschaften umfasst und der Nutzersicht, welche die Wirkung des Companion auf den Nutzer untersucht. Dabei steht in diesem Kapitel die Systemsicht im Fokus, da diese einen ersten Einblick für Lösungen in komplexe multimodale Systeme liefert.

Funktionsweise eines Companion-Systems

In [Biundo und Wendemuth \(2010b\)](#) wird die Funktionsweise eines Companion-Systems folgendermaßen beschrieben: Ein Nutzer kann auf vielfältiger Weise explizite und implizite Anforderungen an das System stellen. Das Companion-System versucht, über einen Dialog die genaue Intention des Nutzers zu erfassen, um daraufhin die Anforderungen zu erfüllen. Dafür werden Sprache, Gestik, Mimik sowie psychobiologische Merkmale durch Komponenten des Systems erfasst und klassifiziert. Zudem wird die Umgebung mit der Sensorik untersucht, um die Umgebungsparameter zu bestimmen. Der Gesamtzustand der Situation wird durch die Gesamtheit der beobachteten Merkmale der Umgebung und des Nutzers repräsentiert. Dieser Gesamtzustand kann durch die Zusammenführung mit dem Wissensmodell in eine symbolische Zustandsbeschreibung überführt werden. Diese symbolische Zustandsbeschreibung liefert die Basis für die Steuerung der Planungs- und Interaktionskomponenten sowie für die Dialogsteuerung. Der Dialog zwischen System und Nutzer erfolgt durch multimodale Komponenten, welche die Nutzeraktionen, konkret die expliziten Funktionsanforderungen, über verschiedene Kanäle erfassen und somit die Interaktion ermöglichen. Durch die neuen Funktionsanforderungen werden Änderungen des Zustands der Anwendung und des Modells der Gesamtsituation vorweggenommen. Daraufhin können die Systemantworten auf Basis der aktualisierten Modelle durchgeführt werden. Zudem werden die Komponenten der multimodalen Interaktion dazu genutzt, um über verschiedene Geräte und Präsentationsformen die Ausgabe von Informationen an den Nutzer zu realisieren. Diese Zusammenhänge werden in [Abbildung 2.17](#) dargestellt.

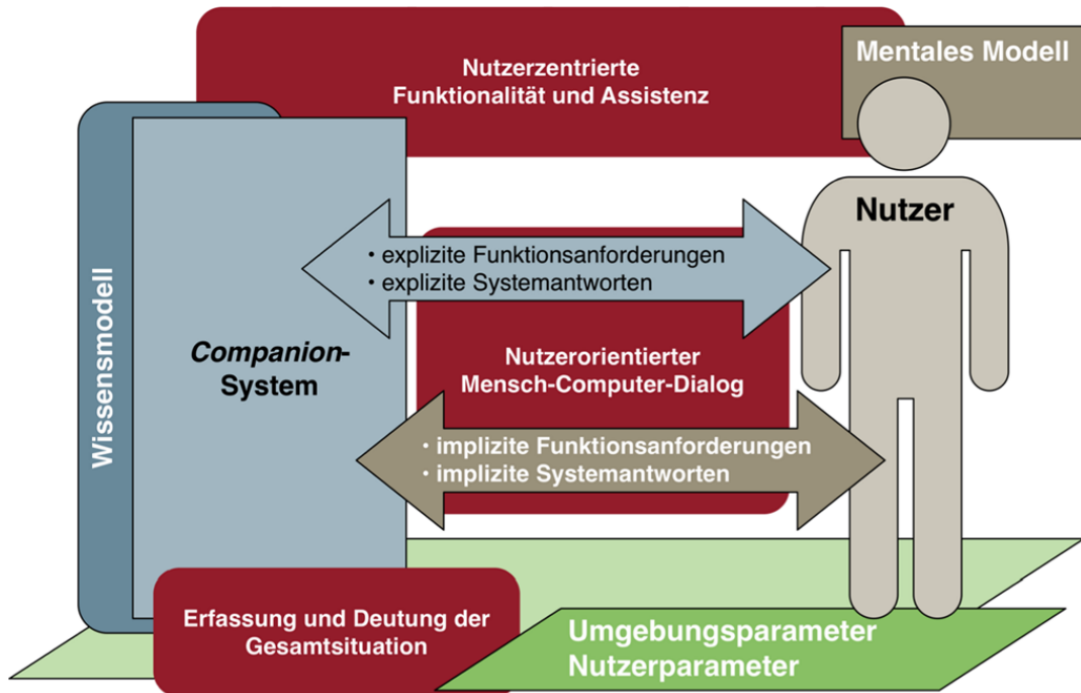


Abbildung 2.17: Hauptfunktionseinheiten eines Companion-Systems nach dem SFB/TTR 62 [Biundo und Wendemuth (2010b)]

2.8 Vergleichbare Arbeiten Multimodale Plattformen

Im Folgenden soll auf vergleichbare Arbeiten im Rahmen der Plattformarchitektur für multimodale Systeme eingegangen werden.

2.8.1 Companion Technology for Multimodal Interaction

Die Abhandlung [Honold u. a. \(2012a\)](#) zeigt die Entwicklung und den Funktionsumfang eines Prototyps für ein Assistenzsystem auf. Das Assistenzsystem soll den Nutzer bei der Einrichtung eines umfangreichen Heimkinos unterstützen. Dafür wurde ein adaptives multimodales User Interface entwickelt. Zudem hat das System verschiedene Sensoren, um Gesten- und Sprachbefehle zu erkennen und verfügt über einen Touchscreen sowie eine Schnittstelle für mobile Geräte. Das System hält interne Zustände für den Nutzer und die Umgebung und kann durch Kameras die Position des Nutzers erkennen. Die Architektur wird durch folgende vereinfachte Darstellung verdeutlicht:

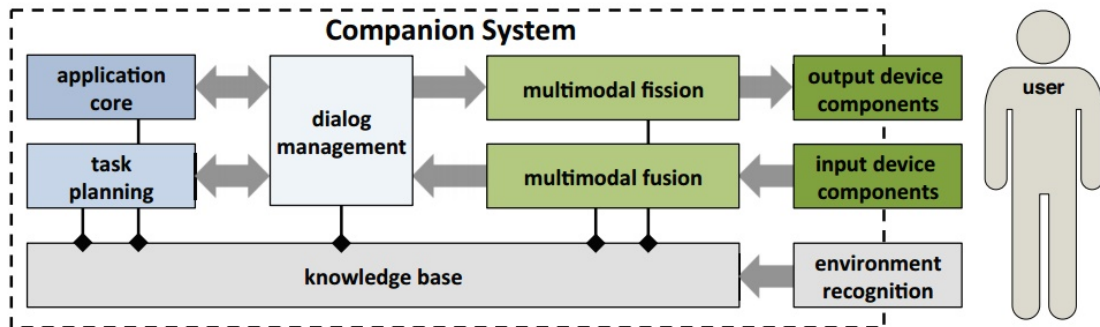


Abbildung 2.18: Vereinfachte schematische Darstellung der Architektur [Honold u. a. (2012a)]

Um die ständige Verfügbarkeit zu gewährleisten wurden ein umfassendes, flexibles User Interface Konzept entwickelt. Dies ermöglicht dem Nutzer - je nach Situation - über verschiedene Geräte mit dem Companion in Interaktion zu treten. Dies soll gewährleisten, dass die Vorlieben, Möglichkeiten und der emotionalen Zustand des Nutzers berücksichtigt werden können.

Funktionsweise des Prototypen

Der Prototyp enthält eine Wissensdatenbank, wodurch er Kenntnis über die Verkabelung der verschiedenen Komponenten des Heimkinos sowie über die verschiedenen Kabeltypen und deren Eigenschaften vorhält. Sobald die "Task Planning" Komponente die nötigen Aufgaben identifiziert und sortiert hat, wird die nächste auszuführende Aktion der "Dialog Management"-Komponente übergeben. Die "Dialog-Management"-Komponente ist dafür verantwortlich, die Systemausgaben sowie die Nutzereingaben und den Umweltzustand in eine abstrakte und modalität-unabhängige Form zu bringen. Auf Basis dieser modalität-unabhängigen Information kann die "Multimodale-Fission"-Komponente eine auf Wahrscheinlichkeit basierende Schlussfolgerungs-Strategie nutzen, um zu entscheiden, über welches Interface - der Situation entsprechend - mit dem Nutzer interagiert werden soll.



Abbildung 2.19: Der Nutzer stellt eine explizite Anforderung an das System. Der Prototyp überprüft durch die Sensorik, über welche Modalität bzw. über welchen Kanal mit dem Nutzer interagiert werden soll. [Honold u. a. (2012a)]

Zur gleichen Zeit observiert die “*Multimodal-Fusion*”-Komponente über die multimodale Sensorik, ob der Nutzer Aktionen wie sprach- oder gestenbasierte Befehle ausgeführt hat. Die erfassten Aktionen werden interpretiert und der “*Dialog-Management*”-Komponente bereitgestellt. Das System lässt eine semantische Vermengung von Gesten- und Sprachbefehlen zu. So kann z.B. ein Item durch eine Zeigegeste und den Sprachbefehl “*this one*” selektiert werden. Durch das in Abbildung 2.19 abgebildete Tablet können zur Laufzeit Änderungen am Zustandsmodell des Nutzers oder der Umgebung vollzogen werden. Dies führt zu einer direkten Änderungen des User Interfaces.

Multimodal Fission

Die Entscheidung für die zu verwendende Modalität (Gerätewahl) sowie die Art der Darstellung wird von der “*Multimodal-Fission*”-Komponente durchgeführt. Diese nutzt eine erweiterbare Menge von Regeln um die geeignetste Form der Ausgabe in der aktuellen Situation und unter Berücksichtigung der aktuell zur Verfügung stehenden Geräten zu berechnen und umzusetzen. Dabei werden aktuelle Verfahren der auf Wahrscheinlichkeit basierenden Schlussfolgerung,

um unsicheres und mehrdeutiges Wissen zu verarbeiten, verwendet. Dazu wurden die Ansätze in [Costa und Duarte \(2011\)](#) sowie in [Rousseau u. a. \(2006\)](#) verwendet. Der Prototyp zeigt zudem, dass sich die erwähnten Ansätze von [Costa und Duarte \(2011\)](#) und [Rousseau u. a. \(2006\)](#) durch ihre Performanz für die Echtzeitberechnung der Schlussfolgerung in interaktiven-Systemen eignen.

Multimodal Fusion

Die “*Multimodale-Fusion*”-Komponente basiert auf einem Verfahren von [Reddy und Basir \(2010\)](#) aus der Robotik, welches ein “*transferable belief model*”¹⁰(TBM) nutzt.

Durch die Fusion verschiedener Sensoren, erlaubt das System dem Nutzer eine natürliche multimodale Interaktion. Zudem soll die Zuverlässigkeit der Aktions-Erkennung durch die umfangreichen Möglichkeiten durch das von [Smets \(2000\)](#) entwickelte TBS-Verfahren stark verbessert worden sein.

Rückschlüsse zur Companion Technology for Multimodal Interaction

In der Abhandlung *Companion Technology for Multimodal Interaction* von [Honold u. a. \(2012a\)](#) wurde ein erweiterbares Assistenzsystem entwickelt, welches mit dem Nutzer über verschiedene Modalitäten interagieren kann. Dabei liegt der Fokus auf dem Forschungsbereich des *Planens und Entscheidens* (Abschnitt [2.7.2](#)) sowie der *Interaktion und Verfügbarkeit* (Abschnitt [2.7.2](#)).

Es wurde dargestellt, wie das System intern eine Aufgabe auf Basis der Wissensdatenbank in verschiedene Arbeitsschritte aufteilt und diese durch die *Multimodale Fission*-Komponente über einen geeigneten Kanal dem Nutzer mitteilt. Durch die Implementierung verschiedener Geräte und Kanäle wird zudem die Verfügbarkeit in verschiedenen Situationen gewährleistet.

Zusätzlich kann der Nutzer auch bei einzelnen Arbeitsschritten weitere Empfehlungen von dem System einfordern, wie z.B. die Wahl des Protokolls, wenn mehrere Typen verwendet werden können. Daher kann das System auf die unterschiedlichen Fähigkeiten des Nutzers eingehen. Diese können von dem System jedoch nicht im Vorfeld ermittelt oder erfragt werden, da das System lediglich auf explizite Anforderungen eingehen kann.

Besonders erwähnenswert ist auch die semantische Vermengung verschiedener Modalitäten, welche dem Nutzer eine natürliche Interaktion mit dem Assistenzsystem ermöglicht. Diese

¹⁰Das Transferable Belief Model ist eine ausführliche Darstellung der Evidenztheorie von Dempster und Shafer. Dabei werden verschiedene Informationen aus unterschiedlichen Quellen zu einer Gesamtaussage zusammengefasst. Zudem wird die Glaubwürdigkeit der Quellen in die Berechnung zur Bewertung einer Information mit einbezogen [vgl. [Shafer \(1992\)](#)].

konnte durch die *Multimodale Fusion*-Komponente des Systems umgesetzt werden.

Die Abhandlung hat dazu einen guten Einblick in die umfangreiche Architektur des Systems geben können. Außerdem konnte die grundsätzliche Funktionsweise eines Echtzeitplanungs- und Entscheidungsprozesses beschrieben werden. Die Umsetzung des Designs bzw. die Architektur eignet sich auch für andere komplexe multimodale Systeme, wie beispielsweise die Interaktion mit einem Smart Home.

2.8.2 Human Gesture Analysis using Multimodal Features

In der Ausarbeitung [Dan u. a. \(2012\)](#) wurde eine Plattform für Gestenerkennung, welche die Amerikanische Gestensprache (American Sign Language) ASL erkennen sollte durch weitere Features wie Emotionserkennung verbessert. Für die Erkennung der Gestik wurde ein Hidden Markov Modell HMM (2.6.1) sowie Conditional Random Fields - CRFs (2.6.1) benutzt.

Die Emotionserkennung wurde durch Modified Census Transformation (MCT) umgesetzt. Durch MCT werden Augen und Gesicht in den Aufnahmen gefunden. Im Anschluss wird das Farbbild in ein grayscale Bild umgewandelt und Discrete Cosine Transform (DCT) genutzt, um Emotionen in den Aufnahmen zu finden.

Der Schwerpunkt dieser Arbeit lag auf der Erkennung der amerikanischen Gebärdensprache durch Gestenerkennung unterstützt durch Mimikerkennung. Die meisten Gesten der Gebärdensprache sind den ikonischen Gesten zuzuordnen, während für die Steuerung von Smart Environments meist andere Arten von Gesten wie beispielsweise symbolisch-manipulative Gesten genutzt werden. Zudem sollte die Verbindung von Sprache und Gestik auch eine semantische Vermischung der genutzten Modalitäten ermöglichen, wie beispielsweise die Selektion per Gestik und die Manipulation per Sprache, welche bei ASL grundsätzlich nicht vorkommt.

2.8.3 Combining Speech, Gaze, and Micro-Gestures for the Multimodal Control of In-Car Functions

In der Ausarbeitung [Nesselrath u. a. \(2016\)](#) wurden verschiedene Sensoren zusammengeführt um die Steuerung im Auto für beispielsweise Fenster und Radio zu vereinfachen. Dafür wurde das selbst entwickelte Framework SIAM dialog platform genutzt.

SIAM-DP nutzt unter anderem Eyetracking, Sprachsteuerung und Gestenerkennung um den Fahrer während der Fahrt bei der Steuerung der Peripherie im Innenraum des Autos zu unterstützen. Eine semantische Vermischung der verschiedenen Modalitäten ist in SIAM-DP

möglich.

Auch wenn smarte Cockpits inzwischen dem Bereich Smart Environments zugeordnet werden können, unterscheidet sich die spezielle Cockpitsituation von vielen anderen Anwendungsfällen wie beispielsweise einem Smart Home durch die eingeschränkte Bewegungsfreiheit des Nutzers (2.1.1). Dennoch bietet die SIAM-DP einige Konzepte für die Verwendung von multimodalen Frameworks.

2.8.4 Rapid Development of multimodal Interactive Systems: A Demonstration of Platform for Situated Intelligence

In der Ausarbeitung [Bohus u. a. \(2017\)](#) wird Entwicklern für *human-centered Computing* ein User Interface Toolkit an die Hand gegeben um vereinfacht multimodale Systeme zu entwickeln. Dabei liegt der Schwerpunkt der Plattform auf der Fusion und dem Visualisieren verschiedener Datenströme welche durch Aktoren erhoben werden.

Drei verschiedene Anwendungsfälle werden in der Ausarbeitung demonstriert, von denen einer relevant für dieser Arbeit ist. In diesem Anwendungsfall wird eine einfache Spracherkennung von der Aufnahme über die Interpretation bis zur Zuordnung einer Person in dem Raum mithilfe der Plattform realisiert. Die Plattform bietet ein Toolkit zur Datenverarbeitung und zum maschinellen Lernen, welches das Entwickeln von multimodalen System vereinfachen soll. Dazu bietet es, neben der Visualisierung der Datenströme, Schnittstellen zu Ros oder OpenCV für eine hohe Konnektivität an.

Die Bereitstellung eines Toolkits für Entwickler erleichtert die Umsetzung neuer Ansätze und kann daher zur Vergleichbarkeit verschiedener Ansätze beitragen. Daher sollte in Betracht gezogen werden, ein solches Toolkit in ein System zur Gestenerkennung zu integrieren.

2.9 Rückschlüsse auf Basis des Forschungsstands

Die Untersuchung des aktuellen Forschungsstands konnte einige Teillösungen für Herausforderungen bei der Gestenerkennung aufzeigen.

2.9.1 Interaktionsrepertoire und Bedienungskonzept

Bei dem Wizard-of-Oz-Experimenten [Anastasiou u. a. \(2013\)](#) und [Anastasiou u. a. \(2012\)](#) konnte gezeigt werden, dass Nutzer häufig auf intuitive einfache Zeige- und Wischgesten zurückgreifen. In Fällen, bei denen keine Zeigegeste genutzt wurde, musste aus dem Kontext (Nähe,

Blickrichtung oder Kopfposition) erschlossen werden, welches Gerät der Nutzer steuern will. Zeigegesten, wie sie auch in [Kulshreshth und LaViola \(2014\)](#) genutzt wurden, lassen sich jedoch nur auf Objekte in unmittelbarer Nähe des Nutzers verwenden, da auf größere Entfernung die Auswahl zu ungenau wird. Die metaphorischen/abstrakten Gesten, welche in [Kulshreshth und LaViola \(2014\)](#) und [Gustafson u. a. \(2010\)](#) genutzt wurden, müssten erst noch auf ihre konkrete Anwendbarkeit im Smart Home untersucht werden. Sie würden jedoch eine Trainingsphase des Nutzers voraussetzen.

Grundsätzlich lässt sich die Ansteuerung eines Objekts in Selektion und Manipulation aufteilen. Daher wird geplant, ein erweiterbares System aufzubauen, welches verschiedene Arten der Selektion (z.B. durch Sprachauswahl, Zeigegesten, metaphorische Gesten oder Kontextinformationen) und der Manipulation (durch Sprachbefehle, ikonische sowie metaphorische Gesten) dem Nutzer zur Verfügung stellt. Dieser Ansatz nimmt auch die Aussagen von 20 der 26 Probanden der Untersuchung von [Anastasiou u. a. \(2013\)](#) auf, welche angegeben haben, dass sie die Wahl der Steuerungsart von den Aufgaben abhängig machen würden. Zudem zeigten die Beobachtungen in [Anastasiou u. a. \(2012\)](#), dass dem Nutzer Sprach- und Gestenbefehle für den Abbruch oder das Zurücknehmen eines falsch interpretierten Befehls zur Verfügung gestellt werden sollten.

Die in Ausarbeitung [Honold u. a. \(2012a\)](#) für Companion Systeme entwickelte Architektur, scheint durch die Einteilung des Systems in verschiedene logische Schichten extrem flexibel und daher auch für multimodale Systeme wie Systeme zur Gestenerkennung einsetzbar zu sein. Diese Plattformarchitektur sollte bei dem Design eines Systems zur Gestenerkennung berücksichtigt werden [siehe Abschnitt [2.8.1](#)].

2.9.2 Identifizierte Herausforderungen und Fragestellungen der Gestenerkennung

Des Weiteren konnte eine Vielzahl von Herausforderungen und Fragestellungen identifiziert werden die im Folgenden mit den entsprechenden Abschnitten, in denen sie diskutiert werden aufgelistet sind:

- Wie kann man in freien Interaktionsräumen, eine große Sensorabdeckung erreichen? (Abschnitte: [2.1.1](#), [2.2](#))
- Müssen Gestik und Sprache zusammen betrachtet werden? Wie kann dies umgesetzt werden? (Abschnitte: [2.1.1](#), [2.4.4](#), [2.5.4](#), [2.8.3](#), [2.8.2](#))
- Wie lässt sich das *Midas Touch Problem* lösen? (Abschnitte: [2.1.1](#), [2.2.2](#), [2.5.1](#), [2.5.2](#))

- Wie lässt sich der Kontext zur Intentionserkennung einbeziehen? (Abschnitte: 2.4, 2.8.1)
- Eignen sich mächtige (methaphorische) oder intuitive (Zeige-, ikonische) Gesten besser zur Ansteuerung in Smart Environments? (Abschnitte: 2.2, 2.5.2)
- Wie lassen sich personalisierte Interfaces für Smart Environments umsetzen? (Abschnitte: 2.5.2, 2.5.4)
- Wie kann ein Feedback für den User bei einer erkannten indirekten Interaktion umgesetzt werden? (Abschnitte: 2.5.3, 2.5.4)
- Welche Verfahren zur Gestenerkennung mit weichen Echtzeitanforderungen passen für die Interaktion mit Smart Environments? (Abschnitt: 2.6)
- Wie lassen sich sinnvoll multimodale Systeme für Smart Environments konzipieren und nutzen? (Abschnitte: 2.8.1, 2.8.2, 2.8.3, 2.8.4)
- Inwieweit müssten Ansätze des *Human-Centered Design* bei der Konzeption und Entwicklung von gestenbasierten Systemen eine höhere Bedeutung bekommen? (Abschnitt: 2.12)

Auch wenn die Analyse des aktuellen Forschungsstands eine Vielzahl von möglichen Ansätzen und Konzepten für bestehende Probleme der Gestenerkennung liefern konnte (Abschnitt 2.9.1), zeigte sich durchgehend ein grundsätzliches Problem: Verschiedene Ansätze aus unterschiedlichen Abhandlungen lassen sich aufgrund der Diversität der verschiedenen Aspekte nur sehr schwer vergleichen. Dies zeigt sich z.B. dadurch, dass in einigen Abhandlungen für die Gestenerkennung kein kontinuierlicher Datenstrom genutzt wird und daher das *Midas Touch Problem* nicht beachtet wird [vgl. Hoffman u. a. (2010)]. Ebenso wird durch die Nutzung verschiedener Sensoren zum Motion Tracking die Beurteilung und die Gegenüberstellung von verschiedenen konzeptionellen Ansätzen und Verfahren zur Gestenerkennung stark erschwert, da sich meist mehrere der in der Fragestellung (Abschnitt 2.1.3) vorgestellten Aspekte (Gestik, Sensorik, Algorithmen) sowie der Anwendungsfall unterscheiden.

Ferner muss betrachtet werden, dass die vorgestellten Studien keine längere Nutzung der Systeme von dem Probanden forderten bzw. es keine umfangreiche Evaluation bezüglich der physischen Belastung und der Ergonomie der gewählten Gestik bei einer längeren Nutzung des Systems gab.

Auch wenn die Analyse gezeigt hat, dass einige dieser Herausforderungen und Fragestellungen bereits für bestimmte Anwendungen und Umgebungen in Teilen gelöst sind, lässt sich aufgrund der schweren Vergleichbarkeit der verschiedenen Ansätze nicht eindeutig ein Ansatz

als optimal identifizieren. Zu ähnlichen Rückschlüssen bzgl. des Vergleichs verschiedener Algorithmen zur Gestenerkennung kommt auch [LaViola und J. \(2014\)](#).

Auf Grund dieses Problems, scheint die Entwicklung eines weiteren Ansatzes nicht sinnvoll, da ein umfassender Vergleich mit anderen Systemen nur schwer möglich ist.

2.10 Integrationsplattform als Lösungsansatz

Basierend auf dieser Erkenntnis (Abschnitt 2.9.2) wäre die Entwicklung eines weiteren Systems, welches nur ein Teilproblem löst und keine Vergleichbarkeit zu anderen Systemen herstellt, unzureichend.

Demgegenüber wäre die Entwicklung einer Integrationsplattform, welche einen vertikalen Schnitt durch die Architektur von der Ebene der Sensorik bis zur Erkennung einer Geste umsetzt, ein sinnvoller Beitrag. Dies, kombiniert mit Ansätzen des *Human-Centered Design* zur Identifizierung und Überprüfung der Gestik sowie des Gesamtsystems, könnte einen aktiven Beitrag zur Entwicklung von multimodalen Systemen zur Gestenerkennung leisten.

Daher werden im Folgenden die Anforderungen und Herausforderungen für eine Integrationsplattform zur Untersuchung von Gestensteuerung in Smart Environments ermittelt.

Anforderungen

Die zu entwickelnde Integrationsplattform soll es Entwicklung ermöglichen Systemen zur Gestenerkennung und eine Vergleichbarkeit verschiedener Ansätze herzustellen und Untersuchungen im Bereich der Mensch-Maschinen-Interaktion durchzuführen.

User-Centered-Design-Ansatz auf Basis einer Integrationsplattform So könnte ein geeigneter Workflow beispielsweise folgendermaßen umgesetzt werden: Ein Entwickler möchte ein System für eine gestenbasierte Steuerung von Fenstern innerhalb eines Smart Homes umsetzen. Hierzu entwickelt er ein Wizard-of-Oz-Experiment, um die geeignete Gestik für diese Anwendung zu identifizieren. Im Anschluss des Experiments kann der Entwickler die Sensorausstattung der Integrationsplattform nutzen oder diese gegebenenfalls durch weitere Sensoren zur Gestenerkennung erweitern. Auf Basis dieses Motion Tracking kann der Entwickler bestehende Frameworks und Toolkits zur Gestenerkennung nutzen oder selbstständig eigene Algorithmen umsetzen. Dabei können unterschiedliche Sensorsysteme sowie unterschiedliche Algorithmen auch parallel ausgeführt werden um eine Vergleichbarkeit dieser

herzustellen.

Daher soll es ermöglicht werden, neue Sensoren zum Motion Tracking leichtgewichtig einzubinden, effizient eine Gestenerkennung zu implementieren und diese im Rahmen von Studien schnell zu erproben, validieren und weiter zu entwickeln. Auf Basis der weiterentwickelten Ansätze kann der Entwickler nun eine weitere Nutzerstudie durchführen. Die Daten werden auf allen Schichten der Integrationsplattform gespeichert, sodass nicht nur eine Auswertung im Anschluss an der Studie möglich ist, sondern zusätzlich das Testen der Variation bzw. neue Algorithmen auf Basis dieser Daten.

Um ein Szenario wie dieses zu gewährleisten, werden im Folgenden einige Anforderungen für die Integrationsplattform definiert:

Erweiterbarkeit Um die Integration verschiedener Sensoren zur Erfassung des Motion Tracking zu gewährleisten, muss die Integrationsplattform leicht erweiterbar sein. Zudem sollte sie dem Entwickler ermöglichen, den Kontext einer Geste mit einzubeziehen. Dies kann durch das Erstellen von zusätzlichen Komponenten zur Kontexterkenkung umgesetzt werden. Der erkannte Kontext kann mit den weiteren Sensordaten der konkreten Gestenerkennung fusioniert werden, um eine höhere Robustheit der Erkennung zu erreichen.

Skalierbarkeit und dezentraler Aufbau Da der Interaktionsraum möglichst das gesamte Smart Environment umfassen soll, muss eine hohe Abdeckung von verschiedenen Seiten durch die Sensorik gegeben sein. Um eine große Sensorabdeckung zu gewährleisten, kann es nötig sein, verschiedene Sensoren miteinander zu verbinden. Diese Fusion der Sensordaten kann auf eine ähnliche Weise umgesetzt werden, wie die Einbeziehung des Kontextes. Dies bedeutet ebenfalls, dass das System für eine hohe Skalierbarkeit für die Verwendung einer großen Anzahl von Komponenten der Integrationsplattform ausgelegt sein muss.

Reaktionszeit und User Feedback Zwei weitere funktionale Anforderungen lassen sich aus der Studie des BAAL Projekts ableiten. Da sich zeigte, dass Nutzer bei dem Wizard-of-Oz-Experiment [vgl. [Anastasiou u. a. \(2013\)](#)] verwirrt waren, wenn das System nicht direkt auf eine Geste reagierte, muss sichergestellt werden, dass der Nutzer direkt nach Verwendung einer Geste ein Feedbacks darüber bekommt. Daher lässt sich die Umsetzung eines passenden Feedbacks sowie eine maximale Reaktionszeit von 100 ms [vgl. [Miller \(1968\)](#)] des Gesamtsystems definieren, um eine hohe Usability für den Nutzer zu erreichen. Konkrete Angaben für die Reaktionszeit variieren stark in unterschiedlichen Untersuchungen. Für die Integrations-

plattform wird auf Basis einer vergleichbaren Arbeit [Anastasiou u. a. \(2013\)](#), in einem Smart Home eine Reaktionszeit von weniger als 100 ms angesetzt.

Portierbarkeit Damit die Integrationsplattform auch in anderen Umgebungen integriert und eingesetzt werden kann, muss diese leicht portierbar sein und sollte daher auf gängigen Systemen einsetzbar sein.

Datenpersistenz und Monitoring Um dem Entwickler das Durchführen von Nutzerstudien zu ermöglichen, muss eine umfangreiche Datenaufnahme und ein Echtzeit-Monitoring der Datenströme gewährleistet werden. Neben den produktiv genutzten Daten zur Gestenerkennung werden weitere Daten wie normale Kameraaufnahmen zur manuellen Auswertung bereitgestellt werden müssen.

Validierung der Integrationsplattform

Um zu überprüfen, ob die Integrationsplattform die Anforderungen, welche hier definiert worden sind, erfüllen kann, soll dies konzipiert und entwickelt werden. Des Weiteren sollen einige Komponenten beispielhaft implementiert werden, um zu evaluieren, ob die Plattform den gewünschten Nutzen für Entwickler und die Untersuchung von Gesten in Smart Environments erfüllt. Dies kann gegebenenfalls durch Beta-Tests oder eine HCI Studie auf Basis der Integrationsplattform unterstützt werden.

Im Folgenden wird auf die Rahmenbedingungen der Integrationsplattform im Labor sowie auf relevante Vorarbeiten in der Forschungsgruppe der HAW Hamburg eingegangen.

2.11 Laborumgebung Living Place Hamburg

Als Anwendungsbeispiel für die Integrationsplattform bietet sich das Living Place Hamburg der HAW Hamburg an. Dies ist ein Smart Home Labor, welches im Rahmen der Ubicomp Forschungsgruppe aufgebaut wurde und inzwischen Teil des Creative Space for Technical Innovation ist. Im Rahmen des Creative Space und des Living Place werden aktuell angewandte Forschungsfragen in den Bereichen HCI und Smart Systems durch interdisziplinäre Teams bearbeitet. Die Wohnung ist als modernes Einperson loft mit Wohnbereich, Küche, Bad, Essbereich und Schlafzimmer ausgelegt. Zusätzlich gibt es einen Schnittraum, eine Werkstatt und zwei Büros, welche von der Wohneinheit getrennt sind.

Für den Living Place stehen konkrete Fragestellungen aus dem Bereich Smart Home, Ubiquitous Computing, Ambient Intelligence und Affective Computing im Vordergrund.



Abbildung 2.20: 3D-Modell des Living Place Hamburg [Karstaedt (2012)]

Das Living Place bietet durch seine auf dem Blackboard basierende Infrastruktur verschiedene Sensoren und Aktoren, welche über ActiveMQ-Nachrichten austauschen. Durch verschiedene Komponenten wird eine Steuerung der Fenster, Vorhänge, Rollos sowie des Lichts ermöglicht. Somit liefert das Living Place Hamburg eine hervorragende Forschungsplattform für beispielsweise Wizard-of-Oz-Experimente und andern HCI Studien und ermöglicht eine schnelle Umsetzung von gestenbasierter Ansteuerung der installierten Haustechnik. Da sich die konkrete Infrastruktur des Smart Homes verändert und in Rahmen dieser Arbeit weiterentwickelt hat, wird diese konkret in der Realisierung (Abschnitt 4.3, 4.3) erläutert.

2.12 HCI Vorstudie zu Gesten im Smart Home

Auf Basis der zuvor beschriebenen Infrastruktur wurde von Karloina Bernat eine Vorstudie [vgl. Bernat (2017)] zur Identifizierung von intuitiven Gesten durchgeführt. Mittels eines

Wizard-of-Oz-Experiments konnten anhand von 15 Probanden erste Präferenzen der Teilnehmer zur Steuerung der Haustechnik wie Rollos und Fenster festgestellt werden. Diese Daten liefern erste Beschreibungen von intuitiven Gesten, welche für die zu konzipierende Plattform genutzt werden sollen.



Abbildung 2.21: Fensterrollo schließen [Bernat (2015)]

So wurden für die Geste, um die Rollos runter zu fahren, beide Hände oder jeweils eine Hand gleichmäßig von oben nach unten bewegt, während sie dem Fenster zugewandt waren (siehe Abb. 2.21). Zudem zeigte sich, dass die Probanden für die Manipulation von Geräten, welche sie nicht direkt physisch steuern können, diverse Gesten genutzt haben. Dies zeigte sich beispielsweise bei der Aufgabe, das Licht an/auszuschalten, da grundsätzlich nicht mit der Lampe direkt sondern mit einem Schalter interagiert wird (siehe Tabelle 2.22).

Aus dieser Vorstudie ergeben sich unterschiedliche Rückschlüsse für die Umsetzung der Integrationsplattform:

- Geste zum Öffnen und Schließen der Rollos ist intuitiv und eignet sich als erste Umsetzung zur Steuerung der Haustechnik.
- Unterschiedliche Nutzer haben für einige Aktoren unterschiedliche Präferenzen bezüglich der Geste zur Ansteuerung. Die Umsetzung dieser sollte schnell, einfach umsetzbar und evaluierbar sein.
- Das Living Place ist zur Umsetzung und Evaluierung der Integrationsplattform gut geeignet, da die Probanden sich sicher im Umgang mit Haustechnik fühlen und die nötige Infrastruktur zur Evaluierung von Versuchen vorhanden ist.

Switch the light off		Switch the light on	
Gesture	Number of attendees	Gesture	Number of attendees
Clap with both hands	5	Clap with both hands	6
Flip/Snap with fingers	1	Flip/Snap with fingers	2
Touch with one hand on the bedside lamp	2	Touch with one hand on the bedside lamp	2
Point with an arm to the ceiling lamp and move the arm to the side	2	Point with an arm to the ceiling lamp and move the arm to the side	1
Put the palm of your hand horizontally in front of your face and move it down your face	1	Put the palm of your hand horizontally in front of your face and move it up your face	1
Point with your hand to the ceiling lamp and rotate your hand (similar as a light bulb)	1	Point with your hand to the ceiling lamp and rotate your hand (similar as a light bulb)	2
		Point with one arm at the bedside lamp	1
Total	12	Total	15

Abbildung 2.22: Tabelle zur präferierten Steuerung des Lichts [Bernat u. a. (2017)]

Diese Studie gibt erste Hinweise dazu, welche Gesten von Nutzern in einem Smart Home zur Ansteuerung von Haustechnik nutzen und kann daher als Grundlage für die Entwicklung von Gestenerkennungssystemen im Living Place dienen.

2.13 Zusammenfassung der Analyse

Nach der Einordnung dieser Arbeit (Abschnitt 2.1.3) wurde eine Analyse von Gestik (Abschnitt 2.2) und Gestenerkennung (Abschnitt 2.3.1) durchgeführt und eine Betrachtung des Kontextes (Abschnitt 2.4) für eine gestenbasierte Interaktion in Smart Environments diskutiert. Des

Weiteren wurden einige bestehende Algorithmen zur Gestenerkennung und eine Vielzahl von Konzepten, Ansätzen und multimodalen Systemen dargestellt.

Im Anschluss wurde erläutert, dass eine weitere Umsetzung eines neuen Ansatzes bei der großen Menge an bestehenden Lösungsansätzen nur einen geringen Mehrwert für das Thema der gestenbasierten Interaktion hätte, da die Vergleichbarkeit mit anderen Konzepten und Ansätzen kaum gegeben ist (Abschnitt 2.9.2).

Um diese Problematik zu lösen, wurde die Konzeption und Entwicklung einer Integrationsplattform für Mensch-Maschinen-Untersuchungen vorgeschlagen und diskutiert (Abschnitt 2.10). Im Anschluss wurden Anforderungen an eine solche Plattform definiert (Abschnitt 2.10) und kurz auf die bestehende Infrastruktur des zur Verfügung stehenden Smart Home Labors (Abschnitt 2.11) sowie eine Vorstudie (Abschnitt 2.12) eingegangen.

3 Design

In diesem Kapitel soll die Konzeption der Integrationsplattform auf Basis der in der Analyse gesammelten Rückschlüsse und Anforderungen beschrieben werden. Zudem wird eine durchgeführte Vorstudie im Bereich der Gestenerkennung vorgestellt welche erste Rückschlüsse auf die Machbarkeit einer solchen Integrationsplattform zulässt. Ferner wird eine Verfahrenswahl bzgl. der Algorithmenklasse (lernende Verfahren/heuristische Verfahren) für die Gestenerkennung getroffen, welche beispielhaft für die Integrationsplattform umgesetzt werden soll.

3.1 Konzeption

Das Gesamtkonzept besteht aus drei Teilen: der Plattformarchitektur, der Systemarchitektur des Interaction-Management-Systems und der Integration der Integrationsplattform in die Architektur des Smart Homes.

Da, wie bereits in der Analyse ausgeführt, die Infrastruktur des Living Place im Rahmen dieser Arbeit weiterentwickelt worden ist (Abschnitt 2.11), wird auf diesen Teil der Architektur im Rahmen der Beschreibung der entwickelten Komponenten in der Realisierung (siehe Abschnitt 4.3) eingegangen.

3.1.1 Plattformarchitektur

Die Plattformarchitektur wird aufgrund der Anforderungen bezüglich Verteilung, Skalierbarkeit und Nebenläufigkeit auf Basis des Aktor-Modells umgesetzt. Das Aktor-Modell [vgl. Hewitt u. a. (1973)] setzt ähnlich wie das Architektur Muster Blackboard [vgl. HNi (1986)] auf ein schlankes Grundprinzip und das Versenden von Nachrichten. Einzelne Aktoren können selber Kanäle, über welche sie Daten versenden, veröffentlichen oder Kanäle anderer Aktoren abonnieren, um deren Datenströme zu empfangen.

Die Eigenschaften der Plattform bezüglich der Anforderungen hängen stark mit der konkreten Implementierung der Architektur zusammen. Daher wird in der Realisierung ausführlich auf die eingesetzten Frameworks sowie die konkrete Implementierung eingegangen (Abschnitt 4.1.1).

3.1.2 Systemarchitektur des Interaktion-Management-Systems

In Anlehnung an die Architektur des Companion Systems [vgl. Honold u. a. (2012b)], welche in Abschnitt 2.8.1 erläutert wurde, wird die Plattformarchitektur durch das Architekturmuster des Interaction Management Systems für multimodale Systeme ergänzt. Durch diese Erweiterung wird die Kommunikation der Aktoren, des ansonsten lose definierten Aktor-Modells, in eine konkretere Struktur gebracht.

Diese Rahmenbedingungen führen zu einer Trennung der Komponenten bzw. der Aktoren in verschiedene logische Einheiten (siehe Abb.: 3.1).

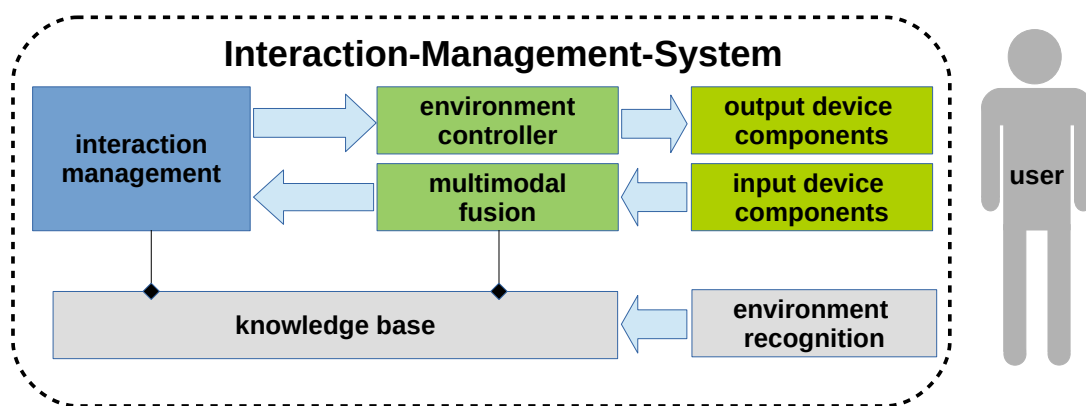


Abbildung 3.1: Systemarchitektur des Interaktion-Management-Systems nach Anpassung der Architektur von Honold u. a. (2012b) für die Integrationsplattform

Zur Erläuterung werden im Folgenden die fünf zentralen logischen Einheiten erläutert:

Interaction management Das Interaction Management bildet die zentrale Einheit, welche zwischen der Wahrnehmung der Umgebung und der Manipulation dieser agiert. Sie beinhaltet die konkrete Anwendungslogik auf Basis der in der multimodalen Fusion erhobenen Befehle. Durch das Einbeziehen weiterer Informationen der Wissensdatenbank, sollen konkrete Rückschlüsse auf die Aktionen des Nutzers gezogen werden. Diese Rückschlüsse bilden die Grundlage für die Generierung von Steuerbefehlen, welche an den Environment Controller weitergeleitet werden.

Multimodal Fusion Auf der Ebene der multimodalen Fusion kommen die verschiedenen Modalitäten zur Erfassung der Intention des Nutzers zusammen. Dies kann beispielsweise durch die Zusammenführung verschiedener Sensoren und eine darauf basierende Gestenerkennung

umgesetzt werden. Ebenso kann das Einbeziehen des Kontextes durch weitere Sensoren auf dieser Ebene umgesetzt werden. Dabei werden Befehle generiert, welche unabhängig von der konkreten, vom Nutzer gewählten Modularität (bspw. Sprache oder Gestik), erfasst werden. Diese Befehle werden in aggregierter Form an das Interaction Management weitergeleitet.

Die Fusion verschiedener Modalitäten soll zu präzisieren und zuverlässigen Aussagen über den Nutzer führen.

Environment Control/User-Feedback Komponente Komponenten, welche auf der Ebene des Environment Control einzuordnen sind, sind für die Manipulation der Umwelt des Nutzers verantwortlich. Ferner sollen diese ebenfalls dem Nutzer ein Feedback geben im Falle einer erkannten Interaktion oder bei Unklarheiten der Intention des Nutzers. Durch Kontextinformationen, welche in der Wissensdatenbank vorliegen, wird auf dieser Ebene entschieden, welche Modalität für die Interaktion mit den Nutzer gewählt werden soll. So kann z.B. der aktuelle Ort des Nutzers oder die Tätigkeit bei der Auswahl des Feedbacks mit einbezogen werden. Die Environment Controller Ebene gewinnt an Bedeutung, wenn das System verschiedene Möglichkeiten zur Interaktion mit dem Nutzer bereitstellt, da die Wahl der Modalität von ihr getroffen wird.

Wissensdatenbank Die Wissensdatenbank enthält Informationen über den aktuellen Zustand aller steuerbaren Elemente des Smart Environments. Außerdem liefert sie dem Interaction Management eine Schnittstellenbeschreibung über mögliche Manipulationen dieser Elemente. Des Weiteren kann die Wissensdatenbank zusätzlich Informationen über den aktuellen Kontext des Nutzers speichern und somit anderen Komponenten zusätzliche Informationen zur Auswertung der Daten liefern. Ob die Umsetzung einer vollständigen Wissensdatenbank nötig ist, ist vom konkreten Anwendungsfall abhängig, da nicht in jedem Szenario das Vorhalten von unstrukturierten Wissen nötig ist.

Input/Output Device Components Für die Sensorik sowie für die Aktorik werden Aktoren erstellt, welche die Konnektivität von der genutzten Hardware zur Integrationsplattform ermöglichen.

Systemaufbau

Eine beispielhafte Umsetzung der Systemarchitektur könnte wie in Abbildung 3.2 aussehen.

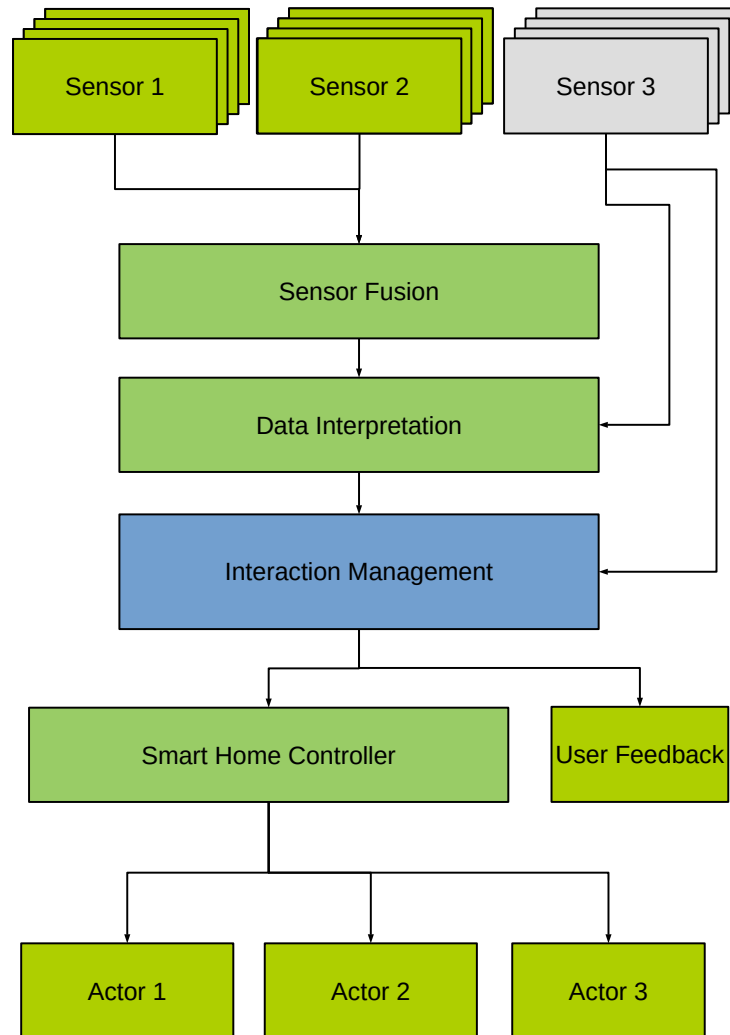


Abbildung 3.2: Komponenten des Interaktion-Management-Systems

Hellgrün: Input/Output

Dunkelgrün: Environment Controller und Multimodal Fusion

Grau: Environment Recognition

Blau: Interaction Management

Die Farben der Komponenten in Abbildung 3.2 zeigen die entsprechende Zugehörigkeit zu den Ebenen in Abbildung 3.1 auf.

Eine Vielzahl von Sensoren kann eingesetzt werden, um das Motion Tracking des Nutzers in einem größeren Interaktionsbereich zu erfassen. Dabei können auch verschiedene Sensoren zum Motion Tracking zum Einsatz kommen. Weitere Sensorik kann genutzt werden, um allgemeine Informationen über die Umgebung oder den Kontext zu erfassen. Die darüber erfassten Daten werden gegebenenfalls miteinander fusioniert und entsprechend interpretiert. Das Interaction Management kann auf Basis der Interpretation sowie des Kontextes Entscheidungen darüber fällen, ob der Nutzer die Intention einer Ansteuerung des Smart Homes aufweist. Entsprechend der Interpretation des Nutzerverhaltens, kann das Interaction Management dem Nutzer ein Feedback geben und die entsprechenden Befehle der Steuerung der Haustechnik an den Smart Home Controller weitergeben. Dieser kennt die Schnittstellen der verschiedenen Aktoren und übernimmt deren konkrete Ansteuerung.

3.2 Vorversuch: Evaluation of an Omnidirectional Walking-in-Place User Interface

Für den Vorversuch “Evaluation of an Omnidirectional Walking-in-Place User Interface with Virtual Locomotion Speed Scaled by Forward Leaning Angle” [vgl. Langbehn u. a. (2015)] wurde auf Basis eines zur Auswahl stehenden Frameworks eine erste Komponente für das Motion Tracking mithilfe der Kinect v2 entwickelt sowie eine weitere Komponente zur Erkennung der Gestik der Versuchsperson. Zudem wurde in Zusammenarbeit mit Tobias Eichler eine weitere Komponente zur Fusion von mehreren Kinect v2 Adaptern umgesetzt.

Dadurch wurde ein omnidirektionales Kamerasystem zur Erkennung der Bewegungen einer Person innerhalb eines Bereichs umgesetzt (siehe Abb.: 3.3). In Zusammenarbeit mit der Universität Hamburg wurde dieses System mit einer VR-Brille gekoppelt, um die Bewegungen des Probanden in die virtuelle Welt zu übertragen. Dabei sollten die Probanden sich durch eine virtuelle Nachstellung der Hammaburg bewegen (siehe Abb.: 3.4). Dieses System wurde in zwei verschiedenen Modalitäten von mehreren Probanden getestet und die Ergebnisse bei der GIVRAR 2015 vorgestellt.



Abbildung 3.3: Versuchsaufbau: Vier Kinects v2 beobachten den Probanden von allen Seiten. Der Proband trägt eine VR-Brille sowie einen Rucksack mit einem Notebook, an das die Brille angeschlossen ist. [Langbehn u. a. \(2015\)](#).

Konkret wurde bei der Studie untersucht, ob das Beugen des Oberkörpers nach vorne und die damit einhergehende Geschwindigkeitszunahme in der virtuellen Welt, in Anlehnung des nach vorne Beugens beim Rennen, die Immersion der virtuellen Welt erhöht.



Abbildung 3.4: Die virtuelle Welt, durch die der Nutzer laufen musste, wurde auf Basis der Hammaburg des neunten Jahrhunderts modelliert [[Langbehn u. a. \(2015\)](#)].

Durch diesen Vorversuch konnten ein Framework auf die Nutzbarkeit für die Integrationsplattform geprüft sowie erste Ideen für Komponenten getestet werden. Ferner konnten weitere Anforderungen für die Integrationsplattform identifiziert werden. Diese umfassen ein live Monitoring sowie einen Lifecircle der Komponenten und die Erfordernis zur Fernkonfiguration der verschiedenen laufenden Komponenten.

In dieser Vorstudie wurde gegenüber dem Badezimmerspiegel (Abschnitt 2.1.1) ein größerer Bereich von mehreren Sensoren abgedeckt, sodass ein Bewegen innerhalb des Bereiches sowie das Erkennen von Gesten, Sprache oder Mimik aus verschiedenen Winkeln möglich wurde. Die Studie diente zudem als Stresstest für die verwendete Middleware, da sie den Datenstream von vier Kinect v2 Kameras sowie die Auswertung mit weichen Echtzeitanforderungen durch eine Komponente zur Gestenerkennung abwickeln musste.

Zusätzlich konnte der Vorversuch Einblicke in die Problematik der Entwicklung eines gestenbasierten Systems für HCI-Studie liefern. So konnte durch die Fusion der Skelette eine 360°-Erkennung des Nutzers gewährleistet werden, welche nicht mit einem einzelnen optischen Sensor umgesetzt werden kann. Weitere Details zum Tracking System folgen im Abschnitt 4.2.2 zum Fusion Agent oder können der Arbeit Langbehn u. a. (2015) entnommen werden.

Verfahrenswahl der 3D-Gestenerkennung

Für die beispielhafte Implementierung einer Gestenerkennung scheinen sich heuristische Verfahren besser zu eignen, da die Entwicklung keine umfangreichen Trainingsdaten benötigt. Außerdem lassen sich die in der HCI-Vorstudie (Abschnitt 2.12) identifizierten intuitiven Gesten leicht mittels heuristischer Ansätze erkennen. Diese Einschätzungen konnten bei der Umsetzung der Gestenerkennung für den Vorversuch (Abschnitt 3.2) bestätigt werden. Des Weiteren besteht der Vorteil, dass für neue oder abgewandelte Gesten keine neuen Trainingsdaten erhoben werden müssen. Dieser Vorteil überwiegt gegenüber dem Risiko, dass beim Hinzufügen neuer Gesten ggf. Anpassungen an die heuristische Erfassung zuvor implementierter Gesten vorgenommen werden müssten (siehe 2.6.2).

Sollte sich dennoch die Notwendigkeit einer auf maschinellem Lernen basierenden Gestenerkennung ergeben, kann diese dem System zusätzlich hinzugefügt werden, ohne dass dies Einfluss auf die bestehende Umsetzung heuristischer Verfahren hat.

3.3 Zusammenfassung des Designs

Im Design konnte mit Hilfe eines Vorversuchs eine geeignete Plattformarchitektur für die Integrationsplattform identifiziert und für die Anforderungen an ein Smart Home angepasst

werden. Außerdem konnten erste Erfahrungen mit gestenbasierten HCI Studien gesammelt werden, bei welchen die ersten Implementierungen sowie die Middleware durch einen Nutzer-test evaluiert werden konnten.

Durch die Verwendung des Aktor Modells, die Aufteilung der Komponenten in verschiedene logische Ebenen sowie die Verwendung modalitätsunabhängiger Befehle soll eine lose Kopplung der Systemkomponenten erreicht werden. Diese, für multimodale Systeme ausgelegte Systemarchitektur des Intermanagement-Systems, ermöglicht eine semantische Vermengung von verschiedenen Kommunikationskanälen und gestattet daher einen transparenten Austausch von Komponenten. Das soll den Nutzern der Integrationsplattform ermöglichen, leichtgewichtig verschiedene Gesten, Ansätze, Sensoren und Algorithmen zu verwenden und zu vergleichen.

Die konkrete Umsetzung der Architektur sowie ein Modell der Komponenten folgt im anschließenden Kapitel, der Realisierung.

4 Realisierung

Die Integrationsplattform, welche auf der Middleware von [Eichler \(2014\)](#) aufbaut, besteht aus einer Vielzahl von Aktoren. Um die Machbarkeit der Integrationsplattform, wie sie im Design konzipiert wurde, zu zeigen, wird im Folgenden eine beispielhafte Implementierung dieser realisiert.

Dabei liegt der Fokus bei der Implementierung auf der Umsetzung eines vertikalen Schnitts der logischen Ebenen von der Datenaufnahme der Sensorik bis zur Ansteuerung des Smart Homes bzw. Interaktion mit dem Nutzer durch ein Feedback.

Dazu wird im Folgenden auf die entwickelten Komponenten zur Hardware Ansteuerung, die entwickelten Agenten sowie die Integration in die bestehende Infrastruktur eingegangen.

4.1 Framework und Middleware

4.1.1 Framework

Das zugrundeliegende Aktor-Modell wurde durch das Toolkit akka.io, welches auf einer Java Virtual Machine aufbaut, umgesetzt. Das Toolkit ist darauf ausgelegt, hochgradig nebenläufige, verteilte Nachrichten basierte Systeme umzusetzen.

Akka zeichnet sich durch außerordentlich niedrige Latenzen beim Messaging sowie eine außerordentliche hohe Skalierbarkeit durch die Leichtgewichtigkeit der Umsetzung der Aktoren aus.

Die Aktoren werden innerhalb eines Aktor-Systems ausgeführt. Diese Aktor-Systeme beinhalten jeweils einen eigenen Thread Pool. Dieser verteilt die Ausführungszeit mithilfe eines Dispatchers auf die verschiedenen Threads. Für die Erstellung eines Aktors ist kaum mehr Aufwand nötig als für die Initialisierung eines Objekts. Dies führt neben leichtgewichtigen Aktoren ebenfalls zu einem sehr geringen Speicherbedarf für diese. Das ermöglicht einem Aktor-System, eine enorme Anzahl von Aktoren (bis zu 2,5 Millionen Aktoren pro GB Hauptspeicher¹) zu verwalten.

¹<https://akka.io/>

4.1.2 Middleware

Auf Basis des akka Toolkits wurde die Middleware zur Entwicklungsunterstützung im Bereich Smart Environments von [Eichler \(2014\)](#) erstellt. Dabei soll die Middleware Anforderungen wie Fehlertoleranz, Skalierbarkeit und Reaktionsfreudigkeit sowie die Unterstützung von Heterogenität und Kontextverarbeitung für spezifische Anwendungen in Smart Environments ermöglichen. Dies soll wiederum Entwicklern ermöglichen, mit geringem Aufwand einen Überblick über das System zu gewinnen und bei Bedarf in das System einwirken zu können. Daher überwacht die Middleware wichtige Teile des Systems und kann auf Fehler von Agenten entsprechend reagieren.

Die Akka-Aktoren wurden um zusätzliche Funktionen, beispielsweise ein vereinfachter Zugriff auf die Gruppenkommunikation und die Interaktionsmöglichkeit mit der Middleware, erweitert und werden im Folgenden als Agenten bezeichnet. Zur Evaluierung des Mittelwerts wurde ein Test im Living Place Hamburg zur Messung von Latenzzeiten durchgeführt [vgl. [Eichler \(2014\)](#)]. Dabei konnte gezeigt werden, dass die Middleware mit einer sehr großen Anzahl von Agenten, Nachrichten und Gruppen umgehen kann. So wurde beispielsweise ein Testszenario aufgebaut, in dem gezeigt werden konnte, dass durchschnittlich 27.000 Nachrichten pro Sekunde bei 120.000 Testagenten und sieben Middleware-Konten zugestellt werden konnten. Dabei wurden 90% der Nachrichten innerhalb von 29 ms zugestellt. Die durchschnittliche Antwortzeit betrug unter 12 ms. Dies zeigt, dass das System eine große Anzahl von Nachrichten und Agenten bewältigen kann. Durch die Möglichkeit, weitere Middleware-Knoten mit eigener Laufzeitumgebung dem System hinzuzufügen, konnte ebenfalls gezeigt werden, dass das System stark skalierbar ist.

Daher zeichnet sich die Middleware als Basis für die Integrationsplattform nicht nur aufgrund der Spezialisierung für Smart Environments, sondern ebenfalls aufgrund der schnellen Reaktionszeit sowie der hohen Skalierbarkeit, welche in der Analyse als funktionale Anforderungen an die Integrationsplattform gestellt worden sind (siehe Abschnitt [2.10](#)) aus. Zudem können somit Agenten, die für das Smart Home Labor Living Place entwickelt werden, direkt in Verbindung mit der Integrationsplattform verwendet werden.

Die im Abschnitt [4.2](#) vorgestellten Systemkomponenten sind, mit Ausnahme des Living Place Controllers, als Agenten der Middleware umgesetzt.

4.2 Systemkomponenten der Integrationsplattform

Während bereits im Abschnitt 3.1.1 die System-Architektur und das Design thematisiert wurden, soll im Folgenden auf die konkreten Softwareagenten, welche in Abbildung 4.1 zu sehen sind, sowie auf deren Umsetzung eingegangen werden.

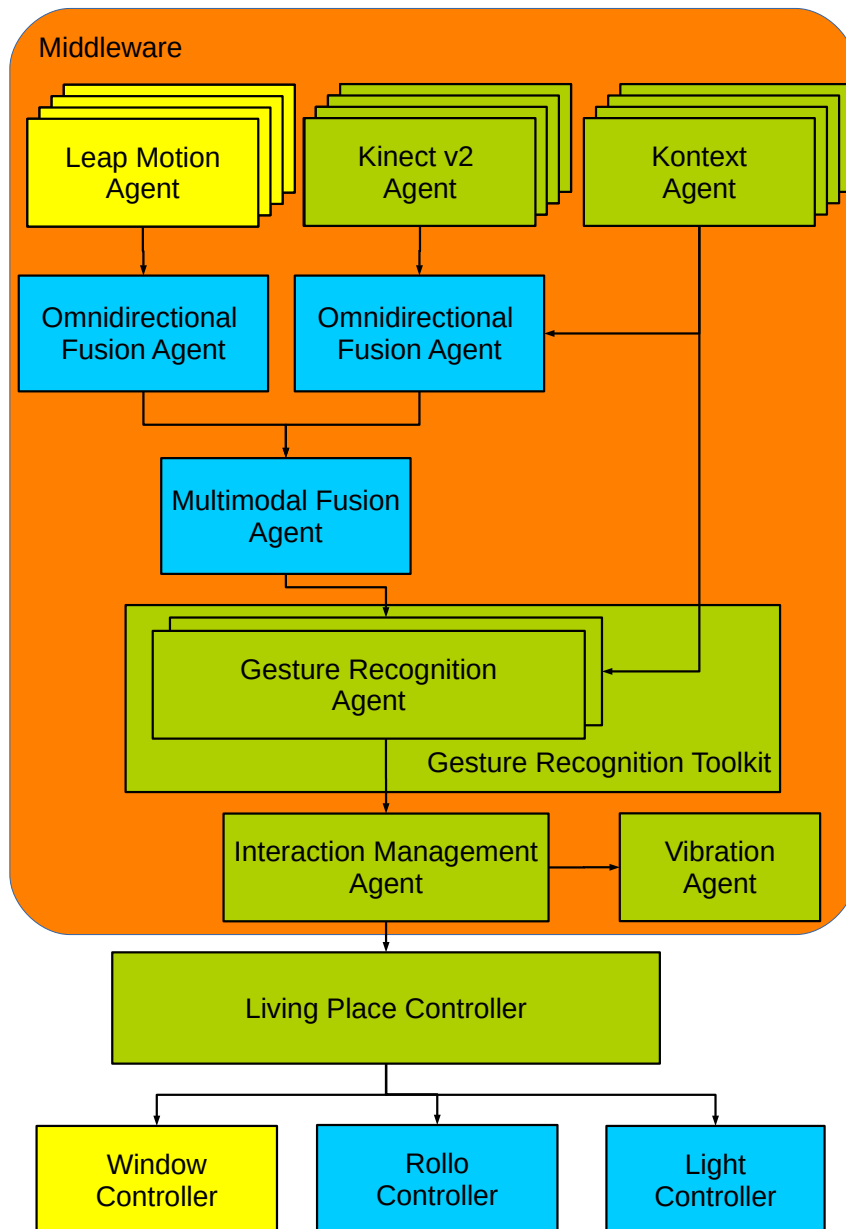


Abbildung 4.1: Systemkomponenten der Integrationsplattform

Alle Komponenten, die in Abbildung 4.1 in der Farbe Grün dargestellt sind, wurden im Rahmen dieser Arbeit konzipiert und implementiert. Die Komponenten, welche blau eingefärbt sind wurden in Zusammenarbeit mit weiteren Personen aus der UBICOM Arbeitsgruppe umgesetzt. Komponenten, welche gelb dargestellt sind, wurden von anderen Teilnehmern der Arbeitsgruppe für den Living Place bereitgestellt.

Im Folgenden soll auf die verschiedenen Komponenten des Gesamtsystems eingegangen werden.

4.2.1 Kinect v2 Agent

Wie bereits in Abschnitt 2.3.2 beschrieben, verfügt die Kinect v2 (Abbildung 4.2) über die Möglichkeit, Skelette von bis zu sechs Personen zu erkennen. Diese werden mit Hilfe von 25 Joints beschrieben, welche menschlichen Gelenken und Knochen entsprechen. Im Rahmen der Arbeitsgruppe UBICOM um den Living Place wurde ein Skeleton API für eine generisches Skelett-Modell eines Menschen definiert. Dieses Modell orientiert sich an den anatomischen Gelenken eines Menschen und verfügt über einen höheren Detailgrad als das Skelett-Modell der Kinect v2. Im Rahmen dieser Arbeit musste ein Agent entwickelt werden, welcher die Daten der Kinect v2 entgegennimmt und diese in das generische Skelett-Modell überführt. Diese Daten wurden dann mit Hilfe der Middleware anderen Agenten bereitgestellt.



Abbildung 4.2: Kinect v2 von Microsoft [Evan-Amos (o.J)]

Die erste Version dieses Agenten zur Datenabnahme der Kinect v2 wurde im Rahmen der Vorstudie, welche im Abschnitt 2.12 vorgestellt worden ist, genutzt und im weiteren Verlauf der Entwicklung der Integrationsplattform verbessert.

Im Folgenden wird das J4KSDK vorgestellt, welches ermöglicht, Daten aus der Kinect in einer Java-Umgebung auszulesen.

Kontext Agent Die Kinect v2, ebenso wie das zugehörige SDK und Toolkit, sind prioritäre Systeme, welche sich lediglich mit C# und C++ verwenden lassen. Da das Framework, die Middleware und die dazugehörigen Agenten in der JVM ausgeführt werden, wurde das J4KSDK genutzt, um die Daten zu extrahieren. Das J4KSDK nutzt dafür das Java Native Interface (JNI), um den Zugriff auf die Datenstrukturen der Kinect v2 umzusetzen.

Das von der University of Florida bereitgestellte J4KSDK, wurde zum Zeitpunkt der Entwicklung noch nicht vollständig auf die Kinect v2 umgestellt. So fehlten zu Beginn der Entwicklung der Integrationsplattform noch weitere Kontextinformationen wie beispielsweise über die Ebene (Boden), das Gesicht des Nutzers (Emotionen, Brille, Mund auf/zu, etc.) und weitere Features der Kinect v2 für Windows. Daher wurden im Rahmen dieser Arbeit weitere Konnektoren zur Middleware entwickelt.

Konkret wird jeweils ein Agent pro Kinect v2-Sensor gestartet. Dies ermöglicht eine hohe Skalierbarkeit beim Einsetzen einer Vielzahl von Kinect v2-Sensoren. Aufgrund des hohen Datendurchsatzes einer Kinect sollten diese auf verschiedene Geräte verteilt werden. Auf weitere technische Rahmenbedingungen des Sensors wurde bereits im Grundprojekt eingegangen [vgl. Ghose (2015)].

Leap Motion Agent

Der Leap Motion Agent wurde im Rahmen der UBICOM Gruppe erstellt und basiert ebenfalls auf der generischen Skeleton API. Der Agent versendet für jeden Frame, in welchem eine Hand (oder Pointable Object) gefunden worden ist, eine Nachricht an die entsprechenden Subscriber des Agenten. Die Einbindung des Agenten kann bestimmten Interaktionsräumen erlauben, wie beispielsweise am Couchtisch oder an der Herdplatte, detailliertere Fingergesten für eine umfangreichere Interaktion zu erfassen. Auf weitere Besonderheiten des Sensors wird bei dem Multimodalen Fusion Agent eingegangen, welcher die Fusion der Kinect v2 und Leap Motion Daten umsetzt.

4.2.2 Omnidirectional Fusion Agent

Das SDK der Kinect v2 ist nur für die Verwendung einer Kinect zurzeit ausgelegt, daher gibt es keine vom Hersteller vorgesehene Möglichkeit, mehrere Kinect-Sensoren miteinander zu verbinden und in einem Weltkoordinatensystem zusammenzuführen. Um dies zu ermöglichen, wurde der Omnidirectional Fusion Agent entwickelt.

Der Omnidirectional Fusion Agent wurde im Rahmen des Vorversuchs (siehe 3.2) zusammen mit Tobias Eichler entwickelt und beschrieben [vgl. Langbehn u. a. (2015)]. Basierend auf der generischen Skeleton API wurden die erkannten Joints (bzw. Skelette) verschieden positionierter Sensoren des gleichen Typs fusioniert, wie beispielsweise in Abbildung 4.3 dargestellt.

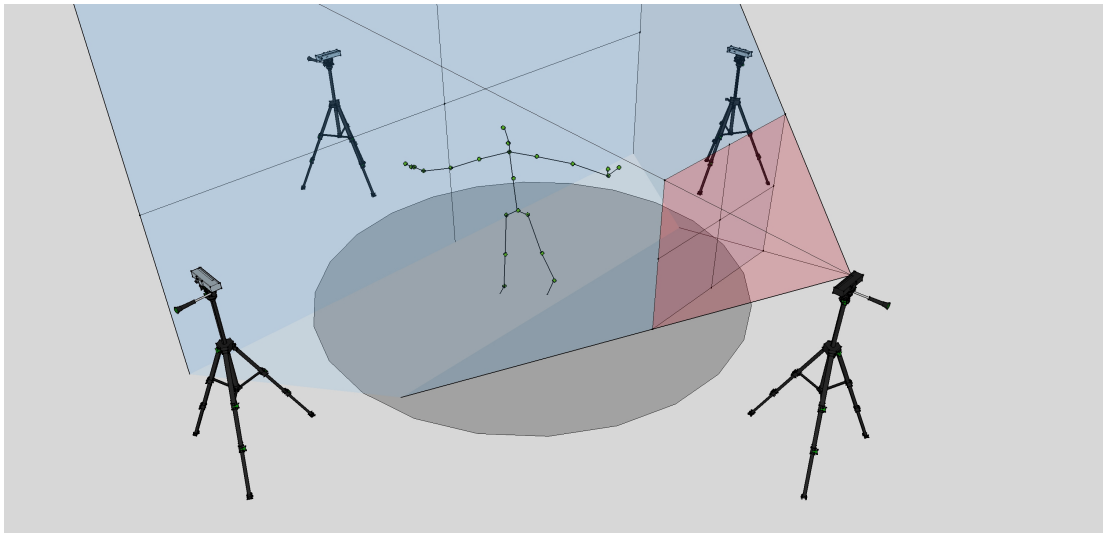


Abbildung 4.3: Laboraufbau: Während der Nutzer sich bewegt, wird er von mehreren Kinects erfasst [Langbehn u. a. (2015)]

Die Fusion der Sensordaten wurde in drei Schritten umgesetzt. Die 3D-Koordinatensysteme der verschiedenen Sensoren wurden durch eine initiale Kalibrierung auf einem Weltkoordinatensystem umgerechnet. Auf Basis der Informationen des Kontext Agent wurde eine Transformation der Ebene des Bodens zum Weltkoordinatensystem berechnet. Die Fusion dieser wurde durch die in der Kalibrierung erstellten Transformationsmatrizen, unter Betrachtung verschiedener Randbedingung, zur Sicherstellung der Validität der Skelette durchgeführt.

Die konkrete Fusion bestand wiederum aus 3 Teilen:

1. Als erstes wurde der Mittelpunkt aller Skelette in einem Weltkoordinatensystem auf einen Punkt gelegt.
2. Daraufhin wurde die Rotation der linearen Gleichungssysteme mittels einer Singulärwertzerlegung (*Singular value decomposition*, kurz SVD) umgesetzt.
3. Abschließend wurden verschiedene Randbedingungen, welche beispielsweise das Zusammenhängen der Joints garantieren sollten, sichergestellt.

Da die Kinect v2 lediglich für die Erkennung von frontal vor dem Sensor stehenden Personen ausgelegt ist, mussten noch weitere Vorbedingungen überprüft werden. Sollte eine Person mehr als 90° bzw. weniger als 270° vor einer Kinect stehen, also rückwärtsgewandt, erkennt dies die Kinect fehlerhaft und geht von einer frontal vor dem Sensor stehenden Personen aus. Um dieses Problem zu lösen, wurde das Skelett-Modell, nach der Transformation in das Weltkoordinatensystem, jeweils einmal korrekt und einmal am Mittelpunkt des Skeletts horizontal gespiegelt, mit dem Skelett des primären Sensors verglichen. Das Skelett-Modell, welches die geringsten Abweichungen von den Joints zu dem primären Skelett-Modell aufwies wurde für die weitere Verarbeitung genutzt.

In Abbildung 4.4 wird dargestellt, wie durch diese Spiegelung der Skelett-Modelle eine Falscherkennung identifiziert wird.

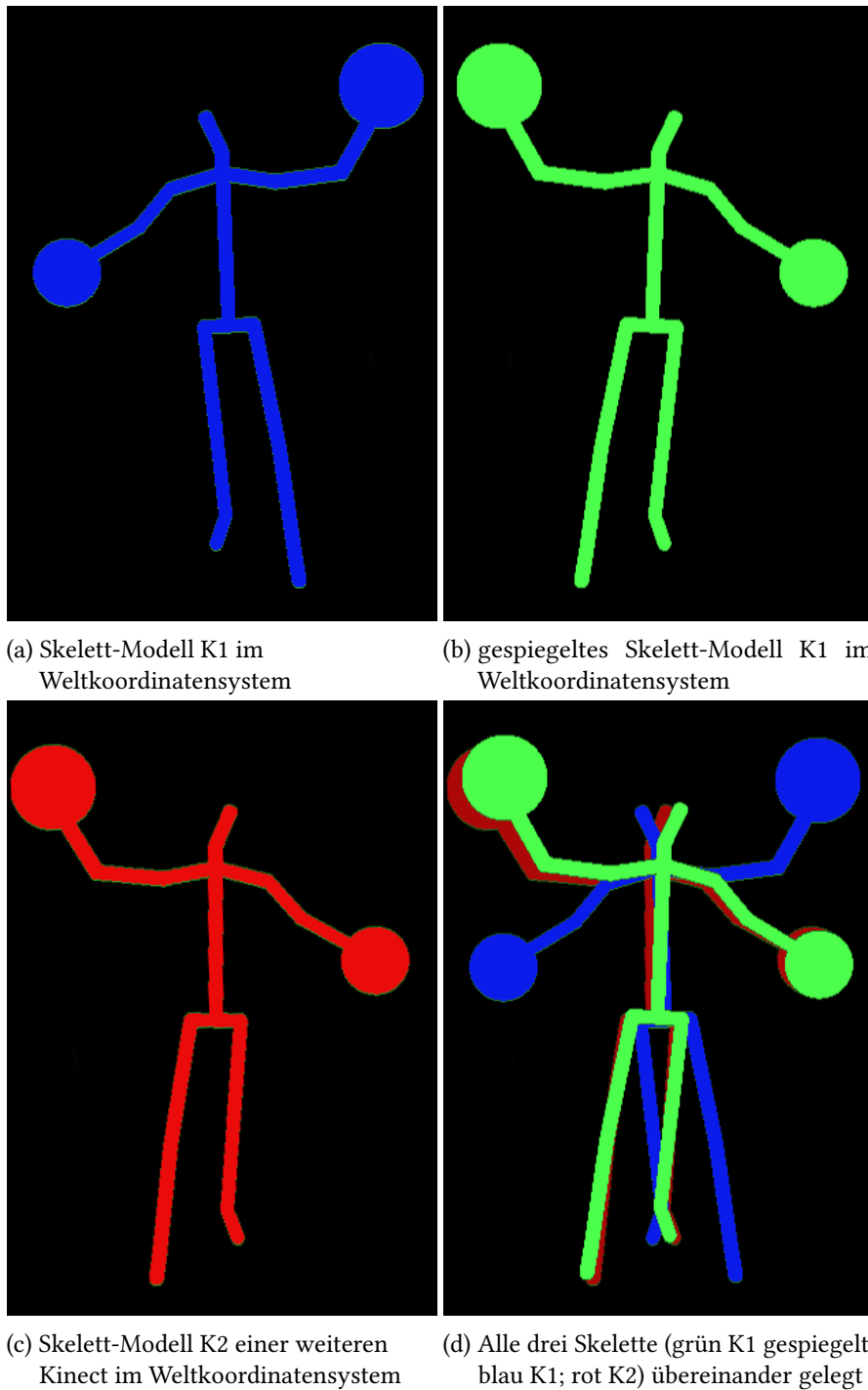


Abbildung 4.4: Der Abstand der Joints des gespiegelten Skeletts (grün) ist zu den Joints des zweiten Skelett-Modells geringer, als der Abstand der Joints des gespiegelten Skeletts. Dies weist darauf hin, dass die Kinect K1 die Person von hinten sieht, dies aber nicht selbstständig erkannt hat.

Umso weniger symmetrisch eine Person steht, indem diese beispielsweise die rechte Hand nach oben und die linke Hand nach unten hält, desto zuverlässiger lässt sich unterscheiden, ob der Sensor die Person richtig erkannt hat.

Eine weitere Möglichkeit zur Validierung der Kinect-Skelette wäre, statt einer einzelnen Frame-Betrachtung, die Drehung der Person über mehrere Frames zu verfolgen, beispielsweise durch Nutzung des Kalman Filters. Ein Ansatz wie dieser wurde beispielhaft von [Gutmann und Nebel \(1997\)](#) umgesetzt. Zudem würde eine fehlerhafte Erkennung der Kinect dazu führen, dass innerhalb von wenigen Frames die Seiten der erkannten Skelette gespiegelt wären.

Ein weiteres Problem bestand darin, dass sich die Zuverlässigkeit der Erkennung einer Person verringert, um so weiter diese rechtwinklig zu einem Kinect Sensor steht. Dies wurde durch eine Gewichtung der verschiedenen Skelett-Modelle in Abhängigkeit des Winkels der Person zu den entsprechenden Sensoren gelöst. Die Kinect, welche die höchste Erkennungsrate und den geringsten Winkel zu der Person aufweist, wurde als primäre Kinect für die Fusion der Skelett-Modelle gewählt.

Der Omnidirectional Fusion Agent wurde konkret für einen Anwendungsfall, bei welchem die Kinect v2 von Microsoft zum Einsatz kam entwickelt. Da dieser aber auf dem generischen Skelett-Modell basiert, kann der Agent auch für weitere Sensoren wie der Leap Motion eingesetzt werden. Anpassungen werden lediglich nötig, sollte der Sensor ähnliche Probleme wie die Kinect bei der Erkennung einer Person von vorne und hinten aufweisen.

4.2.3 Multimodal Fusion Agent

Der Multimodal Fusion Agent verbindet Sensordaten verschiedener Sensoren. Konkret angewendet wird er für die Kinect und die Leap Motion, um ein umfangreicheres und genaueres Skelett-Modell einer Person zum Motion Tracking zu gewinnen. Der Fusion Agent basiert auf einer Umsetzung von [Gutjahr \(2015\)](#), welche im Rahmen der UBICOM Arbeitsgruppe erstellt worden ist. Die Umsetzung der Fusion basiert konkret auf einem Kalman Filter.

Der Fusion Agent soll es Smart Home Anwendungen ermöglichen, die Vorteile beider Sensoren miteinander zu verknüpfen. Die Leap Motion zeichnet sich durch eine hohe Genauigkeit und eine stabile Erkennung von Bewegungen der Hand und der Finger aus, während die Kinect v2 sich für die Erkennung ganzer Skelette bei Entfernungen bis zu 4m bewährt hat.

Die erste Umsetzung des Multimodal Fusion Agent unterliegt zwei Rahmenbedingungen. Es können in der aktuellen Version nur zwei Sensoren miteinander fusioniert werden, eine Fusion mit weiteren Sensoren kann jedoch auf Basis des Omnidirectional Fusion Agent ([4.2.2](#)) umgesetzt werden. Zudem müssen die beiden Sensoren in der aktuellen Implementierung

parallel zueinander auf der x-Achse stehen sowie den Winkel der Sensoren um genau 90° auf der x-Achse gedreht haben (z.B. liegt die Leap Motion nach oben ausgerichtet auf dem Tisch und die Kinect steht horizontal davor).

Die Umsetzung des Kalman Filters für zwei Sensoren wurde folgendermaßen umgesetzt:

Kalman Filter für zwei Sensoren

$$x = x_1 * \frac{\sigma_2^2}{(\sigma_1^2 + \sigma_2^2)} + x_2 * \frac{\sigma_1^2}{(\sigma_1^2 + \sigma_2^2)}$$

- x fusionierter Wert von beiden Sensoren
- x_1 Sensor Wert 1
- x_2 Sensor Wert 2
- σ Standardabweichung
- σ^2 Varianz

Standardabweichung Kinect Für die Berechnung des Kalman Filters für zwei Sensoren wird die Standardabweichung dieser benötigt. Die Standardabweichungen für die Tiefeninformation (3D-Punktwolke der Kinect v2 sind leider nicht vom Hersteller angegeben. Daher wurden zwei wissenschaftliche Studien hinzugezogen, um diese zu bestimmen.

In der Studie [Livingston u. a. \(2014\)](#) wurde eine Standardabweichung von 0,75 mm bei einer Entfernung von 1,2 m sowie eine Standardabweichung von 5,6 mm bei einer Entfernung von 3,5 m angegeben. Die durchschnittliche Standardabweichung aller Messungen von 1 bis 5 m lag bei einer Standardabweichung von 8,1 mm.

In der Studie [Gonzalez-Jorge u. a. \(2015\)](#) wurde die Standardabweichung als eine Formel in Abhängigkeit zur Entfernung angegeben:

$$0.5417 * Z^2 - 0,885 * Z + 2,7083$$

Dabei steht Z für die Entfernung zum Sensor. In Vergleich zur ersten Studie ergibt sich daher bei einer Entfernung von 1,2 m eine Standardabweichung von 2,4 mm und bei den Abstand von 3,5 m eine Standardabweichung von 6,2 mm.

Da der Interaktions-Bereich der Leap Motion gegenüber der Kinect v2 stark eingeschränkt ist, kann von einer maximalen Entfernung von 2 m vom Sensor ausgegangen werden. Daher wurde für den Prototypen eine Standardabweichung von 5 mm festgelegt. Diese könnte in einer Weiterentwicklung des Multimodalen Fusion Agent durch eine Formel in Abhängigkeit zur Entfernung umgesetzt werden.

Standardabweichung Leap Motion Die Standardabweichung der Leap Motion wird vom Hersteller mit 0,001 mm angegeben [vgl. [Leap Motion \(2014\)](#)]. In einem eigenen unabhängigen Versuch konnte der Wert von 0,05 mm bei stillstehenden Objekten und 1,0 mm bei Objekten in Bewegung als Standardabweichung ermittelt werden. Da beim Motion Tracking von bewegten Händen und Fingern ausgegangen wird, wurde für die Standardabweichung ein Wert von 1,00 mm für den Agenten definiert.

4.2.4 Gesture Recognition Toolkit und Agenten

Das Gesture Recognition Toolkit soll es Entwicklern von gestenbasierten Interaktionen für Smart Environments ermöglichen, mit geringem Aufwand und wenig Vorwissen neue Gesten für die Integrationsplattform zu entwickeln und miteinander zu vergleichen.

So ermöglicht das Gesture Recognition Toolkit die Definition von verschiedenem Posen, welche zur Ausführung einer Geste eingenommen werden. Die Definition dieser ist mit geringem Aufwand der Implementierung möglich. Zudem können bereits umgesetzte Posen und Gesten verwendet und weiterentwickelt werden. Diese Posen werden auf Basis des Modells der Skeleton API umgesetzt. Der Entwickler kann angeben, welche Posen aufeinander folgen sowie die Zeitintervalle, für die gesamte Geste oder den Übergang zwischen einzelnen Posen. Eine Pose wird durch die Positionierung und Orientierung verschiedener Joints zueinander oder zum Boden definiert.

Dieser heuristisch basierte Ansatz zur Gestenerkennung ermöglicht die schnelle Entwicklung von neuen Gesten ohne die Erhebung umfangreicher Trainingsdaten, was die Entwicklung und das Testen von Gesten oder Varianten dieser unterstützt.

Die Erkennung der verschiedenen Gesten kann in jeweils einem eigenen Gesture Recognition Agent pro Geste umgesetzt werden. Dies ermöglichen eine hohe Skalierbarkeit des Erkennungssystems und das parallele Ausführen verschiedener Implementationen einer Geste zur gleichen Zeit. Das wiederum ermöglicht den Entwicklern, verschiedene Implementationen der gleichen Geste sowie unterschiedliche Gesten für den gleichen Anwendungsfall zeitnah zu implementieren und zu testen.

Neue Gesten können der Gesture API hinzugefügt werden, welche dann entsprechenden Subscribern, wie beispielsweise dem Interaction Management System, die erfolgreiche Erkennung einer Geste mitteilen. Ebenfalls können über diese Gesture API weitere Attribute einer Geste wie beispielsweise "linke/rechte" Seite oder die Geschwindigkeit der Ausführung mitgeliefert werden.

Die Umsetzung der Gestenerkennung auf Basis eines heuristischen Ansatzes in Kombination mit der Umsetzung einer Geste durch verschiedene Posen, ermöglicht zudem den von **Boetzer (2008)** in der Analyse (Abschnitt 2.3.1) vorgestellten Ansatz: das Erkennen einer Geste bereits bei 75% der Ausführung dieser, um so die Reaktionszeit des Systems zur Gestenerkennung zu verkürzen.

4.2.5 Interaction Management Agent

Der Interaction Management Agent ist die zentrale Steuereinheit der Integrationsplattform. Er nimmt die Interaktion des Nutzers auf und steuert entsprechend das Smart Home an. So wird beispielsweise die konkrete Interpretation der Intention einer erkannten Geste erst vom Interaction Management getroffen. Das Interaction Management weiß um verschiedene Modalitäten des Smart Homes wie beispielsweise die Ansteuerung des Lichts. Aktuell kann das Interaction Management Komponenten und Agenten für Licht, Fenster, Rollos und Gardinen ansteuern. Des Weiteren übernimmt das Interaction Management die Steuerung des Feedbacks für den Nutzer. Ein Feedback kann beispielsweise visuell, per Audio oder durch ein haptisches Feedback umgesetzt werden. In Abschnitt 4.2.6 wird auf den Vibration Agent, welcher durch ein haptisches Feedback dem Nutzer eine Rückmeldung gibt, eingegangen.

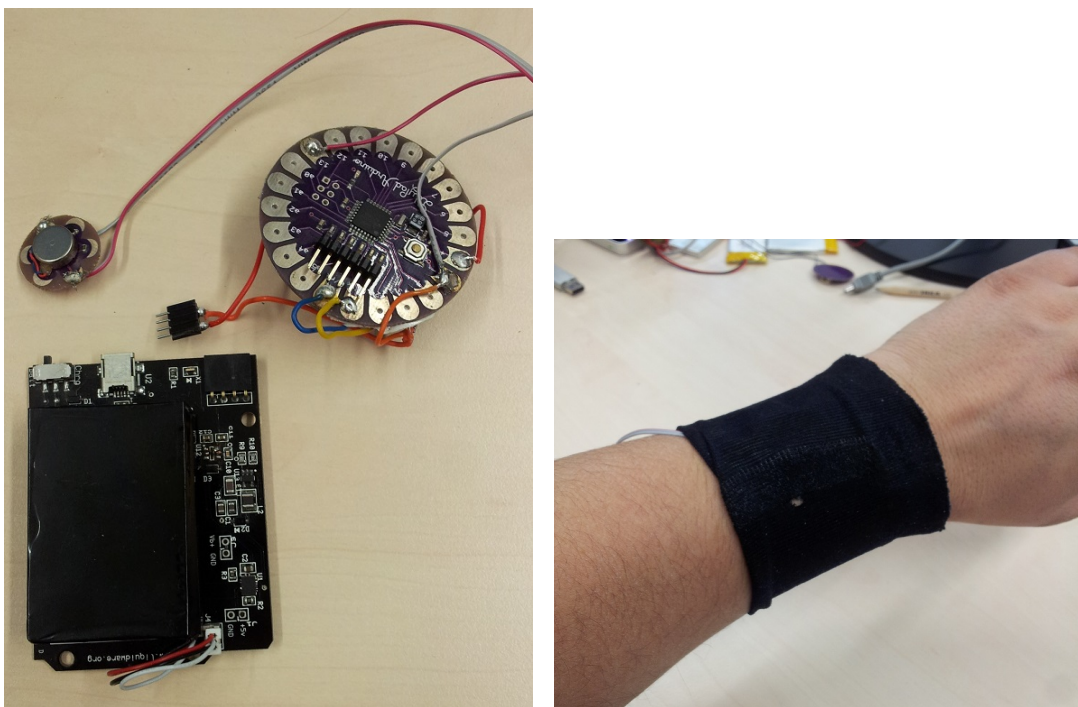
Da zum Zeigen der Machbarkeit einer Integrationsplattform zur Gestenerkennung die Vorhaltung von komplexen oder umstrukturierten Daten und Wissen nicht nötig ist, wurde von der Umsetzung einer umfangreichen eigenständigen Wissensdatenbank abgesehen, da diese im Normalfall auch stark von der aktuellen Anwendung abhängig ist. Die relevanten Informationen wie z.B. Zustand des Lichts in bestimmten Räumen des Smart Homes oder Informationen über den Kontext wurden innerhalb der Interaction Management Komponente umgesetzt. Sollten umfangreichere Informationen für die Interaktion mit dem Nutzer, beispielsweise die Anleitung zur Installation eines bestimmten Gerätes, nötig werden, sollte die Wissensdatenbank in eine eigene Komponente ausgelagert werden.

4.2.6 Vibration Agent

Teile der Infrastruktur des Living Places, wie beispielsweise die Fenster, weisen eine hohe Reaktionszeit bei direkter Ansteuerung auf. Die Ansteuerung dieser Komponenten kann daher die geforderten Reaktionszeiten, welche im Abschnitt 2.10 dargestellt worden sind, nicht erfüllen. Um den Nutzern ein zeitnahes Feedback über das erfolgreiche Ansteuern eines Elements, welches eine hohe Verzögerung aufweist, zu geben, wurde in Anlehnung an den HapRings [vgl.

Oscar u. a. (2015)], welcher die Interaktion im dreidimensionalen Raum mit nicht physischen Objekten durch ein haptisches Feedback unterstützt, ein Vibrations-Armband entwickelt. Außerdem ermöglicht die Interaktion durch das haptische Feedback, dem Nutzer bereits eine Rückmeldung zu geben, sollte dieser durch eine Zeigegeste ein Element der Haustechnik auswählen. Dies wäre mit einem highlighting von Elementen, welche aktuell angesteuert werden, in grafischen Oberflächen vergleichbar.

Mittels eines Arduino LilyPad, XBee und gnu.io wurde ein erster auf Java basierender Prototyp des Vibrations-Armbands umgesetzt (siehe Abb.:4.5).



(a) Prototyp 1 - Xbee Empfänger

(b) Prototyp 1 - Vibrations-Armband

Abbildung 4.5: Sender und Empfänger auf XBee und Arduino Basis

Zur Ansteuerung des drahtlosen Vibrations-Armbands wurde eine API für Vibrationsbefehle sowie ein Agent zur Ansteuerung dieser entwickelt.

Die Vibration API ermöglicht es, Vibrationsmuster zu definieren. Diese bestehen aus zwei Parametern. Der erste Parameter definiert die Länge eines Vibrationssegments. Der zweite Parameter besteht aus einer Liste von Booleans welche definieren ob in dem entsprechenden Segment das Armband vibrieren soll oder nicht.

4 Realisierung

Um beispielsweise das Vibrations-Armband zweimal für 120 ms mit einer Pause von 60 ms vibrieren zu lassen, kann der Befehl folgendermaßen umgesetzt werden:

```
1 ...vibrate.api.VibratePattern(60, Seq(true,true,false,true,true))
```

Der erste Prototyp auf Basis von XBee hat im Living Place nur eine geringe Reichweite und Zuverlässigkeit bei der kabellosen Ansteuerung des Vibrations-Arbands aufgewiesen. Daher wurde ein zweiter Prototyp (siehe Abb.: 4.6) auf Basis des BLE Nano, einem Bluetooth Low Energy Chip von Nordic (Nordic nRF51822 SoC **Semiconductor (o.J)**) umgesetzt. Der BLE Nano ist zum einen wesentlich kleiner als ein Arduino LilyPa und weist gegenüber dem Xbee eine höhere Reichweite und Zuverlässigkeit der drahtlosen Verbindung auf.



Abbildung 4.6: Nordic nRF51822 SoC, Vibrationsmotor und Akku, welche in das Armband gelegt werden können.

4.3 Integration in die Living Place Architektur

Teile der Infrastruktur des Living Place wurden im Rahmen dieser Arbeit erneuert und weiterentwickelt. So wurde beispielsweise der für die Kommunikation des Blackboard genutzte Active MQ durch die von Eichler (2014) entworfene Middleware ersetzt. Diese Middleware wurde bereits im Labor Creative Space for Technical Innovations integriert und weiterentwickelt. Zusätzlich wurde diese ebenfalls als Grundlage der Gestenerkennung und des Datenaustauschs bei dem Vorversuch in Abschnitt 3.2 erprobt.

Zur Entwicklung für die Laborumgebung und auf Basis der bereitgestellten Middleware, wurden u.a. folgende Technologien eingesetzt:

- Git: Versions Control System für große Projekte
- Scala: Funktionale Programmiersprache, welche auf die JVM (JAVA Virtuelle Maschine) aufsetzt
- SBT (Simple Build Tool): Compilierung von Scala sowie Teil des Continuous Integration Prozesses
- Nexus: Als Build Artifact Repository Manager
- Jenkins: Als Continuous Integration Server

Zeitgleich zur Entwicklung der Integrationsplattform wurde die Lichtanlage und Teile der Haustechnik des Living Place erneuert. Dabei wurde die Steuerung des Lichts von einem Phaos Controller² auf einen Raspberry Pi mit einem Dali sowie einem DMX Controller von einem externen Dienstleister umgestellt. Im Gegensatz zum zuvor eingesetzten Phaos Controller, bietet diese Lösung keine integrierte API zur Ansteuerung der Aktoren an. Daher musste in Rahmen der Entwicklung der Integrationsplattform eine API zur Ansteuerung des Lichts sowie der Rollos und Gardinen für das Living Place entwickelt werden, um zu ermöglichen, dass HCI Studien im Smart Home umgesetzt werden können.

Living Place Controller

Zur Ansteuerung der Haustechnik des Living Place, wurde im Rahmen dieser Arbeit eine RESTful API zur Ansteuerung des Lichts, der Gardinen sowie der Rollos umgesetzt sowie ein Web-Interface (siehe Abb.: 4.7) für Tests und manuelle Ansteuerung der API entwickelt.

²<https://www.pharoscontrols.com/>

Zudem wurden die bestehenden Python Bibliotheken zur Ansteuerung des Dali Controllers für warmweiß und des DMX Controllers für RGB-Farben weiterentwickelt.

Living Place - Web Controller



Abbildung 4.7: Web Interface der RESTful API für Licht, Fenster, Gardinen und Rollos

Diese Weiterentwicklung bestand zum einen in der logischen Gruppierung der verschiedenen Lichtquellen zur benutzerfreundlichen Ansteuerung, zum anderen in der Erweiterung der API für den Übergang von verschiedenen Farben. Auf Basis diese Erweiterung wurden einige

beispielhafte Lichtszenen implementiert.

Die RESTful API wurde auf Basis von Flask³ umgesetzt. Die Weboberfläche wurde in die bestehende Infrastruktur des Living Place integriert und löst damit größtenteils die von Eichler (2014) entwickelte Fernbedienung zur Ansteuerung des Living Place ab. Der Living Place Controller zur Ansteuerung der Haustechnik, welcher die bestehenden und weiterentwickelten Python Libraries nutzte und auf der kompakten Hardware eines Raspberry Pi aufsetzte, ist eine Komponente welche nicht als Agent der Middleware umgesetzt worden ist.

4.4 Zusammenfassung der Realisierung

Als Basis der Integrationsplattform konnte mithilfe der Vorstudie Langbehn u. a. (2015) die Middleware (Abschnitt 4.1.2) identifiziert und getestet werden. Basierend auf dieser Middleware sowie der im Design beschriebenen Systemarchitektur (Abschnitt 3.1.2), konnte eine Vielzahl an Agenten (Abschnitt 4.2) als Komponenten der Integrationsplattform für den Anwendungsfall des Smart Homes umgesetzt werden. In Verbindung mit bereits bestehenden Teilen der Infrastruktur des Living Place sowie der Verwendung von weiteren, auf der Middleware basierenden Agenten, konnte ein Gesamtsystem von der Sensorik bis zur Steuerung der Haustechnik (Abschnitte 4.3, 4.3) realisiert werden.

³<http://flask.pocoo.org/>

5 Evaluation

Die grundsätzliche Machbarkeit einer Integrationsplattform für Smart Environments, konnte in Teilen bereits durch die Realisierung gezeigt werden. Im folgenden Kapitel der Evaluation wird auf konkrete Versuche sowie weitere Anforderung an die Integrationsplattform für Studien im HCI-Bereich eingegangen. Dabei ist zu beachten, dass die Nutzer der Integrationsplattform, wie in den Anforderungen beschrieben (Abschnitt 2.10), Entwickler für gestenbasierte Mensch-Computer-Interaktions-Systeme sind.

Konkret wird zuerst auf einen Alpha-Test, in welchem die Zusammenarbeit verschiedener Komponenten evaluiert wird, eingegangen. Im Anschluss wird die Nutzbarkeit der Integrationsplattform für weitere Entwickler durch einen Beta-Test evaluiert. Folgend wird auf die in den Anforderungen (Abschnitt 2.10) definierten, funktionalen Anforderungen eingegangen. Abschließend wird die Eignung der Integrationsplattform zur Nutzung für HCI Studien durch die Durchführung einer solchen evaluiert und diskutiert.

5.1 Alpha-Test

Zur Überprüfung, ob die Entwicklung von gestenbasierter Interaktion mit dem Smart Home auf Basis einer Integrationsplattform grundsätzlich möglich ist, wurden in einem Alpha-Test zwei verschiedene Gesten zur Steuerung der Haustechnik umgesetzt.

Eine erste Geste zur Ansteuerung des Lichts und eine zweite Geste zur Selektion und Manipulation von Fenstern. Zur Ausführung dieses Alpha-Tests sowie den folgenden Umsetzungen einer Gestenerkennung für das Living Place, wurde als Interaktionsbereich das Wohnzimmer mit mehreren Kinect v2 Sensoren ausgestattet.

In Anlehnung an die Metapher einer Begrüßung bzw. des Verabschiedens des Smart Homes, wurde eine Wink-Geste implementiert, welche das Licht an- bzw. ausschaltet. Da die Fenster nur zwei Zustände haben (offen/geschlossen), reicht die Selektion durch das Zeigen mit ausgestreckter Hand auf ein Fenster aus, um den Zustand des Fensters zu ändern.

Da die Ansteuerung der Fenster, unabhängig von der Integrationsplattform, eine hohe Reaktionszeit aufweist, wurde mittels des Vibrations Agent ein User Feedback bei erfolgreicher

Erkennung der Geste umgesetzt.

Durch den Alpha-Test konnte erfolgreich gezeigt werden, dass die Konzeption der im Design vorgestellten Architektur und der in der Realisierung vorgestellten Umsetzung für die Ansteuerung des Smart Homes ideal geeignet ist.

5.2 Beta-Test: Nutzbarkeit für Entwickler

Nach dem erfolgreichen Alpha-Test, wurden für einen Beta-Test zwei Softwareentwicklerinnen, welche keine Vorerfahrung bei der Implementierung von Systemen zur Gestenerkennung hatten, zum Testen der Integrationsplattform hinzugezogen. Beide Entwicklerinnen hatten nach einer kurzen Einführung in das Gesture Toolkit die Aufgabe, die Implementierung von Gesten durchzuführen.

Die erste Entwicklerin hat die Erkennung von zwei verschiedenen Techniken aus dem Taekwondo umgesetzt. Dem Low- sowie dem High-Kick für die rechte und linke Seite.

Die zweite Entwicklerin hat auf Basis der in Abschnitt 2.12 vorgestellten Vorstudie das Erkennen des Klatschens zum An- bzw. Ausschalten des Lichts umgesetzt.

Die Umsetzung der Taekwondo-Techniken verlief reibungslos und schnell. Innerhalb von 60 min konnte eine robuste Erkennung beider Techniken umgesetzt werden, was ohne genauere Vorkenntnisse eine rapide Umsetzung der Gestenerkennung darstellt.

Die logische Umsetzung des Klatschens verlief ebenfalls schnell. Jedoch zeigte sich, dass die Kinect v2, auf deren Basis die Gestenerkennung erfolgte, die Hände, wenn diese beim Klatschen zusammengeführt wurden, nicht stabil erkennen konnte. Dieses Problem konnte auf die relativ hohe Standardabweichung der Kinect v2 (Abschnitt 4.2.3) zurückgeführt werden. Nach Identifizierung des Problems konnte durch Anpassungen der Gestenerkennung das Klatschen erfolgreich erkannt werden.

Beide Beta-Testerinnen gaben an, dass die Implementierung auf Basis des Toolkits nach Einführung effektiv und schnell umgesetzt werden konnte. Die bereits erstellten Gesten und Posen aus dem Alpha-Test waren hilfreich, um das Zusammenspiel der verschiedenen Komponenten zu verdeutlichen.

Trotz der Schwierigkeiten bei der Erkennung des Klatschens zeigte sich, dass die Abstraktion der Sensorebene sich als Vorteil erwiesen hat, da beide Beta-Testerinnen in kurzer Zeit eine Gestenerkennung umsetzen konnten. Dies zeigt weiterhin, dass die Integrationsplattform ebenfalls erfolgreich durch den Beta-Test evaluiert werden konnte.

5.3 Diskussion der Erfüllung der funktionalen Anforderungen

Im Folgenden sollen die expliziten Anforderungen an die Integrationsplattform, welche in der Analyse (Abschnitt 2.10) beschrieben sind, besprochen werden.

Erweiterbarkeit Die leichte Erweiterbarkeit ist durch die lose Koppelung der Agenten untereinander grundsätzlich gegeben. So konnten beispielsweise weitere Agenten wie der Leap Motion Agent (Abschnitt 4.2.1), welche von anderen Mitgliedern der UBICOM Gruppe umgesetzt worden sind, integriert werden. Zudem konnten eigene entwickelte Agenten für die Kontexterkennung oder für weitere Sensoren und Aktoren schnell und einfach umgesetzt und interagiert werden.

Skalierbarkeit und dezentraler Aufbau Die hohe Sensorabdeckung, um einen größeren Interaktionsbereich zu erfassen, setzt einen dezentralen Aufbau sowie eine hohe Skalierbarkeit voraus. Die Erfüllung dieser Anforderungen konnte durch den Vorversuch (Abschnitt 3.2) und der prototypischen Implementierung der HCI Studie (Abschnitt 5.4) gezeigt werden. Die Integrationsplattform kann eine sehr hohe Anzahl von Agenten gleichzeitig ausführen. Sollte die Last auf einem Gerät zu stark sein, können Agenten dieses Geräts auf mehrere Geräte verteilt werden. Dies kann aufgrund der eingesetzten Middleware transparent für die Agenten sowie den Nutzer zur Laufzeit umgesetzt werden [vgl. Eichler (2014)]. Weitere Informationen zur Skalierung des Systems wurden im Abschnitt 4.1.2 behandelt oder sind in dem Test der zugrundeliegenden Middleware beschrieben [vgl. Eichler (2014)].

Reaktionszeit und User Feedback Die grundsätzliche Reaktionszeit der Middleware wurde unter anderem in Langbehn u. a. (2015) und Eichler u. a. (2017) getestet und liegt deutlich unter 100 ms. Die Reaktionszeit des Gesamtsystems ist jedoch nicht nur von den entwickelten Agenten, sondern zusätzlich von verschiedenen Komponenten, die in einigen Fällen von außerhalb eingebunden werden, abhängig. Beispielsweise von der Latenz der Kinect v2 (ca. 80 ms [vgl. Sheridan und Ferrell (1963)]) oder der Reaktionszeit der Fenster und Rollos, welche bei direkter Ansteuerung ohne Integrationsplattform eine Reaktionszeit von mehr als 100 ms aufweisen. Um daraus resultierende Nachteile, wie beispielsweise die Verwirrung von Nutzern bei nicht direkter Ausführung eines Steuerungsbefehls [vgl. Anastasiou u. a. (2013)], zu kompensieren, konnte durch einen Agenten, welcher dem Nutzer durch Vibration ein haptisches Feedback gibt (4.2.6) oder das Erkennen einer Gesten bevor die Bewegung vollständig ausgeführt wurde, eine Reaktion des Systems in unter 100 ms umgesetzt werden.

Portierbarkeit Da die Plattform auf dem offen zur Verfügung stehenden akka.io Framework basiert, ist eine grundsätzliche Portierbarkeit gegeben. Dies zeigt sich beispielsweise dadurch, dass einzelne Agenten der Integrationsplattform bereits im Labor Creative Space for Technical Innovation ¹ eingesetzt worden sind [vgl. Graczyk (2016)].

Für eine umfangreiche Portierung der entwickelten Agenten müssen jedoch die entsprechenden Sensoren und Aktoren zur Verfügung stehen. Durch die leichte Erweiterbarkeit des Systems ist der Aufwand, neue Komponenten wie diese einzubinden, sehr gering.

Datenpersistenz und Monitoring Die zugrundeliegende Middleware der Integrationsplattform ermöglicht ein umfangreiches Monitoring sowie self-healing Funktionalität bei Ausfall eines Agenten. Die Persistenz der Daten wird durch das Speichern der Nachrichten oder das Erstellen von Snapshots des Systems durch die Middleware gewährleistet. Durch die Implementierung verschiedener Agenten sowie durch den Beta-Test und die prototypische Implementierung der HCI Studie (Abschnitt 5.4) konnte gezeigt werden, dass die Entwicklung und das Testen von neuen Agenten oder deren Zusammenspiel durch entsprechende Datenaufnahme und das simulierte Abspielen dieser, stark erleichtert worden ist.

5.4 Gestenbasierte HCI Studie auf Basis der Integrationsplattform

Im Alpha-Test konnte erfolgreich gezeigt werden, dass die Implementierung einer gestenbasierten Ansteuerung der Haustechnik mit Hilfe der Integrationsplattform umgesetzt werden kann. Im anschließenden Beta-Test konnte weiterführend gezeigt werden, dass Entwickler ohne Vorerfahrung im Bereich der Gestenerkennung effektiv eine Gestenerkennung umsetzen können. Daher soll im Folgenden evaluiert werden, ob die Integrationsplattform ebenfalls für die Durchführung von HCI Studien geeignet ist. Zu diesem Zweck wurde in Zusammenarbeit mit Karolina Bernat auf Basis der Integrationsplattform eine HCI Studie konzipiert, umgesetzt und auf dem *IEEE Symposium Series on Computational Intelligence 2017* veröffentlicht [vgl. Bernat u. a. (2017)].

5.4.1 Aufbau der Studie

Für die Entwicklung einer durch *Human-Centered-Development* konzipierten Gestensteuerung ist eine erste explorative Studie, wie beispielsweise ein Wizard-of-Oz-Experiment, zur Erfassung der intuitiven Steuerung von Probanden sinnvoll.

¹<https://csti.haw-hamburg.de/>

Auf Basis der Erkenntnisse der explorativen Nutzerstudie kann eine Identifikation des Interaktionsrepertoires durchgeführt werden.

Dieses Interaktionsrepertoire kann dann mit Hilfe des Gesture-Toolkits implementiert werden. Durch die Verwendung des Interaktion Managements kann die Ansteuerung der Haustechnik durch die in der explorativen Studie identifizierten Gesten umgesetzt werden.

Zur Verifizierung der Implementation und Robustheit sowie der Zuverlässigkeit der Gestenerkennung, kann erneut eine Studie mit Probanden durchgeführt und damit die Entwicklung iterativ fortgeführt werden.

5.4.2 Explorative Vorstudie

Das in der Analyse vorgestellte Wizard-of-Oz-Experiment (Abschnitt 2.12), welche im Living Place durchgeführt worden ist, kann als explorative Vorstudie für die *Human-Centered-Development* Methodik betrachtet werden [vgl. Bernat (2017)]. Aus dieser Vorstudie wurde die Steuerung der Rollos als Beispiel für die Umsetzung der HCI Studie herangezogen. Diese Ansteuerung wird in Abbildung 2.21 dargestellt.

Interaktionsrepertoire

Das Wizard-of-Oz-Experiment hat gezeigt, dass ein Großteil der Probanden (10 von 11, siehe Abb.: 5.1) zum Hochfahren der Rollos eine oder beide Hände gleichmäßig vor dem Körper von unten nach oben bewegen. Zum Herunterfahren der Rollos wird die Geste umgekehrt, dementsprechend eine oder beide Hände gleichmäßig von oben nach unten bewegt.

5.4.3 Implementation des Interaktions Repertoire

Die prototypische Umsetzung eines Agenten zur Gestenerkennung für die Steuerung der Rollos, wurde im Wohnzimmerbereich des Living Place umgesetzt. Die Gestenerkennung zum Hochfahren der Rollos wurde in fünf Posen aufgeteilt:

1. Der Nutzer steht den Rollos zugewandt.
2. Eine oder beide Hände befinden sich unterhalb der Brust.
3. Die entsprechende Hand oder Hände sind auf Höhe der Brust.
4. Die entsprechende Hand oder Hände sind oberhalb der Brust.
5. Die entsprechende Hand oder Hände werden für einen kurzen Moment oberhalb der Brust gehalten.

Close down the roller blinds		Open up the roller blinds	
Gesture	Number of attendees	Gesture	Number of attendees
Slide with one or both arms stretched out before you from above your chest to beneath it. Armes are oriented towards the roller blinds	10	Slide with one or both arms stretched out before you from below your chest to above. Armes are oriented towards the roller blinds	10
Grab in imaginery roller blind cord and pull it down repeatedly	1	Grab in imaginery roller blind cord and pull it down repeatedly	1
Total	11	Total	11

Abbildung 5.1: Tabelle zur präferierten Steuerung des Lichts [Bernat u. a. (2017)]

Nach erfolgreichem Erkennen der Geste, wurden die Rollos hochgefahren. Zudem wurde ein definiertes Zeitintervall gewartet, bevor die Gestenerkennung zum Herunterfahren der Rollos aktiviert wurde, um zu verhindern, dass das Herunternehmen der Hände erneut als Geste erkannt wird und die Rollos wieder herunterfahren.

Die Umsetzung zum Herunterfahren der Rollos besteht aus der umgekehrten Reihenfolge der Posen zum Hochfahren der Rollos.

Weitere Details zur Implementierung können aus Bernin u. a. (2017) und Bernat (2017) entnommen werden.

5.4.4 Validierung der prototypischen Umsetzung

Um die Implementierung sowie die Gestik zu validieren, wurde eine Nutzerstudie mit fünf Probanden durchgeführt. Diese sollten jeweils fünf mal die Rollos öffnen und schließen.



(a) Probandin führt Geste aus, um die Rollos runter zu fahren.



(b) Proband führt Geste aus, um die Rollos hochzufahren.

Abbildung 5.2: Probanden steuern die Rollos im Wohnzimmer des Living Place mittels Gesten auf Basis der Integrationsplattform.

Dementsprechend wurden beide Gesten 25 mal ausgeführt. Beide Gesten wurden jeweils einmal nicht korrekt erkannt, sodass diese wiederholt werden mussten. Dies ergibt eine Erkennungsrate von 96%.

5.4.5 Rückschlüsse der Studie für die Integrationsplattform

Die Entwicklerin, welche die Implementierung der Gestenerkennung durchgeführt hat, hatte mit Ausnahme der Erfahrungen durch den Beta-Test, zuvor noch keine Systeme zur Gestenerkennung entwickelt.

Die Entwicklerin hat die Umsetzung mittels der Integrationsplattform wie folgt beschrieben:

Mit Hilfe der Validierung mit Teilnahme von Testpersonen wurde gezeigt, dass die leichtgewichtige, prototypische Implementierung eines gestenbasierten Interfaces einen schnellen Einstieg in die Entwicklung natürlicher User Interfaces ermöglicht. Das erzielte Ergebnis ist vielversprechend. Es dient ebenfalls als Basis zur weiteren Untersuchungen mit Nutzern, um ein umfangreiches Feedback zur Weiterentwicklung und Verbesserung zu sammeln. Mit dem Test wurde gezeigt, dass sich die ursprünglichen Vorstellungen und Ideen der Benutzer bezüglich einer natürlichen Interaktion mit der Wohnumgebung in kurzer Zeit als Prototyp validieren lassen. Die Umsetzung, die angelehnt an den User-Centered-Design-Ansatz entwickelt wurde, profitiert stark vom direkten User-Feedback.

Bernat (2017)

Auf Basis dieser Studie wurde in Zusammenarbeit eine agile User-Centered-Design-Methode entwickelt, um die Herausforderungen der gestenbasierten Interaktion mit Smart Environments zu meistern. Daher konnte die Studie zeigen, dass die Anforderung in Abschnitt 2.10 erfolgreich umgesetzt werden konnte und der geplante User-Centered-Development-Workflow wurde erfolgreich realisiert.

5.5 Fazit der Evaluierung

Zur Validierung der Integrationsplattform wurde ein vertikaler Architekturschnitt prototypisch implementiert (Kapitel 4).

Durch den ersten Alpha-Test der Plattform sowie den Beta-Test mit weiteren Entwicklern konnte gezeigt werden, dass die leichtgewichtige, prototypische Implementierung eines gestenbasierten Interfaces einen schnellen Einstieg in die Entwicklung natürlicher Interaktion mit einem Smart Home ermöglicht. Das erzielte Ergebnis ist vielversprechend, da es ebenfalls

als Basis für weitere Untersuchungen mit Nutzern, um ein umfangreiches Feedback zur Weiterentwicklung und Verbesserung zu sammeln, dienen kann. Mit dem Test wurde gezeigt, dass sich die ursprünglichen Vorstellungen und Ideen der Benutzer bezüglich einer natürlichen Interaktion mit der Wohnumgebung in kurzer Zeit als Prototyp umsetzen, testen und validieren lassen. Die Umsetzung, die angelehnt an den User-Centered-Design-Ansatz entwickelt wurde, profitiert stark vom direkten User-Feedback.

Ferner konnte eine gestenbasierte HCI Studie auf Basis der Integrationsplattform erfolgreich umgesetzt und eine neue agile Methodik für User-Centered-Design-Ansätze entwickelt und getestet werden [vgl. [Bernat u. a. \(2017\)](#)]. Dies zeigt die erfolgreiche und vielversprechende Umsetzung der Integrationsplattform.

6 Schlussbetrachtung

Diese Arbeit hat sich intensiv mit Fragestellungen der Interaktion mit Smart Environments, vorrangig per Gestik, auseinander gesetzt. Hierzu wurde der aktuelle Forschungsstand zu Gestik, Motiontracking und der 3D-Gestenerkennung analysiert und ein Kernproblem bei der Verwendung und Weiterentwicklung von Systemen zur Gestenerkennung identifiziert. Für diese Herausforderung wurde daher eine Lösung konzipiert, umgesetzt und evaluiert.

6.1 Zusammenfassung

Die Einleitung (Kapitel 1) lieferte die grundsätzliche Einordnung dieser Arbeit in der Informatik und der Mensch-Maschinen-Interaktion. Anschließend wurde auf die Motivation (Abschnitt 1.1) der Rechnerallgegenwärtigkeit sowie der nahtlosen Interaktion eingegangen. Abschließend wurden die Ziele (Abschnitt 1.2) sowie die Betrachtung des aktuellen Forschungsstands für 3D-Gesten-Interaktion mit Smart Environments vorgestellt und eingeführt.

In der Analyse (Kapitel 2) wurde zu Beginn auf die grundsätzliche Fragestellung der Arbeit (Abschnitt 2.1.3) eingegangen. Dabei wurde erläutert, dass die aktuellen Herausforderungen und Lösungen für multimodale Gestensteuerung in Smart Environments das Kernthema der Arbeit bilden.

Im Anschluss wurde die Gestik (Abschnitt 2.2) sowie verschiedene Kategorisierungen und Arten von Gesten (Abschnitt 2.2) zur Steuerung von Computersystemen erläutert. Auf dieser Basis wurden die grundsätzlichen Möglichkeiten der Gestenerkennung (Abschnitt 2.3.1) sowie die hohe Bedeutung des Kontextes des Nutzers und der Aktion diskutiert (Abschnitt 2.4).

Bei den vergleichbaren Arbeiten der Gestenerkennung (Abschnitt 2.5) wurden aktuelle Arbeiten oder Studien zu Gestenerkennung vorgestellt und auf diese im Kontext dieser Arbeit eingegangen. Anschließend wurden die beiden grundsätzlichen Möglichkeiten zur Gestenerkennung durch Verfahren des maschinellen Lernens (Abschnitt 2.6.1) oder heuristischen Verfahren (Abschnitt 2.6.2) erläutert und eine Vielzahl von Beispielen für diese gegeben.

Auf Basis dieser Analyse wurde der aktuelle Forschungsstand zusammengefasst (Abschnitt 2.9) und ein aktuelles Kernprobleme, die Vergleichbarkeit verschiedener Ansätze zur Gestensteuerung, aufgezeigt und diskutiert (Abschnitt 2.9.2). Daraufhin wurde das Thema der Arbeit, die Entwicklung einer Integrationsplattform zur Untersuchung von Gestensteuerung für *Human-Computer-Interaction* Studien in Smart Environments, definiert (Abschnitt 2.10), um eine bessere und effizientere Möglichkeit zum Vergleich verschiedener Sensoren, Algorithmen, Gesten und Ansätze zu ermöglichen.

Im Design (Kapitel 3) wurde auf die grundsätzliche Konzeption wie der Plattformarchitektur (Abschnitt 3.1.1) und der Systemarchitektur des Interaction Managements (Abschnitt 3.1.2) für die Integrationsplattform eingegangen. Zudem wurde durch einen Vorversuch erfolgreich die Basis für die Integrationsplattform getestet und gewählt (Abschnitt 3.2).

In der Realisierung (Kapitel 4) konnte die prototypische Umsetzung der Integrationsplattform erfolgreich vorgestellt und diese mit einem kompletten vertikalen Architekturschnitt durch eine Vielzahl von Agenten (Abschnitt 4.2) implementiert werden. Des Weiteren wurde die erfolgreiche Umsetzung der Integrationsplattform in die Architektur des Forschungslabors Living Place vorgestellt (Abschnitt 4.3).

Zur Evaluierung (Kapitel 5) der Integrationsplattform wurde ein Alpha-Test (Abschnitt 5.1) und ein Beta-Test (Abschnitt 5.2) erfolgreich durchgeführt sowie die Erfüllung der funktionalen Anforderungen vorgestellt und diskutiert (Abschnitt 5.3). Abschließend wurde zur Untersuchung der Integrationsplattform eine vollständige HCI Studie auf Basis der Integrationsplattform zur 3D-Gestensteuerung im Smart Home erfolgreich durchgeführt, ausgewertet und der effektive Nutzen der Integrationsplattform für HCI Studien vorgestellt (Abschnitt 5.4).

6.2 Fazit und Ausblick

Um schnelle Innovationen und eine effektive Fortentwicklung auf diesem Gebiet zu gewährleisten, wurde die Entwicklung von Systemen zur Gestenerkennung auf Basis einer Integrationsplattform vorgestellt. Dies geschah außerdem, um ein agiles und nutzerzentriertes Vorgehen zu ermöglichen und die Vergleichbarkeit verschiedener Ansätze, Gesten, Sensoren und Algorithmen zu ermöglichen.

Mithilfe des Living Place und auf Basis der Integrationsplattform konnte daher zusammen mit

Karolina Bernat eine neue agile User-Centered-Development-Methode entwickelt werden [vgl. [Bernat u. a. \(2017\)](#)]. Auf der Basis dieser neuen Methode aus [Abbildung 6.1](#) konnte eine HCI Studie durchgeführt, evaluiert und veröffentlicht werden (siehe [Abschnitt 5.4.4](#)).

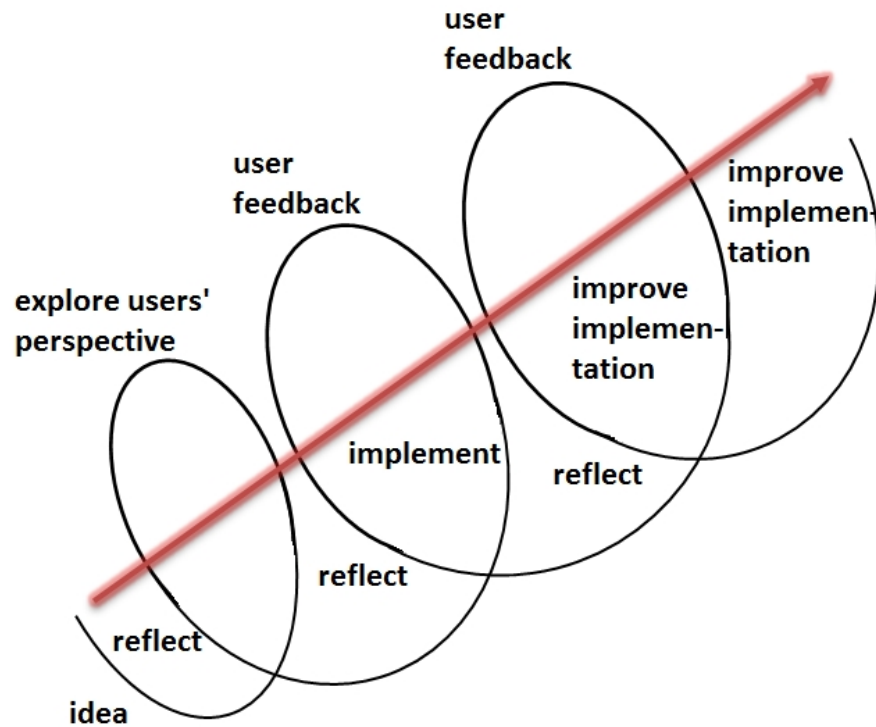


Abbildung 6.1: Agile User-Centered-Development-Methode auf Basis der Integrationsplattform [[Bernat u. a. \(2017\)](#)].

Die Integrationsplattform ermöglicht eine schnelle und iterative Entwicklung von Interaktionskonzepten, deren Implementierung sowie deren Verbesserung.

Das positive Feedback der Entwickler, die schnelle und effektive Entwicklung einer 3D-Gestenerkennung, die leichtgewichtige Durchführung einer HCI Studie, die guten Ergebnisse der ersten Implementierung zur Ansteuerung der Rollos im Living Place sowie die erfolgreiche Veröffentlichung eines effektiven User-Centered-Development-Ansatzes zeigen, dass die gewählte Lösung, die Entwicklung einer Integrationsplattform zur Gestenerkennung in Smart Environments zur Begünstigung von HCI Studien, ein neuartiger und vielversprechender Ansatz ist.

Die Integrationsplattform ist darauf ausgelegt, zukünftig neben der Gestik weitere Modalitäten wie Sprache, Blickrichtung und eine Emotionserkennung zu unterstützen, um die

natürliche Interaktion mit Smart Environments zu verbessern. Eigene Demonstrationen der Ansteuerung von Licht, Fenstern und Rollos per Gestensteuerung bei der Nacht des Wissens¹ und Laborführungen sowie die aktuellen Entwicklungen² der Nutzung von Sprachassistenten wie beispielsweise Alexa³ zeigen, dass der Wunsch nach einer intelligenten Interaktion mit der Umgebung bei vielen Menschen Begeisterung erzeugt. Jedoch bilden jüngste Ereignisse wie beispielsweise Datenlecks⁴ oder die Analyse von Sprachbefehlen von Alexa durch Angestellte bei Amazon⁵ das Risiko, dass die Akzeptanz der Nutzer gegenüber solchen Systemen auch schnell abnehmen kann.

Dies verdeutlicht, dass zum Erreichen einer positiven Zukunftsvision für intelligente und rechnerallgegenwärtige Umgebungen, nicht nur technische sondern auch gesellschaftliche Herausforderungen gelöst werden müssen. Eine Vision, wie technische Systeme dazu angelegt sein müssen, hat bereits der *SFB/TTR 62*, welcher sich intensiv mit der multimodalen Interaktion auseinandersetzt⁶, in einer Vision zu Companion Systemen, beschrieben:

“Technische Systeme der Zukunft sind Companion-Systeme - kognitive technische Systeme, die ihre Funktionalität vollkommen individuell auf den jeweiligen Nutzer abstimmen: Sie orientieren sich an seinen Fähigkeiten, Vorlieben, Anforderungen und aktuellen Bedürfnissen und stellen sich auf seine Situation und emotionale Befindlichkeit ein. Sie sind stets verfügbar, kooperativ und vertrauenswürdig und treten ihrem Nutzer als kompetente und partnerschaftliche Dienstleister gegenüber.”

Biundo und Wendemuth (2010b)

Dies kann als Weiterentwicklung der von **Weiser (1991)** entwickelten Vision aufgefasst werden und gibt Hinweise darauf, dass für die Fortentwicklung von Ubiquitous Computing Systemen eine Individualisierung, Kooperativität sowie Vertrauenswürdigkeit der Systeme sowie eine multimodale Basis zur Entwicklung dieser nötig ist. Dies kann auch eine Weiterentwicklung unserer Vorstellung der Interaktion mit Smart Homes bzw. Environments bedeuten und hin zu einer bewussten, intentionaleren Interaktion mit unser Umgebung führen. Die

¹<https://www.hamburg.de/weiterbildung/4117128/nacht-des-wissens/>

²<https://www.heise.de/newsticker/meldung/Fast-jeder-Dritte-in-Deutschland-nutzt-Sprachassistenten-4443365.html>

³<https://www.amazon.de/b?node=12775495031>

⁴<https://www.heise.de/newsticker/meldung/Amazon-gibt-intime-Sprachdateien-preis-4254716.html>

⁵<https://www.heise.de/newsticker/meldung/Amazon-Mitarbeiter-tippen-zum-Teil-Alexa-Sprachbefehle-ab-4374871.html>

⁶<https://www.uni-ulm.de/in/sonderforschungsbereich-transregio-62/research/projects/project-area-b/project-b3/>

konkrete Ausgestaltung der Interaktionsmodalitäten für zukünftige Systeme lässt sich aktuell nur schwer erahnen, daher sind weitere Studien und Forschungsprojekte erforderlich. Die entworfene agile, nutzerzentrierte Methode sowie die erstellte Integrationsplattform bieten eine neue Möglichkeit, sich dieser weiterentwickelten Vision Schritt für Schritt zu nähern.

Literaturverzeichnis

- [ACM 2019] ACM, Association for Computing M.: *ACM Gesture Search*. 2019. – Online erreichbar unter <https://dl.acm.org/results.cfm?query=gesture>; zuletzt besucht am 05.04.2019.
- [Affectiva 2017] AFFECTIVA: *What if technology could adapt to human emotion?* 2017. – Online erreichbar unter <http://www.affectiva.com/>; zuletzt besucht am 06.02.2017.
- [Anastasiou u. a. 2013] ANASTASIOU, Dimitra ; JIAN, Cui ; STAHL, Christoph: *A German-Chinese Speech-gesture Behavioural Corpus of Device Control in a Smart Home*. 2013. – URL <http://doi.acm.org/10.1145/2504335.2504402>
- [Anastasiou u. a. 2012] ANASTASIOU, Dimitra ; JIAN, Cui ; ZHEKOVA, Desislava: *Speech and Gesture Interaction in an Ambient Assisted Living Lab*. 2012. – URL <http://dl.acm.org/citation.cfm?id=2392866.2392870>
- [Aviezer u. a. 2008] AVIEZER, Hillel ; HASSIN, Ran R. ; RYAN, Jennifer ; GRADY, Cheryl ; SUSSKIND, Josh ; ANDERSON, Adam ; MOSCOVITCH, Morris ; BENTIN, Shlomo: Angry, disgusted, or afraid? Studies on the malleability of emotion perception. In: *Psychological Science* 19 (2008), Nr. 7, S. 724–732
- [Benyon und Mival 2010] BENYON, David ; MIVAL, Oli: From Human-computer Interactions to Human-companion Relationships. In: *Proceedings of the First International Conference on Intelligent Interactive Technologies and Multimedia*. New York, NY, USA : ACM, 2010 (IITM '10), S. 1–9. – URL <http://doi.acm.org/10.1145/1963564.1963565>. – ISBN 978-1-4503-0408-5
- [Bernat u. a. 2017] BERNAT, K. ; GHOSE, S. ; LUCK, K. von ; VOGT, F.: *A Method for an Agile, User Centered Development of Natural User Interfaces*. SSCI17. 2017. – Online erreichbar unter <https://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/papers/SSCI17b.pdf>; zuletzt besucht am 11.05.2019.

- [Bernat 2015] BERNAT, Karolina: *Entwicklung einer Gestensteuerung in einer Smart-Home Umgebung*. Website. 2015. – Online erreichbar unter <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2015-proj/bernat.pdf>; zuletzt besucht am 06.03.2019.
- [Bernat 2017] BERNAT, Karolina: *Konzeption und Nutzerevaluation von Gestensteuerung in Smart-Home Umgebung*. website. 2017. – Online erreichbar unter <https://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/master/bernat.pdf>; zuletzt besucht am 11.01.2017.
- [Bernin u. a. 2017] BERNIN, Arne ; MÜLLER, Larissa ; GHOSE, Sobin ; LUCK, Kai von ; GRECOS, Christos ; WANG, Qi ; VOGT, Florian: *Towards More Robust Automatic Facial Expression Recognition in Smart Environments*. 2017. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/papers/petra2017.pdf>. – online erreichbar unter <http://www.haw-hamburg.de/index.php?id=32258l> Stand 17.02.2018
- [Biundo und Wendemuth 2010a] BIUNDO, Susanne ; WENDEMUTH, Andreas: *Sonderforschungsbereich zur Innovation im Mensch-Technik Dialog: Companion- Systeme*. 2010. – URL https://www.uni-ulm.de/fileadmin/website_uni_ulm/iui.sfbtrr62/Presseinformationen/SFB-Uebersicht.pdf
- [Biundo und Wendemuth 2010b] BIUNDO, Susanne ; WENDEMUTH, Andreas: Von kognitiven technischen Systemen zu Companion-Systemen. In: *KI - Künstliche Intelligenz* 24 (2010), Nr. 4, S. 335–339. – URL <http://dx.doi.org/10.1007/s13218-010-0056-9>. – ISSN 0933-1875
- [Boetzer u. a. 2008] BOETZER, J. ; RAHIMI, M. ; VOGT, M. ; WENDT, P. ; K.V.LUCK: *Gestenbasierte Interaktion mit Hilfe von Multitouch und Motiontracking*. Website. 2008. – Online erreichbar unter <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/papers/WIWITA2008.pdf>
- [Boetzer 2008] BOETZER, Joachim: *Bewegungs- und gestenbasierte Applikationssteuerung auf Basis eines Motion Trackers*. Website. 2008. – Online erreichbar unter <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/bachelor/boetzer.pdf>; zuletzt besucht am 23.10.2019.
- [Bohus u. a. 2017] BOHUS, Dan ; ANDRIST, Sean ; JALOBEANU, Mihai: Rapid development of multimodal interactive systems: a demonstration of platform for situated intelligence, 11 2017, S. 493–494

- [Cassell 1998] CASSELL, Justine: *A Framework For Gesture Generation And Interpretation: Computer Vision in Human-Machine Interaction*; Cambridge University Press. Website. 1998. – Online erreichbar unter http://www.media.mit.edu/gnl/publications/gesture_wkshop.pdf; zuletzt besucht am 25.07.2019.
- [Chen u. a. 2003] CHEN, Feng-Sheng ; FU, Chih-Ming ; HUANG, Chung-Lin: Hand gesture recognition using a real-time tracking method and hidden Markov models. In: *Image and Vision Computing* 21 (2003), 08, S. 745–758
- [Costa und Duarte 2011] COSTA, David ; DUARTE, Carlos: Adapting Multimodal Fission to Users Abilities. In: STEPHANIDIS, Constantine (Hrsg.): *Universal Access in Human-Computer Interaction. Design for All and Inclusion* Bd. 6765. Springer Berlin Heidelberg, 2011, S. 347–356. – URL http://dx.doi.org/10.1007/978-3-642-21672-5_38. – ISBN 978-3-642-21671-8
- [Criminisi u. a. 2012] CRIMINISI, Antonio ; SHOTTON, Jamie ; KONUKOGLU, Ender: Decision Forests: A Unified Framework for Classification, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning. In: *Foundations and Trends in Computer Graphics and Vision: Vol. 7: No 2-3, pp 81-227* (2012). – URL <http://research.microsoft.com/apps/pubs/default.aspx?id=158806>
- [Dan u. a. 2012] DAN, L. ; EKENEL, H. K. ; JUN, O.: Human Gesture Analysis Using Multimodal Features. In: *2012 IEEE International Conference on Multimedia and Expo Workshops*, July 2012, S. 471–476
- [Devineau u. a. 2018] DEVINEAU, G. ; MOUTARDE, F. ; XI, W. ; YANG, J.: Deep Learning for Hand Gesture Recognition on Skeletal Data. In: *2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018)*, May 2018, S. 106–113
- [Dorau 2011] DORAU, Rainer: *Gesten in der Mensch- System-Interaktion*. S. 14–53. In: *Emotionales Interaktionsdesign: Gesten und Mimik interaktiver Systeme*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2011. – URL https://doi.org/10.1007/978-3-642-03101-4_1. – ISBN 978-3-642-03101-4
- [Dourish 2004] DOURISH, Paul: What We Talk About when We Talk About Context. In: *Personal Ubiquitous Comput.* 8 (2004), Februar, Nr. 1, S. 19–30. – URL <http://dx.doi.org/10.1007/s00779-003-0253-8>. – ISSN 1617-4909
- [Eichler 2014] EICHLER, Tobias: *Agentenbasierte Middleware zur Entwicklerunterstützung in einem Smart-Home-Labor*. Website. 2014. – Online erreichbar unter

- <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/master/eichler.pdf>"; zuletzt besucht am 22.04.2019.
- [Eichler u. a. 2017] EICHLER, Tobias ; DRAHEIM, Susanne ; GRECOS, Christos ; WANG, Qi ; LUCK, Kai von: *Scalable Context-Aware Development Infrastructure for Interactive Systems in Smart Environments*. PCAM17. 2017. – Online erreichbar unter <https://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/papers/PCAM17.pdf>"; zuletzt besucht am 22.04.2019.
- [Ekman 1988] EKMAN, P.: *Gesichtsdruck und Gefühl: 20 Jahre Forschung von Paul Ekman*. Junfermann-Verlag, 1988 (Reihe innovative Psychotherapie und Humanwissenschaften). – URL <http://books.google.de/books?id=HUWzAgAACAAJ>. – ISBN 9783873872806
- [Ekman und Friesen 1978] EKMAN, P. ; FRIESEN, W.: *Facial Action Coding System: A Technique for the Measurement of Facial Movement*. 1978. – Online erreichbar unter <https://www.paulekman.com/wp-content/uploads/2013/07/Facial-Sign-Of-Emotional-Experience.pdf>; zuletzt besucht am 10.02.2018.
- [Ekman 2005] EKMAN, Paul: *What the Face Reveals: Basic and Applied Studies of Spontaneous Expression Using the Facial Action Coding System (FACS) (Series in Affective Science)*. Oxford University Press, 2005. – ISBN ISBN 0-19-517964-1
- [Evan-Amos o.J.] EVAN-AMOS: *Xbox-One-Kinect*. Website. o.J. – Gemeinfrei, online erreichbar unter <https://commons.wikimedia.org/w/index.php?curid=33217678>; zuletzt besucht am 18.01.2019.
- [Fleer und Leichsenring 2012] FLEER, David ; LEICHSENRING, Christian: *MISO: A Context-sensitive Multimodal Interface for Smart Objects Based on Hand Gestures and Finger Snaps*. 2012. – URL <http://doi.acm.org/10.1145/2380296.2380338>
- [Fraunhofer Institute for Integrated Circuits 2011] FRAUNHOFER INSTITUTE FOR INTEGRATED CIRCUITS: *SHORE Sophisticated High-speed Object Recognition Engine*. 2011. – Online erreichbar unter <http://www.iis.fraunhofer.de/en/bf/bsy/produkte/shore.html>; zuletzt besucht am 02.07.2014.
- [Gelderblom u. a. 2010] GELDERBLOM, Gert J. ; BEMELMANS, Roger ; SPIERTS, Nadine ; JONKER, Pieter ; DE WITTE, Luc: *Development of PARO Interventions for Dementia Patients in*

- Dutch Psycho-geriatric Care. In: *Proceedings of the Second International Conference on Social Robotics*. Berlin, Heidelberg : Springer-Verlag, 2010 (ICSR'10), S. 253–258. – URL <http://dl.acm.org/citation.cfm?id=1948708.1948739>. – ISBN 3-642-17247-4, 978-3-642-17247-2
- [Ghose 2014] GHOSE, Sobin: *Konzeption und Evaluation eines interaktiven Badezimmer-spiegels*. Website. 2014. – Online erreichbar unter <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/bachelor/ghose.pdf>; zuletzt besucht am 15.5.2019.
- [Ghose 2015] GHOSE, Sobin: *Projekt 1 Ausarbeitung - Multimodale Haussteuerung*. Website. 2015. – Online erreichbar unter <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2015-proj/ghose.pdf>; besucht am 13.02.2019.
- [Gonzalez-Jorge u. a. 2015] GONZALEZ-JORGE, H. ; RODRÍGUEZ-GONZÁLVEZ, P. ; MARTÍNEZ-SÁNCHEZ, J. ; GONZÁLEZ-AGUILERA, D. ; ARIAS, P. ; GESTO, M. ; DÍAZ-VILARIÑO, L.: Metrological comparison between Kinect I and Kinect {II} sensors. In: *Measurement* 70 (2015), Nr. 0, S. 21 – 26. – URL <http://www.sciencedirect.com/science/article/pii/S0263224115001888>. – ISSN 0263-2241
- [Graczyk 2016] GRACZYK, Krystian: *Eignungsuntersuchung von 2D und 3D Gesten für Kollaboration*. Website. 2016. – Online erreichbar unter <https://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/bachelor/graczyk.pdf>; zuletzt besucht am 15.5.2019.
- [Gustafson u. a. 2010] GUSTAFSON, Sean ; BIERWIRTH, Daniel ; BAUDISCH, Patrick: *Imaginary Interfaces: Spatial Interaction with Empty Hands and Without Visual Feedback*. 2010. – URL <http://doi.acm.org/10.1145/1866029.1866033>
- [Gutjahr 2015] GUTJAHR, Christian: *Modellbasierte Sensorfusion von kamera-basierter Bewegungsverfolgung*. Website. 2015. – Online erreichbar unter <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/bachelor/gutjahr.pdf>; zuletzt besucht am 18.04.2019.
- [Gutmann und Nebel 1997] GUTMANN, Jens-Steffen ; NEBEL, Bernhard: Navigation Mobiler Roboter Mit Laserscans. In: *Autonome Mobile Systeme 1997, 13. Fachgespräch*. London, UK, UK : Springer-Verlag, 1997, S. 36–47. – URL <http://dl.acm.org/citation.cfm?id=647723.733855>. – ISBN 3-540-63513-0

- [H Nii 1986] H NII, Penny: The Blackboard Model of Problem Solving and the Evolution of Blackboard Architectures, 06 1986, S. 38–53
- [Hewitt u. a. 1973] HEWITT, Carl ; BISHOP, Peter ; STEIGER, Richard: A Universal Modular ACTOR Formalism for Artificial Intelligence. In: *Proceedings of the 3rd International Joint Conference on Artificial Intelligence*, Morgan Kaufmann Publishers Inc., 1973 (IJCAI'73), S. 235–245. – URL <http://dl.acm.org/citation.cfm?id=1624775.1624804>
- [Hilliges u. a. 2012] HILLIGES, Otmar ; KIM, David ; IZADI, Shahram ; WEISS, Malte ; WILSON, Andrew: HoloDesk: Direct 3D Interactions with a Situated See-through Display. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA : ACM, 2012 (CHI '12), S. 2421–2430. – URL <http://doi.acm.org/10.1145/2207676.2208405>. – ISBN 978-1-4503-1015-4
- [Hoffman u. a. 2010] HOFFMAN, Michael ; VARCHOLIK, Paul ; J. LAVIOLA, Joseph: Breaking the status quo: Improving 3D gesture recognition with spatially convenient input devices, 04 2010, S. 59 – 66
- [Honold u. a. 2012a] HONOLD, Frank ; SCHÜSSEL, Felix ; NOTHDURFT, Florian ; KURZOK, Peter: Companion Technology for Multimodal Interaction. In: *Proceedings of the 14th ACM International Conference on Multimodal Interaction*. New York, NY, USA : ACM, 2012 (ICMI '12), S. 67–68. – URL <http://doi.acm.org/10.1145/2388676.2388696>. – ISBN 978-1-4503-1467-1
- [Honold u. a. 2012b] HONOLD, Frank ; SCHÜSSEL, Felix ; NOTHDURFT, Florian ; KURZOK, Peter: *Companion Technology for Multimodal Interaction*. 2012. – URL <http://doi.acm.org/10.1145/2388676.2388696>
- [Huang u. a. 2009] HUANG, D. ; HU, W. ; CHANG, S.: Vision-Based Hand Gesture Recognition Using PCA+Gabor Filters and SVM. In: *2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, Sep. 2009, S. 1–4
- [Ick-Dietl 2012] ICK-DIETL, Carmen: Die Robbe Paro sorgt für leuchtende Augen, URL <http://www.hallo-muenchen.de/muenchen/m-ost/ramersdorf-perlach/hv-die-robbe-paro-sorgt-fuer-leuchtende-augen-2616451.html>, 2012
- [Karstaedt 2012] KARSTAEDT, Bastian: *Kontextinterpretation in Smart Homes auf Basis semantischer 3D Gebäudemodelle*. Website. 2012. – Online erreichbar unter

- <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/master/karstaedt.pdf>"; zuletzt besucht am 29.3.2015.
- [Knigge o.D] KNIGGE: *Business-Knigge - China*. o.D. – URL <http://www.knigge.de/archiv/artikel/der-auslands-knigge-7078.htm>. – online erreichbar unter <http://www.knigge.de/archiv/artikel/der-auslands-knigge-7078.htm> Stand 18.06.2019
- [Kulshreshth und LaViola 2014] KULSHRESHTH, Arun ; LAVIOLA, Joseph J.: Exploring the Usefulness of Finger-based 3D Gesture Menu Selection. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA : ACM, 2014 (CHI '14), S. 1093–1102. – URL <http://doi.acm.org/10.1145/2556288.2557122>. – ISBN 978-1-4503-2473-1
- [Labs 2014] LABS, Thalmic: *Myo Gesture Control Amrband*. 2014. – Online erreichbar unter <https://www.myo.com/>; zuletzt besucht am 06.02.2017.
- [Lafferty u. a. 2002] LAFFERTY, John ; MCCALLUM, Andrew ; PEREIRA, Fernando: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: *Proc ICML (2002)*, 01
- [Langbehn u. a. 2015] LANGBEHN, Eike ; EICHLER, Tobias ; GHOSE, Sobin ; LUCK, GerdBruder Kai von ; STEINKE, Frank: Evaluation of an Omnidirectional Walking-in-Place User Interface with Virtual Locomotion Speed Scaled by Forward Leaning Angle, 2015
- [LaViola und J. 2014] LAVIOLA, Jr. ; J., Joseph: *An Introduction to 3D Gestural Interfaces*. New York, NY, USA : ACM, 2014 (SIGGRAPH '14). – 25:1–25:42 S. – URL <http://doi.acm.org/10.1145/2614028.2615424>. – ISBN 978-1-4503-2962-0
- [Leap Motion 2014] LEAP MOTION, Inc.: *This is precisely about accuracy*. Website. 2014. – Online erreichbar unter <https://www.leapmotion.com/product>; zuletzt besucht am 10.6.2019.
- [Leap Motion 2016] LEAP MOTION, Inc: *Developer Portal*. Website. 2016. – Online erreichbar unter <https://developer.leapmotion.com/>; zuletzt besucht am 18.07.2016.
- [Lee u. a. 2013] LEE, Jinha ; OLWAL, Alex ; ISHII, Hiroshi ; BOULANGER, Cati: SpaceTop: Integrating 2D and Spatial 3D Interactions in a See-through Desktop Environment. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA : ACM, 2013 (CHI '13), S. 189–192. – URL <http://doi.acm.org/10.1145/2470654.2470680>. – ISBN 978-1-4503-1899-0

- [Livingston u. a. 2014] LIVINGSTON, Mark A. ; SEBASTIAN, Jay ; AI, Zhuming ; DECKER, Jonathan W.: *Performance Measurements for the Microsoft Kinect Skeleton*. Website. 2014. – Online erreichbar unter http://www.nrl.navy.mil/itd/imda/sites/www.nrl.navy.mil.itd.imda/files/pdfs/2012_VRposter_kinectSkelPerf.pdf zuletzt besucht am 27.04.2015.
- [Maghoumi und LaViola 2018] MAGHOUMI, Mehran ; LAVIOLA, Joseph J.: DeepGRU: Deep Gesture Recognition Utility. In: *CoRR* abs/1810.12514 (2018)
- [Maurizio u. a. 2011] MAURIZIO, Caon ; YONG, Yue ; CARSTEN, Maple ; ELENA, Mugellini ; OMAR ABOU, Khaled ; JULIEN, Tscherrig: *Advanced System for Gesture and Activity Recognition and Detection*. 2011. – Online erreichbar unter <https://project.eia-fr.ch/asgard/Pages/Overview.aspx>; zuletzt besucht am 20.05.2014
- [McNeill 1992] MCNEILL, D.: *Hand and Mind: What Gestures Reveal about Thought*. University of Chicago Press, 1992. – URL <https://books.google.de/books?id=3ZZAfNumLvwC>. – ISBN 9780226561325
- [META 2013] META: *The First Holographic Interface*. 2013. – Online erreichbar unter <https://www.metavision.com/>; zuletzt besucht am 06.02.2017.
- [Microsoft 2016] MICROSOFT: *Meet Kinect for Windows*. Website. 2016. – Online erreichbar unter <https://www.microsoft.com/en-us/kinectforwindows/>; zuletzt besucht am 18.07.2016.
- [Microsoft 2017a] MICROSOFT: *JointType Enumeration*. 2017. – Online erreichbar unter <https://msdn.microsoft.com/en-us/library/microsoft.kinect.jointtype.aspx> Stand 02.02.2019
- [Microsoft 2017b] MICROSOFT: *Microsoft HoloLens*. 2017. – Online erreichbar unter <https://www.microsoft.com/microsoft-hololens/en-us>; zuletzt besucht am 06.02.2017.
- [Miller 1968] MILLER, Robert B.: Response Time in Man-computer Conversational Transactions. In: *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I*. New York, NY, USA : ACM, 1968 (AFIPS '68 (Fall, part I)), S. 267–277. – URL <http://doi.acm.org/10.1145/1476589.1476628>
- [Motion 2013] MOTION, Leap: *Understanding Latency: Part 1*. 2013. – URL <http://blog.leapmotion.com/understanding-latency-part-1/>. – online erreichbar unter <http://blog.leapmotion.com/understanding-latency-part-1/> Stand 18.06.2019

- [Müller 2013] MÜLLER, Larissa: Emotionale Modellierung in Mensch-Maschine-Interaktionen, 2013. – Online erreichbar unter <http://users.informatik.haw-hamburg.de/ubicomp/arbeiten/master/mueller.pdf> stand 18.02.2015
- [Nesselrath u. a. 2016] NESSELRATH, R. ; MONIRI, M. M. ; FELD, M.: Combining Speech, Gaze, and Micro-gestures for the Multimodal Control of In-Car Functions. In: *2016 12th International Conference on Intelligent Environments*, Sep. 2016, S. 190–193. – ISSN 2472-7571
- [Oscar u. a. 2015] OSCAR, Ariza ; PAUL, Lubos ; FRANK, Steinicke: *HapRing: A Wearable Haptic Device for 3D Interaction*. 2015. – URL <http://basilic.informatik.uni-hamburg.de/Publications/2015/ALS15>
- [Pavlovic u. a. 1997] PAVLOVIC, V.I. ; SHARMA, R. ; HUANG, T.S.: Visual interpretation of hand gestures for human-computer interaction: a review. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 19 (1997), Nr. 7, S. 677–695. – ISSN 0162-8828
- [Plaisant u. a. 2000] PLAISANT, Catherine ; DRUIN, Allison ; LATHAN, Corinna ; DAKHANE, Kapil ; EDWARDS, Kris ; VICE, Jack M. ; MONTEMAYOR, Jaime: A Storytelling Robot for Pediatric Rehabilitation. In: *Proceedings of the Fourth International ACM Conference on Assistive Technologies*. New York, NY, USA : ACM, 2000 (Assets '00), S. 50–55. – URL <http://doi.acm.org/10.1145/354324.354338>. – ISBN 1-58113-313-8
- [Potratz 2011] POTRATZ, Olaf: *Ein System zur physikbasierten Interpretation von Gesten im 3D-Raum*. Website. 2011. – Online erreichbar unter <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/bachelor/potratz.pdf>; zuletzt besucht am 20.8.2018.
- [Reddy und Basir 2010] REDDY, B. S. ; BASIR, Otman A.: Concept-based Evidential Reasoning for Multimodal Fusion in Human-computer Interaction. In: *Appl. Soft Comput.* 10 (2010), März, Nr. 2, S. 567–577. – URL <http://dx.doi.org/10.1016/j.asoc.2009.08.026>. – ISSN 1568-4946
- [Rousseau u. a. 2006] ROUSSEAU, Cyril ; BELLIK, Yacine ; VERNIER, Frédéric ; BAZALGETTE, Didier: A framework for the intelligent multimodal presentation of information. In: *Signal Processing* 86 (2006), Nr. 12, S. 3696 – 3713. – URL <http://www.sciencedirect.com/science/article/pii/S0165168406001423>. – Special Section: Multimodal Human-Computer Interfaces. – ISSN 0165-1684

- [Semiconductor o.J] SEMICONDUCTOR, Nordic: *nRF51822*. Website. o.J. – Online erreichbar unter <https://www.nordicsemi.com/eng/Products/Bluetooth-low-energy/nRF51822>; zuletzt besucht am 01.11.2016.
- [Shafer 1992] SHAFER, Glenn: Dempster-shafer theory. In: *Encyclopedia of artificial intelligence* 1 (1992), S. 330–331
- [Sheridan und Ferrell 1963] SHERIDAN, T. B. ; FERRELL, W. R.: Remote Manipulative Control with Transmission Delay, Sep. 1963, S. 25–29. – ISSN 0096-249X
- [Smets 2000] SMETS, P.: Data fusion in the transferable belief model. In: *Information Fusion, 2000. FUSION 2000. Proceedings of the Third International Conference on* Bd. 1, July 2000, S. PS21–PS33 vol.1
- [Sylvia Le Hong 2010] SYLVIA LE HONG, User Interface Design G.: *Weltweit berührt - Studie zur Untersuchung kultureller Unterschiede und Gemeinsamkeiten bei der gestenbasierten Bedienung von Multitouch-Oberflächen*. Website. 2010. – Online erreichbar unter http://www.gm.fh-koeln.de/~hk/lehre/sgmci/ss2012/material/UID_Studie_Weltweit_beruehrt.pdf; zuletzt besucht am 15.02.2018.
- [Weiser 1991] WEISER, Mark: The computer for the 21st century. In: *Scientific American* 265 (1991), September, Nr. 3, S. 66–75
- [Williamson u. a. 2011] WILLIAMSON, B. ; WINGRAVE, C. ; LAVIOLA, J. ; ROBERTS, T. ; ; GARRITY, P.: Natural full body interaction for navigation in dismounted soldier training. In *Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC 2011)*, 2103–2110. (2011)
- [Yorick 2010a] YORICK, Wilks: *Close Engagements with Artificial Companions: Key social, psychological, ethical and design issues*. Kap. Introducing Artificial Companions, S. 35–51, John Benjamins, 2010
- [Yorick 2010b] YORICK, Wilks ; WILKS, Y. (Hrsg.): *Close Engagements with Artificial Companions: Key social, psychological, ethical and design issues*. John Benjamins, 2010

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 12. Juli 2019

Sobin Ghose