Integrationsplattform für HCI Untersuchungen in Smart Environments

Sobin Ghose[1]

Hamburg University of Applied Sciences, Hamburg 20099, Germany,
sobin.ghose@haw-hamburg.de,
WWW home page: http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2015-gsem/vortraege.html

Abstract. In dieser Ausarbeitung werden verschiedene Problemstellungen von Gestenerkennung diskutiert werden. Dabei wird auch auf die Schwierigkeiten von Vergleichen zwischen verschiedenen Systemen und Interaktionsrepertoires eingegangen. Zudem wird der Ansatz einer Integrationsplattform zur Untersuchung und Evaluation verschiedener Gesten und Sensoren zu vorgestellt.

Keywords: Kontextsensitivität, Smart Environments, Smart Home, Mensch-Computer-Interaktion (HCI), Gesten, 3D-Gestensteuerung, Gestenerkennung, Plattform, Integrationsplattform, HCI Studien

Table of Contents

Integrationsplattform für HCI Untersuchungen in Smart Environments

1 Einführung

In dieser Masterseminar Ausarbeitung soll das Thema der Masterarbeit erläutert werden, sowie die Abgrenzung innerhalb des Forschungsfeldes vorgenommen werden. Dabei baut diese Ausarbeitung auf Erkenntnissen der "Anwendungen 1" [12], "Anwendungen 2" [13] und dem "Projek 1"[14] Ausarbeitungen auf.

1.1 Motivation

Mark Weiser hat durch seine in "The computer for the 21st century"[26] beschriebenen Vision das Bild, wie wir mit Computern umgehen, maßgeblich geprägt.

Durch die heutige Rechnerallgegenwärtigkeit (engl. ubiquitous computing) bieten natürliche Schnittstellen eine intuitive Möglichkeit mit unserer Umgebung zu interagieren. So wird aktuell bereits Sprache zur Steuerung des Autonavigationsgeräts oder des Smartphones genutzt, sowie 2D-Gesten zur Steuerung eines Touchscreens oder 3D-Gesten als Alternative zum Controller bei Spielkonsolen.

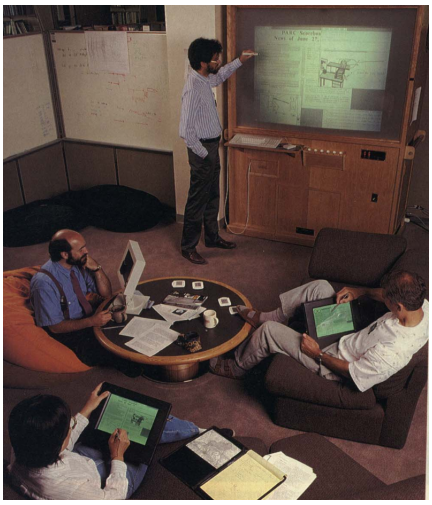


Fig. 1: Smart Board und Tablets als neue Bedienungskonzepte [26]

Grundsätzlich können einzelne natürliche Schnittstellen lediglich für beschränkte Aufgaben genutzt werden, da umfangreichere Anwendungen weiteres Wissen über den Kontext benötigen. So zeigte bereits 1992 David McNeill [21] auf, dass ein Rückschluss über die Intention einer Geste oder eines gesprochenen Satzes häufig nur unter Betrachtung des anderen Kommunikationskanals möglich ist. Dies ist insbesondere bei der Interaktion mit Smart Environments und Companion-System zu beachten. Auf diesen Zusammenhang wurde bereits detailliert in der Anwendungen 1 Ausarbeitung [12] eingegangen. Der Fokus dieser Arbeit liegt darin ein System zu Beschreiben mit welchem das Testen und Vergleichen verschiedener kontextsensitiver HCI Interaktionen umgesetzt werden kann.

Ziele Ziel dieser Ausarbeitung ist die Vorstellung einer Integrationsplattform für HCI Untersuchungen in Smart Environments mit welchen Ansätzen der 3D-Gestensteuerung sowie multimodale Ansätze zu untersuchen, um Anhaltspunkte

für neue Bedienungskonzepte zu gewinnen. Dabei soll das System leicht erweiterbar sein und den Entwicklern eine unkomplizierte Möglichkeit für HCI Untersuchungen bieten.

1.2 Gliederung der Ausarbeitung

Am Anfang wurde kurz auf die *Motivation* (1.1) und die *Ziele* (1.1) für diese Ausarbeitung eingegangen. Im Anschluss dieser *Gliederung* wird auf die *Grundlagen der Gestenerkennung* (2.1) sowie auf die *Problemstellungen* (2.1 , 2.3) bei der 3D-Gestenerkennung eingegangen. Nach der folgenden Analyse verschiedener *Verfahren zur Gestenerkennung* (3) wird auf *vergleichbare Arbeiten* (4) aus dem Bereich der Gestenerkennung sowie einem Gesamtsystem zur Kontexterkennung eingegangen. Im folgenden Kapitel wird das *Forschungsvorhaben* beschrieben (5) .Dabei wird auf einen Vorversuch sowie die *Plattformarchitektur*(5.2) sowie Möglichkeiten zur *evaluation des Systems* (5.3) eingegangen. Abschließend wird im Kapitel *Evaluation* (6.1) auf für die Integrationsplattform identifizierten *Problemstellungen* und *Risiken* (6.1) eingegangen. Am Ende des Kapitels wird ein *Ausblick* (6.2) auf weitere Schritte zur Masterarbeit gegeben.

2 Gestik und Gesten

2.1 Gestensteuerung

In der Mensch-Computer-Interaktion (eng.: Human-Computer-Interaction, kurz: HCI) werden aktuell größtenteils die lautsprachersetzenden Gesten, in Abgrenzung zu lautsprachbegleitenden Gesten, zur Bedienung genutzt. Diese lautsprachersetzenden Gesten können zur nonverbalen Kommunikation zwischen Menschen und von Menschen zu Tieren genutzt werden.

Klassifizierungen von Gesten Für die HCI sind nach einer Klassifizierung von [21] besonders folgende Gestentypen für die Steuerung geeignet:

- Zeigegesten: Diese Gesten werden durch das Ausstrecken des Zeigefingers in Richtung eines Objekts vollzogen. Dabei können Erwachsene auch auf imaginäre Orte und Objekte zeigen. Die Zeigegesten ist eine der Ersten, die von Kindern erlernt wird.
- Ikonische Gesten: Ikonische Gesten bilden die Eigenschaften oder Handlungen realer Objekte nach, wie es auch bei Pantomime gestikuliert wird. Ikonische Gesten werden auch oft als sprachbegleitende Gesten genutzt, um Gesprochenes zu verdeutlichen.
- Metaphorische Gesten: Durch metaphorische Gesten werden abstrakte Bilder dargestellt. Beispielsweise wünscht man jemanden 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”.

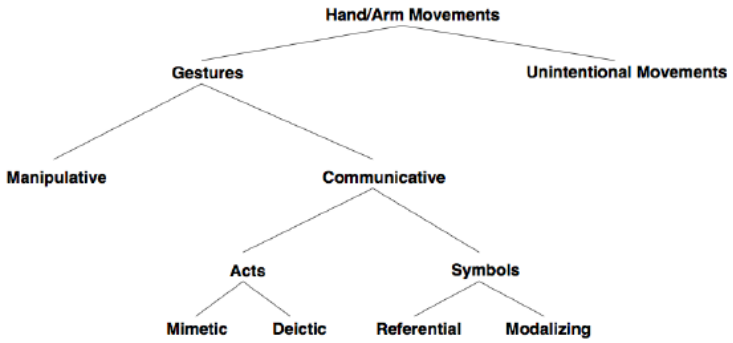


Fig. 2: Klassifizierung von Gesten [23]

Eine weitere Klassifizierung wurde durch u.a. von Pavlovic 1997 [23] (siehe Fig. 2) durchgeführt, welche auf den 6. Wismarer Wirtschaftsinformatik-Tage durch eine Arbeitsgruppe erweitert wurde [28]. Die in Fig. 2 beschriebenen manipulativen Gesten wurden in drei weitere Unterkategorien aufgeteilt, durch die eine genauere Unterscheidung bei Untersuchungen bezüglich der Usability möglich wurde.

- Bewegungsverfolgende Gesten: Diese Gesten sind äußerst intuitiv, da sie meist einer Manipulation von Objekten in der realen Welt nachempfunden werden.
- Kontinuierliche Gesten: Sie haben zwar keine direkte Entsprechung in der realen Welt, sind jedoch stark metaphorisch an die reale Welt angelehnt und dadurch leicht erlernbar.
- Symbolisch-manipulative Gesten: Diese Gesten können für Manipulationen in einer virtuellen Welt genutzt werden, für Aktionen, die in der realen Welt nicht möglich sind. Daher sind diese Gesten abstrakter und müssen vom Nutzer erlernt werden.

Start- Endproblem Bei kontinuierlicher Erfassung von Bewegungen ist insbesondere das Start-Endproblem (auch bekannt als "Midas touch problem"[1]) eine der größten Herausforderungen bei 3D-Gesten. Dabei gilt es einfache Bewegungen von den intentionalen Gesten zu unterscheiden (2) (vgl. [23]). Im Gegensatz zu den meisten 2D-Gesten ist es bei 3D-Gesten schwierig den Start- und Endpunkt einer Geste zu ermitteln. Insbesondere dadurch, dass sich aus der reinen Bewegungserfassung nur stark bedingt Rückschlüsse auf die Intention der Person ziehen lassen. Bei 2D-Gesten wird der Kontext meist durch die Position der berührten Fläche direkt vorgegeben, daher ist die Interpretation einer 2D-Geste

[1] Über König Midas, der bis 695 v. Chr. Herrscher von Phrygien war, gab es mehrere antike mythische Anekdoten von seiner Gier und Dummheit. So soll er sich die Gabe gewünscht haben, dass alles was er berührt zu Gold wird. Dies wurde schnell zu einem Fluch statt einem Segen, da er auch die Nahrung, die er zu sich nehmen wollte oder Verwandte, die er anfasste, zu Gold wurden.

in den meisten Fällen trivial (vgl. [25], [24]). Viele Ansätze zur Lösung dieses Problems für 3D-Gesten, wie z. B. das Drücken eines Knopfes des WiiMote Controllers oder das Stillstehen zwischen zwei Gesten, benötigen weitere Hilfsmittel oder machen die Interaktion unnatürlich bzw. sind nicht für eine kontinuierliche Erfassung der Bewegungen ausgelegt (vgl. [5]).
Aktuelle Ansätze zur Lösung dieses Problems werden in Kapitel vergleichbare Arbeiten 4 vorgestellt.

2.2 Kontext einer Geste

Anwendungskontext Wie im vorangegangen Kapitel 2.1 beschrieben, spielt der Kontext für die Interpretation eine zentrale Rolle. Aktuell wird der softwareseitige Anwendungskontext als Hauptkriterium für das Ausführen eines Befehls einer erkannten Geste genutzt. Dieser Ansatz bietet sich besonders an, wenn dem Nutzer nur ein kleines Gestenrepertoire im aktuellen Anwendungskontext zur Verfügung steht.
Dem Nutzer ist unter Umständen nicht klar, welche Gesten aktuell erlaubt sind, da beispielsweise bei der Bedienung eines Smart Homes keine virtuelle Oberfläche zur Verfügung steht wie bei einem Touchscreen.

Kontextsensitivität (Context-aware Computing) Zusätzlich zum Anwendungskontext der Software bzw. des Systems spielt der reale Kontext einer Geste eine wichtige Rolle. Für eine gute Annahme über die Intention einer Bewegung ist Wissen über die Umgebung, die Person selbst, sowie die aktuelle Tätigkeit der Person, Kernvoraussetzung (vgl. 2.1), welche nicht durch herkömmliche Gestenerkennungssysteme erfüllt werden. Da der Unterschied zwischen einer Geste und einer Bewegung nur die Intention des Nutzers zugrunde legt, werden in diesem Rahmen die individuellen Eigenschaften wie Mimik, Sprache und Blickrichtung als Kontext einer Geste aufgefasst[2].
Besonders im Zusammenhang mit einem Smart Environment muss das System eine gute Theorie darüber entwickeln, auf welches Objekt der Nutzer zeigt bzw. wo sich der Nutzer befindet (Location Awareness), oder in welche Richtung er guckt (Gaze Awareness). Außerdem ist die Mimik des Gesichts ein wichtiger Faktor, um die richtige Bedeutung von Gesten zu erfassen. So lässt sich eine ironisch gemeinte Geste meist lediglich an der Mimik oder der Betonung von begleitender Sprache erkennen (vgl. [21]). Umgekehrt erschließt sich oft nur aus dem Kontext die Emotion wie in Fig. 3 zu sehen ist[3].

Ebenso könnten Mimik und Emotionen im Anschluss einer Geste erste Rückschlüsse auf die korrekte Interpretation liefern. Aktuell lassen sich bereits Verwirrtheit, Frustration sowie Ärger durch kamerabasierte Emotionserkennung feststellen (vgl. [10]). Aus diesem Grund wird auch aus der Perspektive von Companion-Systemen das Feld der Posen- und Gestenerkennung untersucht.

[2] Für eine detaillierte Auseinandersetzung mit dem *Kontext* Begriff eignet sich "What We Talk About when We Talk About Context" von Paul Dourish (2004) [9]

[3] Ein weiterer Aspekt sind die kulturell bedingten unterschiedlichen Verwendungen und Bedeutungen von Gesten (vgl. [6], [8])

Fig. 3: Die Gesichtsmimiken von Ekel und Ärger lassen sich häufig nur durch den Kontext erschließen [3]

Außerdem können sprachbegleitende Gesten nur im Kontext des gesprochenen Satzes interpretiert werden ([21]). Beim Nutzen einer Zeigegeste 2.1 wird deutlich, dass dies auch umgekehrt für einen Satz gelten kann, in dem beispielsweise die Geste mit dem Wort "da" verwendet wird.
Weitere Daten, deren Nutzen für die Interpretation einer Geste untersucht werden muss, sind Terminkalender, Uhrzeit, Wetter sowie vorhergehende Tätigkeiten.

2.3 Schlussfolgerungen für die kontextabhänige Gestensteuerung

Wie durch die Einführung in die Gestensteuerung (2.1) und die Erläuterung des Kontextes für 3D-Gesten (2.2) aufgezeigt wurde, reicht eine Gestenerkennung rein auf Basis von Motion-Tracking-Verfahren nicht aus, da diese Daten keinen Rückschluss auf den Kontext der Geste oder die Intention des Nutzers geben. Daher ist eine semantische Interpretation bezüglich weiterer Informationen über den Kontext der Geste und der Person unabdingbar, um eine Unterscheidung von unbewussten Bewegungen und beabsichtigten Gesten durchzuführen. Dies ist eine Voraussetzung für eine nahtlose, natürliche und intuitive Interaktion. Daher ist die leichte Erweiterbarkeit der Plattform für Sensoren und Aktoren eine zentrale Anforderung an das System. Weitere Details zu dieser Schlussfolgerung finden sich in der *Anwendungen 1* ([12]) Ausarbeitung.

3 Verfahren zur Echtzeit Gestenerkennung

Für die Erkennung und Analyse von Gesten gibt es grundsätzlich zwei verschiedene Herangehensweisen: Maschinenlernen und Heuristik basierte Verfahren. Eine gute Übersicht der Gestenerkennung liefert die Abhandlung *An introduction to 3D Gesture Interfaces* von *Joseph J.* und *LaViola Jr.* [20]. Die beiden Herangehensweisen sollen im Folgenden kurz vorgestellt werden.

3.1 Lernende Verfahren

Üblicherweise werden Maschinen-Lernen-Verfahren zur Klassifizierung von 3D-Gesten genutzt. Dafür müssen wichtige Charakteristiken (Features) der aufgenommenen Geste extrahiert werden und an den Klassifizierungsalgorithmus zur Einordnung übergeben werden [20]. Wie bei den meisten lernenden Verfahren muss das System mit einer Vielzahl von hochwertigen Beispieldaten trainiert werden, um zuverlässige Klassifizierungen durchführen zu können. Die Auswahl aussagekräftiger Features ist eine der zentralen Herausforderungen der 3D-Gestenerkennung durch Maschinenlernen.

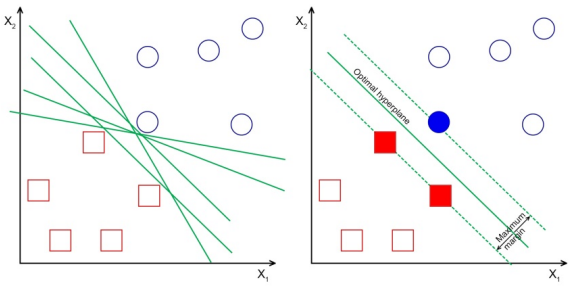


Fig. 4: SVM: Beispiel einer optimalen Hyperebene in einem zweidimensionalen Vektorraum [20]

Folgende konkrete Verfahren des Maschinenlernens werden häufig genutzt:

- Support Vector Machines (SVMs): SVMs gehören zu den Überwachtes-Lernen Verfahren. Jede Geste aus den Testdaten wird durch ein Objekt in einem mehrdimensionalen Vektorraum dargestellt. Dabei wird grundsätzlich jedes Feature in einer Dimension des Vektorraums beschrieben. Aufgabe der SVM ist es mehrere Hyperebenen in dem Raum zu definieren, welche auf Basis der Trainingsdaten, den maximalen Abstand zwischen den verschiedenen Gesten beschreibt. Diese Grenzen der Hyperebenen werden für die Klassifizierung von neuen Daten genutzt. (Siehe Figure 4)
- Hidden Markov Models (HMMs): HMMs sind stochastische Modelle, welche auf einer Markov-Kette mit endlichen Zuständen(für welche die Markow-Annahme besteht) aufgebaut sind und einer Menge zufälliger übergangsfunktionen für die Zustände. HMMs können auf verschiedene Art und Weise für die Klassifizierung eingesetzt werden. Einige Ansätze zur 3D-Gestenerkennung werden in [20] erläutert.
- Conditional Random Fields (CRFs): CRFs sind wie HMMs stochastische Modelle. Jedoch sind bei CRFs die vorangegangenen Zustände gegenüber von HMMs nicht versteckt. Zudem haben sie nicht die potentzielle Gefahr des “labeling bias” Problems, welches bei HHMs durch eine geringe Entropie der Übergangswahrscheinlichkeit der Zustände auftritt [18]. CRFs können ebenso wie HMMs auf verschiedene Weisen für die Gestenerkennung eingesetzt werden [20].
- Decision Trees (DTs): Entscheidungsbäume sind geordnete und gerichtete Graphen, bei denen Knoten, mit Ausnahme der Blätter, Entscheidungsregeln beschreiben. Konkret bedeutet dies, dass jeder Knoten eine Entscheidung über ein Feature trifft. Durch die Aneinanderreihung von Entscheidungen,

von der Wurzel bis zum Blatt, wird die Klassifizierung durchgeführt. Die Entscheidungsbäume werden in der Gerstenerkennung durch rekursive Verfahren wie das "top-down induction of decision trees" (TDIDT) erlernt bzw. aufgebaut.
- Decision Forests (DFs): DFs sind eine Erweiterung von DTs. DFs sind eine Menge von zufällig trainierten Entscheidungsbäumen. Die Nutzung von DFs, gegenüber zu DTs, führt meist zu einer hören Robustheit und zu einer besseren Generalisierung der Ergebnisse. Genauere Informationen zu Decision Forests findet man u.a. in [7].
- Weitere lernende Verfahren: Es können auch weitere Verfahren wie Template Learning, Maschienenlernen durch endliche Automaten oder Neuronale Netze eingesetzt werden. Weitere Informationen dazu liefert [20].

3.2 Heuristik basierte Verfahren

Die auf Heuristik basierenden Verfahren werden seltener verwendet, obwohl sie in Ansätzen wie [27] sehr gute Ergebnisse liefern. Heuristische Verfahren eignen sich besonders, wenn es lediglich eine kleine Anzahl von einfachen Gesten gibt. Zudem haben sie gegenüber den lernenden Verfahren den Vorteil, dass sie keine Trainingsdaten benötigen und durch einfache Regeln umgesetzt werden können [20].

Ein Beispiel dazu liefert [27], wo z. B. auf Basis des Skelettmodells der Kinect von Microsoft durch eine simple Regel ein Springen des Nutzers erkannt werden konnte.

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

Dabei steht H_y für die aktuelle Höhe des Kopfes (y-Achse der Kopfposition), $\bar{H}_y$ für die kalibrierte normale Höhe des Kopfes der Person. C ist eine konstante Größe, die den Schwellwert zwischen Stehen und Springen beschreibt. Das Ergebnis J ist $wahr$ oder $falsch$, je nachdem, ob die Differenz zwischen H_y und $\bar{H}_y$ C übersteigt.

Probleme können jedoch auftreten, wenn neue Gesten dem Interaktionsrepertoire hinzugefügt werden. Dies kann eine Anpassung andere Gesten benötigen, um eine Verwechselung auszuschließen. Zudem ist die Erfolgsrate der Erkennung ebenfalls stark von verschiedenen Parametern abhängig wie beispielsweise der erlaubten Varianz und verscheiden Zeitüberschreitungen zwischen den Übergängen von verschiedenen Teilen einer Geste.

3.3 Verfahrenswahl

Für intuitive Gesten scheinen heuristische Verfahren zur Gestenerkennung besser geeignet zu sein. Zudem können die von aktuellen Sensoren bereitgestellten Skelett- und Handmodelle unmittelbar verarbeitetet werden, ohne zusätzliche Feature Bildung. Da heuristische Verfahren keine umfangreichen Trainingsdaten benötigen, lassen sich somit auch schneller weitere Gesten dem System hinzufügen.

Fig. 5: RealEdge Prototyp auf Basis heuristischer Verfahren [27]

Andererseits besteht die Gefahr bei heuristischen Verfahren, dass durch das Hinzufügen einer neuen Geste eine manuelle Anpassung der anderen Gesten nötig ist. Dies kann auftreten, wenn Effekte der neuen Geste zuvor nicht in Betrachtung gezogen worden sind.

Zumal die in dieser Ausarbeitung vorgestellten komplizierteren metaphorischen Gesten keine Veränderungen auf der z-Achse berücksichtigen, muss im Rahmen des Hauptprojekts geprüft werden, ob bereits bekannte Frameworks, wie z. B. OpenCV zur Gestenerkennung auf Basis von 2D-Bilder, ausreichen. So können z. B. die erkannten Punkte bei Imaginery Interfaces [15] einfach durch eine SVM klassifiziert werden.

Eine nachträgliche Erweiterung zusätzlicher Agenten, welche andere Verfahren zur Gestenerkennung nutzen, stellt auch bei gleichzeitiger Nutzung kein Problem dar, ferner ist diese zur Evaluation verschiedener Ansätze erwünscht.

4 Vergleichbare Arbeiten

Im Folgenden werden aktuelle, Themen relevante Abhandlungen vorgestellt und die Konzepte auf Mächtigkeit, Usability, Lösung des Start-End-Problems sowie ihrer Tauglichkeit im Smart Environments analysiert. Die daraus gewonnenen Erkenntnisse werden im abschließenden Kapitel 6.1 erläutert.

4.1 Bremen Ambient Assisted Living Lab

Das Bremen Ambient Assisted Living Lab (BAAL) der Universität Bremen hat verschiedene Studien im Bereich der multimodalen Interaktion durchgeführt. Ein relevantes Wizard-of-Oz-Experiment[4] wird im Folgenden vorgestellt. Eine weitere WoO-Studie des BAALs wurde in der Anwendungen 2 Ausarbeitung betrachtet [13].

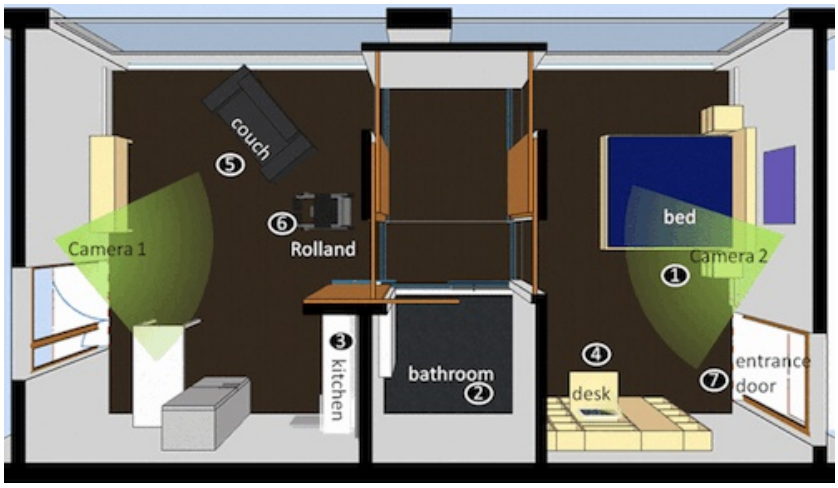


Fig. 6: Bremen Ambient Assisted Living Lab

A German-Chinese Speech-gesture Behavioural Corpus of Device Control in a Smart Home Die Studie[1] zur multimodalen Steuerung von Elektrogeräten und Haustechnik in einem Smart Home zeigt eine große inhaltliche Nähe zu eigenen Zielen auf und liefert wichtige Erkenntnisse für eigene Experimente und Untersuchungen. An der Studie nahmen 20 deutsche und 6 chinesische Probanden teil. Diese mussten vier verschiedene Aufgaben erfüllen:

1. Schließen und öffnen einer automatischen Schiebetür (diskreter Parameter: Auf/Zu)
2. An- und Ausschalten einer Deckenlampe (diskreter Parameter: An/Aus)
3. Hoch- und Runterfahren des Kopf- und Fußteils eines Bettes (kontinuierlicher Parameter: Höher/Tiefer)
4. An- und Ausschalten einer Bettlampe (diskreter Parameter: An/Aus)

Jeder Proband durchlief drei verschiedene Tests:

1. Er darf nur Gesten zur Steuerung nutzen

[4] Wizard-of-Oz-Experiment 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. Er darf nur Sprache nutzen
3. Er kann frei zwischen Sprache und Gesten wählen

Die hier vorgestellte Abhandlung [1] fokussierte die Steuerung durch Gesten.

Durchlauf 1: Gestensteuerung Die Mehrzahl der Probanden (24 von 26) nutzt die gleiche bzw. gespiegelte Geste zum Ein-/Ausschalten von Lichtern oder Öffnen/Schließen von Türen. Dabei wurde die Wischgeste stark präferiert. Ein Viertel der deutschen Probanden nutzte Zeigegesten, jedoch keiner der chinesischen Probanden. (In China deutet man auf Personen oder Gegenstände grundsätzlich mit einer offenen Hand und nicht mit einem Finger) Die chinesischen Probanden haben die Gesten meist schnell und abrupt gegenüber den deutschen Probanden durchgeführt. Zudem haben sich die chinesischen Probanden nicht im Raum bewegt, währenddessen die deutschen Probanden sich häufig näher an das anzusteuernde Objekt bewegt haben (alle Probanden wurden darauf hingewiesen, dass sie sich frei im Raum bewegen dürfen). Allgemein haben die Nutzer ein schnelles Feedback erwartet. War die Verzögerung zu lang, wurden weitere Gesten ausprobiert. Dies hat zu Verwirrung geführt, da der Nutzer nicht wusste, welche Geste vom System erkannt wurde. Da die Nutzer nicht wussten, dass sie an einem Wizard-of-Oz Experiment teilnahmen, waren sie größtenteils erstaunt darüber, dass die Wohnung ihre Befehle erkannte. Einige Nutzer gaben an, dass sie sich bzgl. der Gestensteuerung unsicher waren, da sie nicht wussten, wo sich der Sensor zur Gestenerfassung befand.

Durchlauf 2: Sprachsteuerung Fast die Hälfte der, von den deutschen Probanden, verwendeten Sprachbefehle waren kontextsensitiv. Daher müssen Blickrichtung (gaze Tracking) und Körperposition zur Interpretation des Befehls mit einbezogen werden. Viele der deutschen Probanden bewegten sich in die Nähe des anzusprechenden Objekts. Von den sechs chinesischen Probanden nutzte keiner einen einzigen kontextsensitiven Sprachbefehl, sie gaben daher jeweils den Ort oder die Art des Objekts an, z. B. Deckenlampe statt Lampe.

Durchlauf 3: Multimodale Steuerung Ein Großteil der Probanden (20 von 26) gaben an, dass sie die Wahl für Gesten- oder Sprachsteuerung von der Aufgabe abhängig machten. Dabei war die persönliche Präferenz, für welche konkrete Aufgabe welche Steuerart gewählt werden sollte, unterschiedlich. Fünf der Probanden bevorzugten grundsätzlich die Sprachsteuerung, während lediglich zwei die Gestensteuerung präferierten.
Einige Probanden gaben an, dass die Sprachsteuerung für die kontinuierlichen Parameter zu ungenau sei und daher Gesten bevorzugt würden. Zudem nutzten eine Reihe von Nutzern eine Zeigegeste, um ein Objekt zu selektieren und einen Sprachbefehl, um die Art der Manipulation des Objekts zu definieren.

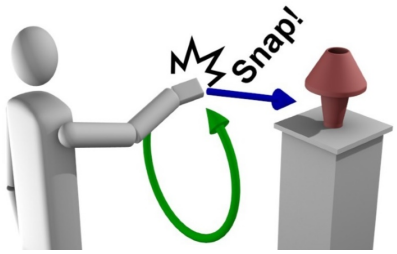
Kulturelle Unterschiede In der Abhandlung wird darauf hingewiesen, dass umfangreiche Studien nötig sind, um die beiden gebildeten Thesen zu den Unterschieden der Kulturen zu verifizieren.

– Keine großen Unterschiede der Gestenwahl von Chinesen und Deutschen
– Chinesen nutzen gegenüber den Deutschen keine kontextsensitiven Sprachbefehle

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

Ein konkreter multimodaler Ansatz zur Steuerung von Geräten wurde an der Universität Bielefeld entwickelt [11]. Der Nutzer kann über eine Zeigegeste in einer kontextsensitiven Umgebung ein Gerät selektieren. Durch das Schnipsen mit dem Finger wird die Selektion bestätigt und der Start einer weiteren manipulativen Geste signalisiert. Die konkrete Art der Ansteuerung des Gerätes wird durch die manipulative Geste festgelegt. So steht z. B. eine Bewegung der Hand nach oben/rechts für den Befehl “An/Weiter” und eine Bewegung der Hand nach unten für “Aus/Zurück” bei diskreten Parametern. Das Repertoire der manipulativen Gesten beschränkt sich auf die Bewegung der Hand nach oben/unten bzw. rechts/links und Kreisbewegungen mit/gegen dem Uhrzeigersinn. Die Kreisbewegungen mit den Uhrzeigersinn werden für kontinuierliche Parameter wie z.B. “Lauter” und gegen den Uhrzeigersinn mit der entsprechenden Negation “Leiser” interpretiert.

(a) Proband wählt durch Zeigegeste aus



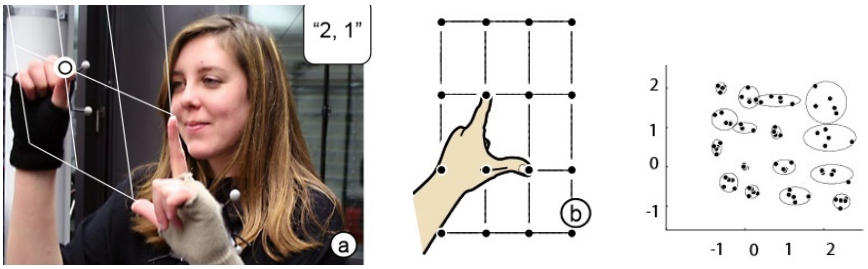
(b) Nach der Zeigegeste signalisiert der Proband durch ein Fingerschnipsen den Start einer Manipulation

Da hier bekannte und einfache Gesten genutzt werden, ist die Steuerung ohne großen Aufwand erlernbar. Jedoch können so 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.

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

In der folgenden Abhandlung[15] wird das Konzept und die Möglichkeiten von Imaginary Interface vorgestellt. Imaginary Interfaces wurden vom The Human

Computer Interaction Lab des Hasso Plattner Institute entwickelt und beschreiben eine Methode Computersysteme durch Gesten auf ein lediglich vorgestelltes Interface zu bedienen.
Dabei hält der Nutzer eine Hand vor sich und streckt den Zeigefinger und Daumen in 90° Winkel aus. Nun kann sich der Nutzer ein Koordinatensystem um seine Hand herum vorstellen (Abbildung 8b), wobei die Länge des Zeigefingers eine Einheit auf der y-Achse definiert und die Länge des Daumens entsprechend eine Einheit auf der x-Achse darstellt. Mit der zweiten Hand kann der Nutzer nun Punkte/Koordinaten auswählen oder Linien/Gesten in das Koordinatensystem zeichnen. Dies tut er durch das Berühren des Zeigefingers und Daumen der zweiten Hand (Start einer Geste). Durch das Öffnen der Finger wird das Ende einer Geste, des Zeichnens oder der Auswahl eines Punktes definiert.



(a) Proband wählt den Punkt 2,1 aus

(b) Koordinatensystem

(c) Klassifizierung der erkannten Punkte

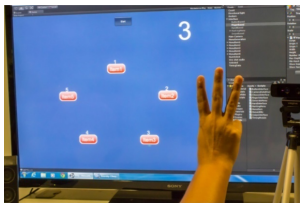
Fig. 8: Imaginary Interfaces

Erste Untersuchungen konnten bereits zeigen, dass das Koordinatensystem, welches durch die erste Hand gebildet wird, maßgeblichen Einfluss auf das visuelle Gedächtnis einer Person hat. Konkret konnte z. B. nachgewiesen werden, dass Nutzer im Vorfeld gemalte Linien und Ecken wesentlich zielgenauer wiederfinden konnten.
Wie bereits beschrieben, liefert dieses Konzept eine gute Lösung für das Start-End-Poblem. Aktuell wird als Sensorik eine kleine Kamera an der Brust getragen (Wearable Computing), was die Usability allgemein einschränkt. Die metaphorischen Gesten bieten eine enorme Mächtigkeit für mögliche Gesten und Steuerbefehlen. Diese müssten jedoch vom Nutzer erst aufwendig erlernt werden. Das System bietet allerdings die Möglichkeit, dass jeder Nutzer die verschiedenen Funktionen individuell in sein personalisiertes vorgestelltes Interface einbettet. Dies könnte die Dauer des benötigten Trainings verringern.

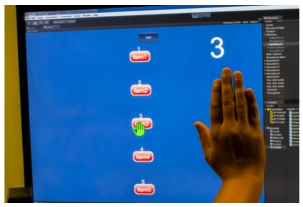
4.4 Exploring the Usefulness of Finger-Based 3D Gesture Menu Selection

Eine weitere Möglichkeit um metaphorische Gesten zu nutzen, liefert das Paper[17] von Arun Kulshreshth, welches bei der ACM CHI 2014 vorgestellt worden ist. In der Studie wurden vier verschiedene Arten von fingerbasierenden

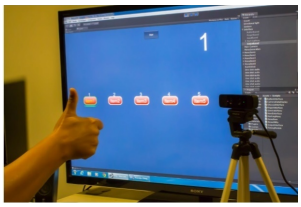
Gesten verglichen, welche die Auswahl eines Menüitems ermöglichen. Als effektivste und stabilste Methode zur Auswahl eines Menüpunktes stellte sich das einfache Aufzeigen einer bestimmten Anzahl der Finger raus. Da die Menüpunkte nummeriert sind, kann so die Anzahl der Finger einem Menüpunkt zugeordnet werden. Gegenüber den Imaginary Interfaces ist diese Art der Gestik einfach zu erkennen und unkompliziert durchzuführen und kann auch mit nur einer Hand durchgeführt werden. Allerdings kann eine höhere Mächtigkeit nur durch das Navigieren durch verschiedene Menüs erreicht werden. Dies kann zu komplexen Strukturen führen, welche ohne visuelles Interface kaum bedienbar sind. Der Initialzustand der Hand ist bei allen Modellen die Faust. Durch das Zurückkehren in diesen Zustand bestätigte der Nutzer seine Selektion (stop der Geste).



(a) Proband wählt durch die Anzahl der Finger den entsprechenden Menüpunkt aus



(b) Proband wählt durch Bewegen der Hand in der Horizontale einen Menüpunkt aus



(c) Proband wählt durch Bewegen der Hand in der Vertikalen einen Menüpunkt aus

Fig. 9: Drei verschiedene Arten der Auswahl eines Menüpunkts durch fingerbasierte Gesten

4.5 Fazit der vergleichbaren Arbeiten

Die im diesem Kapitel vorgestellten Ansätze wurden ausgewählt, da sie unterschiedliche Interaktionsrepertoires vorstellen und teilweise Lösungsansätze für das Start-Ende-Problem aufzeigen. Ein grundsätzliches Problem bei einer Gegenüberstellung verschiedener Gesten besteht darin, dass andersartige Ansätze aus unterschiedlichen Abhandlungen sich nur schwer vergleichen lassen. So wird z. B. in einigen Ausarbeitung für die Gestenerkennung kein kontinuierlicher Datenstrom genutzt und daher das Start-End-Problem nicht beachtet. Zudem werden häufig grundverschiedene Sensoren zum Motion Tracking verwendet. Dies erschwert die Beurteilung und die Gegenüberstellung von verschiedenen konzeptionellen Ansätzen und Verfahren zur Gestenerkennung beträchtlich.
Eine weitere wichtige vergleichbare Arbeit, welche nicht die Gestensteuerung im Fokus hat, sondern die Architektur des Gesamtsystems ist *Companion Technology for Multimodal Interaction* [16], welche im Rahmen des SFB Transregio 62 entstanden ist. Diese Ausarbeitung wurde bereits in Projekt 1 [14] Vorgestellt,

die Grundzüge der Architektur sind stark in das Design der Plattform eingeflossen, welche im Kapitel 5.2 erläutert wird.

5 Forschungsvorhaben

Im Folgenden soll auf die konkreten Forschungsfragen und das Design der zu entwickelnden Plattform eingegangen werden.
Aufgrund der im Fazit des vorigen Kapitels beschriebenen Problemstellungen bzgl. der Vergleichbarkeit verschiedener Ansätze, soll die Plattform dem Nutzer ermöglichen verschiedene Ansätze schnell zu implementieren, um eine Evaluation auf Basis eigener Messungen durchzuführen zu können.
Zur Identifikation der Anforderungen an eine solches System wurde neben der bereits dargestellten Recherche in Kapitel 4 mit verschiedenen Entwicklern und Nutzern ein Alphatests durchgeführt, sowie Vorversuche einer HCI-Studie in Zusammenarbeit mit der Uni Hamburg durchgeführt.

5.1 Vorversuch: Evaluation of an Omnidirectional Walking-in-Place User Interface with Virtual Locomotion Speed Scaled by Forward Leaning Angle

In zusammen Arbeit mit Tobias Eichler wurde ein omnidirektionales Kamerasystem zur Erkennung von Skeletten auf Basis von mehrerer Kinect v2 Sensoren entwickelt. Auf Grundlage dieser Daten wurde eine Gestenerkennung für Schritte und den Neigungswinkel des Oberkörpers entwickelt um die Geste "gehen" und "rennen" zu erkennen. In einer Kooperation mit der Uni Hamburg wurde das System um eine VR-Brille erweitert, durch welche es möglich wurde sich durch eine virtuelle Welt zu bewegen.

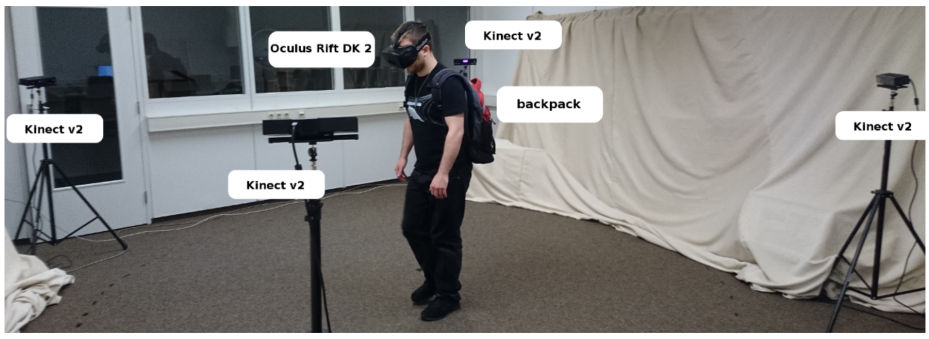


Fig. 10: Laboraufbau: Der Nutzer trägt eine VR-Brille mit einem Notebook im Rucksack. Dabei wird er von vier Kinects v2 getracket [19]

Auf Basis des Motion Tracking Systems und einer weiteren Komponente zur Bodenerkennung konnte eine Schritterkennung durchgeführt werden.

Diese Daten, sowie die Ausrichtung des Skeletts und ein beuge Winkel des Skeletts wurden als Controller für die Bewegung in der virtuellen Welt genutzt.

Konkret sollte untersucht werden ob eine Beschleunigung in Abhängigkeit des beuge Winkels des Nutzer nach vorne intuitiv ist und die Immersion erhöht.
Das Paper wurde bei der GIVRAR 2015 Konferenz vorgestellt, und kann als ein Vorversuch für die Integrationsplattform gesehen werden. So konnten im Rahmen der Untersuchung verschiedene Komponenten der Plattform, wie z. B. Kinect 2 Agent, Fusion Agent oder die Middleware geteste werden, und weitere Anforderungen an das Gesamtsystem, wie ein umfassendes Monitoring, Agenten Lifecycles und eine Erfordernis der Möglichkeit zur Fernkonfiguration der Agenten identifiziert werden.

Fig. 11: Virtuelle Welt: Modell der *Hammaburg* im 9. Jahrhundert [19]

5.2 Plattformarchitektur

Im Folgenden soll kurz auf die Plattform Architektur eingegangen werden. Eine ausführliche Erläuterung der Komponenten ist im Bericht des Grundprojekts zu finden [14]. Die Systemarchitektur lehnt sich an die Architektur eines Companion-Systems [16] an, dies soll zukünftige Weiterentwicklungen und das Hinzufügen von weiteren Komponenten zur Kontexterfassung und -Interpretation vereinfachen.

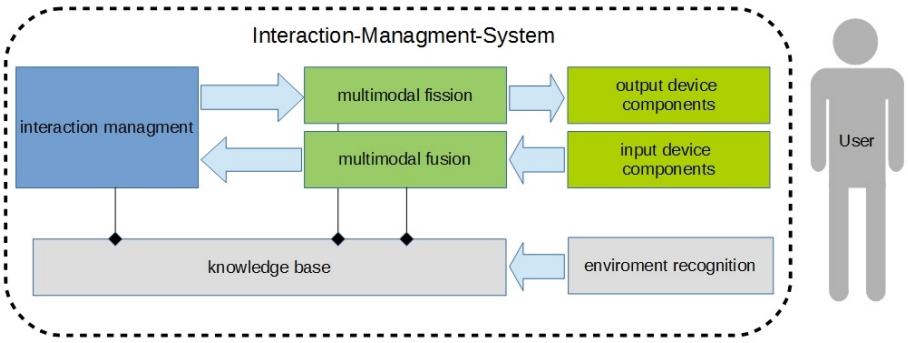


Fig. 12: Komponenten des Interaktion-Management-Systems (vgl. [16])

Das Gesamtsystem besteht aus folgenden Komponenten:

Interaktion-Managment Komponente Diese Komponente beinhaltet die Anwendungslogik, welche auf Basis der Daten der multimodalen Fusion Komponente und der Wissensdatenbank Entscheidungen über die konkreten Zustände des Systems, der Umwelt und vor allem des Nutzers, trifft. Auf dieser Basis kann eine Interpretation der Nutzeraktionen durchgeführt werden. Zudem werden die Steuerungsbefehle an die Multimodal-Fission Komponente weiter geleitet.

Multimodale-Fusion Komponente Diese Komponente interpretiert die Daten der Gesten- und Sprachsensoren. Diese Daten werden im Vorfeld fusioniert um präzisere und zuverlässigere Aussagen über den Nutzer treffen zu können. Die Ergebnisse werden in Aggregierter Modalität-unabhängiger Form an die Interaktion-Management Komponente weitergeleitet.

Multimodal-Fission/User-Feedback Komponente Auf Basis der Modalität-unabhängigen Information der Interaktion-Management Komponente kann die Multimodal-Fisson Komponente durch Nutzung der Kontextinformationen der Wissensdatenbank entscheiden, über welches Interface, der Situation entsprechend, mit dem Nutzer interagiert werden soll. Dies kann besonders wichtig sein, sollte ein Befehl des Nutzers nicht eindeutig erkannt worden sein.

Wissensdatenbank Die Wissensdatenbank hält Informationen über die verschiedenen Aktoren des Smart Environment sowie weitere, für den Kontext relevante, Informationen bereit.

Design Zusammenfassung Das Design soll dem Nutzer - je nach Situation - ermöglichen über verschiedene Geräte und Kanäle mit dem System in Interaktion zu treten. Dies soll gewährleisten, dass die Vorlieben und Möglichkeiten des Nutzers berücksichtigt werden.
Durch die Trennung der verschiedenen Bereiche sowie der Austausch von modalitätsunabhängigen Informationen soll eine lose Koppelung der Systemkomponenten gewährleisten. Dies ermöglicht eine semantische Vermengung von verschiedenen Kommunikationskanälen. Zudem kann ein Austausch der Sensorik oder der Aktoren Transparent für das restliche System geschehen. Die Multimodal-Fission Komponente soll dem Nutzer, durch die Wahl eines geeigneten Kommunikationskanals für das Userfeedback, bei der Steuerung unterstützen. Dies ist besonders wichtig, da in Studien wie z. B. [1] gezeigt wurde, dass Nutzer eine direkte Reaktion des Systems erwarten. In der Studie wie auch in [2] wurde zudem gezeigt, dass die semantische Vermischung verschiedener Steuerungskanäle zur natürlichen Interaktion mit Smart Objects dazugehören. Die Multimodal-Fusion Komponente soll diese semantische Vermischung durch eine Transformation zu aggregierten modalitäts-unabhängigen Nachrichten ermöglichen. Die Interaktion-Managment Komponente hält intern ein Modell des Systems, der Umwelt sowie des Nutzers bereit um eine Interpretation der Nutzereingaben vorzunehmen. Des Weiteren werden weitere Logging-, Monitoring- und Kontrollkomponenten entwickelt um Entwickler bei der Durchführung von Versuchen zu unterstützen.

5.3 System Evaluation und Tests

Ein erweiterbares Gesamtsystem zu testen ist eine komplexe Problemstellung. Um diese Komplexität zu beherrschen, soll das Grundprinzip "Teilen und Herrschen" verwendet werden. Dabei werden den Agenten im System konkrete Vorgaben eines zu implementierenden Lifecycles gegeben. Eine neue Komponente sollte dann auf die korrekte Umsetzung dieses Lifecycle getestet werden. Das Monitoring soll zudem Informationen über den Status der verschiedenen Agenten abfragen um Aussagen über den Zustand des Gesamtsystems tätigen zu können.
Zudem soll die Logging-Komponente um eine Rekordfunktion um einen Datendump welcher alle oder einer Auswahl (z. B. First Level Sensodata) der Nachrichten aufnehmen und abspielen kann erweitert werden. Dadurch kann, eingeschränkt durch unterschiedliche Netzwerklaufzeiten, gezeigt werden, ob das System deterministisch arbeitet. Diese Funktion kann zudem, dem Entwickler helfen im Nachhinein das Systemverhalten mit anderen Parametern, Implementationen oder neuen Komponenten zu testen.
Diese Werkzeuge bieten erste Ansätze um sicherzustellen das die Funktionalität, die Systemzuverlässigkeit und die Performanzanforderungen der Plattform evaluiert werden können.

6 Evaluation

Durch den Vorversuch und weiteren Test im Living Place konnten die zugrunde liegenden Konzepte der Architektur bereits testet werden. Im Folgenden soll genauer auf konkrete Problemstellungen für die Entwicklung des Systems zur Masterarbeit eingegangen werden.

6.1 Problemstellungen und Risiken

Neben den allgemein bekannten Problemstellungen wie die Berücksichtigung unterschiedlicher Körpergrößen, kulturell unterschiedlich genutzten Gesten und der bereits ausgeführten Start-End-Problematik soll der Fokus auf neu identifizierte Risiken bzgl. der Integrationsplattform hingewiesen werden.

Reaktionszeit und User Feedback Die Wizard-of-Oz Studien (vgl. [1], [2]) zeigten auf, dass die Probanden eine schnelle Reaktion des Systems erwarteten. So wird in verschiedenen Untersuchungen dazu angegeben, dass eine Antwortzeit von weniger als 100 ms dem Nutzer fließend vorkommen. Längere Reaktionszeiten führen bei vielen Nutzern zu Verwirrung, da sie sich nicht sicher sind, ob das System ihre Geste erkannt hat. Die schnelle Reaktionszeit muss durch eine gute Architektur und Implementierung des Systems sichergestellt werden. Eine zusätzliche Möglichkeit dies zu gewährleisten kann durch eine Täuschung des Nutzers erreicht werden. So wird dem Nutzer beispielsweise mitgeteilt, dass das System eine 360°-Kreisbewegung mit der Hand erkennt. Dabei erkennt das

System bereits bei 270° die Geste und verhält sich dementsprechend.
Darüber hinaus sollte dem Nutzer durch ein Feedback mitgeteilt werden, dass die Geste erkannt worden ist und die Anfrage bearbeitet wird. Dies ist besonders wichtig, sollte der Nutzer Objekte ansteuern, welche er nicht sehen kann oder bei denen keine direkt bzw. unverzüglich wahrnehmbare Veränderung stattfindet (z. B. Verändern der Temperatur der Bodenheizung). Dieses Feedback kann durch Licht-, Soundsignale oder haptisches Feedback umgesetzt werden. Daher wurde im Rahmen des Projekts ein Vibration-Agent und eine benutzerfreundliche API zur Ansteuerung des kabellosen Vibrationsarmbands umgesetzt. Konzeption und Entwicklung eines Prototypen eines kabellosen Vibrationsarmbands für ein zeitnahes haptisches Nutzerfeedback im Rahmen des Projekts entwickelt (vgl. [22]).

Risiken Ein weiteres Risiko bildet die Möglichkeit der Datenerfassung der Sensorik. Grundsätzlich kann eine kamerabasierte Gestenerkennung nicht alle Bereiche einer Wohnung abdecken. Dies schränkt die Usability ein und kann schnell zur Ablehnung bei den Nutzern führen. Daher soll das System vorerst nur in einem dem Nutzer bekannten Bereich der Wohnung zur Verfügung gestellt werden. Im Living Place Hamburg würde sich z. B. das Wohnzimmer anbieten. Dieser Bereich könnte von mehreren Kinect und Leap Motion Sensoren abgedeckt werden.
Durch die Ergebnisse der von Karolina Bernart durchgeführten Wizard-of-Oz Experiments (vgl. [4]) müssen weitere Gesten dem System hinzugefügt werden. Sollten dabei komplizierte Bewegungsabläufe genutzt werden, besteht die Gefahr, dass diese Gesten sich nur schwer über ein heuristikbasierendes Erkennungsverfahren identifizieren lassen. Ebenso muss überprüft werden, welchen konkreten Technologien für die Selektion eines Objekts bei kontextsensitiven Befehlen zum Einsatz kommen können.
Ferner muss beachtet werden, dass keine der hier vorgestellten Studien eine längere Nutzung des Systems von den Probanden erforderte. Daher können aktuell keine Aussagen über die physikalische Ergonomie oder die Langzeitakzeptanz der Gesten getroffen werden. Dementsprechend sind eigene Vorstudien mit trainierten und untrainierten Probanden durchzuführen. Dabei könnte man ebenfalls einen Vergleich zwischen intuitiven und effektiven Gesten erstellen. Daher muss das System robust über mehrere Stunden zuverlässig Daten aufnehmen und verarbeiten können.

6.2 Fazit und Ausblick

Im Zuge des Masterseminars konnte das Thema der Masterarbeit skizziert werden. Die Mehrheit der Software Komponenten welche einen vertikalen Architektur Schnitt vom Sensor bis zum physikalischen Aktor (Fester, Licht und Rollos im Living Place) bilden, konnten bereits umgesetzt und getestet werden. Dies ermöglicht nach der Fertigstellung des Hauptprojekts, die durch Karolin Bernat identifizierten Gesten (vgl. [4]) zu implementiert und gemeinsam eine erste Nutzerstudie auf Basis der Integrationsplattform durchgeführt. Zudem

sollen weitere Agenten integriert werden um weitere Kontextdaten wie beispielsweise die Position, die Blickrichtung und die Emotion des Nutzers mit zu berücksichtigen. Im Anschluss können verschiedenen Hypothesen bzgl. des Zusammenhangs von Gestik und Kontext sowie vergleiche zwischen verschiedenen Verfahren zur Gestenerkennung evaluiert werden.
In der Masterarbeit wird daher die These untersucht, ob eine leicht erweiterbare Integrationsplattform für HCI-Studien umsetzbar ist, mit welcher verschiedene Verfahren, der Gestenerkennung unter Berücksichtigung des Kontextes, verglichen und evaluiert werden können.

References

1. Dimitra Anastasiou, Cui Jian, and Christoph Stahl. A german-chinese speech-gesture behavioural corpus of device control in a smart home, 2013.
2. Dimitra Anastasiou, Cui Jian, and Desislava Zhekova. Speech and gesture interaction in an ambient assisted living lab, 2012.
3. Hillel Aviezer, Ran R Hassin, Jennifer Ryan, Cheryl Grady, Josh Susskind, Adam Anderson, Morris Moscovitch, and Shlomo Bentin. Angry, disgusted, or afraid? studies on the malleability of emotion perception. *Psychological Science*, 19(7):724–732, 2008.
4. Karolina Bernat. Entwicklung einer gestensteuerung in einer smart-home umgebung. Website, 2015. Erreichbar online unter `http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2015-proj/bernat.pdf`; besucht am 06.09.2015.
5. Joachim Boetzer. Bewegungs- und gestenbasierte applikationssteuerung auf basis eines motion trackers. Website, 2008. "Erreichbar online unter `http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/bachelor/boetzer.pdf`; besucht am 23.10.2013.".
6. Justine Cassell. A framework for gesture generation and interpretation: Computer vision in human-machine interaction; cambridge university press. Website, 1998. "Erreichbar online unter `http://www.media.mit.edu/gnl/publications//gesture_wkshop.pdf`; besucht am 25.07.2014.".
7. Antonio Criminisi, Jamie Shotton, and Ender Konukoglu. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends in Computer Graphics and Vision: Vol. 7: No 2-3, pp 81-227*, 2012.
8. K. Dorscheid. Kultur mal anders: Gesten aus aller welt. Website, o.J. "Erreichbar online unter `http://www.geo.de/GEOlino/mensch/kultur-mal-anders-gesten-aus-aller-welt-59416.html`; besucht am 03.12.2013.".
9. Paul Dourish. What we talk about when we talk about context. *Personal Ubiquitous Comput.*, 8(1):19–30, February 2004.
10. Paul Ekmann. *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.
11. David Fleer and Christian Leichsenring. Miso: A context-sensitive multimodal interface for smart objects based on hand gestures and finger snaps, 2012.
12. Sobin Ghose. Anwendungen 1 Ausarbeitung - Kontextabhaengige Interpretation von 3D-Gesten, 2014.
13. Sobin Ghose. Anwendungen 2 Ausarbeitung - Multimodale Haussteuerung, 2015.
14. Sobin Ghose. Projekt 1 Ausarbeitung - Multimodale Haussteuerung. Website, 2015. Erreichbar online unter `http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2015-proj/ghose.pdf`; besucht am 13.02.2016.
15. Sean Gustafson, Daniel Bierwirth, and Patrick Baudisch. Imaginary interfaces: Spatial interaction with empty hands and without visual feedback, 2010.
16. Frank Honold, Felix Schüssel, Florian Nothdurft, and Peter Kurzok. Companion technology for multimodal interaction, 2012.
17. Arun Kulshreshth and Joseph J. LaViola, Jr. Exploring the usefulness of finger-based 3d gesture menu selection. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 1093–1102, New York, NY, USA, 2014. ACM.

18. John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data, 2001.
19. Eike Langbehn, Tobias Eichler, Sobin Ghose, GerdBruder Kai von Luck, and Frank Steinike. Evaluation of an omnidirectional walking-in-place user interface with virtual locomotion speed scaled by forward leaning angle. 2015.
20. Joseph J. LaViola, Jr. *An Introduction to 3D Gestural Interfaces.* SIGGRAPH '14. ACM, New York, NY, USA, 2014.
21. D. McNeill. *Hand and Mind: What Gestures Reveal about Thought.* University of Chicago Press, 1992.
22. Ariza Oscar, Lubos Paul, and Steinicke Frank. Hapring: A wearable haptic device for 3d interaction, 2015.
23. V.I. Pavlovic, R. Sharma, and T.S. Huang. Visual interpretation of hand gestures for human-computer interaction: a review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):677–695, 1997.
24. Olaf Potratz. Ein system zur physikbasierten interpretation von gesten im 3d-raum. Website, 2011. "Erreichbar online unter `http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/bachelor/potratz.pdf`; besucht am 20.8.2013.".
25. Olaf Potratz. Ein framework für physikbasierte 3d interaktion mit großen displays. Website, 2014. "Erreichbar online unter `http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/master/potratz.pdf`; besucht am 20.2.2016.".
26. Mark Weiser. The computer for the 21st century. *SIGMOBILE Mob. Comput. Commun. Rev.*, 3(3):3–11, July 1999.
27. B. Williamson, C. Wingrave, J. Laviola, T. Roberts, , and P. Garrity. Natural full body interaction for navigation in dismounted soldier training. in interservice/industry training, simulation, and education conference (i/itsec 2011), 2103–2110. 2011.
28. WIWITA. 6. wismarer wirtschaftsinformatik-tage. Website, 2008. Erreichbar online unter `http://www.wi.hs-wismar.de/~cleve/wiwita.html`; besucht am 05.12.2013.