Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Masterarbeit

Arne Bernin

Einsatz von 3D-Kameras zur Interpretation von räumlichen Gesten im Smart Home Kontext

Fakultät Technik und Informatik
Department Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Arne Bernin

Einsatz von 3D-Kameras zur Interpretation von räumlichen Gesten im Smart Home Kontext

Masterarbeit eingereicht im Rahmen der Masterarbeitprüfung
im Studiengang Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. rer. nat. Gunter Klemke
Zweitgutachter : Prof. Dr. rer. nat. Kai von Luck

Abgegeben am 15. September 2011

Arne Bernin

Thema der Masterarbeit

Einsatz von 3D-Kameras zur Interpretation von räumlichen Gesten im Smart Home Kontext

Stichworte

Dreidimensionale Gesten, Usability, Responsiveness, Alltagstauglichkeit, Smart Home, Living Place Hamburg, Latenzen, 3D-Kamera, HCI

Kurzzusammenfassung

Diese Arbeit beschäftigt sich mit dem Einsatz von 3D-Kameras zur Gestenerkennung im Kontext des Smart Home. Schwerpunkt der Arbeit sind dabei die Anforderungen an eine Alltagstauglichkeit aus den Bereichen Datenschutz und Usability und daraus folgend: Die Latenz von 3D-Kameras. Die im Living Place (Smart Home Labor der Hochschule für Angewandte Wissenschaften Hamburg) vorhandenen 3D-Kamerasysteme werden verglichen und ihre Latenzzeiten untersucht. Verfahren und Schnittstellen zur Gestenerkennung werden erläutert und die sich ergebenden Kriterien für die Beurteilung der Alltagstauglichkeit zusammengefasst. Abschließend wird ein Entwurf für ein System zur Erkennung räumlicher Gesten im Living Place Hamburg vorgestellt.

Arne Bernin

Title of the thesis

Employment of 3D-cameras for the interpretation of of spatial gestures in the smart home context

Keywords

3D-gestures, usability, responsiveness, suitability for everyday use, smart home, Living Place Hamburg, latency, 3D-camera, HCI

Abstract

This thesis deals with the requirements for the suitability of 3D gesture recognition in a smart home. The work focuses on the requirements for everyday usage in the area of usability and data privacy protection. Available 3D camera systems in the *Living Place* (Smart Home laboratory at Hochschule für Angewandte Wissenschaften Hamburg) are compared and examined for their latency periods. Procedures and interfaces for gesture recognition are explained and the resulting criteria for assessing the suitability for everyday use are summarized. Finally, a conceptual design for a system to detect spatial gestures in the *Living Place* is presented.

Danksagung

Vielen Dank an Prof. Dr. Gunter Klemke und Prof. Dr. Kai von Luck für die Betreuung dieser Arbeit und die zahlreichen Anregungen.

Danke an Frank, Anja, Siggi, Leif, David, Siva, Imme und Feli für die Hilfe bei der Durchsicht dieser Arbeit.

Und nicht zuletzt: Einen Dank an das Celtic Music Radio[1] für lange gemeinsame Nächte.

[1] http://www.celticmusicradio.net/

Inhaltsverzeichnis

1 Einführung

1.1 Ein etwas anderer Morgen für Sal

Sal wacht auf: Sie riecht frischen Kaffee. Einige Minuten zuvor hatte der Wecker - alarmiert durch den unruhiger werdenden Schlaf - bereits leise gefragt: "Kaffee?", und als Sal nicht reagierte, das Weckgeräusch verstärkt. Die darauf folgende Geste des Werfens mit einem imaginären Gegenstand in Richtung des Weckers wurde von der Haussteuerung korrekt als „Drücken der Schlummertaste“ interpretiert und der Wecker angewiesen, es einige Minuten später noch einmal zu probieren. Dieses Mal mit dem Ergebnis, dass Sal die Geste unter Einsatz eines wirklichen Kopfkissens verstärkt, was ihr weitere fünf Minuten im Bett ermöglicht. Nach einer weiteren Pause ertönt der Wecker erneut, und Sal steht auf.

Das Licht im Badezimmer schaltet sich automatisch ein, Sal, die sich noch zu müde für die Helligkeit fühlt, dimmt es mit einer Handbewegung. Das Wasser beginnt zu fließen als sie ihre Hände unter den Hahn hält, und stoppt genauso wieder, als sie die Hände zurückzieht und das Bad verlässt.

Abbildung 1.1: Blick aus dem Fenster. Quelle: [Decker (2009)]

Sal blickt aus dem Fenster auf ihre Nachbarschaft. Durch eines der Fenster fällt Sonnenlicht und sie blickt auf einen Zaun, doch durch die anderen sieht sie elektronisch markierte Pfade, erzeugt durch das Kommen und Gehen ihrer Nachbarn am frühen Morgen. Datenschutz-Konventionen und praktikable Datenraten verhindern eine Videoansicht, aber durch die Zeitmarkierungen und elektronische Spuren auf der Karte ihres Wohnviertels fühlt sich Sal in ihrer Straße wohl. Etwa anderthalb Meter vor den Fenstern stehend macht Sal eine wegschiebende Geste mit Ihrer Hand, und die projizierten Zeichen auf den Fenstern verschwinden.

1.2 Einleitung

Die vorherige Vision basiert auf dem Auszug eines Textes von Mark Weiser (Weiser (1995)), geschrieben im Jahre 1995. Darin wagt er den Entwurf einer Vision für die Zukunft der Kommunikation zwischen Mensch und Maschine und die Integration von allgegenwärtigen Computern in einen kommenden Alltag (*ubiquitous computing*). Die im originalen Text verwendete Interaktion per Sprache wurde auf Gestensteuerung adaptiert um ein Beispiel für die Integration in den Alltag zu geben.

Die Entwicklung der informationsverarbeitenden Maschinen ist untrennbar verbunden mit der Entwicklung der Schnittstellen zum sie benutzenden Menschen. Die Schnittstellen zwischen Mensch und Maschine entwickelten sich beispielsweise von einfachen mechanischen Schaltern (Abbildung 1.2) zu berührungsempfindlichen Bildschirmen von mobilen Computern (Abbildung 1.3).

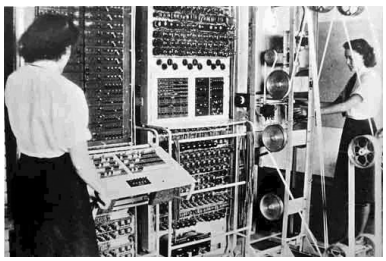


Abbildung 1.2: Colossus. Quelle: [Public record office, London (1943)]

Durch die Omnipräsenz von Computern und ihrer Nutzung hat sich auch die Wahrnehmung durch ihre Benutzer verändert. Der Computer ist nicht länger Arbeitsgerät für Wenige, sondern wird als ins Leben integriertes Gerät des normalen Alltags gesehen. Nach wie vor ist der Mensch dabei jedoch mehr oder weniger dem Diktat der maschinellen Verarbeitung der Kommunikation unterworfen, nicht der Benutzer entscheidet, wie diese abläuft, sondern das Computersystem oder vielmehr seine Entwickler.

Abbildung 1.3: iPad2. Quelle: [Downey (2011)]

Es werden Begriffe wie *intuitiv* oder *natürlich* verwendet, die den Eindruck vermitteln sollen, noch nie sei es so einfach gewesen, einen Computer zu bedienen wie jetzt. Die Frage, die sich aufdrängt, ist: Ist dies wirklich so, und wenn ja, an welchen Kriterien lässt sich dies festmachen?

Die meisten *alten* Eingabegeräte (wie Maus und Tastatur und Pen) werden weiterhin verwendet. Hinzu kamen in jüngerer Zeit *touch-screen* und *2D-Gestik*. Nun wird die Mensch-Maschine-Kommunikation um die Erkennung dreidimensionaler Gesten erweitert. Sind diese eine zusätzliche Möglichkeit oder können sie die bisherigen Eingabegeräte ersetzen?

Die Frage der Relevanz der Gestenerkennung als Teil von zukünftiger Mensch-Maschine-Interaktion wird sowohl im wissenschaftlichen Kontext als auch im industriellen Umfeld diskutiert. Wie dies beispielsweise Steve Ballmer 2010 auf der CES[1] ausführte:

> "I believe we will look back on 2010 as the year we expanded beyond the mouse and keyboard and started incorporating more natural forms of interaction such as touch, speech, gestures, handwriting, and vision— what computer scientists call the 'NUI' or natural user interface."
>
> Steve Ballmer, CEO Microsoft (siehe Ballmer (2010))

Bisher war die dreidimensionale Gestenerkennung – durch den umfangreichen technischen und finanziellen Aufwand für die Beschaffung der notwendigen Geräte – auf einen kleinen Nutzerkreis beschränkt.[2] Seit kurzem sind jedoch auch Produkte für den Endverbraucher kommerziell verfügbar.

1.3 Begriffsklärung

1.3.1 Räumliche Gesten

Räumliche Gesten bezeichnet die Verwendung von physischen Gesten als Ausdrucksform der Kommunikation zwischen Menschen und Maschinen.[3] Konträr zu zweidimensionalen Gesten auf einer Oberfläche (bei einem Surface-Tisch[4] oder einem Smartphone) besteht dabei kein Kontakt zu dieser Oberfläche.

Der Fokus der Gestik liegt dabei auf der Verwendung der Extremitäten, hauptsächlich der Hände. Beispielsweise für Videospiele[5] ist auch der Einsatz weiterer Körperteile zur Steuerung denkbar.

Im Zusammenhang dieser Arbeit sind mit *räumlichen Gesten* Gesten der oberen Extremitäten, also der Arme und Hände, im dreidimensionalen Raum gemeint.

An der Hochschule für Angewandte Wissenschaften Hamburg wird seit dem Jahre 2002 im Bereich Gestik und neue Interaktionstechniken geforscht.[6] Mehr über die Klassifikation von (räumlichen) Gesten findet sich in Abschnitt 7.2.

[1] Consumer Electronics Show, eine Messe in Las Vegas
[2] Die Time-of-Flight-Kamera (Zum Verfahren siehe 5.5) SR4000 kostet mehr als 7000 Euro
[3] Die Begriffe *Maschine* und *Computer* werden in dieser Arbeit synonym verwendet
[4] Microsoft Surface, ein *interaktiver Tisch*
[5] Siehe Szenario 3.1
[6] Siehe Abschnitt 2.3.2 für eine Übersicht der Veröffentlichungen zum Thema

1.3.2 3D-Kamera

3D-Kamera oder *dreidimensionale Kamera* bezeichnet im Kontext dieser Arbeit Kamerasysteme, die in der Lage sind, Informationen über die Entfernung eines abgebildeten Objektes zu erfassen und diese Informationen zur weiteren Auswertung zur Verfügung zu stellen. Dreidimensional bedeutet hier nicht, ein Objekt als dreidimensionales Modell (von allen Seiten) erfassen zu können. Diese im Rahmen von Bewegungserfassungen für Sportwissenschaft oder Spielfilme verwendete vollständige Erfassung sind für die Erkennung von räumlichen Gesten nicht nötig[7].

Im Vergleich mit diesen Techniken, wäre auch der Begriff *2,5-dimensionale* Kameras verwendbar. Einzelne Kameras liefern eine Ebenen-Sicht auf das Objekt (zweidimensional) und können für jeden der zugehörigen Punkte die Entfernung angeben (die *halbe* Dimension) und eben nicht ein dreidimensionales Modell der aufgenommenen Objektes. Nur durch Verwendung mehrerer Kameras ist dies möglich. Die in dieser Arbeit betrachteten Systeme liefern allerdings sehr wohl Koordinaten im dreidimensionalen Raum für jeden Bildpunkt. Somit ist die Bezeichnung *dreidimensional* durchaus gerechtfertigt.

1.3.3 Smart Home

> "A dwelling incorporating a communications network that connects the key electrical appliances and services, and allows them to be remotely controlled, monitored or accessed." intertek.com (2003)

Laut Definition von *intertek*[8] ist ein *Smart Home* also ein Ort zum Leben, in dem die wichtigsten elektrischen Geräte und Einrichtungen durch ein Netzwerk verbunden sind. Dieses kann ferngesteuert kontrolliert, überwacht und benutzt werden. In der *Wohnung der Zukunft* sind also beispielsweise Licht, Küchengeräte, Musikanlage und Terminkalender über gemeinsame Schnittstellen mit vorhandenen Rechnern vernetzt.

Smart Home und die deutsche Übersetzung *intelligente Wohnung* werden in dieser Arbeit synonym verwendet.

[7] Zu Verfahren für die Erkennung von Gesten siehe Kapitel 7

[8] Intertek ist ein weltweit tätiges Forschungsunternehmen, vergleichbar mit der Frauenhofer Gesellschaft in Deutschland

Living Place Hamburg

Der *Living Place Hamburg* ist ein Labor an der Hochschule für Angewandte Wissenschaften Hamburg (HAW), das im Kontext der intelligenten Wohnung anzusiedeln ist. Hier bietet sich die Möglichkeit, Konzepte für zukünftige Interaktion und Dienste von Wohnungen mit ihren Bewohnern zu entwickeln und zu testen. Im Unterschied zu anderen Laboren zur Messung der Usability eines Soft- und Hardwaresystems an der HAW handelt es sich beim Living Place um eine komplett eingerichtete Wohnung.

Abbildung 1.4: Living Place Hamburg, Außenansicht. Quelle: [HAW Hamburg]

Trotzdem bietet es durch eingebaute Kameras die Chance, Interaktionen der Benutzer zu beobachten, aufzuzeichnen und zu analysieren.

Der Living Place bietet die Möglichkeit, Konzepte zu einem Prototyp zu entwickeln und diesen Prototyp in einer realitätsnahen Umgebung zu testen.

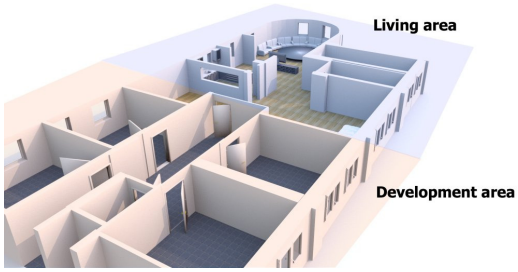


Abbildung 1.5: Lving Place Hamburg, Aufbau. Quelle: [HAW Hamburg]

1.4 Einordnung in den Kontext der Informatik

> "HCI is the study and theory of the interaction between humans and complex technology (usually computers)"
>
> Booth (1980)

Die Begriffe HCI (für *Human Computer Interface* oder *Human Computer Interaction*) sowie MMI (*Man Machine Interaction*) sowie CHI (*Computer and human interaction*) werden synonym verwendet. Im Kern beschäftigt sich HCI mit der Kommunikation zwischen Mensch und

Maschine und den vorhandenen oder zukünftigen Techniken für diese Kommunikation. Dafür werden verschiedene Fachgebiete wie Software Engineering, künstliche Intelligenz und verschiedene Bereiche der Psychologie bemüht (Booth, 1980, Seite 4-18).

1.4.1 Geschichte der Mensch-Computer-Interaktion

Tabelle 1.1 bietet eine Übersicht über die verschiedenen Formen der Mensch-Computer-Schnittstelle durch die verschiedenen Jahrzehnte nach Nielsen (1994).

Tabelle 1.1: Generationen der Mensch-Computer-Schnittstelle, Quelle: Nielsen (1994)

	Zeitraum[a]	Eingabe	Ausgabe	Paradigma
0	bis 1945	Festverdrahtung, Lochkarten	Lampen	Keins (Direkte Interaktion mit der Hardware)
1	1945-1955	tty[b], Schreibmaschine	Drucker	Progammierung, Batch
2	1955-1965	Zeilenorientierte Terminals	Glass tty[c]	Kommandosprachen
3	1965-1980	Tastatur	Ausgabe über Textbildschirm	Full-screen, strikt hierarchische Menüs und Formulare
4	1980-1995	Fenster, Maus und Tastatur	Grafischer Bildschirm	WIMP (Windows, Icons, Menu and Pointing Device)
5	ab 1995	Gesten	Grafischer Bildschirm	Nicht Kommandobasiertes Interface

[a] Zeitraum bezieht sich dabei auf die Zeit der ersten Einführung der Technologie
[b] teletypewriter
[c] Ein Glass tty ist ein primitives Terminal ohne eigene CPU, vergleichbar einem Fernschreiber mit Bildschirm.

Myers (1998) verweist in seiner Betrachtung der Entwicklung von HCI auf den langen Vorlauf, den neue Formen der Interaktion im Rahmen der akademischen und industriellen Forschung vor ihrer Einführung als kommerzielle Produkte haben (siehe Abbildung 1.6). Der Beginn von Gestenerkennung (*Gesture Recognition*) bezieht sich in diesem Fall auf die Verfügbarkeit von Systemen, die auf Eingabegeräten wie dem *Lightpen*[9] basieren. Ein erstes System aus dem akademischen Bereich, basierend auf einem Lightpen, war *Sketchpad* von Ivan Sutherland (Sutherland (1964)).

[9] Ein Eingabegerät zum *Zeichnen* auf dem Bildschirm, dass mit Hilfe einer Fotodiode seine Position am Bildschirm bestimmt.

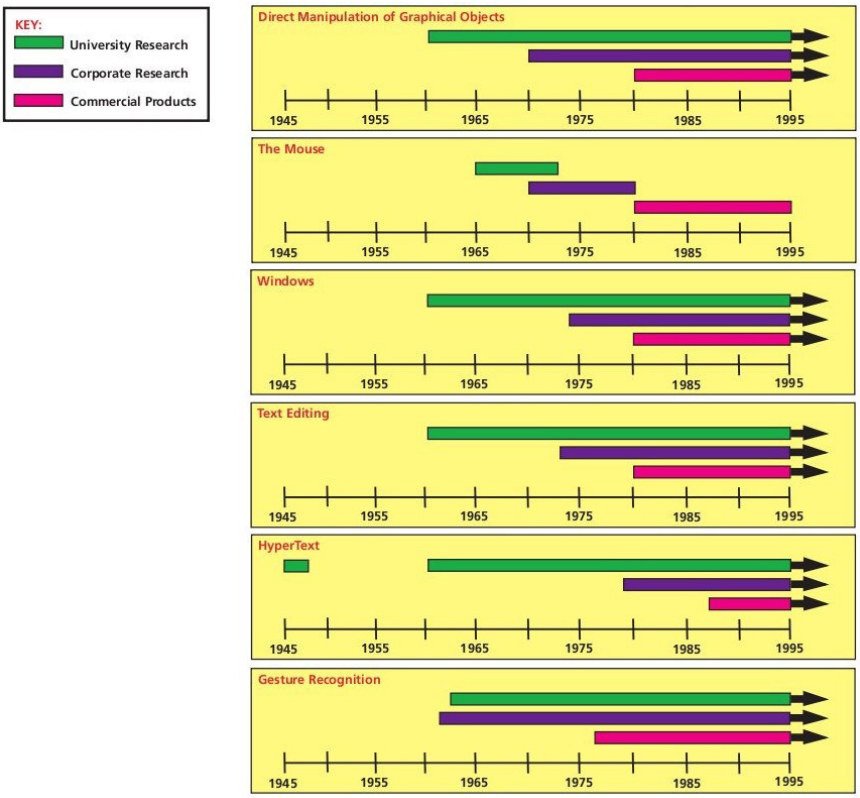


Abbildung 1.6: Entwicklung von User Interfaces nach Myers, Quelle: [Myers (1998)]

1.5 Aufbau der Arbeit

Der Aufbau dieser Arbeit gliedert sich in zehn Kapitel, die im Folgenden kurz vorgestellt werden. Kapitel 2 beschreibt die genaue Aufgabenstellung. Darauf folgt eine Betrachtung (Kapitel 3) der Anwendungsbeispiele im Smart Home Kontext als Grundlage für die weitere Arbeit. Im nachfolgenden Kapitel 4 werden die Anforderungen diskutiert, die sich aus allgemeinen Anforderungen aus dem Bereich Usability und im Besonderen des Anwendungsfelds ergeben. Die zur Erfassung von Gesten im dreidimensionalen Raum verwendbaren Kameras werden in Kapitel 5 diskutiert. Dabei werden die im Living Place vorhandenen Systeme sowie ihre jeweiligen Vor- und Nachteile betrachtet.

Entscheidend für die Beurteilung der Nutzbarkeit der vorhandenen Kamerasysteme ist die Frage der Latenz dieser Kameras. Dieser Frage wird in Kapitel 6 nachgegangen und dabei die durchgeführten Messungen sowie die dafür verwendete Software vorgestellt.

in Kapitel 7 folgt eine Einführung in Verfahren zur Gestenerkennung und ein Vergleich der vorgestellten Algorithmen. Das darauf folgende Kapitel 8 beschäftigt sich mit den technischen Schnittstellen auf Hard- und Software-Ebene. Kapitel 9 fasst die aus der Analyse gewonnen Kriterien für die Beurteilung von Systemen zur dreidimensionalen Gestenerkennung zusammen und listet die erkannten offenen Fragen und Möglichkeiten zukünftiger Forschung im Themengebiet auf. Darauf basierend wird in Kapitel 10.1 ein Design für die weitere Erforschung der Gestenerkennung im Living Place skizziert. In Kapitel 11 wird die Arbeit zusammengefasst und ein Fazit gezogen.

Es folgen ein Abbildungsverzeichnis (Seite 131), Literaturverzeichnis (Seite 134) und Glossar (Seite 149).

2 Einordnung der Arbeit in den Gesamtkontext

2.1 Genaue Aufgabenstellung

Diese Arbeit beschäftigt sich mit den Kriterien und Anforderungen an alltagstaugliche dreidimensionale Gestenerkennung im Smart Home. Die sich daraus ergebenden Fragen sind:

- Was sind die Anforderungen für eine alltägliche Benutzung?
- Welche Anwendungsbeispiele gibt es dafür?
- Welche Problemfelder müssen beachtet werden? Welche Lösungen gibt es für Probleme?
- Welche im Living Place vorhandenen (Teil-)Systeme erfüllen die Anforderungen?
- Welche Kamerasysteme sind am besten geeignet?
- Wie ist der Stand der Forschung? Welche Untersuchungen wurden bereits durchgeführt und wie sollten geeignete Untersuchungen und Verfahren aussehen, um Lücken zu schließen?

Diese Arbeit soll Anworten auf die oben genannten Fragen erarbeiten. Dazu sollen zuerst sinnvolle Beispiele für eine Anwendung im Living Place gefunden und die sich daraus ergebenden Anforderungen and die Hard- und Software ermittelt werden. Von den Anforderungen ausgehend soll eine Evaluierung der vorhandenen Hard-, Software und Verfahren zur Gestenerkennung erfolgen. Abschließend soll ein Designvorschlag für ein System zur Erkennung räumlicher Gesten im Living Place Hamburg entwickelt werden, der als Grundlage für weitere Untersuchungen in diesem Bereich an der HAW dienen kann.

2.2 Relevanz

> „There is strong evidence that future human-computer interfaces will enable more natural, intuitive communication between people and all kinds of sensor-based devices, thus more closely resembling human-human communication.“
>
> Wachs u. a. (2011)

Bisher war die Verfügbarkeit von Systemen zur Erfassung von Bewegungen und Objekten im dreidimensionalen Raum hauptsächlich auf den akademischen Rahmen beschränkt[1]. Mit der Verbreitung des Kinect-Systems für die XBox-Spielekonsole ist ein System kommerziell verfügbar und millionenfach[2] im Einsatz, dass mit den bisherigen Kamerasystemen in der Leistung vergleichbar ist[3]. Die Verwendung der dritten Dimension ist damit, zumindest bei Computerspielen, Realität. Hierdurch rückt die Frage in den Fokus, wie sich dies auf die Kommunikation zwischen Mensch und Maschine auswirkt.

2.3 Verwandte Arbeiten

2.3.1 Allgemein

Es gibt einige Arbeiten, die eine generelle Einführung und Übersicht über die Methoden und Geräte zur Erkennung von Gesten liefern. Hassanpour u. a. (2008) beschäftigen sich skizzenhaft mit Methoden zur Analyse, Modellierung und Erkennung von Handgesten im Kontext der Mensch-Maschine-Interaktion. Deutlich umfangreicher setzen sich Wu u. Huang (1999) mit derselben Thematik auseinander und bieten eine Übersicht über den Stand der Technik und Forschung am Ende der 90er Jahre.

Wachs u. a. (2011) hingegen beschreiben den aktuellen Stand der Technik und ihrer Anwendungen, wobei sie auch auf die Usability eingehen. Einen Weg der Kombination von unterschiedlichen Modalitäten[4] als Ergänzung zur reinen Gestik bieten Jaimes u. Sebe (2007) in ihrer Untersuchung über Techniken zur Multimodal Human Computer Interaction (MMHCI).

[1] Siehe Beispiele für 3D-Stereo in den Abschnitten 5.4.2 und 5.4.5 sowie 5.5 für Time-of-Flight-Kameras

[2] Laut Microsoft wurde die Kinect-Steuerung 8 Millionen mal innerhalb der ersten 60 Tage nach ihrem Erscheinen im November 2010 verkauft (pcgames).

[3] Ein Vergleich der verschiedenen Systeme findet sich in Tabelle 5.3.

[4] Modalität bedeutet in diesem Kontext einen Eingabekanal, also beispielsweise Sprache, Gestik, Tastatur, Maus.

Es werden in der Literatur zahlreiche Systeme zur Erkennung von Handgesten vorgestellt. Beispielsweise stellen Rahman u. a. (2009) ein von Ihnen entwickeltes System zur Gestenerkennung, basierend auf einem Infrarotstrahlung emittierenden Handschuh und einer entsprechenden Kamera vor, das sich mit Hilfe ihres zuvor entwickelten Frameworks für *Ambient Media Services* (Hossain u. a. (2009)) zur Erkennung dreidimensionaler Gesten eignet.

Neßelrath u. a. (2011) integrieren ihr auf einer Wiimote[5] basierendes System zur Gestenerkennung in die *Smart Kitchen* am Deutschen Forschungscenter für künstliche Intelligenz in Saarbrücken. Diese Umgebung ähnelt dem Living Place an der HAW Hamburg. Dabei kommt als Orientierung für die Implementierung der ISO-Standard 24752 zum Einsatz, ein Standard, der die Kommunikation verschiedener Komponenten im Smart Home sicherstellen soll.

Karam (2006) veröffentlichte 2006 eine Doktorarbeit zum Thema “A framework for research and design of gesture-based human computer interactions”. Im Rahmen dieser Arbeit werden, neben einer umfassenden Übersicht über Veröffentlichungen in diesem Bereich, Versuche zur Toleranz von Benutzern gestenbasierter Systeme durchgeführt. Dabei setzt sich Karam mit der Frage auseinander, welche Erkennungsraten ein solches System bieten muss (siehe Abschnitt 4.2.6).

2.3.2 HAW Hamburg

An der Hochschule für Angewandte Wissenschaften Hamburg wurden schon mehrfach Arbeiten in diesem Bereich durchgeführt. Lorenzen (2005) beschäftigt sich im Rahmen seiner Diplomarbeit mit der Erkennung von zweidimensionalen Gesten der Hand auf Grundlage einer Videokamera und dynamischer Programmierung. Senkbeil (2005) beschreibt die Verfolgung von farbigen Objekten zur Erkennung zweidimensionaler Gesten[6].

Heitsch (2008) entwickelte im Rahmen seiner Bachelorarbeit ein System zur Gestenerkennung für “Computer Supported Collaborative Workplaces” mit Hilfe des ARTtrack-Systems[7]. Boetzer (2008a) beschreibt die Entwicklung einer Ansteuerung von Programmen (Google Earth, Google Maps und eines Squash-Spiels) ebenfalls für das ARTtracker System. Sowohl Potratz (Potratz (2011)) als auch Rödiger (Rödiger (2010)) beschäftigten sich gleichwohl auf Grundlage der ARTtracker mit der Erkennung dreidimensionaler Gesten. Boetzer (2008b) fasst erste Forschungsergebnisse an der HAW Hamburg zu dieser Thematik zusammen.

[5] Wiimote ist der Name der Fernsteuerung für die Nintendo Wii Spielekonsole

[6] Das Objekt, im konkreten Fall ein farbiger Handschuh, bewegt sich zwar im dreidimensionalen Raum, die Entfernung von der Kamera wird allerdings nicht als Parameter herangezogen

[7] Ein System zum Motion Capturing der Firma advanced realtime tracking GmbH (http://www.ar-tracking.de/) basierend auf Infrarotkameras mit aktiver Beleuchtung und passiven Markern

2.3.3 Abgrenzung

Alle diese Arbeiten beschäftigen sich zu unterschiedlichen Anteilen mit allgemeinen Fragestellungen der Gesteninteraktion sowie auf dem damals vorhandenen Stand der Technik. Lediglich Wachs u. a. (2011) beziehen Fragen der Usability und neuere Kamerasysteme – wie das Kinect-System – in ihre Überlegungen mit ein, ohne allerdings auf konkrete Anforderungen einzugehen. Im Unterschied dazu beschäftigt sich diese Arbeit mit den anschaulichen Anforderungen sowie Fragen des Datenschutzes und der Integration in ein bestimmtes Smart Home, den Living Place an der HAW Hamburg. Die bisher an der HAW durchgeführte Arbeiten verwendeten außerdem das ARTtracker-System und keine dreidimensionalen Kameras.

3 Anwendungsbeispiele

Dieses Kapitel führt einige mögliche Anwendungsszenarien für die Verwendung von Gestenerkennung auf, die im Living Place Hamburg umsetzbar sind. Dabei wird die Gestensteuerung mit konservativen Formen der Steuerung verglichen. Ziel ist es, zum besseren Verständnis des Anwendungsbereichs, einige konkrete Anwendungsfälle vorzustellen.

3.1 Spiele

Das bekannteste Szenario ist wohl derzeit die Verwendung von Körpergesten im Bereich Spiele. Die klassische Steuerung von Computer-Spielen erfolgt durch Eingabegeräte wie Gamepad, Joystick oder Tastatur. Dabei dient ein Bildschirm oder Beamer zur Anzeige des Spieles. Das Spielprogamm selber wird auf einer Spielekonsole oder einem PC ausgeführt. Es ist möglich, das Eingabegerät durch eine Gestenerkennung zu ersetzen. Dabei kommen Gesten mit Händen, Armen, Beinen oder dem gesamten Körper zum Einsatz. Die ermittelten Steuerinformationen werden entweder direkt verwendet oder auf die klassische Steuerung umgesetzt.

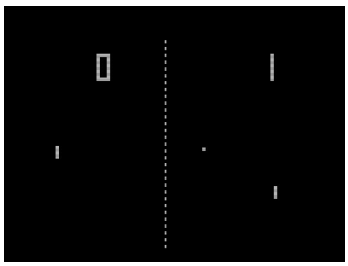


Abbildung 3.1: Computerspiel Pong Quelle: [User:Bumm13 (2006)]

Gestenerkennung kann, durch die direkt auf das Spielgeschehen übertragbaren Bewegungen, die Immersion erhöhen. Durch das verstärkte Eintauchen erhöht sich die Motivation der Spieler, sie fühlen sich mehr im Geschehen. Zudem kann der Zwang zur körperlichen Bewegung an sich schon einen gesundheitlichen Vorteil bieten.

Das gleichzeitige Spielen mit mehreren Spielern ist schwieriger umzusetzen, da die einzelnen Spieler von der Software erkannt werden müssen. Die Steuerung eines Spieles per Joystick oder Gamepad ist präziser und schneller, da die Bewegungen deutlich geringer ausfallen können und ein haptisches Feedback vorhanden ist. Aktuelle Systeme zur Gestenerkennung für Spiele, wie das Kinect-System von Microsoft sind auf eine Framerate von 30 Bildern pro Sekunde (*frames per seconde*, *fps*) limitiert, dies ist insbesondere bei schnelleren Bewegungen unpräzise. Die Steuerung mit dreidimensionalen Gesten ist nicht für alle Spielszenarien gleich

gut geeignet und auch nicht sinnvoll. Langes Spielen ohne viel Bewegung, wie etwa bei einem Strategiespiel, kann ermüden. Für Sportspiele ist sie jedoch geeignet und bietet neue Möglichkeiten zur Interaktion.

3.2 Vortrag/Diashow

Dieses Szenario schließt sowohl einen klassischen Vortrag vor Publikum als auch die *Diashow* (oder vielmehr ihren postmodernen Ersatz) im privaten, häuslichen Umfeld ein. Wir haben also meist einen abgedunkelten Raum, (mindestens) einen Vortragenden und eine Leinwand, auf der Fotos, Animationen oder Bilder zu sehen sind. Der Vortragende erläutert dabei das Gezeigte und ist auch für die Entscheidung verantwortlich, das nächste Bild anzeigen zu lassen. Als Anzeigegerät kommt hierfür häufig ein Beamer mit Computer zum Einsatz. Die Steuerung der Bildfolge wird klassischerweise mit Fernbedienung, Tastatur oder Funkmaus durch den Vortragenden vorgenommen.

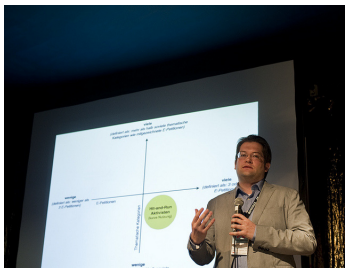


Abbildung 3.2: Vortrag Quelle: [Fischer (2011)]

Diese Eingabegeräte können durch die Verwendung von Gesten (beispielsweise eine *wegwischende* Bewegung der Hand) ersetzt werden. Dies ermöglicht den Verzicht auf Geräte in der Hand des Vortragenden – beispielsweise eine Fernbedienung – und ermöglicht die normale Unterstützung seines Vortrages durch Gesten gegenüber seinem Publikum. Hierbei ist die Trennung von Gestik im Rahmen des Vortrages und Kommandos an den Rechner die Herausforderung. Außerdem kann es durchaus sein, dass der Vortragende die Fernbedienung zum “Festhalten” nutzt und die Nervosität steigt, wenn die Person nichts mehr in der Hand hält.

3.3 Steuerung von Multimediageräten

Dieses Beispiel beinhaltet die Steuerung von Geräten zum Abspielen von Multimediainhalten wie Fernseher, Bluray/DVD-Spieler, Musikanlage oder Mediacenter. Die Geräte werden durch einzelne Fernbedienungen gesteuert, eine Fernbedienung für mehrere Geräte gleichzeitig gibt es nur bei identischem Hersteller oder in Form einer Universalfernbedienung. Typischerweise wird als Medium für die Übertragung Infrarotstrahlung benutzt.

Abbildung 3.3: Fernbedienung Quelle: [Wydra (2008)]

Ein möglicher Einsatz von Gesten wäre die Steuerung der einzelnen Funktionen per Kommando, also beispielsweise, Daumen hoch für mehr Lautstärke, eine „Wischgeste“[1] für das nächste Lied. Von Vorteil ist dabei, dass man die Fernbedienung nicht „verlegen“ kann. Auch ist das Szenario des gefundenen, aber leider mit inzwischen leerem Akku versehenen Bedienteils obsolet.

Problematisch wird es, wenn sich mehrere Personen im Raum befinden. Zum einen steigt die Gefahr von Missinterpretationen der normalen Interaktionen im Raum, zum anderen kann es zum Konflikt kommen, wenn zwei Personen gleichzeitig versuchen, den Fernsehkanal oder die Lautstärke zu ändern[2]. Diese Art der Bedienung findet in Situationen, in denen viele Menschen zugegen sind, wie etwa auf einer Party, seine Begrenzung. Alleine durch das Verstellen des Sichtfeldes durch andere Personen ist eine klare Erkennung von Personen nicht mehr möglich.

3.4 Musikinstrumente

(a) Theremin Etherwave Kit, Quelle:Theremin Etherwave Quelle: Hutschenreuther (2005)

(b) Yamaha DJX Keyboard Quelle: Schmallenberg (2006)

Abbildung 3.4: Theremin und Yamaha DJX

Die Idee, Musikinstrumente berührungslos durch die Veränderung der Position der Hände des Musikers zu bedienen, ist nicht neu. Bereits 1919 entwickelte der russische Physiker Lev Sergejewitsch Termen ein solches Instrument, genannt Theremin (Glinsky u. Moog (2005), siehe Abbildung 3.4a).

[1] Wischgeste bedeutet das schnelle Bewegen der Hand von rechts nach links (oder umgekehrt), als würde man etwas vom Tisch wischen

[2] Die meisten mit Geschwistern aufgewachsenen Menschen dürften noch die Kämpfe um die Fernbedienung des Fernsehers kennen.

Dabei werden die Tonhöhe und die Lautstärke jeweils durch den Abstand einer Hand zu zwei Antennen bestimmt. Zusätzlich beeinflussen die Bewegungen von in der Nähe befindlichen Personen die Tongeneration, die Bewegungen bewirken hierbei Kapazitätsänderungen in den an die Antennen angeschlossenen Schwingkreisen, wobei Antenne und Hand zwei Platten eines Kondensators darstellen.

Im Gegensatz dazu zeigt Abbildung 3.4b ein klassisches, wenn auch elektronisches, Musikinstrument in Form eines Keyboards. Dieses bietet, im Gegensatz zum Theremin, dem Musiker ein haptisches Feedback[3] durch die Tasten. In Form des Theremin, oder eines vergleichbaren Gerätes, das durch Gesten und Körperbewegungen gesteuert wird, erfolgt das Feedback nur durch die erzeugten Töne.

Die im Gegensatz zu Tasteninstrumenten wie einem Keyboard wenig offensichtliche Bedienung wird durch den Umstand ausgeglichen, dass Musikinstrumente sowieso erlernt werden müssen. Deshalb fällt die vorher nötige Erklärung des Konzeptes nicht wirklich ins Gewicht.

Ein Vorteil ist die Möglichkeit, ganz neue Formen der Verschmelzung von Musik und Tanz zu ermöglichen. So könnte sich Musik optimal an Bewegungen des Vorführenden anpassen, da sie durch eben diese erst erzeugt wird.

Castellano u. a. (2007) beispielsweise haben ein solches System auf Basis einer 2D-Videokamera und Software für Multimedia-Interaktion entwickelt.

3.5 Zusammenfassung

Die Eignung von dreidimensionaler Gestenerkennung ist abhängig vom Anwendungsbereich. Die vorgestellten Beispiele sind für die Gestenerkennung geeignet und wurden aus diesem Grunde ausgewählt. (Vermeintliche) Gegenbeispiele zu finden ist nicht schwer. Die Eingabe eines Textes durch Verwendung von Zeichensprache ist ein Beispiel für ein Anwendungsszenario, die für wenig Begeisterung beim *normalen* Benutzer führen dürfte, da diese präziser und schnell über eine Tastatur erfolgen kann. Bei Gehörlosen jedoch, würde ein System zur Erkennung von Gebärdensprache allerdings eine deutliche Erleichterung darstellen können. Es kommt also darauf sehr an, wer die zukünftigen Benutzer sind.

Die hier vorgestellten Beispiele beziehen sich auf die ausschließliche Verwendung von Gesten als Interaktionsform. Der dabei begrenzte Umfang an sinnvollen Anwendungsbeispielen lässt sich durch die Verwendung zusätzlicher Modalitäten erweitern, beispielsweise durch Kombination mit Sprache, Mimik oder dem Greifen eines bestimmten Gegenstandes. In diesen Szenarien ist Gestik dann eine Modalität der Interaktion unter mehreren.

[3] Haptisches Feedback entsteht durch die Berührung eines Gegenstandes.

4 Analyse der Anforderungen für 3D-Gesten

Dieses Kapitel beschäftigt sich mit den Voraussetzungen, die ein System zur Interpretation von räumlichen Gesten im Smart Home erfüllen sollte. Es werden die grundsätzlichen Anforderungen die sich aus dem Anwendungsfeld (Mensch-Maschine-Schnittstelle im Smart Home) ergeben, untersucht.

4.1 Smart Home

4.1.1 Ort zum Leben

Das *Smart Home* als *Wohnung der Zukunft* ist, neben der technischen Ausstattung, vor allem eine Wohnung. Das bedeutet, dass die Bewohner sich dort für lange Zeit aufhalten, teilweise fast ständig[1] und ihren Alltag dort leben. Insofern müssen die verwendeten Systeme auch alltagstauglich sein. Sie müssen entwickelt werden, um im Alltag zu funktionieren und nicht in einer Laborumgebung.

4.1.2 Alltagstauglichkeit

Die folgenden Anforderungen sollte das System für eine Alltagstauglichkeit erfüllen:

Mechanische Robustheit

Um dauerhafte Verwendung finden zu können, sind Anforderungen an die mechanische Ausführung zu stellen. Entweder müssen die Geräte so robust sein, dass ihnen auch Stürze nichts anhaben können, oder sie müssen fest installiert werden. Robustheit beinhaltet auch eine möglichst lange Lebensdauer der Gerätschaften. Alltägliche Tätigkeiten wie eine Reinigung dürfen die Systeme nicht beschädigen.

[1] Variierend, je nach Arbeitssituation, Gesundheitszustand, Alter, etc.

Technische Robustheit

Die Energieversorgung muss so beschaffen sein, dass eine ständige Verfügbarkeit gewährleistet ist. Dies bedeutet, dass keine Akkus verwendet werden können, da diese eine Ladezeit erfordern würden. Zudem muss eine ausreichende Kühlung vorhanden sein, und dies in einer Form, die auch in ruhigen Umgebungen nicht störend wirkt[2].

Die Energieversorgung muss so ausgelegt sein, dass eine Gefährdung durch Überhitzung auch bei dauerhaftem Betrieb ausgeschlossen wird.

Robustheit des Verfahrens

Die verwendeten Verfahren zur Gestenerkennung müssen robust sein. Eine notwendige manuelle Kalibrierung darf nicht häufig notwendig sein. Kalibrierungen müssen vom System möglichst selbständig vorgenommen werden, insbesondere wenn es, bei Verwendung mobiler Komponenten, zur Verschiebung der Positionen kommen kann.

Das verwendete Verfahren muss mit wechselnden Beleuchtungssituationen, wie sie im Vergleich von Tag und Nacht auftreten, problemlos funktionieren. Auch müssen verschiedene Situationen abgedeckt werden. Hierzu gehören: der Benutzer ist alleine zu Haus, er hat wenige Gäste, er hat viele Gäste (Party). Eine möglichst hohe Erkennungsrate (vergleiche Abschnitt 4.2.6) muss gewährleistet sein.

Unterscheidbarkeit von Personen

In einer realen Wohnung sind verschiedene Personen unterwegs, die teilweise Bewohner der Wohnung sind. Allerdings sollten auch Besucher in der Lage sein, die Einrichtungen der Wohnung zu bedienen. Dabei stellt sich die Frage, wie wird die Kontrolle über die Einrichtungen organisiert, wer darf die Steuerung ausüben? Soll diese Kontrolle beschränkt werden, so ist dafür ein Mechanismus, wie etwa eine Gesichtserkennung vorzusehen. Dabei ist der Schutz dieser Daten zu beachten (siehe auch Abschnitt 4.3).

[2] Ein ruhiger Wohnraum hat Abends eine Umgebungslautstärke von etwa 15-30 db, variierend je nach Lage der Wohnung (Moll (2011))

4.1.3 Schnittstellen

Es muss Schnittstellen zu anderen Einrichtungen in der Wohnung geben, da die Steuerung per Gesten in die bestehende Umgebung integriert werden soll. Beispiele hierfür finden sich in Kapitel 3. Die benötigten Schnittstellen müssen dokumentiert und frei verwendbar sein. Der Begriff Schnittstelle bezieht sich dabei sowohl auf die Software, als auch auf die Hardware, wie etwa Bussysteme zum Anschluss der Hardware an das Hausnetz.

Eine Übersicht der Schnittstellen im Living Place findet sich in Kapitel 8.

Hardware

Das System soll zu jeder Zeit für den Benutzer zur Verfügung stehen, das bedeutet, der Benutzer soll keine besondere Hardware am Körper tragen müssen, wie etwa einen *Datenhandschuh* oder ein *Smartphone*.

4.2 Usability

> „User interfaces should be simplified as much as possible, since every additional feature or item of information on a screen is one more thing to learn, one more thing to possibly misunderstand, and one more thing to search through when looking for the thing you want. Furthermore, interfaces should match the users' task in as natural a way as possible, such that the mapping between computer concepts and user concepts becomes as simple as possible and the users' navigation through the interface is minimized."
>
> (Nielsen, 1994, S. 115)

Gestenerkennung ist als Teil einer Benutzerschnittstelle den allgemein gültigen Anforderungen an eine solche unterworfen. *Usability* ist dabei von entscheidender Bedeutung für die Fragestellung, ob und welchen Nutzen ihre Benutzung für den Anwender bietet und somit für die Entscheidung, ob ein Anwender dieses System verwendet.

Dabei ist zu beachten, dass sich die meisten Richtlinien auf *klassische* Benutzerschnittstellen, also auf zweidimensionale Monitore mit Fenstersystemen sowie Maus und Tastatur als Eingabegeräte beziehen. Die vorhandenen Richtlinien müssen also entweder adaptiert werden oder es müssen neue aufgestellt werden.

Aus den sich ergebenden Anforderungen aus Sicht des Benutzers lassen sich die daraus resultierenden technischen Anforderungen ableiten.

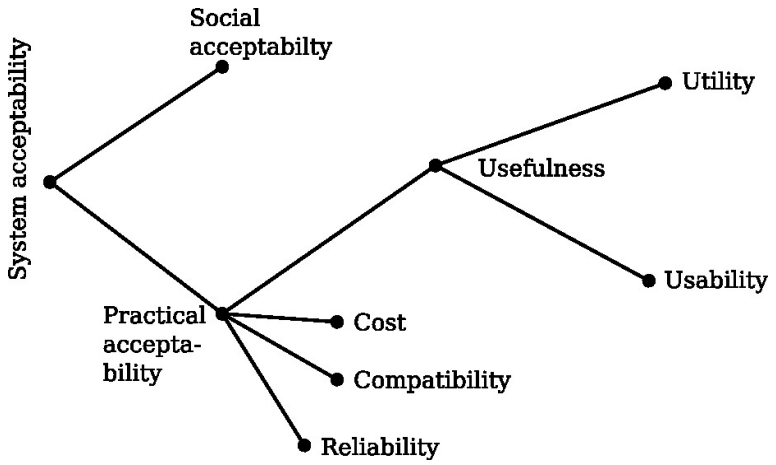


Abbildung 4.1: System acceptability nach (Nielsen, 1994, S. 25)

4.2.1 Was sind die Anforderungen aus Sicht der klassischen *Usability*?

Abbildung 4.1 zeigt die Teilaspekte, die für die Akzeptanz eines Systems durch den Benutzer bedeutsam sind (Nielsen (1994)). Im Folgenden werden die relevanten Aspekte im Hinblick kurz im Kontext von dreidimensionalen Gesten erläutert.

4.2.2 Soziale Akzeptanz

Social acceptability beschäftigt sich mit der gesellschaftlichen Akzeptanz, also der Frage, inwieweit ein System aufgrund seiner sozialen Auswirkungen akzeptiert oder abgelehnt wird. Eine in diesem Zusammenhang relevante Frage der Schutz von anfallenden Daten[3]. Abschnitt 4.3 beschäftigt sich mit dieser Thematik.

4.2.3 Praktische Akzeptanz

Praktische Akzeptanz setzt sich aus den Anforderungen zusammen, die die Nutzung in der Praxis mit sich bringen. Diese werden im Folgenden weiter erläutert.

4.2.4 Kosten

Die Frage der Kosten der einzelnen Komponenten ist für die wissenschaftliche Erprobung zunächst zu vernachlässigen. Im Kontext einer Benutzung im Feld ist sie jedoch relevant. Die Kosten sind abhängig vom angewandten Verfahren und den Stückzahlen, in denen die Hardware produziert wird. Auch der Aufwand der Berechnungen, die die angewandten Algorithmen erfordern ist hier von Bedeutung, er ist die Grundlage für die Abschätzung der benötigten Computerleistung.

Zu den Kosten der Herstellung kommen die Kosten im Betrieb. Diese werden hauptsächlich durch den Energiebedarf verursacht, zusätzlich durch Wartung und Lebensdauer der Geräte (Ersatzbedarf).

[3] spätestens seit der Diskussion über die Volkszählung 1983 und 1987 (siehe Bergmann (2009)) ist das Thema Datenschutz hierzulande von einem breiten gesellschaftlichem Interesse getragen

4.2.5 Kompatibilität

Die einfache Integration in ein bestehendes System zur Haussteuerung oder Steuerung von Multimediageräten ist sowohl ein technisches wie auch ein Usability-Problem. Für die technische Integration müssen die notwendigen Schnittstellen (Bussysteme, Protokolle) vorhanden sein. Aus Sicht der Usability muss eine Integration in ein logisches Gesamtkonzept der Bedienung aller Geräte erfolgen, die Bedienkonzepte der einzelnen Geräte dürfen nicht gegensätzlich sein. Dies ist schwierig, da es keine verbreiteten Standards dafür gibt.

4.2.6 Zuverlässigkeit

Zuverlässigkeit (*Reliability*) besteht aus mehreren Komponenten: Einerseits die Zuverlässigkeit im Bezug auf die Stabilität und Verfügbarkeit, andererseits auch die Zuverlässigkeit im Bezug auf den Prozentsatz der korrekt erkannten Gesten, also der zuverlässigen Reaktion auf Benutzereingaben.

Ein System, das eine beabsichtigte Interaktion des Benutzers zu über 90 Prozent [4] erkennt, ist aus technischer Sicht eine Herausforderung. Aus Sicht eines Benutzers allerdings ist eine Fernbedienung (als Metapher), die jedes zehnte Mal nichts, nicht das was er will oder sogar das Gegenteil vom Beabsichtigten tut, nahezu wertlos.

Karam (2006) führte dazu Experimente durch, die den folgenden Aufbau besaßen: Probanden mussten Aufgaben durchführen, die primär nichts mit der Informatik zu tun hatten, beispielsweise das Lösen eines dreidimensionalen Puzzles. Die Informationen, wie diese Aufgaben zu erfüllen sind, wurden über mehrere Bildschirmseiten vermittelt. *Das Weiterblättern* zwischen einzelnen Seiten erfolgte dabei entweder über eine einfache Geste per Hand, oder über eine ebenfalls vorhandene Tastatur. In zwei unterschiedlichen Szenarien war die Tastatur entweder direkt für die Testpersonen greifbar, oder mehrere Meter entfernt. Die Erkennung der Geste wurde nun im Programm so manipuliert, dass die Fehlerrate entweder 0, 10,20 oder 30 Prozent betrug. Dabei zeigte sich, dass im ersten Szenario (Die Tastatur war direkt am Benutzer) die Toleranz gegenüber fehlerhaften Erkennungsleistungen deutlich geringer war: 10 Prozent tolerierte Fehlerrate gegenüber 30 Prozent bei entfernter Tastatur.

Insgesamt sind Untersuchungen zum Thema „welche Zuverlässigkeit erwartet der Benutzer“ sehr rar, Edwards u. Grinter (2001) beispielsweise haben für ihre Vorschläge betreffend *Reliability* in ihrem Artikel “At Home with Ubiquitous Computing: Seven Challenges” nicht eine Quelle anzubieten. Bei Karam (2006) verhält es sich ebenso.

[4] Beispielsweise messen Xu u. a. (2009) für ihr Videobasiertes Gestenerkennungssystem 93.1 Prozent korrekt erkannte Gesten.

4.2.7 Nützlichkeit

Usefulness betrachtet die Fragestellung, ob ein System für den Benutzer in Bezug auf die Erfüllung einer bestimmten Aufgabe nützlich ist. Zum einen die Frage ob die technische Implementierung für die Erfüllung einer Aufgabe überhaupt geeignet ist (*utility*), zum anderen die Frage wie geeignet genau dieses System für den Benutzer ist (*usability*). Welchen Nutzen hat der Benutzer davon, genau dieses System für einen bestimmten Zweck zu verwenden? Ein hoher Nutzwert ist eine bessere Motivation als der Zwang etwas einsetzen zu müssen, weil es keine Alternative gibt.

Ersetzt das System eine bestehende Interaktionsmöglichkeit, wie beispielsweise eine Fernbedienung oder schafft es neue Möglichkeiten, beispielsweise durch eine Interaktion mit einer Kunstinstallation, die vorher nicht möglich war? Ergibt sich durch den Einsatz von dreidimensionalen Gesten ein zusätzlicher Nutzen, wie etwa die Steigerung der körperlichen Fitness bei Sportspielen?

4.2.8 Benutzbarkeit

Benutzbarkeit oder auch *Gebrauchstauglichkeit* bezieht sich auf die Eignung eines Systems für einen bestimmten Zweck. Nielsen (1994) zufolge ist ein System benutzbar, wenn es die folgenden fünf Kriterien erfüllt:

- **Erlernbarkeit** - Das System soll schnell erlernbar sein.
- **Effizienz** - Das System soll effizient arbeiten und dem Benutzer einen hohen Grad an Produktivität sichern.
- **Einprägsamkeit** - Der Benutzer soll sich leicht an die richtige Benutzung erinnern, auch nach einer längeren Pause bei der Verwendung[5].
- **Fehler** - Das System soll eine niedrige Fehlerrate haben und fehlertolerant sein.
- **Zufriedenheit** - Das System soll angenehm zu bedienen sein und bei dem Benutzer ein Gefühl der Befriedigung auslösen.

Die Frage nach dem Erfassungsbereich einer Kamera (siehe 5.8.4) gehört zu dieser Fragestellung und lässt sich direkt messen. Ebenso gibt es Möglichkeiten, die Produktivität zu messen. Eine Frage, die ebenfalls in diesen Bereich gehört, ist die Frage nach der *Responsiveness*.

[5] Dies schließt Systeme mit schlechtem Usability-Design leider nicht aus.

Gorilla arm syndrome

Ausladende Gesten der Arme können zum Tennisarm-Syndrom (*gorilla arm syndrome*) führen. Dies muss bei der Auswahl der Gesten beachtet werden.Es muss ein Kompromiss gefunden werden zwischen der verfügbaren Auflösung und der minimal signifikanten Bewegung[6].

Barrierefreiheit

Gestenerkennung darf nicht dazu führen Bevölkerungsgruppen von der Verwendung auszuschließen. Inwieweit sind körperlich behinderte Menschen in der Lage, Gestenerkennung zu bedienen ? Was ist, wenn sie keine Arme haben, um diese zur Interaktion zu benutzen? Im Falle von taubstummen Menschen hingegen lässt sich Gestenerkennung vielleicht zum Vorteil nutzen, beispielsweise wenn Zeichensprache als Gestenalphabet genutzt wird (siehe Abschnitt 7.4.2).

4.2.9 Responsiveness

> „The ability of a functional unit, such as an automatic data processing, communications, computer, information or control system to perform an assigned function, such as computer service or a telecommunications service within the required time interval.“ (Weik, 2006, Seite 1984)

Die Frage ist also, wie groß das geforderte Zeitintervall ist, in dem eine Antwort erfolgen muss. Das Zeitintervall, das zwischen Durchführung einer Aktion und der Reaktion des Systems liegt, wird als *Latenz* bezeichnet. Die Grenze hierfür legt (unbewusst) der Benutzer fest. Verhält sich das *Interface* zum Computer aus Sicht des Benutzers nicht so, wie er es erwartet, sinkt die Bereitschaft, diese Schnittstelle zu benutzen[7].

Da es bisher noch keine experimentell bestätigten Zahlen für gestenbasierte Systeme gibt, bietet es sich an, auf Zahlen aus der klassischen Usability zurückzugreifen. Eine experimentelle Bestätigung über die Zulässigkeit dieses Transfers steht allerdings noch aus.

Die ersten Veröffentlichungen zu Antwortzeiten von Benutzerschnittstellen (*Response Time*) geht auf das Jahr 1968 zurück (zum Beispiel Miller (1968)). Umfassende Untersuchungen zu den Grenzen der menschlichen Wahrnehmung aus Sicht der kognitiven Psychologie finden sich bei Card u. a. (1983). Nielsen greift die Ergebnisse in seinem Buch „Usability Engineering“ (Nielsen, 1994, S. 135) auf und teilt die Antwortzeiten in mehrere Klassen ein:

[6] Über den Schauspieler Tom Cruise ist bekannt, dass er sich bei den Filmaufnahmen zu „Minority Report“ über die Ermüdung durch die (nur simulierte) Gestenerkennung beklagte Saffer (2010)

[7] (vergleiche Karam (2006))

- **0-0,1 Sekunden:** Eine Verzögerung ist für den Benutzer nicht wahrzunehmen, die Reaktionen des Systems werden als Echtzeit wahrgenommen.
- **0,1-1 Sekunde:** Die Verzögerung ist merkbar, aber der Benutzer hat eine klare Zuordnung von Aktion zu Reaktion, der Fluß der Gedanken bleibt ununterbrochen.
- **1 - 10 Sekunden:** Der Benutzer hat keine klare Zuordnung von Aktion zu Reaktion.
- **> 10 Sekunden:** Es kommt der Wunsch auf, andere Dinge während der Wartezeit zu tun, eine Zuordnung ist nicht mehr möglich.

Echtzeit

Der Wert, der maximal von einem Benutzer toleriert wird, ohne das Gefühl der Echtzeit zu verlieren, ist umstritten. Die Grenze von 100 Millisekunden findet sich seit Miller (1968) in unterschiedlichen Veröffentlichungen, obwohl sie dort so nicht definiert ist. Miller selbst definiert diese Grenze als Bereich zwischen 100-200 Millisekunden, und experimentelle Untersuchungen (Dabrowski u. Munson (2001)) kommen auf einen Wert von bis zu 195 Millisekunden. Dabei wird die Wahrnehmung der Probanden vom Eingabegerät beeinflusst, die Verzögerung bei Mausbewegungen darf größer sein (195 Millisekunden) als bei Tastatureingaben (150 Millisekunden). MacKenzie u. Ware (1993) sehen den Wert bei weniger als 225 Millisekunden.

Abbildung 4.2: Uhr als Symbol, Quelle: [Seligmann (2003)]

Im Gegensatz dazu beziffern von Hardenberg u. Berard (2001) die maximale Zeitspanne auf 50 Millisekunden, und berufen sich dabei auf die Untersuchungen zur Fähigkeit der Unterscheidung von einzelnen Ereignissen bei Card u. a. (1983) (Siehe Abschnitt 4.2.9).

Laut Wachs u. a. (2011) beträgt diese Grenze sogar bis zu 300 Millisekunden. Experimentelle Untersuchungen über das Zutreffen dieser Annahme, liegen derzeit nicht vor.

Eine offene Frage ist auch, ob ein Gewöhnungseffekt eintritt, ob also auch längere Reaktionszeiten irgendwann als *normal* empfunden und akzeptiert werden.

Grenzen der menschlichen Wahrnehmung

Ein Faktor für die Beurteilung der minimalen Zeit, nach der eine Verzögerung wahrgenommen wird, ergibt sich aus der *Verarbeitungsgeschwindigkeit* im menschlichen Gehirn. Also die Frage, wie kurz Ereignisse aufeinander folgen können, um als getrennt wahrgenommen zu werden, genauso wie die Frage, wann die Verarbeitung eines Ereignisses abgeschlossen ist.

Im 1983 erschienen Buch „The Psychology of Human-Computer Interaction“ (Card u. a., 1983, S. 31-32) wird diese Zeitspanne auf etwa 50-100 Millisekunden festgelegt, zumindest für akkustische wie visuelle Wahrnehmung. Folgen die Ereignisse in kürzeren Abständen, so verschwimmen sie zu einem einzigen Ereignis.

Die Feedback-Schleife von einer Bewegung bis zur Rückmeldung des Ergebnisses liegt bei 200-500 Millisekunden ((Card u. a., 1983, S. 34)), dabei gibt es sowohl Feedback von der optischen Wahrnehmung als auch über die Rückmeldungen über Gelenkstellungen und Muskeln. Für die rein visuelle Wahrnehmung beträgt die Zeit 50-200 Millisekunden Card u. a. (1983).

Die Reaktionszeit auf einen beim Menschen eintreffenden Stimulus beträgt laut (Card u. a., 1983, S. 66) zwischen 100-400 Millisekunden.Schmidt (1988) Gibt den Zeitraum für eine willkürliche Reaktion auf einen äußeren Stimulus mit 120-180 Millisekunden an.

Fazit

Der genaue Wert, den ein System erfüllen muss, um als Echtzeit zu gelten, ist bisher nicht klar definierbar. Vergleicht man die obigen Zahlen, so scheint er irgendwo im Bereich 50-400 Millisekunden zu liegen. Für Gesten als Modalität fehlen Untersuchungen zu diesem Thema. Der oft genannte Wert von 100 Millisekunden (Nielsen (1994)) scheint jedoch zu kurz, in Anlehnung an die Untersuchungen von Dabrowski u. Munson (2001) und in Übereinstimmung mit Miller (1968) sollte eher von einem Wert von 150 Millisekunden ausgegangen werden. Dies erfolgt unter der Annahme, dass die Ansprüche des Benutzers an eine Interaktion mit Gestenerkennung als Schnittstelle vergleichbar mit denen an eine Tastatur sind. Um diesen Wert für Gestik zu verifizieren ist die Durchführung weiterer Messungen sinnvoll.

Messung

Zur Messung der Latenz gibt es unterschiedliche Möglichkeiten:

Bei einem auf Software basierenden System ist es möglich, die einzelnen Komponenten getrennt zu betrachten, insbesondere wenn der Quelltext verfügbar ist. Die Latenzen der verwendeten Kameras und der Verarbeitung in der Software werden so getrennt ermittelt. Dazu kann

spezielle Soft- und Hardware eingesetzt werden, die automatisch die Verzögerung ermittelt (Als Beispiel siehe Kapitel 6).

Bei einem Hardware-System (*black-box*) wie einer Spielekonsole ist dies nur schwer möglich, da auf die einzelnen Komponenten nicht getrennt zugegriffen werden kann. Einen Vorschlag für die Messung eines solchen Systems findet sich in Abschnitt 9.3.3.

Zu beachten ist, dass zur Latenz des Systems bei der Verarbeitung eventuell weitere Latenzen – wie die Berechnung und Darstellung des Bildschirminhalts – hinzugerechnet werden müssen.

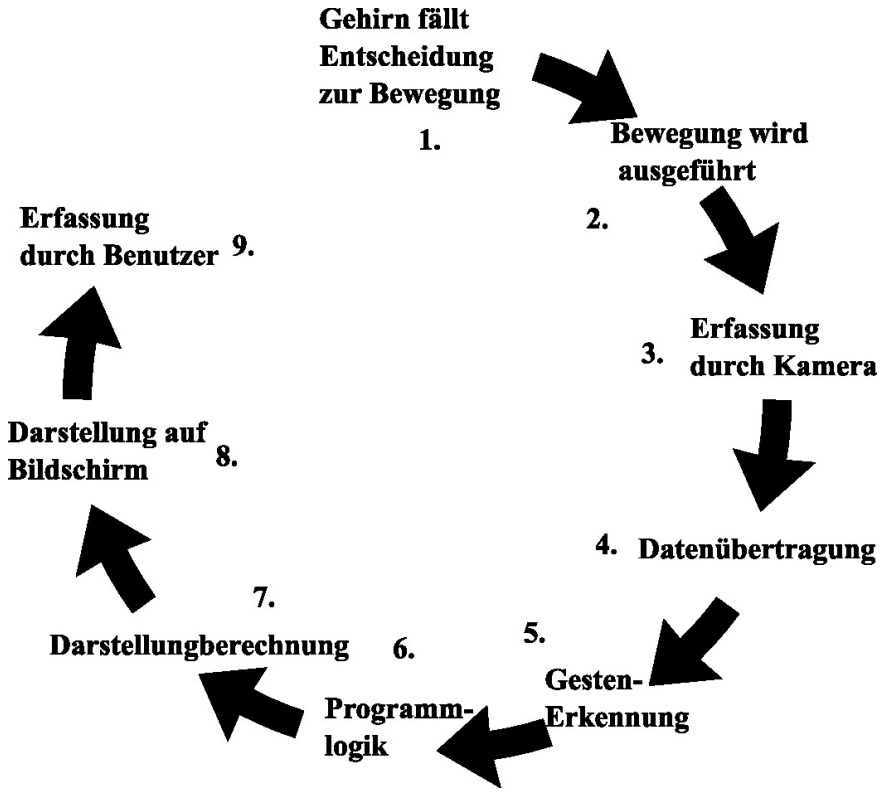


Abbildung 4.3: Ablauf aus Sicht des Benutzers

Der sich aus Sicht des Benutzers ergebende Ablauf ist in Abbildung 4.3 schematisch dargestellt. Dabei wird nach der Entscheidung zur Ausführung der Geste diese zunächst ausgeführt. Nach der Erfassung der Bewegung durch die Kamera werden die Bilddaten gegebenenfalls dort bereits vorverarbeitet. Nach der Übertragung an den angeschlossenen Rechner werden diese Daten verwendet, um eine Gestenerkennung durchzuführen. Die erkannten Gesten steu-

ern dann den Ablauf der Programmlogik, das Programm berechnet die Darstellung auf dem Bildschirm. Nach erfolgter Darstellung kann der Benutzer die angezeigten Inhalte erfassen.

Dieser Ablauf ergibt die für den Benutzer wahrnehmbare Gesamtverzögerung. Dabei sind die Punkte 3-8 durch externe technische Systeme direkt messbar und vergleichbar, da die die Punkte 1,2 und 9 systemunabhänging sind, kann eine Messung unterbleiben[8].

4.2.10 Mentales Modell

Das Konzept des *mentalen Modells* (Zuerst für die HCI bei Carroll (1990) behandelt) besagt, dass ein Benutzer zum Verständnis einer Maschine ein Modell in seinem Kopf bildet, von dem er annimmt, dass es die Reaktionen und Funktionsweise des Systems abbildet.

Wenn eine Funktionsweise für die Bedienung bekannt erscheint, wird das entsprechende Modell verwendet. Die Begriffe der *natürlichen* oder *intuitiven* Schnittstelle zielen genau auf diese Tatsache ab, dass der Umgang aus einem anderen Kontext geläufig ist.

Dies bietet den Vorteil, Verhaltensweisen nicht im einzelnen erklären zu müssen. Dabei besteht jedoch das Risiko, dass der Benutzer ein anderes Modell in seinem Kopf hat, als der Entwickler (siehe auch 4.2.14). Das Aufgreifen bereits vorhandener mentaler Modelle kann die Zeit zum Erlernen des Umgangs mit einem bestehenden System verkürzen.

Beispiel für irreführendes mentales Modell

Im Folgenden wird mit einem einfachen Beispiel einer Mensch-Maschine-Interaktion die Auswirkung eines fehlinterpretierten Bedienungsmodells verdeutlicht.

Die Hamburger S-Bahn setzt in Zügen der Baureihe 474.1/874.1 Knöpfe zum Türöffnen ein, die optisch an Berührungssensoren erinnern, da sie lediglich als plane Fläche ausgeführt sind. (siehe Abbildung 4.4b). Dies führt dazu, dass manche Fahrgäste diese Knöpfe als infrarotempfindliche Sensoren analog zu den Schaltern an Bedarfsampeln (siehe Abbildung 4.4a) wahrnehmen und versuchen, die Tür durch vorhalten der Hand in einigem Abstand beziehungsweise durch einfaches Auflegen ohne Druck zu öffnen. In diesem Fall öffnet die Tür nicht, da es sich um einen Druckknopf innerhalb der Folie handelt.

In neueren Bahnen der Reihe 474.1/874.1 ist dieser Knopf durch einen Knopf (Abbildung 4.4c) ersetzt worden, der klar erkennbar gedrückt werden muss.

[8] Die Zeitspanne für den ersten Punkt, von der Entscheidung bis zur Ausführung im Gehirn liegt bei etwa 110-120 Millisekunden (Schmidt (1988)).

(a) Berührungsschalter an Ampel, Quelle: ADFC Wedel (2010)

(b) Folienschalter als Türöffner, Quelle: Bernin (2011a)

(c) Druckknopf als Türöffner, Quelle: Bernin (2011b)

Abbildung 4.4: Infrarot und Druckschalter als Bedienelemente

Dies ist ein einfaches Beispiel für den Fall, dass ein Benutzer eines Systems (der S-Bahn Tür) eine andere Vorstellung von der Funktionsweise hat als sie in der Realität gegeben ist. Je komplexer das System, desto eher dürften solche Missverständnisse auftreten.

Folgerungen

Das Funktionsmodell sollte entweder klar ersichtlich sein (Druckknopf erkennbar als Druckknopf) oder muss kommuniziert werden. Ein Schild mit der Aufschrift: „bitte drücken“ wäre eine einfache Abhilfe und würde Missverständnisse verhindern.[9]

Die Kommunikation von mentalen Modellen ist allerdings nicht immer so trivial, insbesondere bei komplexen Modellen ist die Gefahr von Missverständnissen gegeben.

4.2.11 Weitere menschliche Einflussfaktoren

Es gibt weitere Faktoren, die die Bereitschaft und Fähigkeiten zur Benutzung beeinflussen.

[9] Zumindest für deutschsprachige Benutzer.

- **Alter:** Im Alter nehmen die sensorischen und motorischen Fähigkeiten eines Menschen ab. Zudem sind die Vorerfahrungen, beispielsweise im Beruf, relevant für das Erlernen neuer (technischer) Fähigkeiten. Die Erfahrung mit bekannten Schnittstellen wirkt sich auf den Umgang mit der Technik und die Selbstsicherheit dabei aus.[10]
- **Berufliche Herkunft:** Menschen, die einen technischem Beruf nachgehen, stehen dem Umgang mit technischen Systemen vielleicht offener gegenüber[11].
- **Soziale Faktoren:** Die generelle Einstellung zu Computern, deren Benutzung sowie der Selbstsicherheit in der Benutzung ist abhängig vom Grad der Benutzung (Levine (1998)).
- **Kultureller Hintergrund:** Gerade Gestik als Ausdrucksform in der zwischenmenschlichen Kommunikation unterliegt großen kulturellen Unterschieden (als Beispiel siehe Abbildung 4.5; weiteres hierzu in Abschnitt 7.4.5).

(a) Cultural posture 1 , Quelle: Rehm u. a. (2008)

(b) Cultural posture 2 Quelle: Rehm u. a. (2008)

Abbildung 4.5: Beispiel für Unterschiede in der Gestik (Posture) zwischen Deutschland (verschränkte Arme) und Japan (verschränkte Hände): abwartende Haltung.

[10] Eine Studie der Nielse Norman Group über Usability von Webseiten bei Senioren weist ein signifikant weniger performantes Abschneiden beim Umgang als bei der jüngeren Kontrollgruppe nach. Verursacht wird dies, nach Meinung der Autoren, durch körperliche Schwächen (Sehen, Motorik) als auch durch mangelnde Erfahrung im Umgang mit den *neuen* Medien in der Berufszeit der Studienteilnehmer (siehe Nielsen u. Pernice (2002)) Der Artikel von Ijsselsteijn u. a. (2007) über das Design von Computerspielen für Senioren kommt zum gleichen Ergebnis.

[11] Auch hier sei als Beleg auf Nielsen u. Pernice (2002) sowie Ijsselsteijn u. a. (2007) verwiesen, die sich mit den Auswirkungen von Erfahrungen während der Berufsjahre auf das Erlernen der Interaktion mit neuen Systemen befassen.

4.2.12 Einschränkungen durch die Situation

Gesten sind Teil der zwischenmenschlichen Kommunikation. sowohl in der Kommunikation untereinander, als auch alleine (Beispielsweise der allein durch die Wohnung laufenden dabei telefonierenden als auch gestikulierenden Mensch). Ein System zur Gestenerkennung muss dies unterscheiden können und den Kontext kennen, in dem sich der Anwender gerade befindet.

4.2.13 Natürliche Benutzerschnittstelle

Ein *natural user interface* ist eine Benutzerschnittstelle, die dem Benutzer ermöglichen soll, mit einem Computer genauso zu kommunizieren wie mit anderen Menschen und/oder Objekten der realen Welt (siehe Rauterberg u. Steiger (1996)).

Gemeint ist damit, dass sich die Schnittstelle an Kommunikationsmustern der realen Welt orientieren soll. Der Begriff *natürlich* ist in diesem Zusammenhang missverständlich. Er ist nicht im kulturhistorischen Sinne gemeint, denn fast alles, was uns in dieser Welt umgibt, ist menschengemacht.

Insofern ist der Begriff *intuitive Benutzerschnittstelle* eher angemessen.

> „Hence an intuitive interface may be defined as an interface, which is immediately understandable to all users, without the need neither for special knowledge by the user nor for the initiation of special educational measures. „
>
> (Bærentsen (2002))

Aber auch dieser ist umstritten. Raskin merkt schon 1994 an, dass es sich bei *intuitiv* eher um *bekannt* (*familiar*) handelt, da die Bedienung von Benutzerschnittstellen für Computer keine Fähigkeit ist, die einem Menschen von Geburt an zur Verfügung steht (Raskin (1994)).

Insofern erfolgt die vielfältige Verwendung der Begriffe *natürlich* und *intuitiv* wohl eher auf Marketinggründen. Eigentlich gemeint ist, dass der Anwender das der Anwendung zugrunde liegende Modell aus einem anderen Zusammenhang kennt (siehe 4.2.10).

In der realen Welt ist die Benutzung eines Gegenstandes, oder seiner *Benutzerschnittstelle* oft ersichtlich. Ein Buch etwa wird durch Umblättern der Seiten bedient, eine andere Bedienung ist zum Lesen nicht möglich. Die Aufgabe, einer anderen Person mitzuteilen, wo sich das Buch befindet, kann jedoch auf verschiedene Arten erfolgen, durch das Zeigen auf das Buch genauso wie durch eine Beschreibung des Ortes erfolgen. Oder auch durch das Holen desselbigen und die Aushändigung an den Fragenden mit der Beschreibung: „Hier!“.

4.2.14 Grenzen der *Natürlichkeit*

Donald A. Norman weist in seinem Artikel “Natural interfaces are not natural” (Norman (2010)) noch auf ein anderes Problem hin: Benutzer neigen dazu, sich zu sehr an einem ihnen bekannten Modell zu orientieren und *intuitiv* oder unbewusst auch Aktionen auszuführen, die bei näherem Hinsehen offensichtlich falsch sein müssen. Im konkreten Beispiel beschreibt der Artikel den Fall des Bowling-Spiels für die Spielekonsole Nintendo Wii, bei dem die Steuerung mithilfe des Wiimote-Kontrollers erfolgt. In diesem Spiel wird das Werfen der Bowlingkugel durch eine Bewegung des Armes symbolisiert. Der Spieler hat dabei den Controller in der Hand, die Geschwindigkeit und der Winkel der Kugel wird anhand der Bewegung und Position berechnet, die der Controller durchführt. Solange dabei eine Taste gedrückt wird, hält die Figur im Spiel die Kugel fest, lässt der Spieler los, so tut dies auch seine Repräsentation im Spiel.

Leider führte diese Art des Interfaces dazu, dass die Spieler neben der Taste auch den Controller losließen und ihn *natürlich* – wie beim realen Bowling – in den Fernseher schleuderten.

Abbildung 4.6: Bowling. Quelle: [Xiaphias (2007)]

Bei anderen Techniken die keinen Controller verwenden, wie Kinect (siehe Abschnitt 5.6), stellt sich die Frage, wie in diesem Fall das Loslassen der Kugel überhaupt dargestellt werden soll. Eine Möglichkeit wäre, in diesem Moment die Hand zu öffnen. Im Vergleich zum drücken oder loslassen eines Schalters ist dies allerdings zeitlich unpräzise. Zudem fehlt bei allen Systemen ohne Controller eine Form des haptischen Feedbacks. Dies dürfte sich negativ auf das Eintauchen in das Geschehen (*Immersion*) auswirken.

Ein Beispiel für die Abwesenheit einer Entsprechung in der realen Welt ist die *Vergrößerungsgeste*. Bei Smartphones wie dem iPhone erfolgt das Vergrößern von Bildern durch eine zweidimensionale Geste, bei der zwei auf dem Bildschirm aufliegende Finger das Bild *grossziehen*. Diese Art der Bedienung ist zwar leicht merkbar und verständlich, sie hat aber keine Entsprechung in der realen Welt: niemand käme auf die Idee, die Fotos in einem Fotoalbum durch diese Geste vergrößern zu können. Sie ist somit ein Beispiel für eine sinnvolle neue Entwicklung ohne *Natürlichkeit*.

4.2.15 Feedback

Ein generelles Problem bei der Verwendung von Gesten resultiert aus den mangelnden Möglichkeiten, dem Benutzer in gewohntem Maße ein Feedback seiner Aktionen zu bieten. Feed-

back ist eine der Forderungen der Usability an ein System, dies wird im Bereich der Gestenerkennung deutlich erschwert.[12]

Visuelles Feedback

Bei der Verwendung eines Mauszeigers kann der Benutzer sehen wohin er klickt. Klickt er neben einen Knopf, so erfolgt zwar keine Aktion, aber er kann erkennen, wo das Problem liegt. Bei Gesten im zwei- oder dreidimensionalen Raum hat er diese Möglichkeit nicht. Die Geste wird nicht erkannt — wo genau das Problem liegt — ist schwer ersichtlich.

Eine Menüführung bietet die Möglichkeit, die ausführbaren Aktionen zu erkunden. Bietet die Anzeige der Schnittstelle auch eine Menüfunktion, ist Ähnliches bei der Gestensteuerung machbar. Ansonsten ist nicht erkennbar, wie die Steuerung funktioniert.

Bei der Integration in ein Smart Home kann der Fall auftreten, dass ein System erst angesteuert werden muss, es also erst sehr spät ein Feedback durch die Aktion gibt. In diesem Fall sollte eine vorherige Information an den Benutzer erfolgen.

Haptisches Feedback

Bei einem Touchscreen hat der Benutzer ein haptisches Feedback. Zum einen merkt er, wann er den Bildschirm berührt, zum anderen gibt es die Möglichkeit, durch Vibrationen anzuzeigen, dass das System das Drücken einer virtuellen Taste erkannt hat. Bei Gesten im Raum entfällt diese Möglichkeit bei derzeitigem Stand der Technik komplett.

4.2.16 Undo

Auch das Vorhandensein einer *undo*-Funktion, zum Rückgängig machen der vorherigen Operation (*non-destructive operations*) ist etwas, was seit Langem zum Standard bei Benutzerschnittstellen gehört (Norman u. Nielsen (2010)). Wie realisiert man so etwas mit einer Geste? Und was macht man, wenn diese Geste nicht erkannt wird? Eine mögliche Abhilfe ist das Verwenden einer anderen Modalität, wie eines Sprachkommandos.

[12] Norman u. Nielsen (2010) beziehen sich in ihrem Artikel „Gestural interfaces: a step backward in usability“ zwar auf zweidimensionale Gesten, die Schwierigkeiten bei dreidimensionalen Gesten sind allerdings noch schwerwiegender, da Möglichkeiten zum haptischen Feedback komplett fehlen.

4.2.17 Messbarkeit

Ein Großteil der Anforderungen aus der Usability sind nicht technisch messbar. Lediglich die Latenzen eines Systems lassen sich durch Messung ermitteln, siehe dazu Abschnitt 4.2.9. Die Messung der Fehlerrate gestaltet sich schwieriger, da der Benutzer feststellen muß, ob das Ergebnis einer Aktion seinen Erwartungen entspricht[13].

Die Effizienz lässt sich messen, indem man eine Aufgabe zum zuerst mit Hilfe von herkömmlichen Interaktionsmitteln und danach mit Gestenerkennung erledigen lässt. Der Vergleich der benötigten Zeit lässt Rückschlüsse auf die Effizienz des Systems zu.

Alle weiteren Kriterien sind subjektiv, und müssen durch eine Befragung einer möglichst großen und heterogenen (in Bezug auf Alter, Vorwissen, Herkunft) Benutzergruppe ermittelt werden.

4.2.18 Fazit zur Usability

Manche Anforderungen der klassischen Usability, wie *undo*, *Feedback* und *Zuverlässigkeit* scheinen schwer erfüllbar. Führt dies zu einem Rückschritt führt oder gibt es andere Anforderungen, die sich – als Ersatz – formulieren lassen? Ist die Verwendung räumlicher Gesten wirklich ein Schritt zurück, aus Sicht der Benutzbarkeit[14]?

Die Antwort auf diese Frage ist, wie fast immer: Es kommt darauf an. Dreidimensionale Gesten bieten eine neue Form der Interaktion, aber man sollte sehr genau betrachten, ob sie für den Anwendungsfall geeignet sind, oder ob herkömmliche Arten der Interaktion in diesem Bereich besser funktionieren. Die Eingaben per Tastatur, Maus oder Joystick sind sehr viel exakter und schneller, als dies derzeit durch Gesten realisiert werden kann.

In den bisherigen Untersuchungen zur Usability von Gestensteuerung zeigten sich Lücken. Eine weitere Betrachtung sowie Vorschläge für das Schließen dieser Lücken finden sich in Kapitel 9.2.

[13] Eine Ausnahme hiervon bildet das Einfügen künstlicher Fehler, siehe 4.2.6

[14] Diese These wird von Norman u. Nielsen (2010) im Bezug auf zweidimensionale Gesten auf einem Touchscreen aufgestellt.

4.3 Datenschutz

Dieser Abschnitt beschäftigt sich mit den möglichen Auswirkungen von kamerabasierter Gestenerkennung auf den Datenschutz und die Privatsphäre der Nutzer.

> „Wir müssen – als Einzelne und als Gesellschaft – lernen, mit den neuen digitalen Dimensionen unseres Lebens umzugehen. Dazu gehört in erster Linie ein waches Bewusstsein dafür, was mit den preisgegebenen Informationen geschehen kann."
>
> (Schaar (2007))

4.3.1 Einführung

Der Schutz der persönlichen Daten und der Privatsphäre ist ein wichtiger Aspekt in der Informationsverarbeitung. Zum einen aufgrund rechtlicher Rahmenbedingungen, zum anderen im Rahmen einer ethischen Verantwortung des Entwicklers[15] als auch bei der Frage inwieweit potenzielle Benutzer ein System akzeptieren und ihm vertrauen (Spiekermann u. Cranor (2009)).

Alle in dieser Arbeit besprochenen Aufnahmetechniken sind kamerabasiert, es fallen dabei — je nach Technik — Videodaten in unterschiedlicher Qualität an, die mehr oder weniger Rückschlüsse auf den oder die Benutzer zulassen.

4.3.2 Rechtliche Rahmenbedingungen

Da es sich bei dieser Masterarbeit um eine Arbeit im Fachbereich Informatik handelt, sind die folgenden Ausführungen zu in Deutschland allgemein gültigen Gesetzen nur als Anregung aufzufassen, sich auch im rechtlichen Bereich mit den Auswirkungen von kamerabasierter Gestenerkennung zu befassen.

Bundesdatenschutgesetz

> § 1 Abs. 2 Nr. 3 BDSG: Das Bundesdatenschutzgesetz gilt "für die Erhebung, Verarbeitung und Nutzung personenbezogener Daten durch ... nicht-öffentliche Stellen, ... es sei denn, die Erhebung, Verarbeitung oder Nutzung der Daten erfolgt ausschließlich für persönliche oder familiäre Tätigkeiten."(§ 1 Abs. 2 Nr. 3 BDSG) (Störing, 2011, S.29)

[15] Vergleiche die ethischen Grundlagen der Gesellschaft für Informatik (http://www.gi.de/wir-ueber-uns/unsere-grundsaetze/ethische-leitlinien.html)

Für den Einsatz im privaten Rahmen des Smart Home ergibt sich somit, dass das Bundesdatenschutzgesetz nicht anwendbar ist. Vorausgesetzt, die Daten verlassen die Wohnung nicht, wie dies bei der Nutzung von Diensten (beispielsweise eine Gesichtserkennung durch einen externen Dienstleister) in der *Cloud* der Fall wäre. Es gibt jedoch weitere Gesetze, die für den Einsatzbereich im privaten Rahmen zur Anwendung kommen, betreffend der Herstellung von Bild- und Tonaufnahmen.

Verletzung der Vertraulichkeit des Wortes

Dieser Paragraph im Strafgesetzbuch (StGB) beschäftigt sich mit dem Verbot von Aufnahmen von Gesprächen im privaten Umfeld.

> (1) Mit Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe wird bestraft, wer unbefugt
>
> 1. das nichtöffentlich gesprochene Wort eines anderen auf einen Tonträger aufnimmt oder
> 2. eine so hergestellte Aufnahme gebraucht oder einem Dritten zugänglich macht.
>
> (StGB §201StGB (a))

Bei Gestik als einziger Modalität ist dies nicht relevant, bei Kombination mit Sprache jedoch schon. Enscheidend an diesem Paragraphen ist, dass alleine die Aufnahme ausreicht.

Verletzung des höchstpersönlichen Lebensbereichs durch Bildaufnahmen

> Wer von einer anderen Person, die sich in einer Wohnung oder einem gegen Einblick besonders geschützten Raum befindet, unbefugt Bildaufnahmen herstellt oder überträgt und dadurch deren höchstpersönlichen Lebensbereich verletzt, wird mit Freiheitsstrafe bis zu einem Jahr oder mit Geldstrafe bestraft.
>
> (StGB §201a StGB (b))

Für die Absicherung gegenüber den Anforderungen dieses Paragraphen kann es also offenbar notwendig sein, sich die Befugnis für die Erstellung von Bildaufnahmen der sich im Sichtbereich der Kameras aufhaltenden Personen (schriftlich) versichern zu lassen. Im Laborkontext mag dies möglich sein, aber für die Verwendung im Feld ist es schwer vorstellbar.

Recht am eigenen Bild

> Bildnisse dürfen nur mit Einwilligung des Abgebildeten verbreitet oder öffentlich zur Schau gestellt werden.
>
> (KUG §22 Kug)

Zumindest für die systeminterne Verarbeitung sollte dies nicht zutreffen, da die entstehenden Aufnahmen weder verbreitet noch öffentlich gemacht werden.

Fazit

Eine allgemeine Rechtliche Beurteilung zum Thema der rechtskonformen Ausgestaltung eines intelligenten vernetzten Haushaltes findet sich bei Störing (2011). Diese bezieht sich allerdings hauptsächlich auf *intelligente Stromnetze*. Die rechtliche Beurteilung ist wichtig und relevant für die praktische Verwendung der Systeme, auch im Bereich der Forschung.

4.3.3 Begehrlichkeiten

Das Vorhandensein von Daten schafft Begehrlichkeiten, sei es im Rahmen von staatlichen Eingriffen[16],kriminellen Aktivitäten[17] oder dem Wunsch von Medien nach Erkenntnissen aus dem Privatleben prominenter Mitbürger.

4.3.4 Vertrauen

Die Frage, ob die gewonnenen Daten missbraucht werden können, ist für die Benutzer von Bedeutung (Smith u. Milberg (1996)). Das System könnte sich anders verhalten, als nach außen propagiert, der Benutzer muss also dem Hersteller des Systems vertrauen. Alternativ wäre hier eine Open-Source-Lösung möglich, die es versierten Benutzern ermöglicht, ihr eigenes System zu übersetzen, oder von vertrauenswürdigen Leuten zu beziehen.

Die eigene Überprüfung ist nur für einen kleinen Anwenderkreis möglich, da sie umfassende technische Kenntnisse voraussetzt. Und selbst bei diesen kann es zum Einbau versteckter Zugänge (Hintertüren) kommen. Alleine die Diskussion um *vermeintliche* Hintertüren ist geeignet, das Vertrauen der Benutzer nachhaltig zu erschüttern. Als Beispiel hierfür kann die Ende 2010

[16] Die Diskussion um die Einführung der Vorratsdatenspeicherung ist hier ein Beispiel
[17] Der Einbrecher, der vorher über die Netzwerkschnittstelle kontrolliert, ob der Bewohner zu Hause ist.

aufgekommene Diskussion um eine vermeintliche Hintertür im IPsec-Stack[18] des Betriebssystems OpenBSD angesehen werden, die für einige Verunsicherung unter den Anwendern sorge (Paul (2010)).

Aus technischer Sicht kann dem Problem der nachträglichen Veränderung von Programmen durch Zertifizierung beziehungsweise die Verwendung von kryptographisch abgesicherten Checksummen von Programmversionen begegnet werden. Die setzt allerdings voraus, dass die nötige Sachkenntnis vorhanden ist , Manipulationen anhand fehlerhafter Checksummen zu erkennen. Dass diese gegeben ist, ist in der Mehrzahl der Anwendungszenarien eher unwahrscheinlich.

4.3.5 Datenschutz im Design

Der Datenschutz sollte beim Design bedacht werden, oft sind Lösungen denkbar, die weniger invasiv in die Privatsphäre eingreifen (Spiekermann u. Cranor (2009)). Beispielsweise können Systeme zur Bestimmung des Aufenhaltsortes passiv oder aktiv arbeiten, diese Entscheidung wird allerdings beim Systementwurf getroffen (Mattern u. Langheinrich (2001)). Ein später hinzugefügter, aufgesetzer, Datenschutz versucht dann Probleme zu beheben, die es bei einem anderen Systementwurf gar nicht gegeben hätte. Zudem ist es nicht unüblich, dass ein Prototyp zu einem Produkt wird (Langheinrich (2005)). Deshalb ist es besser, Datenschutz zu beachten.

Beispielsweise liefern Dey u. a. (2002) einen Vorschlag für die Entwicklung eines Frameworks zur Herstellung von alltäglicher Privatsphäre im Smart Home. Senior u. a. (2005) Entwickelten eine Überwachungskamera, die Gesichter erkennt und diese unkenntlich macht. Nur im notwendigen Falle, wie der Strafverfolgung, kann auf diese Daten zugegriffen werden.

4.3.6 Transparenz und Kontrolle

Das System muss transparent sein,der Benutzer muss wissen, welche Daten vom System erhoben werden. Was mit diesen Daten geschieht, wie und wo sie gegebenenfalls gespeichert werden.

Kontrolle aus Sicht des Datenschutzes bezieht sich nicht auf die Kontrolle über den Benutzer, sondern die Kontrolle des Benutzers über seine Daten.

Abbildung 4.7: Not-Aus-Schalter
Quelle: [Stahlkocher (2006)]

[18] IPsec ist eine Erweiterung des IP-Protokolls, dass die Übertragung von Daten kryptographisch absichert (VPN).

Der Benutzer muss erkennen können, wann das System
aktiv ist und Daten erfasst, beispielsweise durch die Verwendung einer Leuchtdiode. Dabei ist zu beachten, das per Software gesteuerte Anzeigen bei einem Einbruch Dritter in das System unzuverlässig sind. Der Einbau eines (Hardware-) Netzschalters behebt dieses Problem.

Einen möglichen Weg zur automatischen Kontrolle von Einstellungen zur Privatsphäre im Ubiquitous Computing zeigt Langheinrich (2005) auf, Privacy-Aware Ubiquitous Computing Systems geben Daten nur frei, wenn dies von der Privacy Policy des Benutzers so eingestellt ist.

4.3.7 Verhaltensänderung

Die Allgegenwärtigkeit von (Kamera-)Überwachung führt zu einer Verinnerlichung dieser (Focault (1976)) und kann dazu führen, dass Menschen ihr Verhalten ändern um den Normen zu entsprechen (Koskela (2000)).

Obwohl dies erwünscht sein kann, ist es fraglich, ob man die letzten kamerafreien Räume verringern sollte.

4.3.8 Allgemeine Lösungsansätze

Allgemeine Lösungsansätze im Bereich Datenschutz beinhalten, die erhobenen Daten auf ein notwendiges Maß zu begrenzen, und die erhobenen Daten sicher zu speichern. Technisch kann letzteres mit Verschlüsselung und Kapselung der Systeme erreicht werden (analog zur PrivacyCam von Senior u. a. (2005)), die Bilddaten werden nur auf einem System zur Gestenerkennung bearbeitet, und diese Daten werden danach gelöscht. An das Smart Home wird dann nur gemeldet, dass eine Geste erkannt wurde.

4.4 Ein Fazit zum Datenschutz

Es zeigt sich zum einen, dass die rechtlichen Rahmenbedingungen für den Einsatz kamerabasierter Gestenerkennung im Smart Home derzeit nicht geklärt sind. Zum anderen sind auch die Fragen nach Kontrolle des Benutzers, dem Umgang mit Daten und ihrer Sicherung ungeklärt. Dieser Zustand mag für die Evaluierung von Techniken im akademischen Umfeld nicht entscheidend sein, für eine tatsächliche Nutzung ist er es. Es sei denn, man hält es mit Mark Zuckerbergs Ausspruch: „The age of privacy is over“[19].

[19] Mark Zuckerberg, der Gründer von Facebook, siehe http://www.readwriteweb.com/archives/facebooks_zuckerberg_says_the_age_of_privacy_is_ov.php

4.5 Fazit

Die Anforderungen an ein System zur Gestenerkennung im Smart Home sind vielfältig, zumindest wenn es alltagstauglich sein soll. Für die Integration in intelligente Wohnungen fehlen derzeit die Standards. Zwischen den Anforderungen des Smart Home und der Usability finden sich Überschneidungen, beispielsweise bei der *Robustheit des Verfahrens* und der *Zuverlässigkeit*. Die *mechanische Robustheit* ist eine Vorstufe zur *Zuverlässigkeit* in der Usability.

Das Thema Datenschutz wird derzeit eher vernachlässigt, die rechtlichen und technischen Fragen sind ungeklärt. Die Vereinbarkeit mit Kriterien der Ergonomie (Gorilla Arm) ist zweifelhaft. Die Anforderungen der Usability sind nur teilweise erforscht und erfüllbar, die Adaption der bestehenden Richtlinien gestaltet sich teilweise schwierig, insbesondere die Frage nach der Zuverlässigkeit des Systems.

Weitere Forschungen in diesem Bereich sind nötig. Eine funktionierende Interaktion mit Gestenerkennung als alleiniger Modalität könnte dann als Grundlage für die Einbeziehung in eine menschengerechtere Kommunikation mit der Maschine unter Einbeziehung von Gestik, Mimik und Sprache dienen.

In Kapitel 9 werden die in diesem Kapitel betrachteten Kriterien zusammengefasst und auf weitere Untersuchungen eingegangen.

5 Dreidimensionale Kameras

5.1 Einführung

Dieser Abschnitt beschäftigt sich mit den zur Verfügung stehenden Hardwarekomponenten für die visuelle Gestenerkennung im Raum. Abbildung 8.6 zeigt, rot markiert, den hier betrachteten Teil des generellen Ablaufs der Gestenerkennnung. Generell basiert das Verfahren darauf, die Kamerainformationen zu nutzen, um ein Modell des betrachteten Objektes im Rechner (Rekonstruktion) zu berechnen.

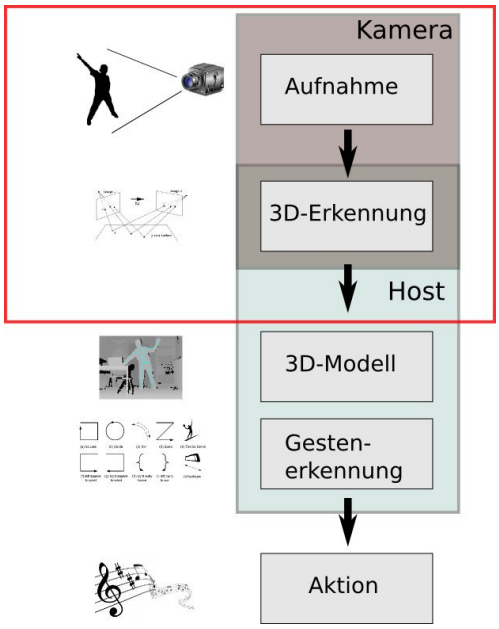


Abbildung 5.1: Übersicht Ablauf Gestenerkennung

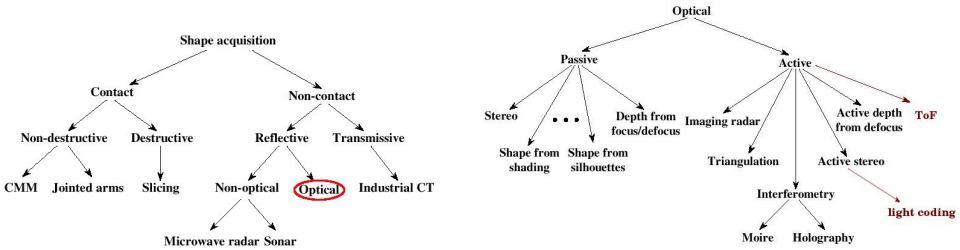
Ob die eigentliche 3D-Erkennung dabei auf dem Kamerasystem oder dem *Hostsystem*[1] erfolgt, hängt vom verwendeten Verfahren ab. Eine Übersicht der Möglichkeiten zur *Rekonstruktion*[2] findet sich in Abbildung 5.3. Die dabei gewonnenen Daten über die Entfernung eines Objektes werden als *Tiefeninformationen* bezeichnet.

Die in dieser Arbeit betrachteten Systeme basieren auf optischen Verfahren und ausschließlich auf Kameras. Weitere Möglichkeiten wie die Verwendung von Laserscannern[3] oder Ultraschall[4] wurden nicht weiter betrachtet, da sie entweder nicht verfügbar sind oder nicht

[1] Als Hostsystem wird hier die Rechnerkomponente bezeichnet, an die die Kamera(s) ihre Daten übertragen und die für die weitere Verarbeitung verwendet wird.
[2] Rekonstruktion bedeutet die Abbildung des realen Objektes im Rechner.
[3] Javier Garcia verwenden einen tragbaren Laserscanner zum Tracken einzelner Finger
[4] Das von Ogris u. a. (2005) verwendete System benötigt umfangreiche Hardware am Körper des Bedienenden, siehe Abbildung 5.2

den Anforderungen (keine Hardware am Körper des Benutzers, siehe Abbildung 5.2) entsprechen.

Eine Anforderung an die Verfahren sind die kommerzielle Verfügbarkeit[5] der entsprechenden Hardwaresysteme. Somit werden rein akademische Möglichkeiten wie Moiré oder Interferenz (siehe Abbildung 5.3) nicht betrachtet.



(a) 3D-Rekonstruktion. Quelle: Rocchini u. a. (2001)

(b) Opische 3D-Rekonstruktion erweitert um ToF und Light Coding. Quelle: Rocchini u. a. (2001)

Abbildung 5.3: Arten der 3D Konstruktion

5.2 Vorteile der *dritten Dimension*

5.2.1 Segmentierungsproblem

Die Erkennung der Gestik eines Benutzers setzt voraus, dass man diesen vom ihm umgebenden Raum unterscheiden kann. Es erfolgt also eine Trennung in Vorder- und Hintergrund. Dies wird als *Segmentierung* bezeichnet.

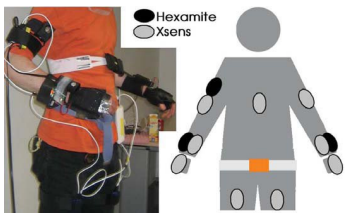


Abbildung 5.2: Quelle: [Ogris u. a. (2005)]

In den 1990er Jahren wurden mehrere Systeme zur Erkennung von Gesten der Hände entwickelt. Diese verwendeten die Hautfarbe als Basis für die Segmentierung. Beispielsweise benutzen Stark u. a. (1995) für ihre ZYKLOP genannte Handerkennung eine Farbsegmentierung im YUV-Farbraum[6], Xu u. a. (2009) konvertierten die Bilder in den

[5] Diese Arbeit beschäftigt sich mit der Evaluierung von Kamerasystemen, nicht deren Entwicklung.

[6] *YUV*: Luminanz Y und Chrominanz bestehend aus U und V, verwendet bei der Videoübertragung im PAL-Standard.

HSL-Farbraum.[7] Probleme bei der Verwendung von Hautfarbe zur Segmentierung treten insbesondere in Umgebungen auf, in denen Gegenstände ähnliche Farbwerte wie Haut aufweisen, beispielsweise Holzmöbel. Ein anderes Problem sind die Unterschiede in der Hautfarbe je nach Herkunft der Benutzer. Ein detaillierter Vergleich der verschiedenen Techniken zur Erkennung von Hautfarben findet sich bei Kakumanu u. a. (2007).

Eine andere Möglichkeit der Segmentierung ist die Erkennung von Bewegung. Die Unterscheidung in Vorder- und Hintergrund erfolgt hierbei durch die Berechnung der Unterschiede von aufeinander folgenden Bildern einer Sequenz (Siehe Bernin (2009)). Die Annahme ist hierbei, dass die Bewegung nur im Vordergrund erfolgt. Fehlerhaftes Erkennen tritt hierbei leicht durch Reflexionen an spiegelnden Oberflächen und durch wechselnde Lichtverhältnisse[8] auf. Zudem ist eine Verwendung für statische Gesten ohne Bewegung nicht möglich.

Die Einbeziehung der Position einer Person im Raum ermöglicht eine einfache Segmentierung. Beispielsweise lässt sich ein Radius um die Person bestimmen und die Betrachtung der Bilddaten anhand dieses Radius einschränken. Dies verringert die Datenmenge und erhöht so die Verarbeitungsgeschwindigkeit.

5.2.2 Neue Arten von Gesten

Neben der Nutzung zur Segmentierung bietet das Vorhandensein von Informationen über die Position im Raum einen weiteren Vorteil. Diese ermöglichen, die Veränderung der Entfernung von Extremitäten gegenüber dem Kamerasystem als weiteren Parameter in die Gestik mit einzubeziehen, beispielsweise also eine Bewegung der Handfläche auf die Kamera hin. Zudem sind Gesten zur zwischenmenschlichen Kommunikation oft dreidimensional, lassen sich so also besser erfassen.

5.3 Kameras für die dritte Dimension

Alle folgenden Verfahren basieren auf der Verwendung von optischen Kameras auf CMOS oder CCD-Basis. Sie unterscheiden sich durch die Anzahl der Kameras (eine oder zwei), in der Wellenlänge des verwendeten Lichts (sichtbar[9] oder infrarot)[10] und in der Art der Beleuchtung (aktiv oder passiv).

[7] HSL: *hue, saturation and luminosity.* also Farbton, Farbsättigung und Helligkeitswert.

[8] Ist die Differenz der Helligkeitswerte zwischen aufeinander folgenden Bildern zu groß, wird dies fälschlicherweise als Bewegung erkannt.

[9] sichtbares Licht im Bereich ca. 400-700 Nanometer

[10] jenseits 700 Nanometer, für die hier betrachteten Systeme 780 beziehungsweise 850 Nanometer

Zum Anschluss von Kameras an einen Computer stehen verschiedene digitale Schnittstellen zur Verfügung, insbesondere FireWire (400-3200 MBit/s), USB (480-5000 MBit/s), und Ethernet (100-1000 MBit/s).

Für die hardwarebasierten Systeme, also diejenigen, bei denen die Berechnung der Tiefeninformationen auf den Kameras erfolgt (siehe Abschnitt 5.5 und 5.6) , werden diese Tiefeninformationen als Graustufen-Bilder in unterschiedlicher Bittiefe ausgegeben. Erfolgt die Berechnung auf dem Hostsystem (beispielsweise bei den Kameras in Abbildung 5.5a), so müssen die Rohdaten in Form von Bildern über Netzwerk übertragen werden. Die dabei entstehenden großen Datenmengen müssen bei der Wahl der Schnittstelle zum Anschluss des Systems beachtet werden.

Tabelle 5.1: Vergleich verschiedener Verfahren zur 3D-Rekonstruktion

Verfahren	3D-Stereo aktiv	3D-Stereo passiv	Time-of-Flight	Light Coding
Verfügbar seit (Jahr)	ca. 1990	ca. 1980	ca. 2000	2010
Anwendungsbereich	Akademisch, Prototypen	Industrie, Robotik	Industrie, Robotik	Spiele, Robotik
Beispiel	siehe Aggarwal u. Wang (1988), Boyer u. Kak (1987)	Barnard u. Thompson (1980)	Lange (2000)	PrimeSensor, Kinect

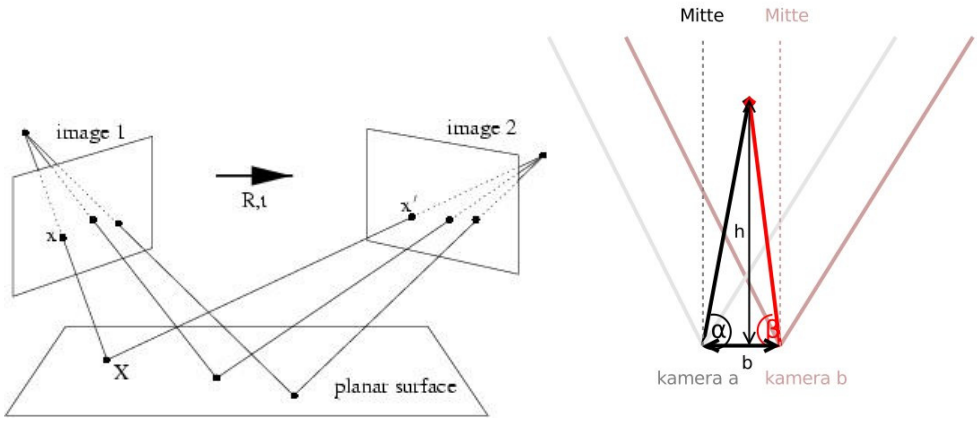
5.4 3D-Stereo Rekonstruktion (Triangulation)

Bei diesem Verfahren gibt es grundsätzlich zwei verschiedene Bauweisen, *aktiv* und *passiv*. Dabei bezieht sich *aktiv* und *passiv* auf die Art der Beleuchtung.

5.4.1 Grundsätzliches Verfahren

Es erfolgt eine Bestimmung der Entfernung eines Objektes durch Triangulation (siehe Abbildung 5.16). Dabei ergibt sich für die Entfernung[11] $h = b\frac{sin(\alpha)sin(\beta)}{sin(\alpha+\beta)}$. b entspricht dabei dem Abstand zwischen den beiden Kameras (bzw. dem Mittelpunkt der Linsen), die Winkel α und β lassen sich durch den Abstand vom Mittelpunkt der Kamerabilder berechnen.

[11] Herleitung: mit $b = \frac{h}{tan(\alpha)}$ und $b = \frac{h}{tan(\beta)}$ ergibt sich $h = \frac{b}{\left(\frac{1}{tan(\alpha)}+\frac{1}{tan(\beta)}\right)}$.
mit $tan(\alpha) = sin(\alpha)/cos(\alpha)$ und $sin(\alpha+\beta) = sin(\alpha)\,cos(\beta) + sin(\beta)\,cos(\alpha)$
ergibt sich daraus $h = b\frac{sin(\alpha)sin(\beta)}{sin(\alpha+\beta)}$



(a) Triangulation 3D **(b)** Triangulation 2D

Abbildung 5.4: Triangulation

5.4.2 Passives 3D-Stereo

Bei einem passiven Aufbau werden zwei gleichartige Videokameras verwendet (siehe Barnard u. Thompson (1980)), die in einem festen Abstand voneinander auf einer Ebene montiert sind und die gleiche Szenerie mit einem um einige Zentimeter verschobenen Ursprung betrachten (analog zu den menschlichen Augen). In beiden Kamerabildern müssen nun gleiche Punkte erkannt werden (*correspondence problem*). Dadurch lässt sich der Abstand des Objektes von der Kamera-Achse durch geometrische Triangulation berechnen (siehe Abbildung 5.4).

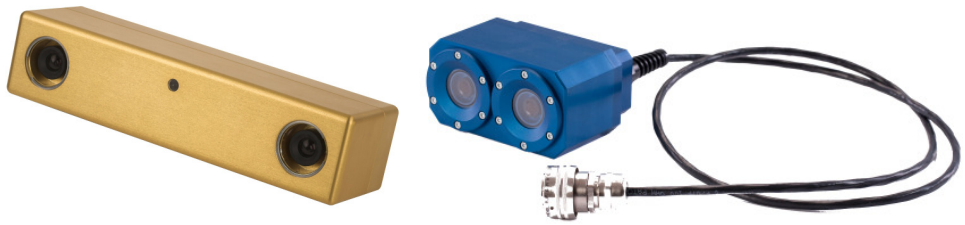
Ein Beispiel für ein solches kommerziell verfügbares System ist die Bumblebee 2 Kamera von *Pointgrey Research Inc.*[12]

Bei einem passiven System erfolgt die Berechnung des Abstandes in der Regel auf dem Hostsystem (siehe Abbildung 5.5a) und nicht in der Kamera selbst. Eine Ausnahme hiervon ist das System Mobile Ranger C3D (Abbildung 5.5b) der Firma MobileRobots. Dieses verwendet einen integrierten FPGA zur Berechnung.

5.4.3 Vorteile

Die Vorteile der Verwendung von passivem 3D-Stereo als Verfahren zur Gewinnung von Tiefeninformationen sind:

[12] http://www.ptgrey.com



(a) Bumblebee 2 , Quelle: Point Grey Research Inc (2011)

(b) MR C3D, Quelle: MobileRobots Inc (2011)

Abbildung 5.5: Bumblebee 2 und Mobile Ranger C3D

Ein Eigenbau ist möglich, damit können günstige Standardkomponenten zum Einsatz kommen.[13]

Das Verfahren funktioniert auch im Sonnenlicht, da die Umgebungsbeleuchtung verwendet wird.

5.4.4 Nachteile

Passive 3D-Stereo Systeme haben die folgenden Nachteile:

Nur der Eigenbau ist wirklich kostengünstig (siehe Tabelle 5.1 für Preise von kommerziell verfügbaren Systemen). Dann besteht allerdings erhöhter Kalibrierungsaufwand[14] und der Einsatz von massiver Rechenleistung ist nötig.[15]

Das Verfahren funktioniert nur bei entsprechenden Lichtverhältnissen.[16]

Passives 3D-Stereo setzt eine Strukturierung der Oberfläche voraus, da sonst die entsprechenden (Referenz-)Punkte in beiden Bildern nicht erkannt werden.

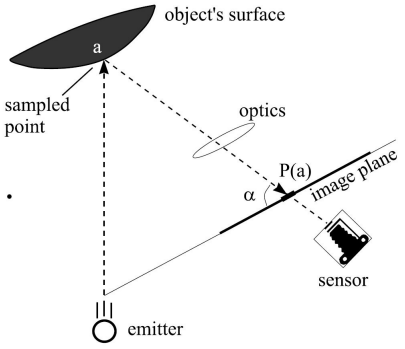
[13] Dabei ist zu beachten, dass einfache Videokameras nicht über die Möglichkeit zur Synchronisation des Zeitpunktes der Bildaufnahme verfügen. Auch ist der Preisvorteil mit der Verfügbarkeit des Kinect Systems zumindest fraglich.

[14] Kommerziell erhältliche Systeme sind vom Hersteller vorkalibriert.

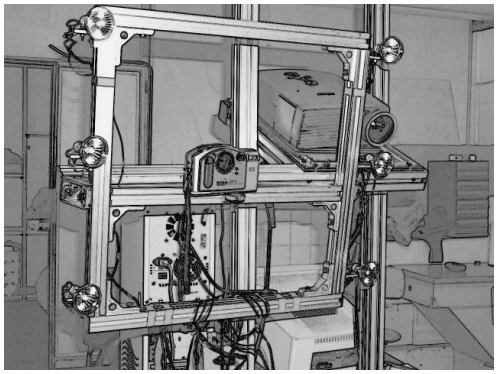
[15] Die Berechnung der Entfernung für korrespondierende Pixel muss für jedes Pixel durchgeführt und diese Pixel vorher in beiden Bildern gesucht werden.

[16] Tageslicht oder eine zusätzliche Beleuchtung mit sichtbaren Lichtes beziehungsweise Infrarotstrahlung.

5.4.5 Aktives 3D-Stereo



(a) Triangulation mit strukturiertem Licht , Quelle: Rocchini u. a. (2001)



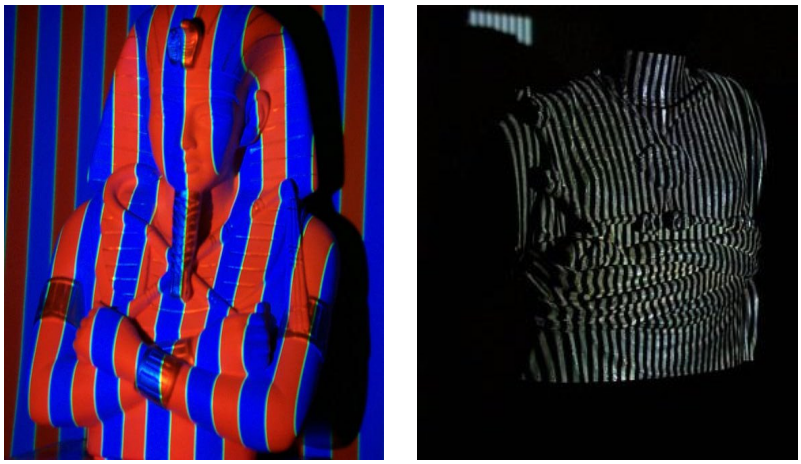
(b) 3D System mit strukturiertem Licht, Quelle: Rocchini u. a. (2001)

Abbildung 5.6: aktives 3D Stereo

Ein aktives System verwendet nur eine Kamera, benötigt jedoch zusätzlich eine Quelle für *strukturiertes Licht* (siehe Aggarwal u. Wang (1988), Boyer u. Kak (1987)). Bei strukturiertem Licht ist ein Muster im Licht vorhanden. Dieses Muster wird bei der Projektion auf eine Fläche sichtbar.

Ein Beispiel für ein solches System, entwickelt zum Scannen kultureller Artefakte wie Statuen, findet sich bei Rocchini u. a. (2001). Es ist ein kostengünstiger optischer 3D-Scanner (siehe Abbildung 5.7a), basierend auf einer handelsüblichen CCD-Kamera und einem Beamer. Die dabei verwendete Triangulation (siehe Abbildung 5.6b) ist vergleichbar mit derjenigen in passiven Systemen mit dem Unterschied, dass hier an der Position einer zweiten Kamera die Lichtquelle installiert ist. Durch die projizierte Struktur werden die Koordinaten bestimmbar und damit lässt sich die Position im Raum errechnen. Dabei gilt wie beim passiven 3D-Stereo-Verfahren für die Entfernung h: $h = \frac{b}{\left(\frac{1}{tan(\alpha)} + \frac{1}{tan(\beta)}\right)}$.

Abbildung 5.7 zeigt zwei Bilder als Beispiele, die mit diesem System aufgenommen wurden. Die Verwendung von Farben (Abbildung 5.7a) als zusätzliche Art der Kodierung bietet gegenüber der Beschränkung auf Schwarz-Weiß (Abbildung 5.7b) den Vorteil, dass durch die Reflexion an der Oberfläche des betrachteten Objektes hervorgerufene Störungen unterscheidbar werden vom an dieser Stelle originalem Muster.



(a) strukturiertes Licht. farbig Quelle: Rocchini u. a. (2001)

(b) strukturiertes Licht, schwarz-weiss. Quelle: Rocchini u. a. (2001)

Abbildung 5.7: Beispiele für 3D-Stereo mit strukturiertem Licht

Ein auf dieser Technik basierendes System zur 3D-Rekonstruktion (Tsalakanidou u. a. (2005)) liefert beim Einsatz eines Einkern-Prozessors (Pentium4) mit 3,2 GHz eine Bildrate von 23 Bildern in der Sekunde ((Tsalakanidou u. a., 2005, S. 44)), ist also prinzipiell für Echtzeit-Anwendungen geeignet. Ein Beispiel für die Ergebnisse dieser Arbeit zeigt Abbildung 5.8.

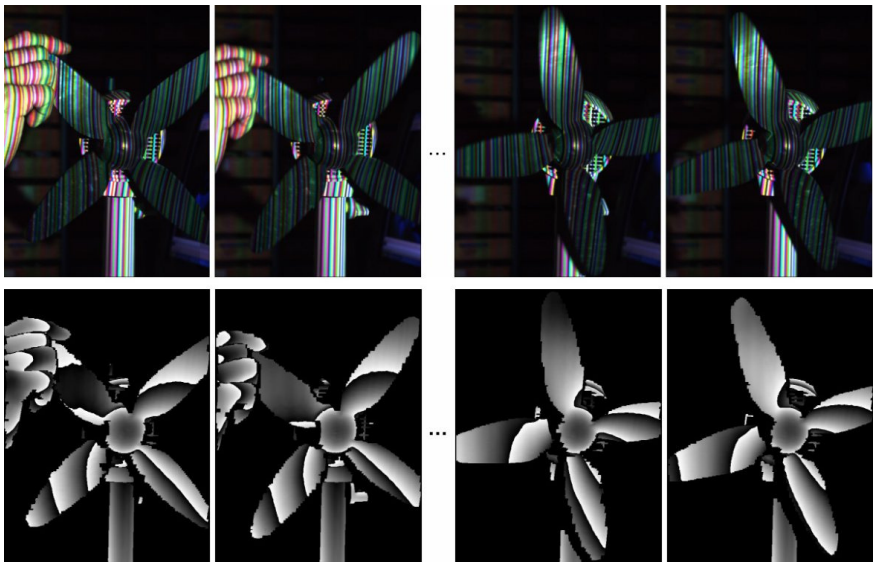


Abbildung 5.8: Aktives 3D-Stereo mit farbigem Linienmuster. Quelle: [(Tsalakanidou u. a., 2005, S. 38)]

Nähert sich ein Objekt der Kamera, so kommt es aufgrund der unterschiedlichen Ursprünge von Kamera und Projektor zu einer Rechtsverschiebung im Kamerabild (siehe Abbildung 5.9). Ist der Wert dieser Verschiebung durch das Aufzeichnen von Bildern in unterschiedlichen Entfernungen im Vorwege bekannt, so kann direkt auf die Entfernung geschlossen werden, ohne dass diese durch Triangulation berechnet werden muss.

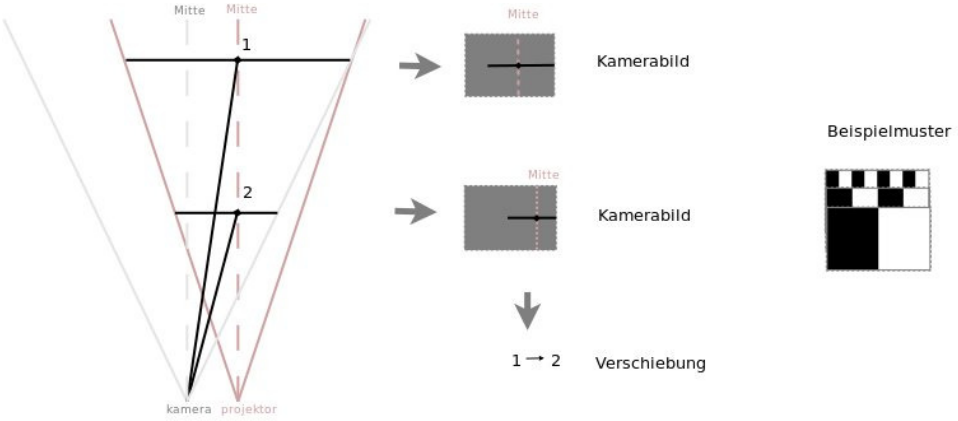


Abbildung 5.9: Rechtsverschiebung bei strukturiertem Licht

Pseudozufällige Kodierung

Eine Erweiterung findet sich bei Morano u. a. (1998). Dort wird als Strukturierung ein pseudozufälliger Code (M-Array, siehe MacWilliams u. Sloane (1976)) verwendet (Abbildung 5.10). Die Kodierung erfolgt dort mit einem Punktmuster, entweder wie im Beispiel mit verschiedenen Graustufen (im Beispiel schwarz, grau, gestreift und weiß dargestellt) oder mit unterschiedlichen Farben. Diese Kodierung ermöglicht eine genaue Positionsbestimmung. Ebenso lässt sich diese Position mehrfach im Muster – und damit fehlertolerant – kodieren.

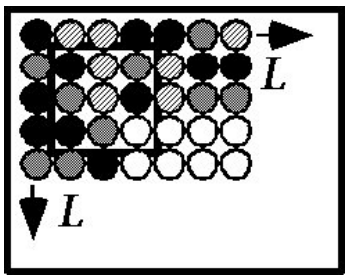


Abbildung 5.10: Codematrix Quelle: [Morano u. a. (1998)]

5.4.6 Vorteile

Die Vorteile von aktivem 3D-Stereo sind:

Im Vergleich zu spezieller Hardware (beispielsweise Time-of-Flight, siehe Abschnitt 5.5) sind die Kosten relativ gering, wenn kommerziell verfügbare Standardkomponenten (Beamer, Kamera) zum Einsatz kommen.

Es können verschiedene Lichtmuster verwendet werden. Da die Muster mit einem Beamer erzeugt werden, ist ein Wechsel leicht möglich und somit der Vergleich verschiedener Muster.

5.4.7 Nachteile

Das Verfahren hat folgende Nachteile:

Beim Einsatz von Farbkodierung kann es zu Problemen bei stark gefärbten Objekten kommen, da diese die Farben des Musters bei der Reflexion verändern.

Es sind keine Komplettsysteme kommerziell verfügbar, nur im Rahmen von akademischen Projekten entwickelte Prototypen.

Das projizierte Muster liegt im sichtbaren Lichtspektrum, blendet also den Benutzer.[17]

Durch die Berechnung in der Software entstehen höhere Latenzen in der Verarbeitung als beim Einsatz spezialisierter Hardware (zum Beispiel ein FPGA).

5.5 Time-of-Flight

Time-of-Flight Kameras (auch *ToF-Kameras*) verwenden eine Laufzeitmessung des ausgesandten Lichts zur Bestimmung der Entfernung eines Objektes. Sie kommen sowohl in industriellen Anwendungen (zum Beispiel zur Beurteilung von Werkstücken) als auch für die Mensch-Maschine-Interaktion zum Einsatz.

5.5.1 Grundsätzliches Verfahren

Das Verfahren basiert auf der Messung der Laufzeit der ausgesandten Lichtimpulse von der zugehörigen Lichtquelle über das reflektierende Objekt zurück zur Kamera. Dabei gibt es zwei Varianten: die Aussendung von dedizierten Pulsen von Licht und die Verwendung eines kontinuierlichen, modulierten Signals. Dabei wird Licht aus dem Infrarotspektrum verwendet.

[17] Dies macht das System eher akademisch als praktisch interessant, da ein Benutzer dies nicht tolerieren würde.

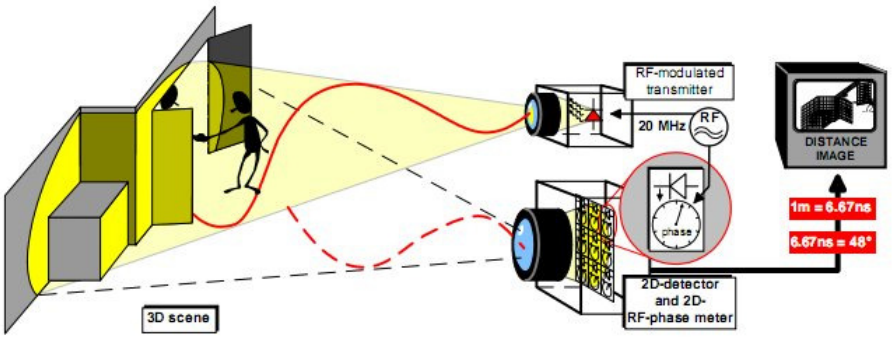
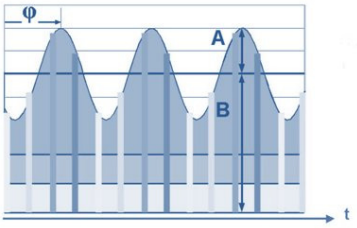


Abbildung 5.11: Modulation bei ToF, Quelle: [(Lange, 2000, S. 39)]

5.5.2 Pulsverfahren

Bei dieser Art der Laufzeitmessung wird ein kurzer Lichtpuls (im Nanosekundenbereich) ausgesandt und die rückstrahlende Lichtmenge mit Hilfe eines Kamerachips gemessen. Lichtpuls bedeutet in diesem Falle, dass die Lichtquelle tatsächlich an- und wieder abgeschaltet wird. Da die Menge des ausgestrahlten Lichtes gering ist, wird dieser Vorgang mehrmals wiederholt, ehe der Chip ausgelesen wird. Das Pulsverfahren erfordert eine sehr präzise Lichtquelle in Form von Infrarot-Dioden oder Lasern. Die Ansteuerung dieser Lichtquelle ist aufwendig.

(a) SR4000 Kamera, Quelle: Mesa Imaging AG (2010)



(b) Time-of-Flight moduliertes Signal Quelle: SR4

Abbildung 5.12: SR4000 Kamera und Signal

5.5.3 Moduliertes Signal

Die zweite Methode ist die Verwendung einer Sinus-Modulation für die Lichtimpulse (siehe Abbildung 5.12b) und die Messung der Phasenverschiebung zwischen ausgesandtem und dem empfangenen Signal. Die Berechnung der Entfernung erfolgt aus deren Diskrepanz (siehe Abbildung 5.11). Der Nachteil dieser Variante ist, dass Objekte, die die maximale Entfernung überschreiten, nicht von Objekten unterschieden werden können, die sich bis zur maximalen Entfernung an der Kamera befinden. Sie erscheinen in diesem Falle wie Objekte, die sich innerhalb der Reichweite der Kamera befinden, ein Objekt in 11 Metern Entfernung befindet sich also aus Sicht der ToF-Kamera in einem Meter Entfernung.

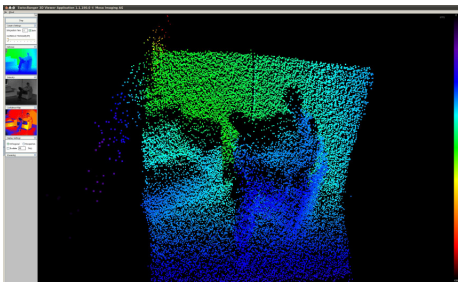


Abbildung 5.13: Swissranger 3D View im Living Place

5.5.4 Störung durch natürliche Infrarotstrahlung

Die SR4000 Kamera verwendet Infrarot-Strahlung im Bereich um 850 Nanometern (siehe SR4). Die dabei insgesamt abgegebene Beleuchtungsstärke beträgt ca. 1 Watt. Die natürliche Beleuchtungsstärke beträgt für 850 Nanometer[18] etwa 0,91 W/m^2 (siehe (NREL), Abbildung 5.14). Bei einer in drei Meter Entfernung von der SR4000 erfassten Fläche von 4,63m^2 (240x193cm, siehe Tabelle 5.1) ergibt sich eine Beleuchtungsstärke von etwa 0,22 W/m^2 , also ungefähr einem Viertel der natürlichen Einstrahlung[19] in diesem Spektrum. Um dies zumindest teilweise auszugleichen, verwendet die SR4000-Kamera ein Verfahren, bei dem die Kamera durch Phasen ohne eigene Beleuchtung die Hintergrundbeleuchtung ermittelt und diese Werte bei der Berechnung berücksichtigt, indem die gemessene Grundbeleuchtung subtrahiert wird.

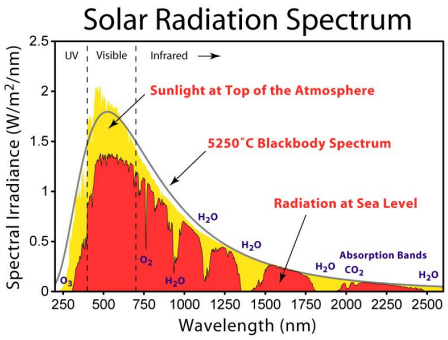


Abbildung 5.14: Spektrum Sonnenlicht, Quelle: Rohde (2007)

[18] Dabei ist zu beachten, dass der Filter in der SR4000 einen größeren Bereich als nur genau 850 Nanometer passieren lässt.

[19] Die reale Beleuchtungsstärke ist im Innenbereich abhängig von der Größe der Fenster und ihren Filtereigenschaften für infrarotes Licht.

5.5.5 Vorteile

Die Vorteile von Time-of-Flight-Kameras sind:

Mehrere Kameras sind problemlos kombinierbar.[20]

Time-of-Flight benötigt keine Hintergrundbeleuchtung.

Es ist bedingt möglich, auch bei direktem Sonnenlicht zu arbeiten.

Das Verfahren benötigt keine strukturierte Oberfläche, da das Verfahren selber für eine Strukturierung sorgt.

Die verwendete Beleuchtung blendet die Anwender nicht.

5.5.6 Nachteile

Dabei haben Time-of-Flight-Kameras die folgenden Nachteile:

Das Time-of-Flight Verfahren funktioniert nicht bei schwarzen Oberflächen, da diese zu viel des ausgesandten Infrarotlichts absorbieren.

Es treten Artefakte (sogenannte *fliegende Punkte*) auf, besonders an den Rändern von Gegenständen.

Das Verfahren ist empfindlich gegenüber zu viel IR-Hintergrundstrahlung (Sonnenlicht) bei wechselnder Beleuchtungslage durch Wolken.

Durch Reflexionen kann das Licht über mehrere Wege zu einem Objekt gelangen und dadurch zu fehlerhaften Werten führen.

[20] Bei der SR4000 von MesaImaging maximal drei

5.6 Light Coding

Light Coding[21] ist ein Verfahren, das von der israelischen Firma PrimeSense[22] in ihren 3D-Kameras verwendet wird.

5.6.1 Verfahren

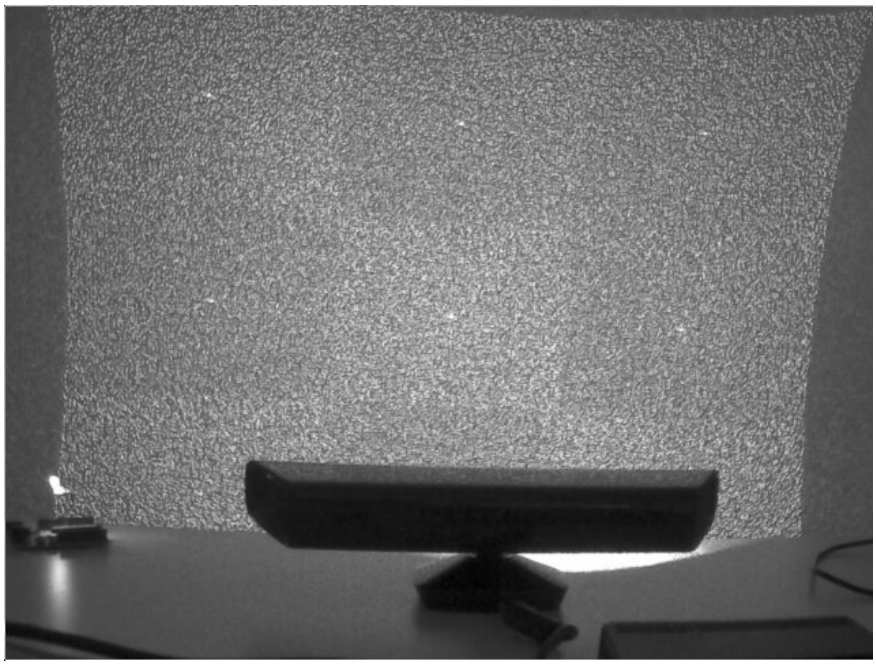


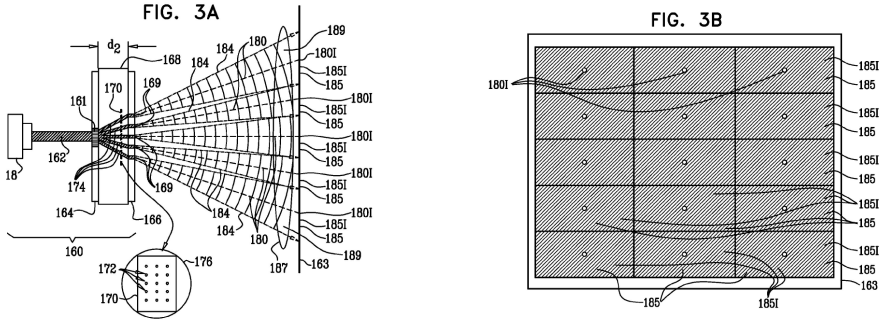
Abbildung 5.15: Kinect Fleckmuster

Der genaue Algorithmus ist nicht dokumentiert, der Ablauf lässt sich jedoch aufgrund von Patenten und eigenen Untersuchungen sowie Untersuchungen Dritter vermuten.

Shpunt u. a. (2010) beschreiben einen möglichen Ablauf der 3D-Rekonstruktion. Grundsätzlich beruht das Verfahren auf der Verwendung einer Quelle für strukturiertes Licht, in diesem Falle ein Infrarot-Laser (*IR-Laser*) mit einer Wellenlänge von 780 Nanometer in Kombination mit einer Infrarot-Kamera (*IR-Kamera*). Das grundsätzliche Verfahren ist analog zu aktivem 3D-Stereo (vergleiche 5.4.5).

[21] Eingetragenes Warenzeichen von PrimeSense Ldt, Israel
[22] http://www.primesense.com

Abbildung 5.15 zeigt das auf eine Wand projizierte Muster mit einer Kinect im Vordergrund. Dabei ist eine Meta-Strukturierung zu erkennen. Das Gesamtmuster ist unterteilt in ein schachbrettartiges Muster, bestehend aus neun *Kacheln*. Im Zentrum ist jeweils ein hellerer Leuchtpunkt zu erkennen. Grund für diese hellen Mittelpunkte scheint die Verwendung eines zweifachen Diffusors zum Aufspalten des emittierten Laserstrahls zu sein (siehe Abbildung 5.16a). Durch die Trennung im ersten Diffusor entsteht ein stärkerer Reststrahl, der im entstehenden Bild deutlich sichtbar bleibt als Mittelpunkt einer der neun *Kacheln*. (5.16b).



(a) Strahl Aufteilung mit 2fachem DOE , Quelle: Shpunt (2009)

(b) Strukturiertes Muster, Quelle: Shpunt (2009)

Abbildung 5.16: Meta-Muster Ursache beim Light Coding

Laut Shpunt u. a. (2010) dient die unterschiedliche Helligkeit in den *Kacheln* (siehe Abbildung 5.15) dazu, den Abschnitt zu identifizieren, in dem sich der Algorithmus gerade befindet. Dies ist wichtig, da sich das Muster in jeder Kachel wiederholt und so nicht anhand des Musters erkannt werden kann, in welcher Kachel sich dieses befindet. Das Muster findet sich in Abbildung 5.17.

Auffallend sind die Parallelen zur in Abschnitt 5.4.5 vorgestellten pseudozufälligen Kodierung.

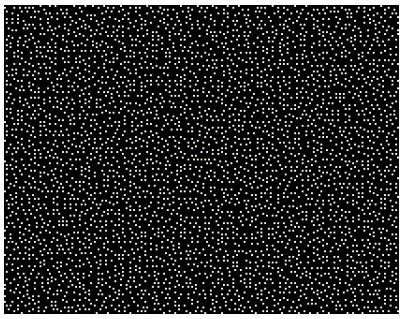
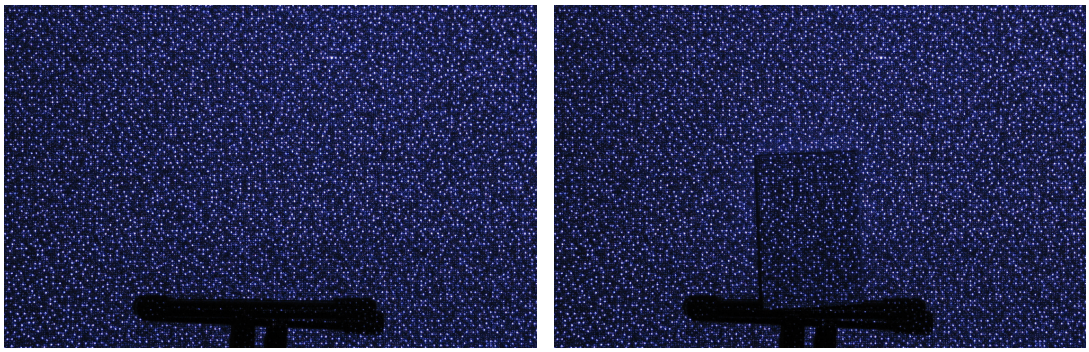


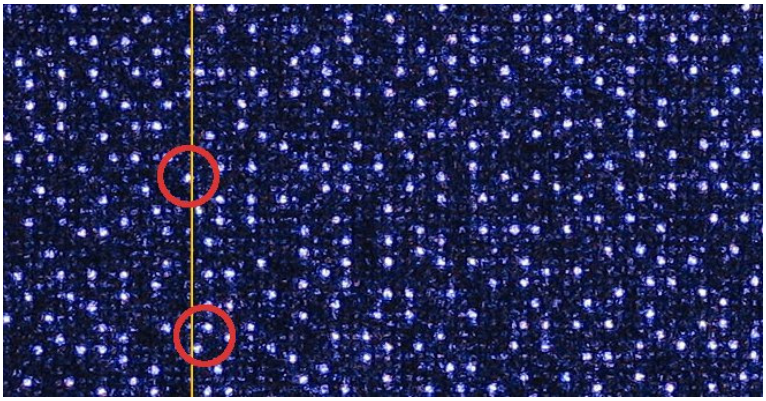
Abbildung 5.17: PrimeSensor-Muster. Quelle: Reichinger (2011)

Durch die Anordnung des Systems, bei der der IR-Laserprojektor um 8 cm versetzt neben der IR-Kamera angebracht ist, ergibt sich bei Verringerung der Entfernung zum Hintergrund eine Verschiebung des Musters nach rechts (siehe Abbildungen 5.18). Durch die Bestimmung dieser Verschiebung, beziehungsweise dem Vergleich mit in unterschiedlichen Entfernungen gespeicherten Referenzmustern, lässt sich die Entfernung von der Kamera ermitteln. Eine Grundvoraussetzung dafür ist, dass bekannt ist, an welchen Koordinaten im Bild sich der Algorithmus zu jedem Zeitpunkt befindet.

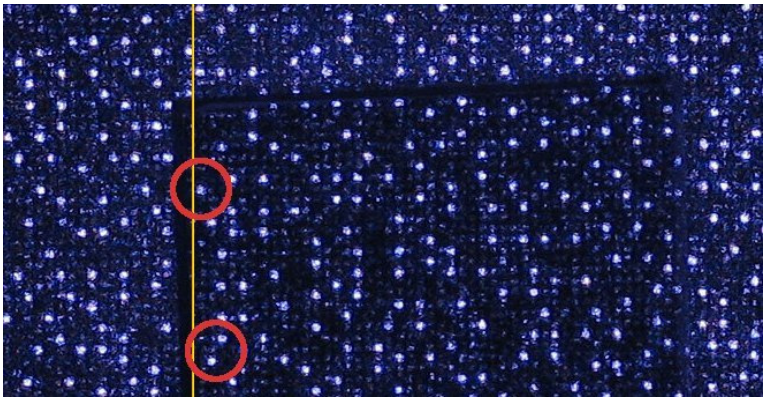


(a) Kinect-Muster. Quelle: Daniel Reetz (2011a) **(b)** Kinect-Muster mit Gegenstand. Daniel Reetz (2011a)

Abbildung 5.18: Kinect Muster mit Gegenstand



(a) Kinect-Muster-Zoom. Quelle: Daniel Reetz (2011a)



(b) Kinect-Muster-Zoom mit Gegenstand. Quelle: Daniel Reetz (2011a)

Abbildung 5.19: *Light Coding*: Muster-Verschiebung bei Veränderung der Position im Raum

Die Aussendung des Lichts erfolgt hierbei kontinuierlich, es erfolgt keine Pulsung der Lichtquelle (siehe Daniel Reetz (2011b)). Die dabei gemessene Stärke des Lasers findet sich in Tabelle 5.2.

Tabelle 5.2: Gemessene Leistung des Kinect IR-Laser, Quelle: Covey u. Chen (2011)

Entfernung	Leistung
61cm	**0,004 mW**
30cm	**0,09 mW**
10cm	**0,25 mW**
5cm	**0,67 mW**
1 cm	**1,5 mW**
0,5 cm	**2,9 mW**

Kalibrierung

Die Kalibrierung des Systems erfolgt während der Produktion der Geräte, oder bei entsprechend geringen Toleranzen in der Fertigung, anhand eines Musters für die gesamte Serie. Angesichts des Preises von unter 100 Euro ist vom zweiten Fall auszugehen.

(a) Kinect Quelle: Amazon Co. Uk (2011)

(b) PrimeSensor Quelle: ASUSTeK COMPUTER INC. (2011)

Abbildung 5.20: Kinect und PrimeSensor (Xtion pro)

5.6.2 PrimeSensor

PrimeSensor ist das originale Produkt der Firma PrimeSense. Das Produkt ist nur für Entwicklungszwecke verfügbar. Zusätzlich wird die Technik in unterschiedlichen Varianten von OEM-Herstellern wie der Microsoft *Kinect* und dem *Asus Xtion PRO* eingesetzt. Letzteres ist im Living Place verfügbar.

5.6.3 Kinect

Das Kinect-System wurde von der Firma Microsoft für die Spielekonsole Xbox 360 im Jahre 2010 eingeführt.[23] Es handelt sich bei dieser Kamera um eine modifizierte Version des PrimeSensors. Die Modifikation besteht in der Integration einer Videokamera (siehe Abbildung 5.21).

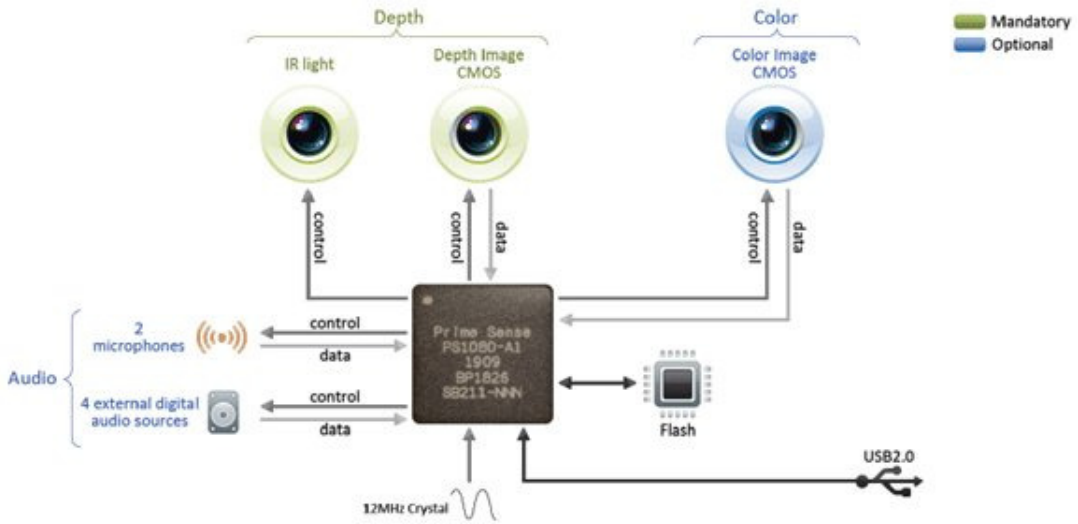


Abbildung 5.21: PrimeSensor Blockdiagramm, Quelle:[PrimeSense Ltd (2010)]

5.6.4 Verwendung mehrerer Systeme

Bei auf *Light Coding* basierenden Kameras ist die Kombination von zwei Systemen möglich. Oliver Kreylos hat dazu Versuche mit zwei Kinects durchgeführt.[24] Die Überlagerung der Muster führt dabei nicht zu einem völligen Systemabbruch. In den meisten Fällen erkennt das System, welche Punkte des Musters zu welcher Kamera gehören. Es kommt dabei allerdings zu Fehlern in Teilbereichen des Bildes. Die Kombination von mehr als zwei Systemen ist nicht getestet.

Als Alternativen sind folgende Möglichkeiten zur Synchronisation der Infrarot-Projektion möglich:

[23] http://www.xbox.com/en-US/Xbox360/Accessories/Kinect/Home
[24] http://idav.ucdavis.edu/~okreylos/ResDev/Kinect/

Internes Pulsen der Infrarotstrahlung

Die Hardware könnte so modifiziert werden, dass die Ausstrahlung des Infrarot-Lasers nicht kontinuierlich, sondern in einer Art Zeitschlitz-Verfahren (Time division multiplexing, TDM) erfolgt. Problematisch könnten hierbei Vorlaufzeiten des Infrarot-Lasers sein. Ob es eine Unterstützung in der existierenden Hardware für eine dann nötige Synchronisation gibt, ist nicht bekannt.

Externes Pulsen der Infrarotstrahlung

Vorgebaute mechanische Shutter, die miteinander synchronisiert werden, können verwendet werden, um das Licht aus den Infrarot-Lasern zeitweilig zu unterbrechen. Ob dies die Probleme löst, ist nicht geklärt, da auch hierbei die Infrarot-Kamera des Systems mit dem abgedeckten Laser die Muster des anderen Systems sehen. Gegebenenfalls müsste auch die Infrarot-Kamera für diesen Zeitraum abgedeckt werden.

5.6.5 Vorteile

Die Vorteile von *Light Coding* sind:

Es ist eine preisgünstige Technik.

Das Verfahren funktioniert auch ohne Hintergrundbeleuchtung.

Es benötigt keine strukturierte Oberfläche, da das Verfahren selber für eine Strukturierung sorgt.

5.6.6 Nachteile

Die Nachteile sind:

Das Verfahren ist nicht offengelegt, eventuelle Schwachstellen sind also nur experimentell zu bestimmen.

Da Infrarot-Licht verwendet wird, ist das Verfahren prinzipiell anfällig gegenüber Infrarotstrahlung aus anderen Quellen, zum Beispiel Sonnenlicht (siehe auch 5.5.4).

Es gibt keine offizielle Unterstützung für mehrere Kameras.

5.7 Vergleich

Tabelle 5.3 fasst die wichtigsten Eigenschaften der betrachteten Kamerasysteme zusammen.

Tabelle 5.3: Vergleich der Eigenschaften verschiedener 3D-Kameras

Kamera	Bumblebee 2	MR C3D	Kinect	PrimeSensor	SR4000
Technik	Stereo passiv	Stereo passiv[1]	Light Coding (Stereo aktiv)	Light Coding (Stereo aktiv)	Time-of-Flight
Sensor	CCD	CCD	CMOS	CMOS	CCD/CMOS
Berechnung der Tiefeninformationen	Host	Kamera (FPGA)	Kamera (SOC)	Kamera (SOC)	Kamera (SOC)
Preis[2]	ca. 6500 €	ca. 6500 €	ca. 100 €	ca. 150 €	ca. 7000 €
Latenz[3]	k.A.	k.A.	k.A.	max. 40ms	max. 150ms
Reichweite	> 10m [4]	> 10m [4]	3,5m	3,5m	10m
maximale Auflösung[5] (B x H)	1024x768	752x480	640x480	640x480	172x144
maximaler Blickwinkel[6] (B x H)	97x73[8] 66x50[9]	75x52	58x45	58x45	43.6x34.6
erfasste Fläche (B x H) 3m abstand[12]	678x444[8] 390x280[9]	460x293	333x249	333x249	240x193
Tiefenauflösung[13]	variabel	6 bit	11 bit (1,7mm)	11 bit (1,7mm)	14 bit (0,6mm)
Pixelgröße (B x H) 3m abstand[13]	6mm x 6mm[8] 4mm x 4mm[9]	6mm x 6mm	5mm x 5mm	5mm x 5mm	14mm x 13mm
Bildrate	20 fps (1024x768) 48 fps (644x488)	30 fps	30 fps	30 fps (640x480) 60 fps (320x240)	54 fps
Schnittstelle (max. Datenrate)	IEEE-1394a (400 MBit/s)	PCI oder PC104+ (1064 MBit/s)	USB 2.0 (480 MBit/s)	USB 2.0 (480 MBit/s)	USB 2.0 (480 MBit/s), Ethernet (100 MBit/s)
max Anzahl[14]	unbegrenzt	unbegrenzt	1	1	3
Außeneinsatz[10]	Ja	Ja	Nein	Nein	Nein
Benötigt externe Beleuchtung	Ja	Ja	Nein	Nein	Nein
benötigte Rechenleistung[15]	hoch	gering	gering	gering	gering

[1] mit internem FPGA
[2] incl. MwSt.
[3] Herstellerangabe
[4] Keine Herstellerangabe, theoretisch Sichtweite
[5] in Pixel
[6] in Grad
[8] 2.5mm Linse
[9] 3.8mm Linse
[10] Kamera ist für Betrieb im Sonnenlicht geeignet
[11] in Grad
[12] in cm (gerundet) bei 3m Entfernung)
[13] gerundet, bei 3m Entfernung
[14] Auf einen Bildbereich
[15] auf dem Hostsystem

5.8 Einschränkende Faktoren

5.8.1 Auflösung

Je nach gewünschter Extremität für die Gesten ist die Auflösung der Kamera und damit verbunden die Auflösung der einzelnen Pixel (siehe Tabelle 5.1) der limitierende Faktor. Die Erkennung einer Hand erfordert eine andere Auflösung als die einzelner Finger. Diese Auflösung ist abhängig vom Abstand des erfassten Objektes von der Kamera.

Es besteht die Möglichkeit, die Auflösung für bestimmte Teilbereiche – etwa eine erkannte Hand – durch Kombination mit einer zweiten hochauflösenden 2D-Kamera zu erhöhen (siehe Kapitel 7.6). Diese benötigt gegebenenfalls eine zusätzliche Beleuchtung.

5.8.2 Bildrate

Eine zu geringe Bildrate führt zu Unschärfe bei der Aufnahme von schnellen Bewegungen, da die wechselnden Positionen des Objektes in der Bewegung gleichzeitig im Bild sichtbar sind. Für die Bildrate gilt die Regel, je höher, desto besser. Die minimale Bildrate sollte 30 Bilder pro Sekunde nicht unterschreiten (Kirishima u. a. (2005)).

5.8.3 Beleuchtung

Einige der technischen Verfahren zur Gestenerkennung setzen eine bestimmte Umgebung voraus. Beispielsweise sind auf infrarotem Licht basierende Kamerasysteme gegenüber Sonnenlicht empfindlich. Da die Leuchtstärke des Sonnenlichts erheblich sein kann (bis zu 100000 Lux, siehe mey (1981)) ist eine Überstrahlung des vom System emittierten Lichtes wahrscheinlich.

Das Gegenteil ist der Fall wenn das Verfahren, wie beispielsweise 3D-Stereo, eine Mindestbeleuchtung voraussetzt. Dies kann zu einer deutlichen Einschränkung der Einsatzmöglichkeiten führen. Bei Einsatzgebieten, die eine geringe Beleuchtung voraussetzen, wie etwa beim Halten eines Vortrags. Wenn keine Farbsicht erforderlich ist, besteht die Möglichkeit, auf Infrarot-Kameras auszuweichen, die zwar weiterhin eine Beleuchtung erfordern, aber eben nicht sichtbar für die Anwender.

5.8.4 Größe des Sichtbereichs

Auch bei 3D-Kameras ist ein *Sichtbereich* vorhanden, also ein Bereich, den die Kamera erfassen kann, ausgehend von der Position der Kamera selber in den Raum hinein. Dieser Sichtbereich kann durch Personen oder Gegenstände ständig oder zeitweise verdeckt werden. In diesem Falle ist unter Umständen die Kombination mehrerer Kameras und die gemeinsame Ausrichtung auf den selben Sichtbereich notwendig, vorausgesetzt, die Kameras unterstützen dies.

Abbildung 5.22: Grafischer Vergleich der Sichtbereiche, Angabe der Größe bei 3 Meter Abstand in Zentimetern, Breite x Höhe

Abbildung 5.22 verdeutlicht die unterschiedliche Größe der Sichtbereiche für die verschiedenen Systeme in einer Entfernung von drei Metern von der Kamera. Drei Meter entsprechen dabei einer typischen, in einem Wohnzimmer realisierbaren Entfernung. Je nach geplantem Anwendungsgebiet der Gesten, wie beispielsweise eine Steuerung für Multimediageräte mit den Händen (siehe Abschnitt 3.3), ist es nicht notwendig, die gesamte Person zu erfassen, sondern nur Teile von ihr.

5.8.5 Situation

Von besonderem Interesse ist die Situation im Raum, der zur Gestenerkennung genutzt werden soll. Es ist leicht vorstellbar, dass es im Gegensatz zu einer Person, die beispielsweise auf einem Sofa sitzt und die Lautstärke der Musikanlage regeln will, im Falle einer Party mit vielen Menschen im Raum und damit im Sichtfeld der Kamera zu Problemen kommen kann.

5.8.6 Pixelgröße

Die Größe der Pixel des Tiefenbildes ist abhängig von der Entfernung zum Benutzer. Auch hier ist die Annahme einer Entfernung von drei Metern sinnvoll. Beim Vergleich der unterschiedlichen Systeme fällt auf, dass die SR4000-Kamera hier einen mehr als doppelt so großen Wert liefert. Abhilfe bietet hier nur eine Kombination mit einer zweiten Kamera (siehe 7.6).

5.9 Fazit

Alle betrachteten Systeme bringen Einschränkungen mit sich. Insbesondere die Beschränkung der Sichtbereichs auf wenige Quadratmeter macht es entweder nötig, nur einen bestimmten Bereich als Interaktionsraum zu definieren oder mehrere Kameras zu verwenden. Obwohl dies beim PrimeSensor nach praktischen Erfahrungen möglich scheint, ist es nur bei den passiven Stereo Varianten und bei Time-of-Flight garantiert. Im Falle der Verwendung des PrimeSensors oder der Kinect sind weiterführende Untersuchungen notwendig.

Aktive 3D-Stereo Systeme eignen sich gut für die Erzeugung von Modellen unbelebter Gegenstände, durch den Blendeffekt sind sie für Gestensteuerung nicht geeignet.

Die geringe Auflösung der ToF-Kameras – mit Pixelgrößen von über einem Zentimeter bei drei Metern Entfernung – macht eine Erkennung von Gesten mit geringen Bewegungen schwierig. Ebenso im Falle der Verwendung von Fingern für die Gestik. Dies kann die Kombination mit einer weiteren Kamera erforderlich machen.

Time-Of-Flight-Kameras sind prinzipbedingt besser zum Betrieb in Umgebungen mit Sonnenlicht geeignet. Sofern es nicht zu ständigem Wechsel von Beleuchtungsstärken kommt, können sie dies besser kompensieren als die Variante von Primesense.

Je nach Einsatzbereich dürfte also eine Kombination mehrerer ToF-Kameras oder das Primesense/Kinect System die sinnvollsten Kandidaten für eine Gestensteuerung im Bereich Smart Home sein. Eine praktische Evaluierung, insbesondere in komplizierten Umgebungen mit verschieden Beleuchtungen, Möbeln und Personen ist notwendig.

Die vom Hersteller für die SR4000 Kamera angegebene Latenz von 150 Millisekunden ist zu groß für eine Erkennung in Echtzeit. Ob diese Latenzzeit in der Praxis diese Werte erreicht, zeigt das folgende Kapitel.

6 Latenzmessungen von Kamerasystemen

Dieser Abschnitt beschäftigt sich mit den im Rahmen dieser Arbeit durchgeführten Latenzmessungen an verschiedenen Kamerasystemen. Dabei wurden folgende Kameras verwendet: Die Tiefenkamera des Kinect-Systems, ASUS Xtion Pro (PrimeSensor), SR4000, Playstation3 Eye und Axis P1344. Bei der Asus P1344 handelt es sich um eine Netzwerk-Kamera, die sowohl für sichtbares Licht als auch Infrarotstrahlung verwendet werden kann. Die Playstation3 (PS3) Eye Kamera wurde zur Überprüfung der Latenzzeiten des Messaufbaus eingesetzt.

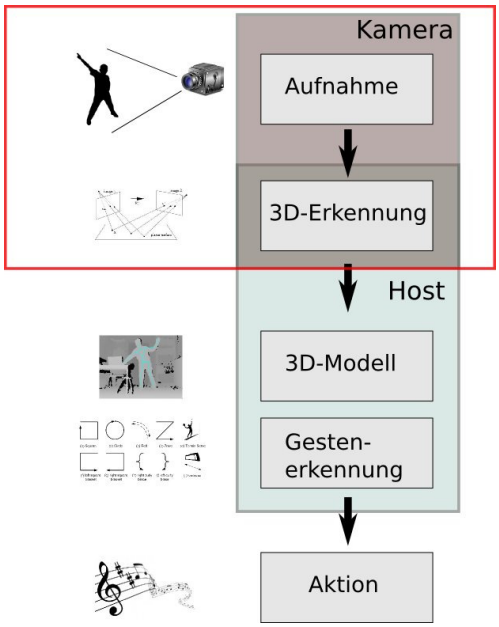


Abbildung 6.1: Übersicht Ablauf Gestenerkennung

Hintergrund der Messungen ist die Fragestellung, ob die betrachteten Systeme zum Aufbau eines Systems zur Gestenerkennung in Echtzeit[1] – mit einer Gesamtlatenz kleiner als 150 Millisekunden – geeignet sind. Die Axis-Kamera wurde in die Messung aufgenommen, um die Frage zu klären, ob die Kamera für eine eventuelle Kombination mit der SR4000 geeignet ist.[2]

Die Messungen erfolgten in einem abgedunkelten Raum, um Störungen durch Hintergrundbeleuchtung auszuschließen[3].

[1] siehe Abschnitt 4.2.9

[2] Zur Kombination von Time-of-Flight und Videokamera siehe Abschnitt 7.6.

[3] Die Axis P1344 ist nach Deaktivierung des Infrarotfilters in der Lage, ebenfalls in dunkler Umgebung zu arbeiten.

6.1 Verwendetes System

Das für die Messungen benutzte System besteht aus einem Ubuntu Linux basiertem PC, verbunden mit den einzelnen Kameras und einem *Arduino Embedded Board*. An dieses ist eine Infrarot-Diode angeschlossen. Das Messverfahren basiert auf der Idee, dass diese Diode bei den normalen optischen Kameras direkt sichtbar ist[4] oder sich durch Störungen bei den Tiefenkameras bemerkbar macht.

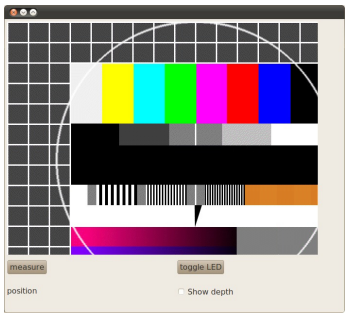


Abbildung 6.2: Screenshot LpLatencyMeasure

6.1.1 Messaufbau

Die Hardware für den Messaufbau besteht aus folgenden Komponenten:

- PC
- Prozessor: Intel Core i5 CPU 750 mit 2,67Ghz
- Arbeitsspeicher: 2GB
- Arduino Duemilanove (2009)
- ATMEGA 328 Mikrocontroller
- Infrarot-LED (770nm)
- Kamera (Kinect, SR4000, Xtion Pro, Axis P1344 oder PS3-Eye)

Als Betriebssystem für die Messungen kommt Ubuntu 10.04 mit einem Linux-Kernel 2.6.32-33 zum Einsatz.

Eine Darstellung des eingesetzten Arduino inklusive Infrarot-LED (IR-LED) findet sich in Abbildung 6.7. Die Anbindung der Arduino erfolgt über USB 2.0 an einem getrennten Bus, so dass es zu keiner Interferenz mit den per USB angebundenen Kameras auf Hardware-Ebene kommt. Die Kommunikation mit dem Arduino erfolgt über die Emulation einer seriellen Schnittstelle (FTDI).

[4] trotz des Infrarotfilters der Kamera passiert ein gewisser Anteil an Nah-Infrarotstrahlung

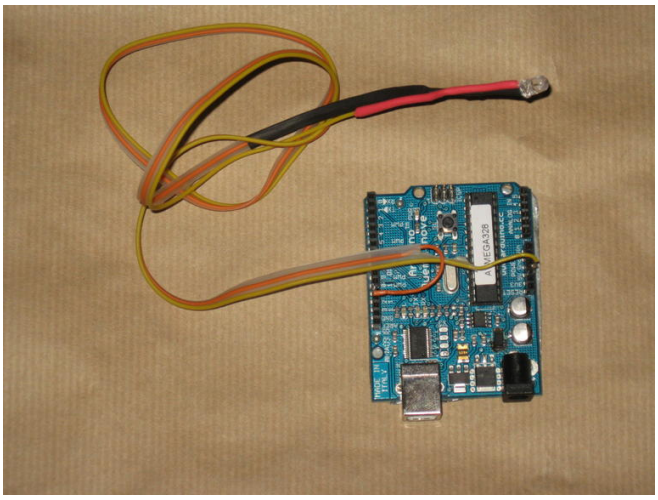


Abbildung 6.3: Arduino mit IR-LED

6.1.2 Software

Das vom Autor dieser Arbeit entwickelte Programm *LpLatencyMeasure* (Living Place Latency Measure) ist in C++ geschrieben und basiert auf den folgenden Bibliotheken:

- **libmesaSR** ist eine Treiberbibliothek für die SR4000-Kamera, siehe Abschnitt 8.3.2.[5]
- **libfreenect** ist eine Treiberbibliothek für die Kinect-Kamera, siehe Abschnitt 8.3.1.[6]
- **wxWidgets** ist eine Bibliothek zum plattformübergreifenden Programmieren von Benutzeroberflächen.[7]
- **OpenNI** ist eine Treiberbibliothek für Kinect und PrimeSensor, siehe Abschnitt 8.3.4.[8]
- **OpenCV** ist eine plattformübergreifende Bibliothek zur Bildverarbeitung.[9]
- **Gstreamer** ist eine Bibliothek zum Verarbeiten von unterschiedlichen Videoströmen.[10]

Gründe für die Entwicklung in C++ waren eine möglichst hohe Ausführungsgeschwindigkeit des Programms sowie die Verfügbarkeit von Programmierschnittstellen zu C++ in den verwendeten Treiberbibliotheken. *OpenNI* wurde zur Anbindung des *ASUS Xtion Pro* Kamerasystems

[5] Version: 1.0.14-653.
[6] Version: 1:0.0.1+201012.
[7] Verwendete Version: 2.8.10, siehe http://www.wxwidgets.org/
[8] Version: 1.3.2.3
[9] Version: 2.1.0-1ppa3, siehe http://opencv.willowgarage.com/wiki/
[10] Version: 0.10.21, siehe gstreamer.freedesktop.org/.

verwendet, da *libfreenect* dies derzeit nicht unterstützt. Libfreenect wurde als ursprüngliche Treiberbibliothek in LpLatencyMeasure integriert, erwies sich jedoch in der derzeitigen Version als unzuverlässig.

Die Verwendung von *Gstreamer* zur Anbindung der Axis-Webcam ermöglicht prinzipiell den Zugriff auf alle Videoquellen, die *Gstreamer* unterstützt. Dazu gehören sowohl auf den Protokollen HTTP und RTP basierende Videoquellen als auch lokale Quellen: Webcams, MPEG und AVI kodierte Dateien.

Die mit LpLatencyMeasure ermittelten Werte wurden mit *GNUPlot* zur grafischen Darstellung bearbeitet.

6.1.3 Architektur

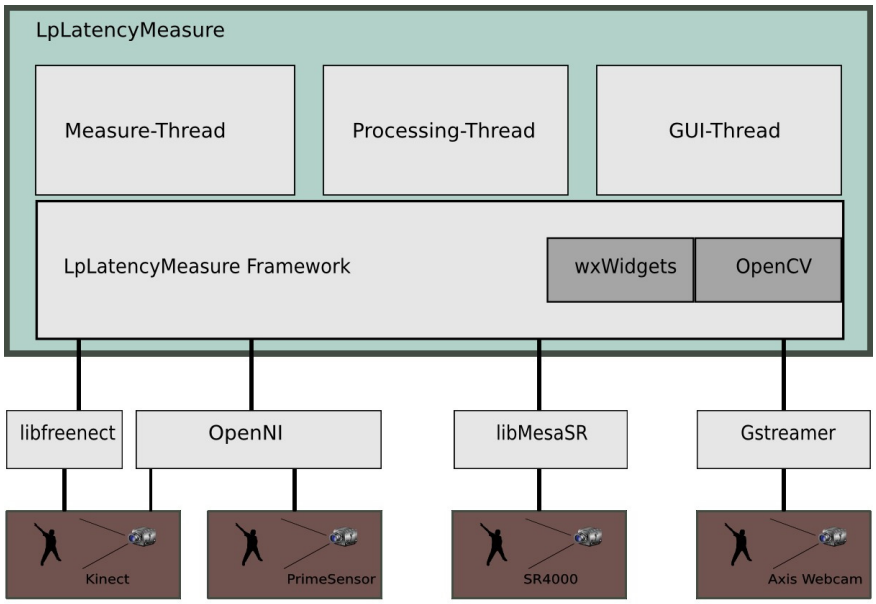


Abbildung 6.4: Architektur LpLatencyMeasure

Zum Einsatz kommt eine Schichtenarchitektur, dargestellt in Abbildung 6.4. Die unterschiedlichen Treiberbibliotheken sind an das LpLatencyMeasure-Framework angebunden, dort sorgt jeweils ein eigener Thread für die Bearbeitung der ankommenden Bilddaten und ihre Speicherung in einer zentral zugreifbaren Datenstruktur.

LpLatencyMeasure ist als Multithreading-Anwendung konzipiert. Das dabei zum Einsatz kommende wxThreads, das dem wxWidgets-Framework entstammt, verwendet unter Linux die Standard-Pthread-Bibliothek.

Die drei Threads der Anwendung werden vom GUI-Thread initiiert und laufen danach parallel. Je nach Art der Bibliothek zum Zugriff auf die Kamera werden von dieser weitere Threads gestartet.

Obwohl alle Bibliotheken prinzipiell plattformübergreifend sind, wurde die Übersetzung des Programms und die Ausführung nur unter Ubuntu Linux getestet.

6.1.4 Ablauf der Messung

Zum Ablauf der Messung siehe Abbildung 6.5. Folgender Algorithmus wurde verwendet:

1. Nach dem Start des Programms erfolgt die Kalibrierung. Dazu wird die LED vom Anwender ein- und wieder ausgeschaltet, und die Stelle im Bild gesucht, die sich verändert.
2. Bei eingeschalteter LED erfolgt die Markierung der LED-Position im Kontrollbild. Dabei wird der Farbwert des Zielpixels gespeichert.
3. Der Messzyklus wird aktiviert.
4. Die LED wird ausgeschaltet.
5. Die LED wird über die USB-Schnittstelle und den Arduino eingeschaltet, dabei wird die aktuelle Zeit gespeichert.
6. Das Zielpixel wird auf Veränderung überwacht.
7. Sobald das Zielpixel den gespeicherten Farbwert erreicht, wird die Zeit erneut gemessen.
8. Die Differenz der Zeitwerte wird bestimmt.
9. Eine (durch Pseudozufall bestimmte) Pause wird eingehalten.
10. Das Verfahren wird an Position 4 fortgesetzt, bis die maximale Anzahl an Durchläufen erreicht ist.
11. Ist die maximale Anzahl an Durchläufen erreicht, werden die gesammelten Werte ausgegeben.

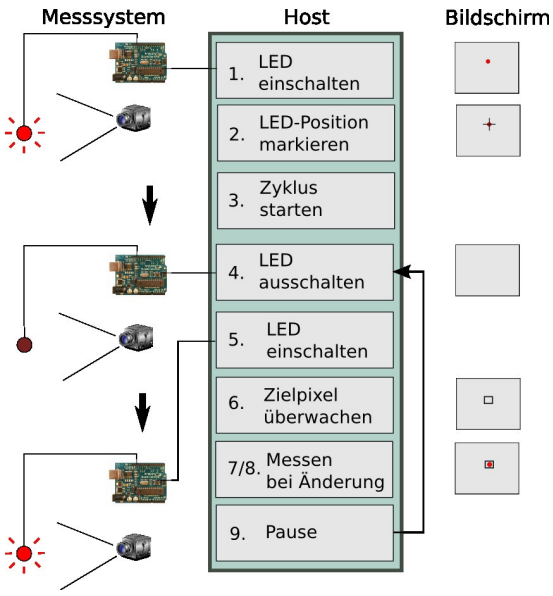
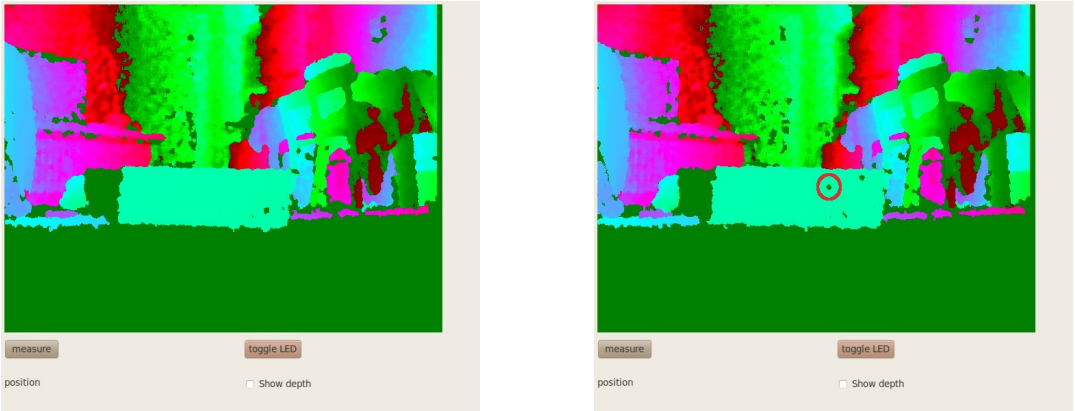


Abbildung 6.5: Programmablauf LpLatencyMeasure

Abbildung 6.6 zeigt die Programmfenster bei der Latenzmessung des ASUS-Systems. Die durch die Infrarot-LED auftretende Störung ist in 6.6b durch den roten Kreis markiert. Die zusätzlich im Vergleich der beiden Bilder erkennbaren Störungen stehen in keinem Zusammenhang mit der Aktivierung mit der Infrarot-LED.



(a) Screenshot mit ausgeschalteter LED

(b) Screenshot mit eingeschalteter LED

Abbildung 6.6: LpLatencyMeasure Screenshots

6.1.5 Bestimmung der Grundlatenzen

Um die durch den Messaufbau auftretenden Latenzen abschätzen zu können, wurden zwei Untersuchungen durchgeführt. Zum einen wurde die Latenz der Kommunikation mit dem Arduino-Board gemessen, zum anderen ein Aufbau mit einer Kamera mit einer hohen Bildrate durchgeführt.

Die in den folgenden Abbildungen verwendete Standardabweichung wurde mit $S := \sqrt{\frac{1}{n-1}\sum_{i=0}^{n}(Xi - \bar{X})^2}$ genähert.

Latenz der seriellen Schnittstelle

Zur Ermittelung der Latenz auf der seriellen Schnittstelle wurde ein Testprogramm[11] verwendet, das Daten über die serielle Schnittstelle an den Arduino schickt und diese wieder empfängt. Im konkreten Fall dient ein Buchstabe zum Ein- und Ausschalten der Infrarot-LED. Dieser Buchstabe wird nach der De- bzw. Aktivierung der LED zurückübertragen. Der Startzeitpunkt der Messung ist vor dem Abschicken an die serielle Schnittstelle, der Endzeitpunkt liegt nach dem Empfang. So ist es möglich, die Latenz zwischen Schreiben und Lesen zu ermitteln. Die Werte finden sich in Abbildung 6.7. Dabei zeigt sich, dass die Abweichungen vom Mittelwert gering sind und die Latenz maximal 5.05 Millisekunden beträgt. Eine Aufteilung, welche Latenz jeweils durch die Übertragung zum und vom Arduino entsteht, ist bei dieser Messung nicht möglich. Insofern wird die auftretende Gesamtlatenz als Latenz des Aufbaus betrachtet.

Die Schaltzeit der eingesetzten IR-Leuchtdioden ist nicht bekannt: bei baugleichen Typen liegt sie unter einer Mikrosekunde, ist also im Vergleich zur oben gemessenen Latenz vernachlässigbar.

Latenz des Gesamtaufbaus

Zur Ermittlung der Latenz des gesamten Messaufbaus kam eine Playstation3 Eye Kamera zum Einsatz, die über Video4Linux mit einer Bildrate von 125 Bildern pro Sekunde angesteuert wurde. Eine grafische Darstellung findet sich in Abbildung 6.8. Der dabei ermittelte minimale Wert zum Auslesen der LED per Kamera liegt bei 4 Millisekunden. Durch die Bildrate von 125 Bildern pro Sekunde entsteht eine theoretische Schwankung von 8 ms. Die ermittelten Werte müssten also zwischen 4 und 12 Millisekunden liegen, tatsächlich liegen sie zwischen 4 und 15 Millisekunden.

[11] modifizierte Version von http://neophob.com/2011/04/serial-latency-teensy-vs-arduino/

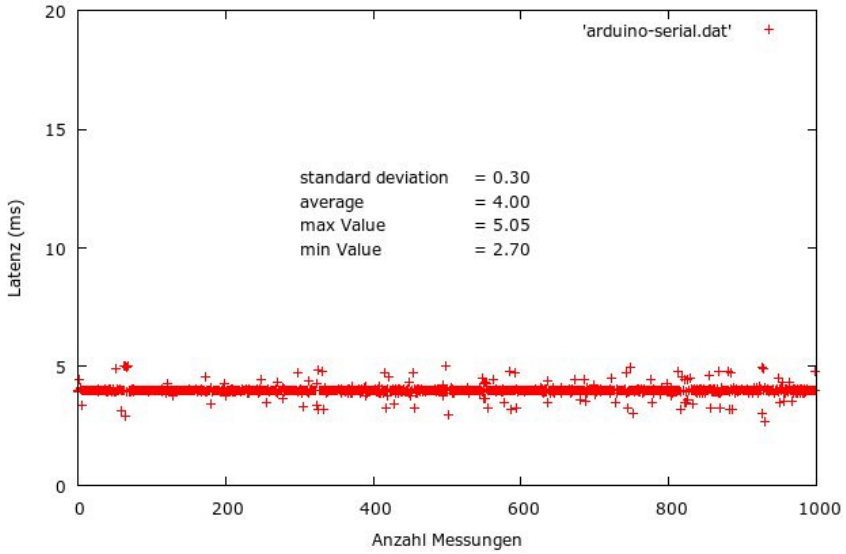


Abbildung 6.7: Latenzmessung serielle Schnittstelle Arduino

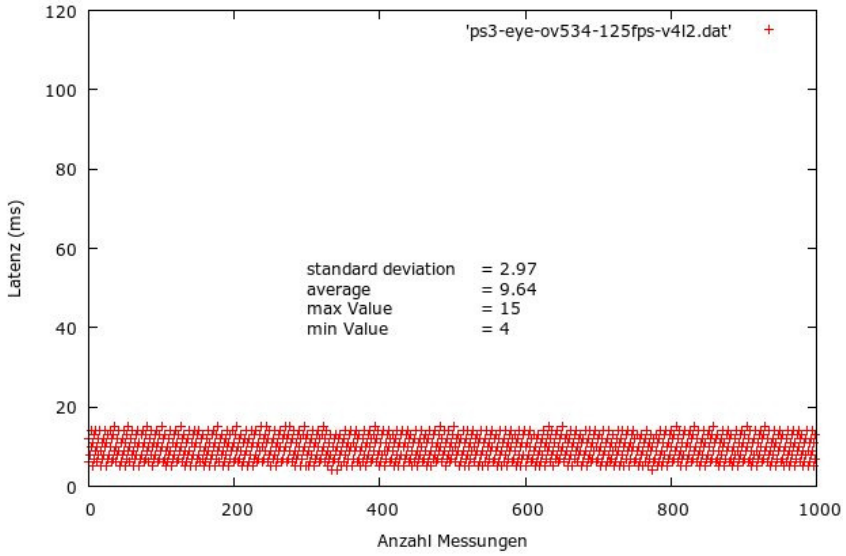


Abbildung 6.8: Latenzmessung PS3Eye mit V4L2 bei 125 fps

6.2 Ermittelte Werte

Für die folgenden Kamerasysteme wurden die angegebenen Werte ermittelt. Der Messzyklus wurde dabei 500-1000 mal durchlaufen. Der Abstand der Infrarot-LED zu den einzelnen Kameras betrug jeweils 60cm.

6.2.1 SR4000

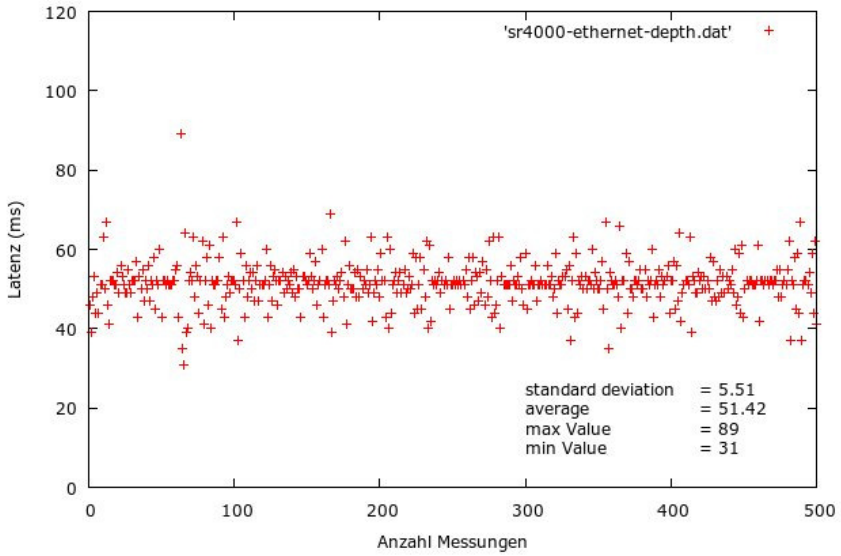


Abbildung 6.9: Latenzmessung SR4000

Die Messung mit der SR4000 (siehe Abbildung 6.9) erfolgte bei einer Bildrate von 54 fps. Die Übertragung erfolgte per Ethernet (100 MBit/s). Um Störungen durch andere Geräte auszuschließen, wurde die Kamera direkt mit einer freien Netzwerkschnittstelle verbunden. Der verwendete Treiber ist libMesaSR (siehe 8.3.2).

6.2.2 PrimeSensor und ASUS Xtion Pro

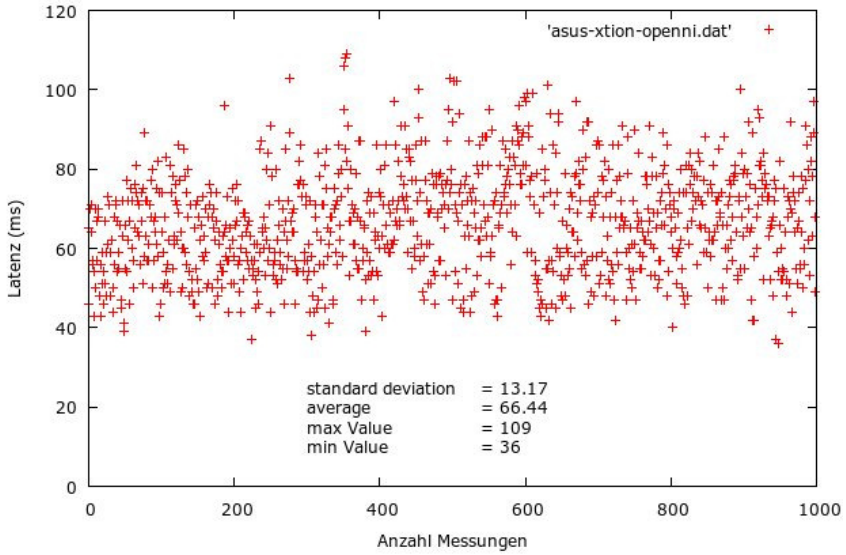


Abbildung 6.10: Latenzmessung ASUS Xtion Pro mit OpenNI

Die Messung (siehe Abbildung 6.10) mit dem ASUS Xtion Pro System erfolgte bei einer Bildrate von 30 fps und einer Auflösung von 640x480 (VGA). Der Versuch, die Kamera mit 60 fps bei einer Auflösung von 320x240 (QVGA) zu betreiben, scheiterte, da die Infrarot-LED in diesem Modus nicht sichtbar war. Bei 30 fps und derselben Auflösung tritt dieses Problem nicht auf. Der Grund hierfür ist unbekannt. Anscheinend wird bei einer Bildrate von 60 fps ein anderer Filter verwendet, der die Störung durch die Infrarot-LED entfernt.Tendenziell dürfte der Modus mit 60 fps allerdings zu einer verringerten Latenz führen, da die Aufnahmezeit pro Bild um etwa 16 Millisekunden zurück geht. Als Treiber kam OpenNI (siehe 8.3.4) zum Einsatz.

6.2.3 Kinect 3D

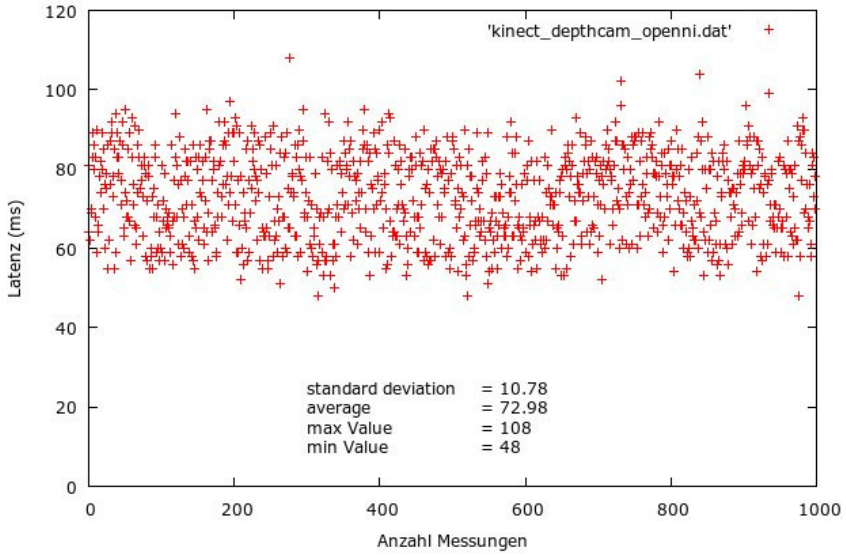


Abbildung 6.11: Latenzmessung Kinect 3D mit OpenNI

Die Messung des Kinect Tiefensensors (siehe Abbildung 6.11) erfolgte mit dem Treiber von OpenNI (siehe 8.3.4). Es wurde eine Bildrate von 30 fps bei einer Auflösung von 640x480 verwendet.

6.2.4 Axis Webcam

Die Axis P1344 Webcam ist eine Kamera, die über Ethernet einen MPEG4-kodierten Videostrom zur Verfügung stellt. Die Anbindung an LPLatencyMeasure erfolgte über das Gstreamer-Framework. Auch wenn die Axis Kamera keine Tiefeninformationen liefert, ist sie für die gemeinsame Verwendung mit anderen Kamerasystemen durchaus sinnvoll (siehe 7.6). Die gemessenen Werte sind in Abbildung 6.12 dargestellt. Die Kamera wurde mit einer Bildrate von 30 fps bei einer Auflösung von 1280x720 (720p) betrieben.

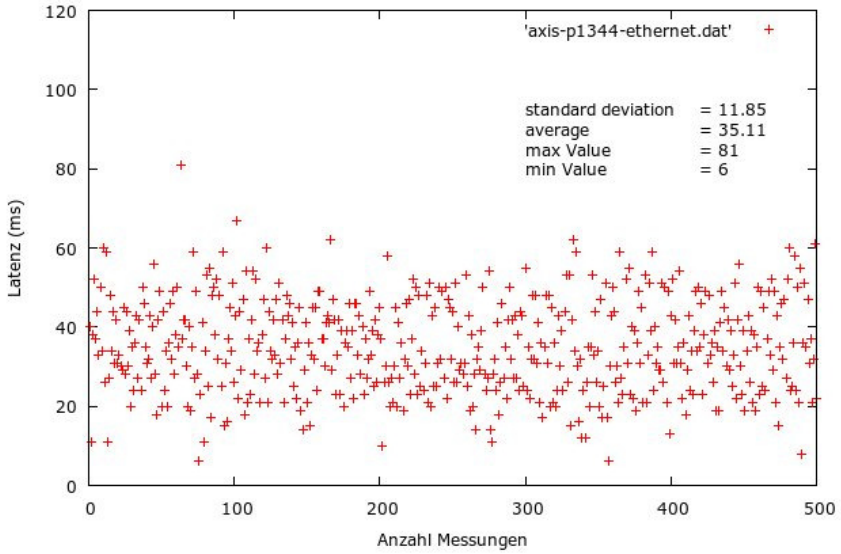


Abbildung 6.12: Latenzmessung Axis P1344

6.3 Diskussion und Fazit

Anhand der Voruntersuchung zur Latenz des Aufbaus ist davon auszugehen, dass die minimale Latenz des Aufbaus bei unter 4 Millisekunden liegt. Der höchste gemessene Wert für die Aktivierung der LED liegt bei 5 Millisekunden, dieser Wert sollte damit als Latenz des Messaufbaus angenommen werden. Die ermittelten Werte für die Kameras sind also maximal um diesen Wert zu hoch. Zur Bestimmung von genauen Latenzzeiten wären diese Werte zu subtrahieren, für eine ungefähre Einschätzung des Zeitraumes ist dies nicht nötig.

Aufgrund der Abweichungen in der Latenz gibt es offensichtlich Unterschiede im ASUS Xtion Pro gegenüber dem Kinect-System.

Die SR4000-Kamera hat gegenüber dem Kinect-System und dem ASUS Xtion Pro eine um 29 bzw 31% verringerte Latenz (15 beziehungsweise 21 msec).

Die gemessenen Werte weichen von den Herstellerangaben ab (siehe Tabelle 5.3). Für die SR4000 liegen sie mit 51 statt 150 Millisekunden deutlich darunter, der PrimeSensor mit 66 gegenüber 40 Millisekunden deutlich darüber. Selbst unter Berücksichtigung der oben erwähnten 5 Millisekunden ist der Wert vom PrimeSensor mit 61 Millisekunden um 53 Prozent erhöht.

Die SR4000 verwendet als Übertragungsmedium Ethernet. Dies scheint prinzipiell nicht schlechter geeignet zu sein als USB, obwohl es mit einer maximalen Datenrate von 100 MBit/s gegenüber 480 MBit/s eine höhere Latenz bei der Datenübertragung aufweist.

Da die SR4000 von den getesteten Systemen die geringste Latenz aufweist, ist sie in dieser Hinsicht am geeignetesten. Um eine Gesamtlatenz von 150 Millisekunden nicht zu überschreiten, bietet sie die größten Reserven. Allerdings bieten auch Zeiten von 66 (Xtion Pro) beziehungsweise 73 (Kinect) Millisekunden eventuell genügend Raum dafür. Der dafür entscheidende Faktor ist die Auswahl des richtigen Verfahrens zur Gestenerkennung.

7 Verfahren zur Gestenerkennung

Dieses Kapitel gibt eine Übersicht über die Grundlagen und Algorithmen zur Erkennung von Gesten, die für den Kontext des Living Place geeignet sind.

7.1 Historisches

Die Benutzung von Gesten zur Kommunikation reicht weit zurück in die Geschichte. Schon bei Plato (428-348 v. Chr.) findet sich ein erster Hinweis auf die Benutzung von Gesten als Kommunikationsmittel von Taubstummen (Bragg, 1997). Das erste Buch mit Abbildungen eines Gestenalphabets, des Internationalen Finger-Alphabets, erschien 1592 (Bragg, 1997).

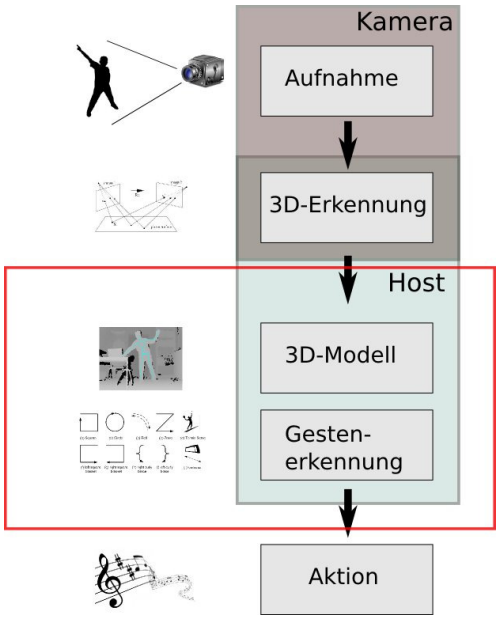


Abbildung 7.1: Übersicht Ablauf Gestenerkennung

Der Beginn der Nutzung von Gesten zur Bedienung eines Rechners reicht (laut Myers (1998), siehe auch Abbildung 7.2) bis in die 1960er Jahre zurück. Das *Sketchpad* von Ivan Sutherland (Sutherland (1964)) war ein erstes System, dass auf Basis eines *Lightpens* arbeitete und für CAD-Anwendungen genutzt werden konnte. Mit einem Lightpen konnte man Bewegungen auf dem Bildschirm ausführen, die dann als Gesten erkannt wurden. So konnten Objekte manipuliert (zum Beispiel vergrößert oder verschoben) werden.

Vergleichbar ist dies mit einer späteren Entwicklung, dem *Grafiktablett*. Dort wird die Position des Stiftes auf einer horizontalen Fläche aufgezeichnet. Beide Techniken sind somit eher als Vorläufer von Touch- und Multitouch-Gesten anzusehen und waren zweidimensional.

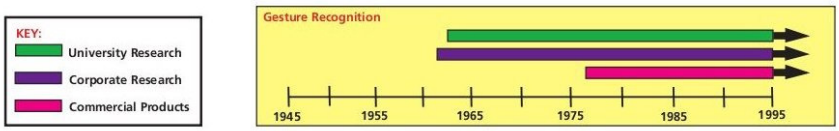


Abbildung 7.2: Entwicklung von User Interfaces nach Myers, Quelle: [Myers (1998)]

Dreidimensionale Gesten verwendet Bolt (1980) für sein *put-that-there* gennantes System Dabei handelt es sich um eine Mischung aus Spracherkennung und Erkennung einer statischen Zeigegeste.

7.2 Klassifizierung von Gesten

> "When you're interacting with the computer, you are not conversing with another person. You are exploring another world."
>
> Rheingold (1991)

Es gibt unterschiedliche Konzepte zur Klassifizierung von Gesten. Eine Klassifizierung ist möglich anhand ihrer Ausführung (siehe 7.2.2) oder aufgrund inhaltlicher Kriterien (siehe 7.2.1).

7.2.1 Klassifizierung nach Inhalt

Ursprünglich aus der Psychologie kommend, klassifiziert Kendo (1988) Gesten anhand des sogenannten *Kendon Continuum*. Dabei wird unterschieden in:

1. **Gestikulation**: Unterstützende Bewegungen der Hände und Arme während des Sprechens.
2. **Sprachähnliche Gesten**: Wie Gestikulation, aber grammatikalisch integriert in die Äußerung.
3. **Pantomime**: Gesten die geeignet sind, ohne Sprache einen Inhalt zu transportieren.
4. **Embleme**: Symbolzeichen (zum Beispiel Lob und Beleidigung).
5. **Zeichensprache**: Ein Satz von Gesten und Posituren[1] die ein vollständiges sprachliches Kommunikationssystem bilden.

[1] Als Posituren werden Körperhaltungen bezeichnet, die eine Symbolik zum Ausdruck bringen.

Darauf aufbauend unterteilt McNeill (1992) Gesten in die Klassen *deiktische Gesten* (Zeigegesten), *ikonische Gesten* (bildhafte Beschreibung eines real existierenden Objektes, einer Handlung), *metaphorisch* (bildhaft, aber abstrakt, beispielsweise verwendet beim Wechsel zwischen Themen während einer Rede) und *Beats*, die das gesprochene Wort rhythmisch unterstreichen. *Beats* sind nur für die zwischenmenschliche Interaktion relevant, da sie Sprache voraussetzen.

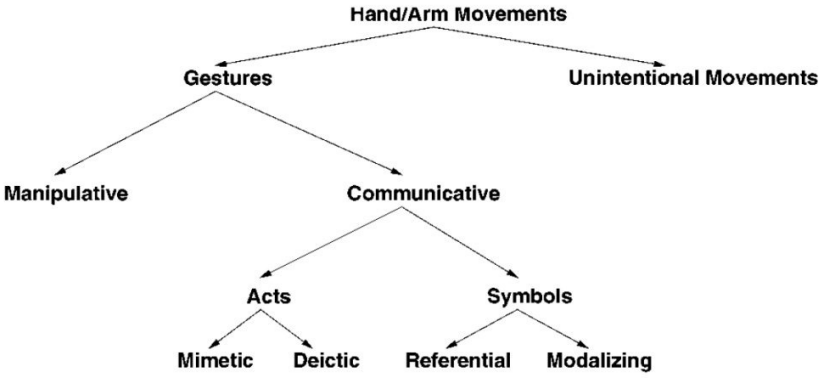


Abbildung 7.3: Klassifizierung von Hand- und Arm-Bewegungen, Quelle: [Pavlovic u. a. (1997)]

Im Sinne der Human-Computer-Interaction besser geeignet, da aus dem Kontext der Informatik stammend, ist der Ansatz von Pavlovic u. a. (1997). Dieser ist in Abbildung 7.3 dargestellt. Dabei werden die Hand/Arm-Bewegungen in *Gesten* und *unabsichtliche Bewegungen* unterteilt. Gesten wiederum in *manipulative* und *kommunikative*. Manipulative Gesten werden zur Handhabung von Objekten verwendet, kommunikative Gesten sind Bestandteil einer Kommunikation. In zwischenmenschlicher Kommunikation werden sie normalerweise durch Sprache begleitet. Diese Gesten werden weiter in *Handlungen* (*Acts*) und *Symbole* unterteilt. Handlungen können dabei entweder *mimisch*, also etwa eine Bewegung nachahmend oder *deiktisch*, also zeigend sein. Symbole haben eine sprachliche Bedeutung und werden in *referenzierende* und *modalisierende* unterschieden. Ein Beispiel für eine referenzierende Geste ist die kreisende Bewegung des Zeigefingers als Symbol für ein Rad. Eine modalisierende Geste zeigt eine bestimmte Eigenschaft an: Nach dem verbalen Hinweis auf einen Motorrad eine Geste die anzeigt, dass der Rahmen vibriert. Wu u. Huang (1999) unterteilen Gesten in drei Klassen: *kommunikative*, *manipulative* und *kontrollierende* Gesten. Kontrollierende Gesten bezeichnet dabei die Kontrolle über ein System mit Hilfe eines (virtuellen) Mauszeigers. Manipulative Gesten ermöglichen die Interaktion mit Objekten, kommunikative sind Teil einer verbalen Kommunikation oder ersetzen diese.

7.2.2 Klassifizierung nach Ausführung

Eine weitere Art der Klassifizierung ist die Unterteilung in statische und dynamische Gesten, vergleichbar mit den *Acts* und *Symbols* bei Pavlovic. Diese Klassifizierung bezieht sich alleine auf die Ausführung der Geste und nicht auf ihre Bedeutung. Dies entspricht einer eher technischen Sichtweise auf die Arten der Bewegung, die beobachtet werden können.

Statische Gesten

Statische Gesten werden auch als *Posture* (das deutsche Wort ist *Positur*) oder symbolische Gesten bezeichnet. Dabei wird mit der Hand eine symbolische Geste gebildet, als Beispiel siehe das „ok"-Zeichen in Abbildung 7.10.

Dynamische Gesten

Dynamische Gesten beinhalten eine Bewegung. Dabei kann die Art und Ausführung der Bewegung selber die Geste darstellen, beispielsweise eine kreisförmige Bewegung in der Luft, oder aber Teil einer dynamisch-statischen Geste sein. Ein Beispiel hierfür ist die *rotating*-Geste in Abbildung 7.8.

Dynamisch-statische Gesten

Dynamisch-statische Gesten beinhalten eine Bewegung und das Enden in einer statischen Geste. Beispielsweise könnte die oben genannte kreisförmige Bewegung mit einer Zeigegeste enden.[2] Eine Sonderform ist die gleichzeitige Interaktion mit zwei Händen, wobei die eine Hand eine statische Geste und die andere eine dynamische ausführt. Dies ist beispielsweise in Abbildung 7.8 in der *panning*-Geste zu finden.

7.2.3 Zusammenfassung

Die Art der Klassifizierung hängt vom Blickwinkel ab. Aus technischer Sicht bietet sich eine Beschreibung nach der Ausführung an, mit einer zusätzlichen Unterteilung von statischen Gesten in *symbolisch* und *zeigend*. Für eine inhaltliche Sicht in der HCI scheint Wu u. Huang (1999) (Einteilung in kommunikativ, manipulativ, kontrollierend) eine gute Lösung zu sein, da sie die

[2] Ein praktisches Beispiel wäre ein kreisförmiges Auswahlmenü, bei dem immer ein Symbol in einem Auswahlrahmen erscheint. Durch die Bewegung wird der Kreis gedreht, das anschließende Verharren in einer Zeigegeste auf das Symbol beschließt die Auswahl.

wichtigen Bereiche von Interaktion mit der Maschine abdeckt, ohne sich in Details zu verlieren. Pavlovic u. a. (1997) und McNeill (1992) scheinen für den Kontext der Informatik weniger geeignet.

I dieser Arbeit wird die eben beschriebene Kombination aus technischer Klassifizierung und Wu u. Huang (1999) verwendet.

7.3 Von der Bewegung zur Geste

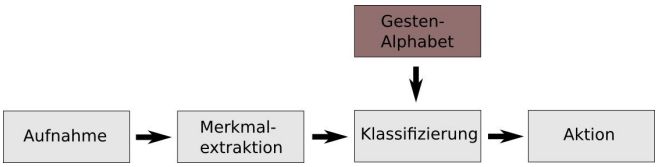


Abbildung 7.4: Ablauf der Erkennung ohne Modell

Um von der Erfassung der Bewegung zur Geste zu kommen, gibt es zwei unterschiedliche Möglichkeiten. Zum einen die Extraktion von Merkmalen zur Klassifizierung von Gesten direkt aus den Kamerabildern, zum anderen die Abbildung auf ein Modell und die anschließende Verwendung von Merkmalen des Modells zur Klassifizierung. Die Abbildungen 7.4 und 7.5 zeigt die jeweiligen Wege.

Bei der Erkennung ohne 3D-Modell erfolgt eine Ermittlung der Merkmale aus den Rohdaten der Kamerasysteme, in der Regel also einem Graustufenbild. Die Anwendung der Algorithmen erfolgt auf diese Daten.

Bei einem modellbasierten Verfahren werden zunächst die Merkmale extrahiert, die für das 3D-Modell notwendig sind. Nach der Berechnung der aktuellen Position des Modells werden wiederum Merkmale extrahiert, die zur Gestenerkennung verwendet werden.

Der Vorteil der modellbasierten Verfahren ist eine größere Sicherheit in der Erkennung, der Nachteil die steigende Latenz durch die Generierung des Modells. Die Gestenerkennung bei modellbasierten Verfahren ist einfacher, da die Menge der markanten Merkmale reduziert ist.

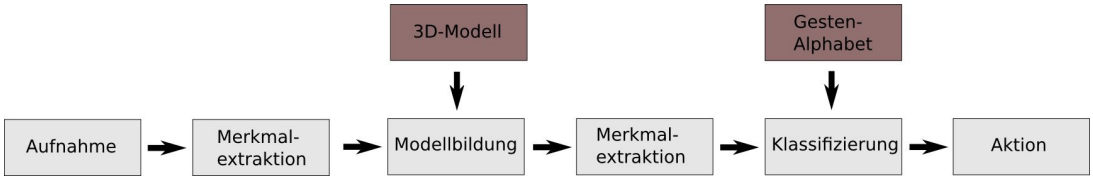


Abbildung 7.5: Ablauf der Erkennung bei modellbasiertem Vorgehen

Extraktion von Merkmalen

Merkmalextraktion (*feature extraction*) ist der Prozess des Extrahierens der markanten Merkmale einer Geste für ein bestimmtes Verfahren. Ein Merkmal kann dabei sehr unterschiedlich sein, zum Beispiel die Position einer Bewegung, die Richtung, die Beschleunigung oder die Farbe an einem bestimmten Punkt. Welches Merkmal benötigt wird, hängt vom verwendeten Algorithmus für die Klassifizierung ab. Welche Merkmale verfügbar sind, hängt unter anderem von der verwendeten Technik und dem Gestenalphabet ab.

7.4 Gestenalphabet

7.4.1 Allgemeines

Die Art des Gestenalphabets ist von mehreren Faktoren abhängig. Zum einen von der grundsätzlichen Frage, ob es sich um dynamische, statische oder eine Mischung dieser beiden Gestentypen handelt. Zum anderen von der Frage, welche Extremitäten zur Klassifizierung genutzt werden, also beispielsweise Hand, Finger, Arm oder einer Kombination.

Die folgenden vier Aspekte von Gesten sind wichtig sind für ihre Bedeutung (Hummels u. Stappers (1998)):

- *Spatial information*: Wo die Geste auftritt, Position auf die sie hinweist,
- *Pathic information*: Der Pfad, den eine Geste nimmt,
- *Symbolic information*: Symbolische Bedeutung der Geste (V für Victory, Daumen über Zeigefinger und Mittelfinger reiben für Geld zählen),
- *Affective information*: der emotionale Teil einer Geste, die Bedeutung für den Benutzer.

7.4.2 Beispiele

Im Folgenden sind einige Beispiele für Gestenalphabete dargestellt:

Unistroke

Castellucci u. Mackenzie verwenden in ihrer Arbeit dynamische Unistroke-Gesten (siehe Abbildung 7.6), die mit Hilfe einer Wiimote-Steuerung in den Raum gezeichnet werden. Unistroke-Gesten wurden 1993 erstmals von Goldberg u. Richardson (1993) zur Benutzung mit einem Lightpen vorgeschlagen. Unistroke bedeutet, dass eine Linie in einer durchgehenden Bewegung gezogen wird. Durch die Verwendung einer Wiimote Steuerung konnten Castellucci u. Mackenzie den Start und Endzeitpunkt durch das Halten des A-Knopfes an der Steuerung bestimmen.[3]

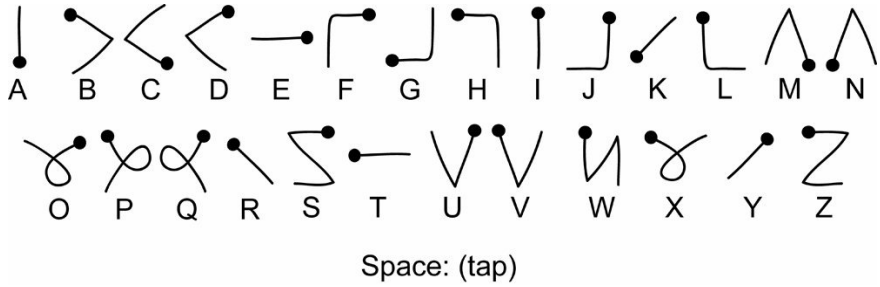


Abbildung 7.6: Unistroke gesten, Quelle: Castellucci u. Mackenzie

Posture

Die Verwendung von symbolischen Zeichenalphabeten hat den Vorteil, dass sie auch mit zweidimensionalen Systemen recht zuverlässig funktionieren, vorausgesetzt, die Umgebung ist kontrolliert und weist einen Kontrast zum Körper des Bedienenden auf.[4] Deshalb wurden solche Systeme recht früh entwickelt, beispielsweise verwendeten Tamura u. Kawasaki (1988) schon 1988 und Murakami u. Taguchi (1991) 1991 Handzeichen der japanischen Zeichensprache. Auch heutzutage wird dieses Alphabet weiterhin in der HCI verwendet, Athitsos u. a. (2010) entwickelten 2010 ein datenbankgestütztes System zur Erkennung der American Sign Language.[5] Bei Postures handelt es sich per Definition um statische Gesten. Sie sind beispielhaft in Abbildung 7.7 dargestellt.

[3] Weiteres zur Start-Stop-Problematik siehe 7.4.6.

[4] Die Segmentierung des Vordergrundes muss möglich sein. Dies wird erreicht durch die Vermeidung störender Farben im Hintergrund und kontrollierte Lichtbedingungen.

[5] Die verwendete Sprache benutzt auch dynamische Elemente, allerdings war ein Teil der Arbeit die Erkennung von Postures.

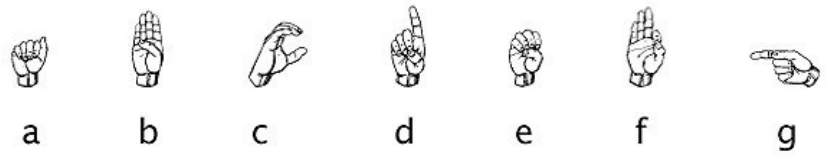


Abbildung 7.7: Beispiel für Posture-Gesten, Quelle: Van den Bergh u. Van Gool (2011)

beidhändiges Gestenalphabet

Van den Bergh u. Van Gool (2011) verwenden für ihre Arbeit Gesten mit beiden Händen. Die Hände bleiben dabei in einer Ebene, die dritte Dimension wird nicht verwendet. Das verwendete Alphabet ist eine Mischung aus statischen und dynamischen Gesten, eine Hand führt eine Posture aus, während die andere sich bewegt. Oder aber beide Hände bewegen sich. Dieses Alphabet ist in Abbildung 7.8 dargestellt.

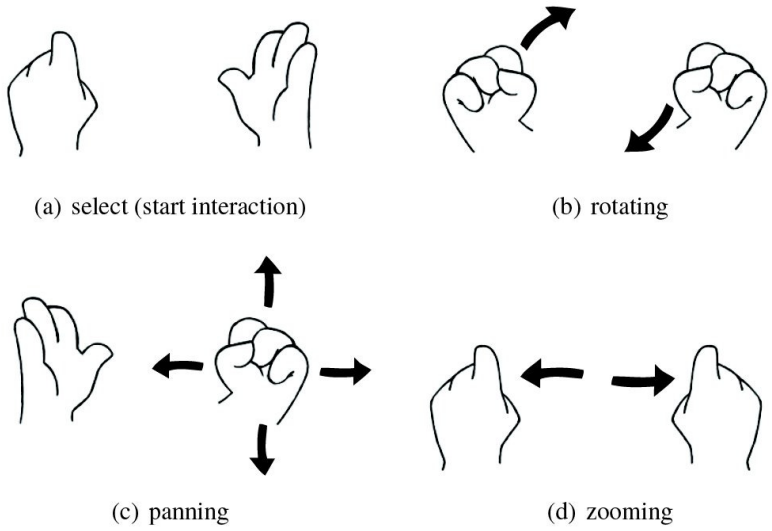


Abbildung 7.8: Beispiel für beidhändige Gesten, Quelle: Van den Bergh u. Van Gool (2011)

3D-Gestenalphabet

Ein Beispiel für ein wirkliches 3D-Alphabet, also unter Einbeziehung aller drei Dimensionen, stellen Hoffman u. a. (2010) vor (siehe Abbildung 7.9). Zur Bestimmung der Position kommt wiederum ein Wiimote Controller zum Einsatz.

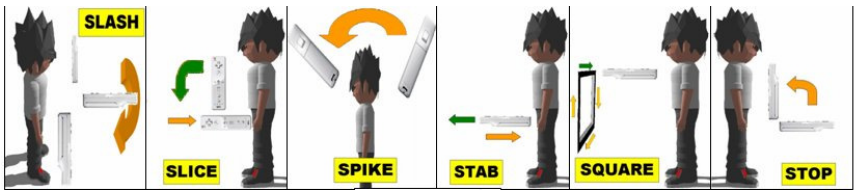


Abbildung 7.9: Beispiel für 3D Gesten, Quelle: Hoffman u. a. (2010)

7.4.3 Unterscheidbarkeit

Die Auswahl des Alphabetes ist für die Erkennungsrate von Bedeutung, es sollten möglichst Gesten verwendet werden, die durch die Algorithmen leicht voneinander unterschieden werden können. Rubine (1991) untersuchte die Erkennungsrate für verschiedene Alphabete basierend auf seinem Algorithmus. Für Zahlen und Buchstaben kam er auf eine Erfolgrate von 98,4-99,5 Prozent, bei der Verwendung sehr einfacher Gesten[6] von 100 Prozent. Je einfacher die Gesten voneinander zu unterscheiden sind, desto bessere Erkennungsraten sind zu erwarten.

7.4.4 Anwendungsgebiet

Auch das Anwendungsgebiet hat Einfluss auf die Auswahl des Alphabets. Beispielsweise würde die Eingabe von Text ein vollständiges Buchstabenalphabet voraussetzen, die Steuerung eines primitiven Musikspielers allerdings nur fünf Gesten.[7] Neben der Anzahl der Gesten ist auch ihre Art unterschiedlich, bei einem Boxspiel etwa virtuelles Boxen als Gestik hingegen bei einem Vortrag eine *Wischgeste* zum Weiterschalten der Folien.

7.4.5 Kulturelle Unterschiede

Gesten können in unterschiedlichen Kulturkreisen eine unterschiedliche Bedeutung haben. Selbst in verschiedenen Ländern des eigentlich gleichen *Kulturkreises* (im folgenden Beispiel, des *westlichen*) haben sie eine unterschiedliche Bedeutung. Beispielsweise steht ein aus Zeige- und Mittelfinger gebildetes “V” in den USA für Victory (Sieg).[8] Die Handfläche zeigt dabei nach aussen. Die gleiche Geste mit umgedrehter Handfläche zeigt in Großbritannien an, dass man sein Gegenüber verachtet (siehe

Abbildung 7.10: Okay Zeichen, Quelle User:Steevven1 (2007)

[6] Strich runter für runter, Strich hoch für hoch als Beispiele
[7] für die Komandos: lauter, leiser, lied vor, lied zurück, start/stop
[8] http://www.encyclopedia.com/doc/1O999-vsign.html

Cassell (1998)). Verwendet man eine dieser Gesten in einer deutschen Kneipe, so wird man daraufhin wohl zwei Bier erhalten, zumindest bei ersterer Geste wohl auch an einer Theke in den USA oder Großbritannien.

Ein anderes Beispiel ist das aus Daumen und Zeigefinger gebildete “o” (Abbildung 7.10. In den USA steht dies für “okay”, in Frankreich oder Italien ist es eine obszöne Geste.(Cassell (1998)).

Aus dem ersten Beispiel ist ersichtlich, dass symbolische Gesten nicht nur vom jeweiligen Land, sondern auch von der Situation abhängig sind, in der sie verwendet werden.

7.4.6 Start-Stop Problematik

Die Start-Stop-Problematik, auch *Midas touch problem* genannt (Jacob (1990)), beschreibt ein grundsätzliches Problem der Erkennung dynamischer Gesten: Wann beginnt die Geste, und wann hört sie auf? Der Name *Midas touch* bezieht sich auf König Midas aus der griechischen Mythologie. Danach soll Midas Silenos[9] gefangen haben und für dessen Freilassung von Dionysos,[10] einem alten Schüler von Silenos, gefordert haben, dass alles, was er fortan berühre, zu Gold werden solle. Dies führte dazu, dass Midas Hunger und Durst litt, da er seine Gabe nicht kontrollieren konnte.

Abbildung 7.11: König Midas verwandelt seine Tochter versehentlich in Gold, Quelle Library of Congress (2009)

Für die Gestenerkennung muss man also eine Möglichkeit finden, dem erkennenden System den Beginn oder das Ende einer Geste mitzuteilen. Möglichkeiten hierfür sind beispielsweise die Einleitung und Beendigung mit einer bestimmten Geste, dem Hinzuziehen einer weiteren Modalität (wie Sprache) oder dem Verbinden mit dem Greifen eines bestimmten Gegenstands. Auch die Veränderung der Entfernung zur Kamera kann als Einleitung und Beendigung der Geste dienen.[11]

7.4.7 Kein Standard

Es gibt bislang keinen Standard für ein allgemeines Gestenalphabet für dreidimensionale Gesten. Sollte ein System eine weite Verbreitung finden, so wird es wahrscheinlich eine ähnliche Entwicklung geben wie bei zweidimensionalen Gesten auf Touchscreens, bei der ein Hersteller

[9] den Sohn der Göttin Gaia.

[10] dem griechischen Gott des Weines, der Freude, der Trauben, der Fruchtbarkeit und der Ekstase.

[11] Die Hand wird 10cm Richtung Kamera geschoben, dann wird die Geste ausgeführt, und die Entfernung wieder vergrößert.

(Apple mit dem iPhone) ein Alphabet vorgibt und andere (wie etwa Google mit Android) dieses übernehmen.

7.5 Verfahren zur Gestenerkennung

Die folgenden Verfahren kommen in der dreidimensionalen Gestenerkennung zum Einsatz. Die erste Frage, die sich vor der Auswahl eines geeigneten Algorithmus stellt ist die, mit welchen Daten gearbeitet werden soll, als mit den originalen Kameradaten oder einem Modell (siehe Abschnitt 7.3).

Da die Auflösung der zur Verfügung stehenden Kameras nicht ausreicht, um die Erkennung einzelner Finger zu leisten, wird angenommen, dass die Bewegung der Hände als Gestik interpretiert wird. Für eine Erkennung von Fingern, etwa zur Einbeziehung statischer Gesten der Hand muss eine zusätzliche Kamera verwendet werden. Ein Beispiel hierfür findet sich in Abschnitt 7.6.

Das Gestenalphabet ist ein weiteres Kriterium zur Auswahl der verwendbaren Algorithmen. Spielt die Entfernung zur Kamera als dritte Dimension keine Rolle (beispielsweise bei Unistroke-Gesten, siehe Abbildung 7.6), so lassen sich Algorithmen im zweidimensionalen Raum verwenden.

7.5.1 Dynamic Time Warping

Historie

Dynamic Time Warping (DTW) ist ein Verfahren, dass ursprünglich für die Erkennung von Sprache verwendet wurde (beispielsweise Sakoe (1978)). Seit 1993 wird es für die visuelle Gestenerkennung eingesetzt (Darrell u. Pentland (1993)). Klassisches DTW hat einen Berechnungsaufwand von $O(N^2)$ (Salvadore u. Chan (2004)).[12]

Eine Beschleunigung von DTW findet sich unter dem Namen FastDTW bei Salvadore u. Chan (2004). Eine Beispielimplementierung für die Programmiersprache Java ist unter der MIT-Lizenz verfügbar.[13] FastDTW verringert den Berechnungsaufwand von DTW auf $O(N)$.

[12] N ist dabei die Anzahl der zu betrachtenden Merkmale.
[13] siehe http://code.google.com/p/fastdtw/

Wöllmer u. a. (2009) erweitern den ursprünglich auf zwei Dimensionen beschränkten DTW-Algorithmus auf eine beliebige Anzahl an Dimensionen (MD-DTW), indem sie die Abstandsfunktion auf die Berechnung über mehrere Dimensionen hinweg erweitern. Da die zusätzlichen Berechnungen linear abhängig von der Anzahl der Dimensionen sind, bleibt der Aufwand grundsätzlich bei $O(N^2)$.

Grundsätzliches Verfahren

DTW wird benutzt, um Signale zu synchronisieren. Dies ist in der Gestenerkennung beispielsweise der Fall, wenn vorliegende Daten mit einer Reihe von Gesten-Templates verglichen werden sollen. Dazu werden immer zwei Signale gleichzeitig betrachtet, in diesem Beispiel also die aufgenommene Sequenz an Bewegungen und die einzelnen Templates der Gesten. DTW berechnet nun den Abstand zwischen allen möglichen Paaren von Punkten der beiden Signale (Werte der Merkmale, siehe Abbildung 7.12). Daraus wird eine kumulative Distanzmatrix berechnet und der Pfad mit der geringsten Entfernung ermittelt. Dieser Pfad wird *ideal Warp* genannt und ist die Synchronisation mit den geringsten Abstandswerten der Merkmale. Durch Vergleich der kumulativen Kosten der Pfade mit verschiedenen Templates kann das Gesten-Template mit den geringsten Pfadkosten (also der am besten passenden Geste) ermittelt werden (Wöllmer u. a. (2009)). Je nach Variante (DTW beziehungsweise MD-DTW) ist dieses Verfahren für die Erkennung von zwei- wie auch dreidimensionalen Gesten geeignet.

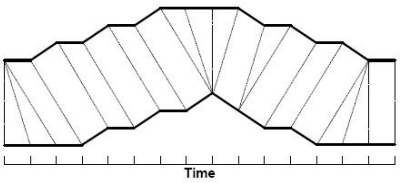


Abbildung 7.12: Abstand zweier Kurven bei DTW, Quelle Fang (2009)

7.5.2 Hidden Markov Model

Historie

Hidden Markov-Modelle (HMM) stammen, wie auch DTW, ursprünglich aus der Spracherkennung. Die ersten Arbeiten zu diesem Thema wurden bereits ab Mitte der sechziger Jahre von Leonard Baum veröffentlicht (Baum u. Petrie (1966)). Es sind zahlreiche Abwandlungen des Modells bekannt.

Grundsätzliches Verfahren

Abbildung 7.14 zeigt ein typisches Hidden Markov-Model. Dies ist in zwei Stufen gegliedert. Die erste Stufe entspricht einer Markov-Kette, bestehend aus Zuständen (X) und den zugehörigen Übergangswahrscheinlichkeiten (a) zwischen den Zuständen. Die Zustände der ersten Kette sind von außen nicht sichtbar (*hidden*). Die zweite Stufe erzeugt die Ausgangssymbole (y), der Übergang von der ersten Stufe zur zweiten erfolgt dabei gemäß einer Wahrscheinlichkeitsverteilung (b).

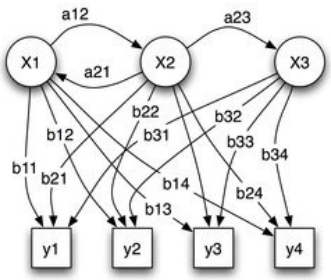


Abbildung 7.13: Hidden Markov Model, Quelle Fang (2009)

Je nach Auswahl der verwendeten Merkmale ist dieses Verfahren sowohl für zwei- als auch für dreidimensionale Gesten geeignet.

HMM beschreibt nur das Modell; für die Umsetzung gibt es unterschiedliche Algorithmen, die im Folgenden kurz vorgestellt werden.

Viterbi-Algorithmus

Der bei Viterbi (1967) beschriebene Algorithmus findet die wahrscheinlichste Sequenz von versteckten Zuständen (bei HMM die mit X bezeichnete Markov-Kette) anhand der beobachteten Werte. Diese Sequenz wird als Viterbi-Pfad bezeichnet. Zur Berechnung werden Techniken der dynamischen Programmierung benutzt. Die Berechnungskomplexität des Algorithmus ist $O(N^2)$, eine Variante mit $O(N\sqrt{N})$ ist bei Patel (1995) zu finden.

Forward-Backward-Algorithmus

Der Forward-Backward-Algorithmus (auch Baum-Welch) berechnet die A-posteriori-Wahrscheinlichkeit der Markov Kette. Die Berechnungskomplexität dieses Verfahrens ist $O(N^2)$ (siehe Klaas u. a. (2006)).

7.5.3 Rubine-Algorithmus

Historie

Der Rubine Algorithmus wurde 1991 von Dean Rubine (Rubine (1991)) zur Erkennung von zweidimensionalen Gesten mit einem Lichtschwert entwickelt. Eine Erweiterung für den dreidimensionalen Bereich, bei dem der ursprüngliche Algorithmus in allen drei Ebenen (für die unterschiedlichen Achsen im Koordinatensystem) ausgeführt wird, nennt sich Rubin3D und ist im iGesture Framework (siehe 8.4.1) enthalten. Blagojevic u. a. (2010) bieten einen Ansatz für die automatische Auswahl von geeigneten Merkmalen bei Rubine für Fälle, in denen der Algorithmus zu etwas anderem als der ursprünglich geplanten Erkennung von zweidimensionalen Gesten mit einem Stift zum Einsatz kommt.

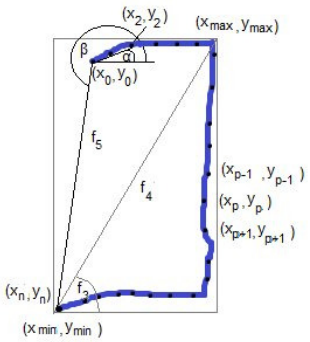


Abbildung 7.14: Rubine Beispiel Features, Quelle Kilan (2011)

Grundsätzliches Verfahren

Rubine verwendet Vektoren von Merkmalen (*Feature Vector*) zur Erkennung der Gesten. Dazu werden in periodischen Abständen aus der aktuellen Position des Lightpens 13 unterschiedliche Merkmale berechnet.[14] Diese 13 Merkmale werden zu einem Vektor zusammengefasst. Die einzelnen Feature-Vektoren der Geste werden nun mit allen Feature-Vektoren aus der Trainingsphase verglichen. Die Geste mit den meisten Übereinstimmungen wird ausgewählt. Die Berechnungskomplexität von Rubine ist $O(2^N)$ (Ferri, 2008, S. 233)

Rubine ist nur bei modellbasiertem Vorgehen sinnvoll, da allein hier einige wenige Punkte im Skelettmodell die Funktion eines *Stiftes* übernehmen können und die Berechnung durch die exponentielle Berechnungskomplexität im Rahmen bleibt.

7.5.4 Dynamic Bayesian Network

Ein mögliches Verfahren zur Berechnung eines Skelettmodells aus den Daten einer Time-of-Flight-Kamera zeigten Ganapathi u. a. (2010). Dort wird ein *Dynamic Bayesian Network* (DBN) verwendet, um aus den Punkten (*Punktwolke*) einer Time-of-Flight-Kamera ein Skelettmodell zu errechnen.

[14] Beispiele für Featurevektoren sind die Entfernung zum Startpunkt, die Dauer, die Geschwindigkeit, verschiedene Winkel und deren Verhältnis zueinander.

Die Berechnung erfolgt dabei auf einer Grafikkarte (GPU). Insgesamt erzielt das System eine Latenz von 100 bis 250 Millisekunden, um die Winkel des 48 Freiheitsgrade umfassenden menschlichen Körpermodells (siehe Abbildung 7.15) zu berechnen. Der dazu eingesetzte Algorithmus AGEX hat eine Berechnungskomplexität von $O(NIog(N))$ (Plagemann u. a. (2010)).

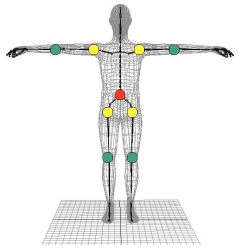


Abbildung 7.15: 3D Modell für DBN, Quelle Ganapathi u. a. (2010)

7.5.5 Weitere Verfahren

Geometric Template Matcher

Wobbrock u. a. (2007) schlagen als Verfahren zur Erkennung von zweidimensionalen Gesten die Verwendung eines *Geometric Template Matchers* vor, der ohne Training fest programmierte grafische Objekte den Gesten zuordnet. Dieses Verfahren verwendet kein 3D-Modell.

Partikelfilter

Partikelfilter oder auch sequenzielle Monte-Carlo-Filter (SMC-Filter) sind ein Verfahren zur Zustandsschätzung von dynamischen Modellen. Beispiele für die Nutzung zur Erkennung von Gesten der Hand finden sich bei Guðmundsson u. a. (2010).

Hauterkennung

Hauterkennung ist, neben Verfahren zur Erkennung des Umrisses, eine klassische Methode zur Erkennung der Hände. Grundsätzlich wird dabei versucht, durch Transformation der Farbbilder in einen geeigneten Farbraum eine Erkennung möglichst nur der Hautfarbe zu erhalten.[15] Einige Verfahren sind bereits in Abschnitt 5.2.1 erwähnt; eine genaue Übersicht über die möglichen Verfahren findet sich bei Kakumanu u. a. (2007). Die Hauterkennung wird heutzutage nicht mehr alleine eingesetzt, sondern im Verbund mit anderen Verfahren (siehe Abschnitt 7.6).

[15] Bei einigen Algorithmen wird neben Haut auch Holz oft als hautfarben erkannt

Support Vector Machine

Eine Support Vector Machine (SVM, auf deutsch auch Supportvektormethode) teilt eine Menge von Merkmalen in Klassen ein. Dabei wird der freie Raum um die Klassengrenzen herum maximiert. Eine Klasse ist dabei beispielsweise eine Geste, oder Teil einer Geste. Eine Implementierung unter Verwendung einer SVM zur Erkennung statischer Gesten findet sich bei Liu u. a. (2008).

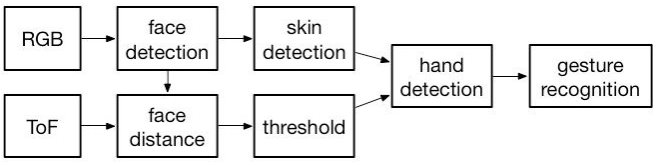
7.6 Kombination von unterschiedlichen Techniken

Als Beispiel für eine Kombination unterschiedlicher Verfahren zur Erkennung von Gesten kann Van den Bergh u. Van Gool (2011) dienen. Diese entwickeln in ihrer Arbeit ein System, das eine Time-of-Flight- mit einer Videokamera kombiniert.

Das Verfahren ist in Abbildung 7.16b zu sehen. Grundsätzlich wird eine Hauterkennung verwendet, um das Gesicht des Benutzers zu identifizieren. Durch eine Überlagerung der Position des Gesichtes mit dem Tiefenbild kann dieses in der Tiefeninformation identifiziert werden. Anschließend erfolgt die Festlegung eines Radius um die Position herum. Dieser Radius wird als Filter für die Hauterkennung verwendet und so die Hände erkannt.

Als Beispielapplikation wurde ein 3D-Modell-Viewer gewählt, das verwendete Gestenalphabet findet sich in 7.8. Laut Van den Bergh u. Van Gool (2011) kommt das System auf eine Erkennungsrate von 99,54%. Für das Verfahren kommt kein 3D-Modell zum Einsatz.

(a) RGB und ToF Kamera Kombination. Quelle: Van den Bergh u. Van Gool (2011)



(b) 3D System mit strukturiertem Licht. Quelle: Van den Bergh u. Van Gool (2011)

Abbildung 7.16: Kombination RGB und ToF

7.7 Fazit

Die Auswahl an Algorithmen ist zu umfassend, um sie hier alle darzustellen. Das Thema Erkennung von Gesten ist seit ungefähr 20 Jahren ein Feld mit zahlreichen Veröffentlichungen, die die unterschiedlichen Algorithmen ausführlich besprechen. Es gibt eine Reihe mit sehr guten Erkennungsraten, je nach Gestenalphabet bis zu 100 Prozent.

Die Reduzierung der Kompexität für die nachfolgende Gestenerkennung ist ein entscheidender Vorteil der Berechnung eines 3D-Modells. Der Preis dafür ist jedoch die Erhöhung der Latenz durch diesen Zwischenschritt, beispielsweise um 100 bis 250 Millisekunden bei der Verwendung von DBN (siehe 4.1.2)[16]. Ob dies tragbar ist, hängt von der sich ergebenden Gesamtlatenz ab. Diese muss im Gesamtsystem gemessen werden; eine abschließende Beurteilung ist nur dann möglich. Ein Vorschlag für weiterführende Untersuchungen findet sich in Abschnitt 9.3.

Die vorgestellten Algorithmen zur Erkennung der Gesten unterscheiden sich in der Komplexität ihrer Berechnung. Ob dies praktisch eine Rolle spielt, hängt sehr vom Umfang der verarbeiteten Daten ab. Mit der Verwendung eines 3D-Skelettmodells sinkt der Aufwand beträchtlich, da nur noch wenige Punkte (oder gar nur einer für eine Hand) einbezogen werden müssen.

Wichtiger als die Festlegung auf einen Algorithmus scheint die Abstimmung des gesamten Ablaufs, vom Kamerasystem über die Gestenerkennung zur Einbindung in das Smart Home zu sein. Dies aufgreifend findet sich in Abschnitt 10.1 ein Vorschlag für ein solches Design.

[16] Durch Vereinfachung des Skelettmodells und die Wahl einer performanteren Grafikkarte sollte sich dieser Wert noch deutlich erniedrigen lassen

8 Technische Schnittstellen

Dieses Kapitel beschäftigt sich mit den technischen Schnittstellen zwischen unterschiedlichen Systemkomponenten zur Gestenerkennung. Dies dient zum besseren Verständnis der technischen Möglichkeiten und Einschränkungen der verschiedenen Schnittstellen auf Hard- und Softwareebene.

Der Fokus liegt hierbei auf der Einbindung der Gestenerkennung in die Infrastruktur des Living Place Hamburg. Der Grund hierfür ist, dass weitere Untersuchungen dort stattfinden werden und eine auf Grundlage des später folgenden Entwurfs entwickelte Software in den Living Place integriert werden soll.

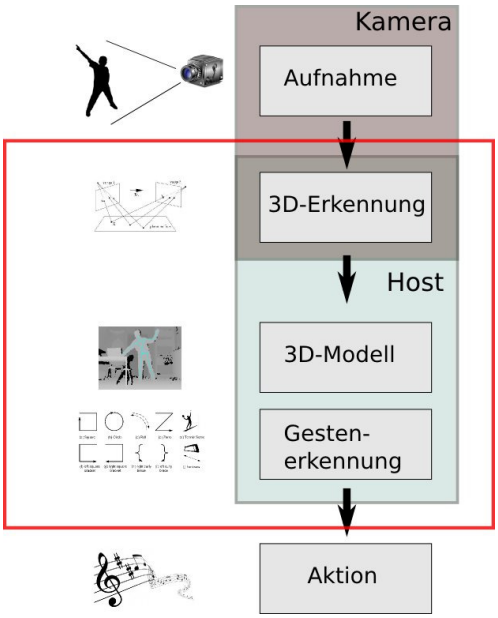


Abbildung 8.1: Übersicht Ablauf Gestenerkennung, Schnittstellen

8.1 Bussysteme

Dieser Abschnitt gibt einen kurzen Überblick über die beiden eingesetzten Bussysteme zur Anbindung von Kameras und Hostsystem.

8.1.1 USB

Universal Serial Bus (USB[1]) ist ein 1996 von der Firma Intel eingeführter Standard zum Anschluss von Peripherie-Geräten an den PC. Der High-Speed genannte Übertragungsmodus im Versionsstandard 2.0 bietet eine Übertragungsrate von 480 MBit/s. Dieser Modus kommt für die Sensoren Kinect und PrimeSensor zum Einsatz.

[1] siehe http://www.usb.org

8.1.2 Ethernet

Ethernet, nach der IEEE-Norm 802.3, ist eine Technologie zur Übertragung von Daten in kabelgebundenen Netzen. Die im Kontext des Living Place verwendeten Datenraten betragen dabei 100 MBit/s (nach dem Standard IEEE 802.3u (1995)) und 1000 MBit/s (nach dem Standard IEEE 802.3ab (1999)). Für den Anschluss der SR4000 ToF-Kamera kommen 100 MBit/s zum Einsatz, für weitere Komponenten[2] 1000 MBit/s.

8.2 Software

Die verfügbare Software gliedert sich in verschiedene Teilbereiche. Zunächst erfolgt eine Übersicht über die Treiber zur Ansteuerung der Kameras. Als Erweiterung hierzu bietet Software wie OpenNI und das Microsoft SDK für das Kinect-System die Koordinaten eines Skelettmodells. Darauf aufsetzend lässt sich eine Erkennung für Gesten durchführen, beispielsweise mit iGesture. Abschließend wird die Software für die Einbindung von Geräten in das Smart Home-Netzwerk betrachtet.

8.3 Treiber

Die folgenden Treiberbibliotheken sind für ToF-Kameras und Kinect/PrimeSensor verfügbar. Die Bibliotheken werden im Userspace ausgeführt.

8.3.1 libfreenect/OpenKinect

Libfreenect vom OpenKinect Projekt[3] ist der erste frei verfügbare Treiber, der für das Kinect-System veröffentlicht wurde. Die dazu nötigen Informationen wurden durch *Reverse Engineering*[4] gewonnen. Die ursprüngliche Version wurde von Héctor Martin entwickelt. Libfreenect bietet Zugriff auf die Tiefeninformationen (als 12-Bit-Graustufenbild mit einer Auflösung von 640x480 Pixeln), ebenso wie auf die Kinect-Videokamera. Die Software läuft unter Windows, MacOS X und Linux[5]. Schnittstellen für C, C++, C#, Java und Python sind verfügbar. Die Software ist derzeit nicht kompatibel zum PrimeSensor (ASUS Xtion Pro).

[2] Zum Anschluss des Hostsystems an das Hausnetz und den Message Broker (siehe 8.5) im Living Place siehe Abschnitt 8.5.1.

[3] http://www.openkinect.org

[4] Als *Reverse Engeneering* wird die Analyse eines bestehenden Systems zur Gewinnung von Informationen über seinen Aufbau bezeichnet.

[5] Ubuntu Pakete sind erhältlich unter https://launchpad.net/~arne-alamut

8.3.2 libMesaSR

LibMesaSR ist die Treiberbibliothek für die SwissRanger Time-of-Flight-Kamera Serie, wie die SR4000. Die Bibliothek ist für Linux und Windows erhältlich und mit den Programmiersprachen C und C++ verwendbar. Die von der Kamera gelieferten Tiefeninformationen stehen als Graustufenbild (14-Bit-Tiefeninformation) in der Auflösung der Kamera (172x144 Pixel) zur Verfügung.

8.3.3 MS SDK

Microsoft hat Mitte 2011 ein Software Development Kit (SDK) für das Kinect-System und die Windows7-Plattform veröffentlicht[6]. Dieses bietet, ähnlich wie die OpenNI-Software, sowohl Zugriff auf die direkt vom System übertragenen Daten, als auch ein Skelettmodell (siehe Abbildung 8.2). Im Unterschied zu OpenNI fällt beim Skelettmodell auf, dass ebenfalls die Ausrichtung und Position der Hände bestimmt wird. Die Positionsbestimmung einzelner Finger ist nicht möglich.

Als Einschränkung gilt, dass das SDK nur auf der Windows7-Plattform und nur mit dem Kinect Sensor funktioniert, das PrimeSensor-System wird nicht unterstützt. Die kommerzielle Nutzung des SDK ist nicht gestattet.

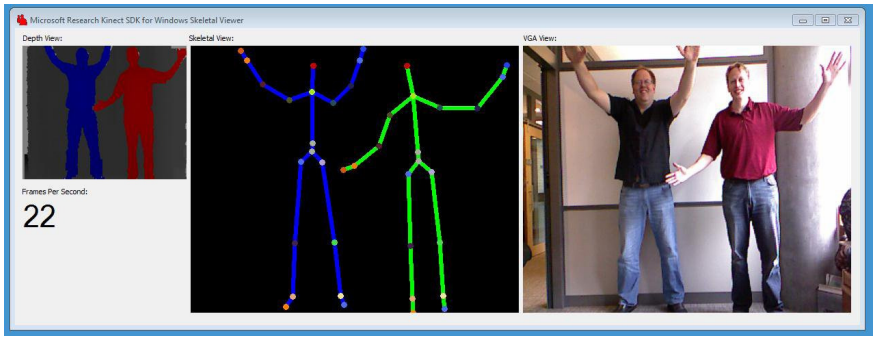


Abbildung 8.2: MS Kinect SDK, Skelettmodell, Quelle: Research (2011)

[6] http://research.microsoft.com/en-us/um/redmond/projects/kinectsdk/

8.3.4 OpenNI

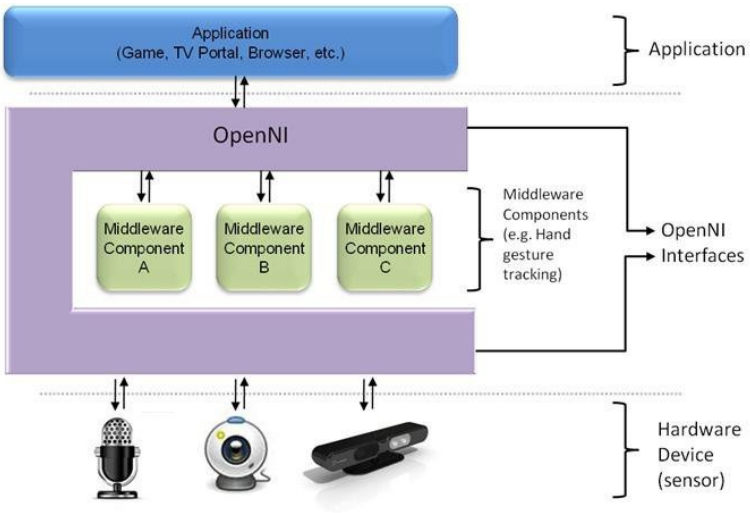


Abbildung 8.3: Architektur OpenNI, Quelle: OpenNI

OpenNI (siehe Abbildung 8.3) ist ein für die Geräte der Firma Primesense entwickeltes Framework. Es beinhaltet sowohl einen Hardwaretreiber für diese Kamerasysteme (PrimeSensor und Kinect), als auch Komponenten zur Modellierung eines Skelettmodells (NITE). Die Kameraschnittstelle ist dabei, genauso wie die API, offengelegt. Das Framework selbst steht unter der GNU LGPL Lizenz und ist damit auch für den kommerziellen Einsatz freigegeben. Die verfügbaren Komponenten für die Skelettmodellierung sind nur als Binärdateien verfügbar und der dabei verwendete Algorithmus ist unbekannt. OpenNI ist für die Plattformen Windows, Linux und MacOS X verfügbar.

Die Verwendung von OpenNI setzt pro Benutzer eine Kalibrierungsgeste (siehe Abbildung 8.5) voraus, erst danach wird das Tracking für diese Person aktiviert. OpenNI bietet neben den Koordinaten der Punkte des Skelettmodells eine primitive Gestenerkennung und Zugriff auf das originale Tiefenbild (in Graustufen). Die unterstützten Gesten finden sich in Tabelle 8.1.

Tabelle 8.1: OpenNI Gesten

Geste	Parameter
Drücken (push)	Beschleunigung (Zoll/Sekunde)
Wischen nach unten (swipe down)	Beschleunigung (Zoll/Sekunde)
Wischen nach links (swipe left)	Beschleunigung (Zoll/Sekunde)
Wischen nach rechts (swipe right)	Beschleunigung (Zoll/Sekunde)
Wischen nach oben (swipe up)	Beschleunigung (Zoll/Sekunde)
Kreis (circle)	Radius (Zoll)
Winken (wave)	-

8.3.5 Zusammenfassung

Tabelle 8.2 fasst die wichtigsten Eigenschaften der für die SR4000, Kinect und PrimeSensor verfügbaren Treiberbibliotheken zusammen:

Tabelle 8.2: Eigenschaften der Treiber

	libMesaSR	libfreenect	openNI	MS Kinect SDK
unterstützte Hardware	SR4000	Kinect	Kinect, PrimeSensor	Kinect
unterstützte Betriebssysteme	Windows, Linux	Windows, Linux, MacOS	Windows, Linux	Windows7
Rohdaten	Ja	Ja	Ja	Ja
Skelettmodell	Nein	Nein	Ja	Ja
Lizenz	proprietär	GPL und Apache	GPL und proprietär	proprietär
Unterstützte Programmiersprachen	C, C++	C, C++, Python, Java	C, C++	C#, C++
Unterstützte Kameras	SR4000	Kinect, PrimeSensor	Kinect, PrimeSensor	Kinect
Eigene Gestenerkennung	Nein	Nein	Ja	Nein

8.4 Frameworks

Die nachfolgenden Frameworks (TUIO, FAAST, VRPN) sind für die Übertragung der Daten im Netzwerk und die Gestenerkennung (iGesture) nutzbar und werden hier kurz vorgestellt.

8.4.1 iGesture

iGesture[7] ist ein in Java implementiertes Framework für die Gestenerkennung (siehe Signer u. a. (2007)). Ursprünglich entwickelt wurde die Software für die Erkennung von zweidimensionalen Gesten auf elektronischem Papier. In den Entwicklungszweig[8] sind die Arbeiten zur

[7] http://www.igesture.org
[8] https://igesture.svn.sourceforge.net/svnroot/igesture/trunk/

Erkennung von dreidimensionalen Gesten bereits eingeflossen. Derzeit ist iGesture auf den Plattformen Windows und BSD-basierten

Systemen wie FreeBSD und MacOS X lauffähig, eine Portierung auf Linux ist in Planung[9].

Durch die in iGesture vorhandene Schnittstelle zu TUIO (siehe Abschnitt 8.4.3) ist eine Verwendung von durch TUIO unterstützte Tracker wie Kinect/PrimeSensor über die OpenNI-Software mögich.

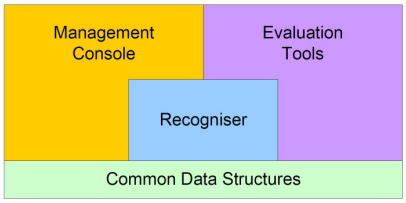


Abbildung 8.4: Architektur iGesture. Signer u. a. (2007)

8.4.2 VRPN

Das *Virtual Reality Peripheral Network* ist ein von Russell M. Taylor II am Department of Computer Science der Universität von North-Carolina entwickeltes Framework zur Verbindung von Geräten für Virtual-Reality-Umgebungen im Netzwerk (siehe Taylor u. a. (2001)).

Eigenschaften

Das *Virtual Reality Peripheral Network* ist ein geräteunabhängiges, netzwerk-transparentes Interface zwischen Applikationen und physikalischen Eingabegeräten (beispielsweise Motiontracker[10]) für Virtual Reality Systeme. VRPN ist als allgemeine Abstraktion von Eingabegeräten in einer Client-Server-Architektur implementiert. Dabei bietet es Zeitsynchronisation zwischen den beteiligten Geräten, genauso wie einen automatischen Wiederaufbau von unterbrochenen Verbindungen.

Laut Taylor u. a. (2001) liegt die durch VRPN verursachte Latenz bei circa 1,7-3,3 Millisekunden für die Übertragung der Nachrichten vom Server zum Client.

Je nach Geräteklasse werden unterschiedliche Parameter wahlweise per UDP oder TCP übertragen, für einen Tracker beispielsweise die Position und Ausrichtung der einzelnen 3D-Objekte, genauso wie die Bewegung und Geschwindigkeit der einzelnen Objekte. Das Hinzufügen neuer Geräteklassen ist einfach möglich.

[9] http://soft.vub.ac.be/soft/edu/mscprojects/2010_2011/gesturedetectionalgorithms
[10] Ein Motiontracker dient zur Erfassung von Bewegungen.

Verwendung

VRPN ist für Virtual-Reality-Umgebungen gedacht, dementsprechend sind Server-Implementierung für Tracker wie den ARTtracker vorhanden. Ebenso gibt es mit FAAST (siehe Abschnitt 8.4.4) einen VRPN-Server für Kinect/PrimeSensor. Interessant für Tests ist die Möglichkeit der Aufzeichnung und Wiedergabe der Kommunikation zwischen Server und Client (*session replay*).

8.4.3 TUIO

TUIO[11] ist ein Framework für *Tangible multitouch surfaces*, das ein Protokoll und eine API definiert (siehe Kaltenbrunner u. a. (2005)). TUIO wurde beispielsweise verwendet, um reacTable zu bauen, ein auf einem Multitouch-Tisch basierendes Musikinstrument (Kaltenbrunner u. Bencina (2007)).

Auch wenn TUIO nicht primär für gestenbasierte Interaktion im dreidimensionalen Raum entwickelt wurde, ist das Protokoll in der Lage, auch dreidimensionale Koordinaten zu übertragen.

Ein einfacher Tracker für das Kinect-Kamerasystem, basierend auf libfreenect und openframeworks, ist verfügbar[12]. Zusammen mit der Verwendung von iGesture (siehe 8.4.1) ist es möglich, ein komplettes System zur Erkennung von Gesten aufzubauen (als Beispiel siehe Kapitel 10.1). Eine Schnittstelle zur OpenNI-Software ist ebenfalls verfügbar.

8.4.4 FAAST

Allgemeines

Flexible Action and Articulated Skeleton (Suma u. a. (2011)) ist ein Programm für das Windows-Betriebssystem, das am Institute of Creative Technologies der *University of Southern California* entwickelt wurde. FAAST verwendet OpenNI für den Zugriff auf die Hardware und nutzt eine Kombination aus eigenen Algorithmen und dem Skelettmodell von OpenNI (Suma u. a. (2011)).

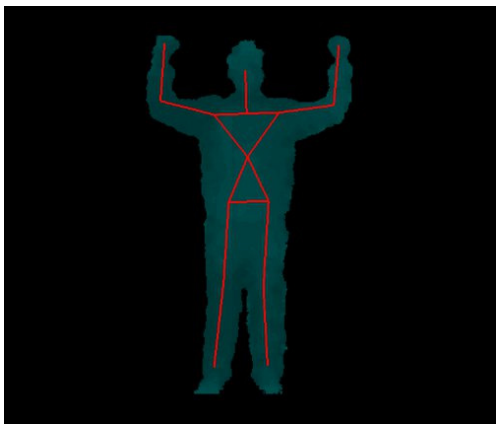


Abbildung 8.5: FAAST / Openni Kalibrierungsgeste

[11] http://www.tuio.org
[12] http://code.google.com/p/tuiokinect/

FAAST verwendet nach eigener Aussage ein *adaptive view-based model* (siehe Morency u. a. (2003)). Die genaue Implementierung ist nicht bekannt, da der Sourcecode derzeit nicht öffentlich verfügbar ist.

Die Software bietet die Emulation eines Treibers für Maus- und Tastaturevents, eine Abbildung einfacher Gesten auf bestimmte Maus- oder Tastatureingaben ist somit möglich. Als Beispiel sei hier die Verwendung der linken Hand als Mauszeiger genannt. Ebenso ist in FAAST ein VRPN-Server enthalten.

8.5 Smart Home Middleware

Für die Interoperabilität der Geräte im Smart Home wird eine Middleware benötigt. Die Anforderungen an eine Middleware in einer Smart-Home-Umgebung finden sich bei Hollatz (2010). Über diese Middleware können angeschlossene Geräte Informationen über Ereignisse in der Wohnung beziehen, zu denen auch erkannte Gesten gehören.

Derzeit existiert kein universeller Standard für eine Middleware in diesen Umgebungen. Es gibt mehrere Vorschläge (siehe 8.5.2). Einen davon, die Software ActiveMQ[13] von der Apache Foundation[14] befindet sich an der HAW Hamburg bereits im Einsatz.

Da es keinen allgemeinen Standard gibt, an den sich Consumergeräte halten könnten, müssen Adapter zu den dort üblichen Standards[15] geschaffen werden.

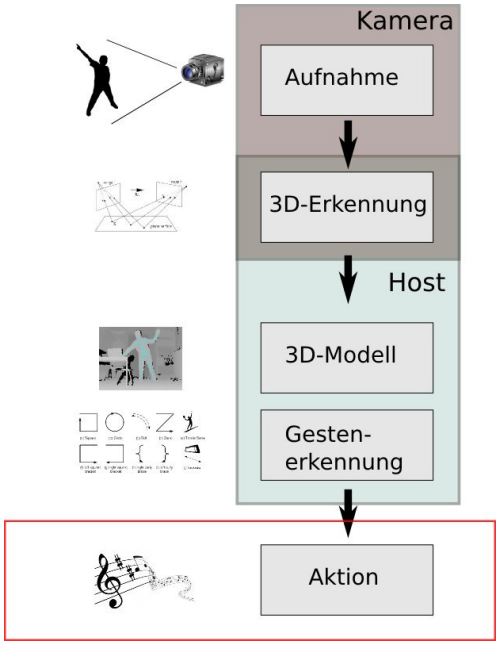


Abbildung 8.6: Übersicht Ablauf Gestenerkennung, ausgelöste Aktionen

[13] http://activemq.apache.org/
[14] http://www.apache.org
[15] Für Mediaplayer beispielsweise UPnP.

8.5.1 ActiveMQ

ActiveMQ ist ein in Java implementierter *Message Broker*[16]. Er kommt seit dem Jahre 2010 im Living Place Hamburg als zentrale Komponente zum plattform- und programmiersprachenunabhängigen Austausch von Nachrichten zum Einsatz (siehe Kjell Otto (2010)). Der Zugriff auf das System ist neben Java auch mit zahlreichen anderen Programmiersprachen wie C#, C/C++ sowie Python und Ruby möglich.

Auch an anderer Stelle wird ActiveMQ genutzt. Huang u. a. (2009) beschreiben ein auf ActiveMQ und OSGI basierendes Framework für intelligente Wohnungen namens *MyHome*. Auch Lu u. a. (2011) verwenden für ihre Experimente zum Tracken mehrerer Personen in einer intelligenten Wohnung diesen Message Broker. Guo u. a. (2010) verwenden ActiveMQ in ihrem *iPlumber* genannten System, zum User-orientierten Management für Ubiquitous Computing-Umgebungen. Dort bietet es die Möglichkeit, den beteiligten Geräten Veränderungen am Kontext anzuzeigen (Context Subscription).

Henjes u. a. (2007) kommen bei ihren Messungen der Performance dieses Message-Broker-Systems zu dem Schluss, dass auch durch das Verarbeiten von mehreren tausend Nachrichten pro Sekunde trotz aktivierter Filterfunktion für einzelne Gruppen von Nachrichten nicht mit Performance-Problemen zu rechnen ist.

8.5.2 Alternativen

Es gibt Alternativen zur Verwendung von ActiveMQ als Message Broker für die Smart Home-Umgebung. Kusznir u. Cook (2010) verwenden in ihrem System ein auf dem XMPP-Protokoll basierendes System. Der früher im Kontext des Living Place als Message Broker evaluierte Eventheap iRos[17] (siehe Hollatz (2008)) wird nicht mehr weiter entwickelt.

Eine weitere Alternative zu ActiveMQ wäre die Verwendung eines Universal Remote Hub Systems im Kontext der Universal Remote Console (ISO 24752, CEA-2014-A). Eine Implementierung für Java ist verfügbar[18].

Da ActiveMQ allerdings im Living Place bereits erfolgreich und performant im Einsatz ist, wäre ein Wechsel der Middleware weder sinnvoll noch ist er angedacht.

[16] Ein *Message Broker* ist eine Middleware zum Austausch von Nachrichten über Systemgrenzen. Der Austausch der Nachrichten erfolgt über das Netzwerk.
[17] http://sourceforge.net/projects/iros/
[18] siehe http://myurc.org/tools/UCH/

8.5.3 URC

Die Universal Remote Console ist ein Standard zum Fernbedienen von Geräten im Smart Home Kontext. URC kommt in „The smart kitchen“ am Deutschen Forschungszentrum für Künstliche Intelligenz Saarbrücken zum Einsatz (siehe Neßelrath u. a. (2011)). Da es keine Geräte gibt, die URC direkt unterstützen, wurden Adapter zu anderen Standards wie z.B. UPnP zur Steuerung eines Mediacenters[19] entwickelt.

Auch die Kombination mit ActiveMQ als Schnittstelle zu Multimedia-Geräten, die die UPnP-Schnittstelle unterstützen, ist möglich.

8.6 Fazit

Für die Anbindung der Komponenten für die Gestenerkennung im Living Place gibt es mehrere Möglichkeiten. Für die Wahl der Treiberbibliothek ist OpenNI derzeit die flexibelste Lösung, da es Kinect und PrimeSensor unterstützt.

Weder für die Skelettmodellierung von OpenNI noch für die des Microsoft Kinect SDK sind die Latenzzeiten bekannt. Eine Beurteilung der Frage, ob ein Skelettmodell als Zwischenschritt zur Gestenerkennung – im Bezug auf die Latenzzeiten – tragbar ist, ist somit nicht möglich. Eine zusätzliche Verzögerung wäre nur zu rechtfertigen, wenn sich die Erkennungsrate gegenüber Verfahren, die keine Modellbildung erfordern, signifikant erhöht. Dies lässt sich nur mit weiteren Untersuchungen (siehe Kapitel 9.2) beantworten.

Wünschenswert wäre eine völlig freie Implementierung auf Basis von libfreenect und einer Erkennung von Skelett und Gesten auf Basis von Open Source-Software. Dies würde zum einen Raum für die Untersuchung und das Hinzufügen von Verbesserungen bieten und andererseits auch für ToF-Kameras verwendbar sein, da nur die Ansteuerungsebene für die Kamera ausgetauscht werden müsste. Außerdem wäre die Verwendung auf allen relevanten Betriebssystemen möglich.

Die Evaluierung von ActiveMQ in Verbindung mit URC ist eine Option, die näherer Evaluierung bedarf. Alternativ muss über das Implementieren eines eigenen Steuerungsservers für Multimedia-Abspielgeräte mit dem UPnP-Standard nachgedacht werden, um diese Geräteklasse einbinden zu können.

[19] Siehe Huang u. a. (2008) für eine beispielhafte Verwendung von UPnP Standards im Smart Home.

9 Beurteilungskriterien für ein System zur Gestenerkennung

In diesem Kapitel werden die Kriterien zur Beurteilung eines Systems zur Gestenerkennung zusammengefasst und Vorschläge für weitere Untersuchungen vorgestellt.

9.1 Zusammenfassung der Kriterien

Die folgende Zusammenstellung ist eine Zusammenfassung der in den vorherigen Kapiteln betrachteten Anforderungen und der sich daraus ergebenden Kriterien zur Beurteilung eines existierenden Systems zur Erkennung von dreidimensionalen Gesten.

Die Gruppierung erfolgt anhand der Möglichkeiten zur Validierung (Analyse, Messung, Benutzerbefragung). Auf Basis dieser Kriterien ist die Ausarbeitung eines Fragebogens zum Vergleich von Systemen möglich.

9.1.1 Schnittstellen

Die hier folgenden Kriterien beziehen sich auf die technischen Schnittstellen. Die Beurteilung der Kriterien kann anhand von technischer Dokumentation oder Analyse eines vorhandenen Systems erfolgen.

Bussysteme - Welche physikalischen Schnittstellen sind zum Anschluss an das Netzwerk im Smart Home vorhanden (siehe 4.1.3 und 4.2.5)?

Softwareschnittstellen Welche Schnittstellen und Protokolle sind auf Anwendungsebene verfügbar (siehe 8.5 und 4.2.5)?

9.1.2 Datenschutz

Im Folgenden werden die Kriterien für die Beurteilung der Belange des Datenschutzes zusammengefasst. Die Beantwortung der Fragen in diesem Bereich erfolgt bei geschlossenen Systemen (*Closed Source*) durch den Hersteller oder durch *Reverse Engeneering* und bei offenen Systemen (*Open Source*) durch Analyse der Abläufe im Programmcode.

Design - Ist das System nach Kriterien des Datenschutzes konzipiert, gibt es eine Kapselung der Daten (siehe 4.3.5)?

Videodaten - Nimmt das System Daten mit einer Videokamera auf oder werden nur – zur Identifizierung von Personen deutlich schlechter geeignete – Tiefeninformationen verwendet (siehe 4.3.6)? Wenn Daten aufgenommen werden, wo und wie werden diese gespeichert? Sollte eine Aufzeichnung erfolgen, verbleiben die Daten auf dem System und wie werden sie dort gespeichert (siehe 4.3.8)?

Daten - Welche Daten gibt das System nach außen (siehe 4.3.6)? Lassen sich darüber einzelne Personen identifizieren (siehe 4.3.5)? Gibt es eine Schnittstelle zum Wartungszugriff von außerhalb? Wenn diese existiert, welche Daten sind darüber verfügbar? Wie ist diese Schnittstelle gegen unbefugte Benutzung abgesichert?

Kontrolle und Transparenz - Ist erkennbar, wann die Kameras Daten aufnehmen (siehe 4.3.6)? Wenn es eine Anzeige dafür gibt, ist es möglich, diese zu umgehen? Ist ein komplettes Ausschalten des Systems für den Benutzer möglich? Ist die Validierung der Software durch den Anwender möglich (siehe 4.3.4)?

9.1.3 Leistung des Systems

In diese Kategorie fallen Kriterien, die die Leistung des Systems beschreiben. Die Ermittlung der einzelnen Variablen erfolgt durch Messungen. Ein Vorschlag zur Messung der Latenzen und Erkennungsraten findet sich in Abschnitt 9.3.1.

Responsiveness - Wie groß ist die durchschnittliche Gesamtlatenz des Systems (siehe Abschnitt 4.2.9)? Liegt dieser Wert innerhalb der Grenzen von Echtzeit (150 Millisekunden)? Welchen Wert haben die minimalen und maximalen Abweichungen?

Zuverlässigkeit - Wie zuverlässig funktioniert die Gestenerkennung, also mit welcher Erkennungsrate arbeitet das System (siehe 4.2.6)? Anzustreben sind hier 100 Prozent.

Sichtbereich - Wie groß ist der Sichtbereich der verwendeten Kameras in unterschiedlichen Entfernungen? Dieser Sichtbereich gibt Aufschluss über das Aktionsfeld, das vor den Sensoren für den Benutzer entsteht, innerhalb dessen eine Kommunikation möglich ist (siehe 5.8.4).

Leistungsaufnahme - Wie hoch ist die elektrische Leistungsaufnahme (siehe 4.1.2)?

Lautstärke - Wie groß ist die Lautstärke im Ruhebetrieb? wie laut unter Last(siehe 4.1.2)?

Mechanische Robustheit - Ist das System mechanisch robust aufgebaut(siehe 4.1.2)?

9.1.4 Usability-Analyse

Die folgenden Kriterien stammen der Usability und sind durch eine externe Analyse des Systems beurteilbar.

Barrierefreiheit - Ist die Benutzung für körperlich behinderte Menschen geeignet (siehe 4.2.8)?

Feedback - Gibt es Feedback bei der Erkennung einer Geste oder im Fehlerfall (siehe 4.2.15)? Wie ist dieses Feedback beschaffen?

Gestenalphabet - Aus welchem Bereich stammt das Gestenalphabet, welche Art von Gesten werden dafür eingesetzt (Klassifizierung siehe 7.2.3)? Sind die Gesten dem Anwendungsbereich angemessen oder führen sie zu einer übermäßigen Ermüdung (siehe 4.2.8)? Ist das Gestenalphabet leicht zu erlernen (siehe 4.2.8)?

Mentales Modell - Ist ein mentales Modell für die Bedienung bekannt (siehe 4.2.10)? Wird dieses kommuniziert, und wenn ja, wie?[1]

Kosten - Wie hoch sind die Anschaffungskosten? Gibt es monatliche Kosten, wenn ja, wie hoch sind diese (siehe 4.2.4)?

9.1.5 Usability-Untersuchungen

Die folgenden Kriterien erfordern ganz oder teilweise die Befragung der Benutzer, beispielsweise durch einen Online-Fragebogen.

Nützlichkeit - Die Frage nach der Nützlichkeit eines Systems kann teilweise durch Analyse beantwortet werden. Inwieweit ist das System geeignet für die Problemstellung? Gibt es alternative Interaktionsmöglichkeiten (siehe 4.2.7)? Gibt es einen zusätzlichen Nutzen wie *körperliche Betätigung*? Die Beantwortung der letzten Frage sollte durch Messungen erfolgen.

Benutzbarkeit - Die Beurteilung der unterschiedlichen Kriterien an die Benutzbarkeit verlangt eine differenzierte Herangehensweise. Die Fehlerrate ergibt sich aus der Zuverlässigkeit (siehe Abschnitt Zuverlässigkeit in 9.1.3). Erlernbarkeit, Effizienz, Einprägsamkeit und Zufriedenheit erfordern die Durchführung von Usability-Studien mit Befragung der Benutzer.

[1] beispielsweise durch einen Trainingsmodus oder ein Handbuch.

9.1.6 Umgebung

Die folgenden Kriterien sind durch Analyse und Messungen zu beurteilen.

Beleuchtung - Ist zum Betrieb des Systems eine bestimmte Beleuchtung erforderlich (siehe 5.8.3)? Neben einer Analyse sollten die Ergebnisse durch Messungen verifiziert werden.

Situation - Gibt es Beschränkungen, in welcher Situation das System funktioniert? Nur für einen einzelnen Benutzer, für mehrere oder eine große Anzahl (siehe 5.8.5)? Neben einer Analyse sollten die Ergebnisse durch Messungen verifiziert werden.

9.2 Offene Fragen

Bei der Literatur-Recherche ergaben sich folgende offenen Fragen, die nicht hinreichend geklärt sind.

9.2.1 Toleranz bei Erkennungsraten

Die Frage nach der Fehlertoleranz bei der Erkennung von Gesten scheint nicht hinreichend geklärt (siehe 4.2.6). Karam (2006) untersucht dies zwar für eine einfache Geste, allerdings nur mit einer kleinen Gruppe (26 Probanden). Wie unterscheidet sich die Toleranz bei unterschiedlichem Anwendungsgebiet? Gibt es hier Unterschiede für verschiedenen Benutzergruppen, beispielsweise Altersunterschiede?

9.2.2 Grenzen der Latenz

Die Frage nach den Latenzzeiten, die Benutzer bei der Interaktion mit Gesten akzeptieren, scheint ebenso nicht hinreichend geklärt zu sein (siehe 4.2.9). Gibt es Unterschiede hinsichtlich der Art der Aufgabenstellung, also für manipulative, kommunikative oder kontrollierende Gesten? Unterscheidet sich die Akzeptanz von Latenzzeiten bei Spielen von der bei der Steuerung von Multimediageräten? Gibt es hierbei Unterschiede in verschiedenen Gruppen von Anwendern?

9.2.3 Präferenz bestimmter Gesten

Gibt es bestimmte Präferenzen von Benutzern hinsichtlich des Gestenalphabets? Werden dynamische Gesten bevorzugt oder eher statische? Unterscheiden sich diese Präferenzen in unterschiedlichen Kulturkreisen oder in unterschiedlichen Benutzergruppen?

9.2.4 Datenschutz im Smart Home

Die Frage nach den Möglichkeiten des Datenschutzes für Smart-Home-Umgebungen scheint ebenso nicht hinreichend geklärt (siehe 4.3). Dies ist unter anderem darauf zurückzuführen, dass es keinen Standard für eine Middleware im Smart Home gibt und somit die Architekturfrage nicht geklärt ist.

9.3 Mögliche Folge-Untersuchungen

Im weiteren Verlauf werden Vorschläge unterbreitet, wie eine Untersuchung der aufgeworfenen Fragen erfolgen kann. Dies ist als Ausblick auf weitere Untersuchungen im Rahmen der Gestenerkennung im Living Place Hamburg zu verstehen.

9.3.1 Messungen der Toleranzgrenzen von Benutzern

Zur Messung der Grenzen in Bezug auf Toleranz und Fehlerrate wird ein Testaufbau analog zum Designvorschlag (siehe 10.1) verwendet. Der Testaufbau sollte dauerhaft im Living Place verbleiben können und einfach zu bedienen sein; der Test selbst nur wenige Minuten dauern. Dies bietet die Möglichkeit, dass normale Besucher diesen Test durchführen können, etwa im Rahmen einer Führung. Dies ermöglicht eine größere Gruppe verschiedener Probanden.

Der Testablauf sollte sich an einem Anwendungsszenario orientieren, beispielsweise die Steuerung eines – fiktiven – Mediaplayers um dem Probanden das Hineindenken in das mentale Modell der Steuerung zu erleichtern.

Eine Möglichkeit zur Ermittlung der Toleranzen wäre zum einen die Befragung der Benutzer. Eine andere wäre ein Testaufbau, der den Testpersonen die Möglichkeit bietet, auf eine alternative Eingabemodalität zu wechseln, beispielsweise eine Konsole mit einem Drehrad und Knöpfen für die verschiedenen Kommando-Gesten zur Bedienung eines Abspielgeräts. Dabei werden die Testpersonen gebeten, die Gesteneingabe zu benutzen. Der Wechsel zur alternativen Eingabemodalität gibt einen Hinweis, ob die Grenze für den Anwender erreicht ist (analog

zu Karam (2006)). Durch einen Testdurchlauf ohne künstliche Erhöhung der Latenz und Fehlerrate wird getestet, inwieweit die Probanden überhaupt mit dem System arbeiten können; wechseln sie bereits dann zur alternativen Modalität, wird der Test abgebrochen. Die zweite Möglichkeit bietet den Vorteil, dass die Subjektivität der Befragung vermieden wird. Da der Benutzer nicht weiß, dass das Messkriterium ist, wann er zur zweiten Modalität wechselt, erfolgt dies *intuitiv* ab dem Punkt, ab dem die Grenze überschritten ist.

Um nicht mit Seiteneffekten bei der Messung konfrontiert zu werden, sollte das System einfach zu bedienen sein, also ein einfaches Gestenalphabet verwenden. Das Alphabet sollte leicht zu merken sein und nur aus wenigen Gesten bestehen, wie das in Abschnitt 10.1.1 vorgestellte. Eine transportable Ausführung des gesamten Testsystems ist sinnvoll, dies ermöglicht die Aufstellung an anderen Orten als dem Living Place, beispielsweise parallel zu einer Ausstellung[2].

9.3.2 Präferenz von Gestenalphabeten

Ein Aufbau wie in 9.3.1 würde sich ebenfalls eignen, um die Präferenzen von Benutzern für ein bestimmtes Gestenalphabet zu untersuchen. Dazu könnten Aufgaben mit unterschiedlichen Gestenalphabeten zu erledigen sein und abschließend eine Befragung nach der Präferenz erfolgen. Oder die Alphabete werden kurz vorgestellt und der Benutzer hat die Möglichkeit eines auszuwählen, das ihm am besten zur Erledigung der Aufgabe erscheint.

9.3.3 Latenzmessungen bei Gestenerkennungssystemen

Zur Frage, welche Latenzen überhaupt toleriert werden, kommt die Frage, wie man die Latenzen eines Komplettsystems (*black box*) misst.

Bei softwarebasierten Systemen gibt es die Möglichkeit, die Schnittstellen der Systemteile untereinander zu nutzen, um dort einen Datenabgriff durchzuführen (*Hook*-Methode). Bei Komplettsystemen ist dies nicht durchführbar. Zudem sollte die Messung automatisiert ablaufen, damit eine hohe Wiederholungszahl und somit eine quantitative Auswertung sowie das Minimieren von Messfehlern möglich ist. Weitere Vorteile der Automatisierung sind die Wiederholbarkeit des Tests bei Softwareänderungen und der Ausschluss der Varianzen, die durch manuelles Durchführen von Gesten entstehen. Die folgenden Abschnitte beziehen sich auf die Untersuchung von Black-Box-Systemen:

[2] Als Beispiel die Ausstellung *reactive landscapes* im Frappant http://livingplace.informatik.haw-hamburg.de/blog/?p=348

Menschersatz

Das Erkennungssystem von OpenNI benötigt zu Beginn der Nutzung eine Kalibrierungsgeste (siehe Abbildung 8.5). Dies kann auch bei anderen Systemen der Fall sein. Da die Messung automatisiert erfolgen soll, wird eine Vorrichtung benötigt, die die vom System zu erkennenden Bewegungen ausführen kann. Ein Vorschlag für den konkreten Fall ist die Verwendung eines Pappaufstellers (verwendet für Werbung für Kinofilme), der die entsprechende Geste *ausführt*. Ein Arm ist dabei beweglich gelagert und kann mit Hilfe eines Schrittmotors von oben nach unten bewegt werden. Für Tests mit komplexeren Gesten ist über die Verwendung einer Puppe mit einem oder zwei Roboterarmen nachzudenken.

Aufnahme

Die Aufnahme erfolgt mit einer Kamera mit Highspeed-Zeitlupenfunktion (HS). Dabei ist die Position der Kamera so zu wählen, dass sie sowohl die Geste als auch die auf dem Bildschirm angezeigte Reaktion auf diese in ihrem Bildbereich erfasst. Die Kamera sollte sowohl den beweglichen Arm, als auch den Bildschirm erfassen. Dies hat den Vorteil, dass eine Synchronisation mehrerer Kameras entfallen kann.

Auswertung

Durch Festlegung von Referenzpunkten (analog zum Verfahren bei LpLatencyMeasure) kann festgestellt werden, wann die Bewegung des Akteurs zu einer Reaktion führt.

9.3.4 Datenschutz im Smart Home

Die im Living Place verwendete Architektur kann als Grundlage genutzt werden, um Konzepte zum Datenschutz im Smart Home zu entwickeln und zu testen. Dies könnte im Rahmen von weiteren Arbeiten an der HAW Hamburg geschehen.

9.4 Fazit

Die vorgestellten Kriterien bieten die Möglichkeit des formalisierten Vergleichs von Systemen zur Gestenerkennung. Sie sind gleichzeitig ein Hinweis auf die Fragen, die bei der Entwicklung eines Systems beachtet werden sollten. Die aufgezeigten offenen Fragestellungen bieten die Möglichkeit für weitere Forschung im Living Place. Dazu ist es nötig, ein System zur Gestenerkennung im Living Place aufzubauen. Einen Entwurf dafür wird im folgenden Kapitel vorgestellt.

10 Designskizze

10.1 Systementwurf für die Gestenerkennung im Living Place

Der hier vorgestellte Entwurf ist ein Vorschlag für ein System zur Erkennung von Gesten im Living Place Hamburg. Er kann als Prototyp für die weiteren Untersuchung dienen. Abbildung 10.1 zeigt eine Übersicht über die Architektur. Grundsätzlich kommt eine Schichtenarchitektur zum Einsatz, wobei einzelne Abschnitte als Client-Server-Architektur ausgeführt sind (TUIO und ActiveMQ sind Client-Server-Protokolle). TUIO wird als Schnittstelle zwischen OpenNI und iGesture eingesetzt.

Als Anwendungsfall ist die Steuerung von Multimediageräten (siehe Abschnitt 3.3) vorgesehen, also die Steuerung eines Multimedia-Players mit Handgesten.

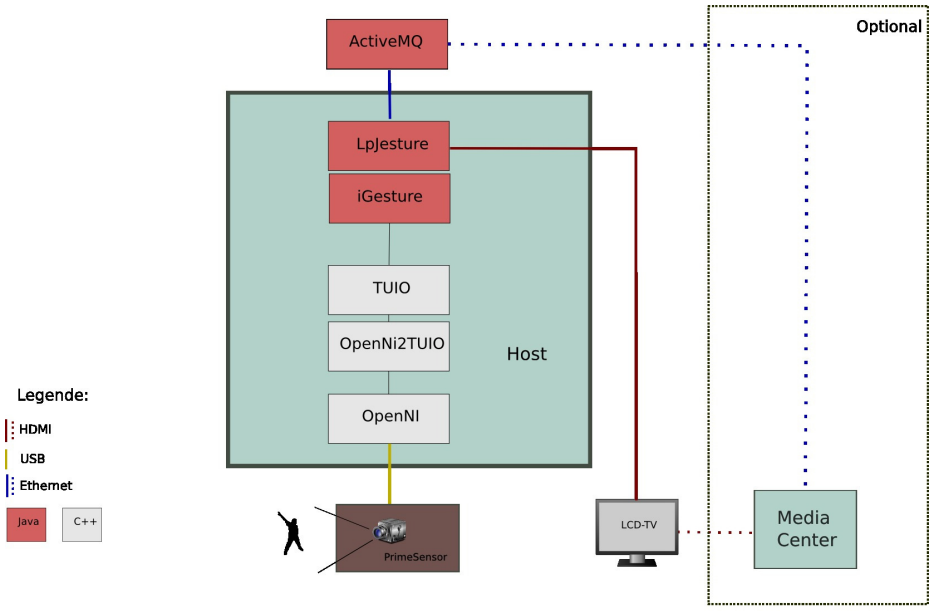


Abbildung 10.1: Architektur Gestenerkennungssystem im Living Place

10.1.1 Komponenten

Die nachfolgenden Komponenten sollen zum Einsatz kommen. Die Begründung für ihre Auswahl findet sich in den entsprechenden Absätzen.

Kamera

Zum Einsatz kommt das ASUS Xtion PRO System. Der Hauptgrund für die Auswahl dieser Kamera ist die hohe mögliche Bildrate von 60 Bildern pro Sekunde, die geringere Latenz gegenüber dem Kinect-System, sowie die Unterstützung durch die OpenNI-Software. Der Anschluss an das Hostsystem erfolgt über USB 2.0.

Host

Zum Einsatz als Hostsystem wird ein Mac-Mini gewählt. Gründe hierfür sind eine ausreichende Performance und ein kleines Gehäuse. Letzteres ermöglicht die Mobilität des Aufbaus. Auf dem Mac-Mini lassen sich sowohl MacOS X, Linux als auch Windows7 betreiben, dies sichert maximale Flexibilität. Zum Einsatz als Betriebssystem soll Windows7 kommen, da iGesture derzeit nicht unter Linux lauffähig ist, und so die Möglichkeit besteht, die Kinect und das MS SDK als Ersatz für OpenNI einzusetzen. Dies würde allerdings die Entwicklung einer Adapterschicht analog zu OpenNI2TUIO erfordern. Die Verbindung zum Hausnetz des Living Place, und damit zu ActiveMQ erfolgt über Ethernet. Zur Ausgabe von weiteren Informationen verfügt das System über einen Anschluß zum LCD-TV über HDMI.

OpenNI

Als Treiber für das ASUS Kamerasystem dient OpenNI mit NITE, dem Modul für die Skeletterkennung. OpenNI kommt deshalb zum Einsatz, da es als einzige Software (derzeit) Skeletterkennung und Unterstützung für PrimeSensor/ASUS Xtion PRO bietet. Skeletterkennung soll zum Einsatz kommen, damit eine simple Gestenerkennung einfach zu implementieren ist, beispielsweise durch Verfolgen der Hände. libfreenect bietet dieses derzeit nicht.

OpenNI2TUIO

Als Adapter zwischen OpenNI und iGesture kommt OpenNI2TUIO[1] zum Einsatz. Dabei werden die Koordinaten der Hände als 3D-Koordinaten mit Hilfe des TUIO-Protokolls über eine IP-basierte Netzwerkschnittstelle übertragen, im vorliegenden Design über das localhost-Interface. OpenNI2TUIO verwendet Bestandteile der openframeworks-Bibliothek[2].

TUIO

TUIO dient als Übertragungsprotokoll und Adapter zwischen OpenNI und iGesture. Näheres zu TUIO findet sich in Abschnitt 8.4.3. Die Auswahl fällt auf TUIO, da das alternativ verfügbare VRPN (siehe 8.4.2) keine Schnittstelle zu iGesture besitzt.

iGesture

Zur Erkennung der Gesten soll iGesture zum Einsatz kommen. iGesture verfügt über eine Workbench zur Erstellung von Gesten. Zusammen mit dem bereits integrierten Rubine3D-Algorithmus lässt sich ein Gestenalphabet mit geringem Aufwand realisieren.

ActiveMQ

Da ActiveMQ bereits im Living Place als Message Broker im Einsatz ist, fällt die Wahl auf diese Software als Middleware. Die in Abbildung 10.1 optional dargestellte Verbindung zu einem Multimedia-Abspieler ist derzeit nicht verfügbar. Sie ließe sich aber mit wenig Aufwand über einen Adapter realisieren, der die Informationen von ActiveMQ auf die Steuerung der Multimedia-Software umsetzt. Für erste Versuche und weitere Untersuchungen (siehe 9.3) ist die Anbindung an ActiveMQ nicht zwingend erforderlich, sie kann daher in einem zweiten Schritt erfolgen.

lpJesture

lpJesture ist als Applikation in Java vorgesehen. Sie dient als zentraler Controller in diesem Aufbau. Java bietet sich als Programmiersprache an, da sowohl ActiveMQ als auch iGesture in Java implementiert sind und direkte Aufrufe von Klassen der iGesture-Bibliothek ermöglicht. Nach der Erkennung einer Geste durch iGesture können sowohl Nachrichten an ActiveMQ

[1] siehe http://dl.dropbox.com/u/335522/web/code/OpenNI2TUIO.zip
[2] siehe http://www.openframeworks.cc/

übermittelt werden, als auch direkt Daten auf dem Bildschirm (LCD-TV) ausgegeben werden. Zur Steuerung von Testprogrammen bietet lpJesture eine Plugin-Architektur, durch die wechselnde Szenarien realisiert werden können.

Mediacenter

Es sind derzeit keine Multimedia-Clients mit Anbindung an ActiveMQ verfügbar. Dies sollte dauerhaft geändert werden, ist für erste Messungen aber nicht erheblich. Deshalb ist das Mediacenter in Abbildung 10.1 als *optional* deklariert.

Begründung für den Aufbau

Ein Schichtenarchitektur bietet einen einfachen Aufbau, der trotzdem flexibel genug ist, um einzelne Komponenten gegen andere austauschen zu können. Die Konzentration der Erkennung auf einen Host bietet zum einen den Vorteil einer geringeren Latenz gegenüber dem Aufteilen der Funktionalität auf mehrere Rechner und die damit einhergehenden Übertragungen über das Netzwerk. Zum anderen erfüllt sie auch den Anspruch nach Schutz der Daten, da hiermit eine Kapselung stattfindet und nur die aggregierten Daten (Geste erkannt) an das ActiveMQ weitergegeben werden.

Gestenalphabet

Das Gestenalphabet ist vom Einsatzgebiet abhängig. Für einfache Tests bietet sich eine Untermenge des Unistroke-Alphabets (beispielsweise „A“,Ë“,"T“,Ï“, siehe Abbildung 7.6) an. Das Anlernen des Systems erfolgt nicht per Training von Beispielen, sondern über die Eingabemöglichkeiten der iGesture-Workbench. Die zum Einsatz kommenden Gesten lassen sich somit als dynamisch-kommunikativ klassifizieren.

10.1.2 Risiken

Die folgenden Risiken für dieses Design lassen sich identifizieren:

Latenz

Beim Entwurf kommen Komponenten zum Einsatz, die auf unterschiedlichen Programmiersprachen beruhen. Insbesondere die Gestenerkennung durch iGesture birgt das Risiko, dass sie zu langsam erfolgt, da iGesture Java-basiert ist. Java weist eine geringere Performance im Vergleich zu nativem Code auf. Sollte sich herausstellen, dass die Latenz zu groß ist, könnte beispielsweise iGesture durch eine Implementierung des Algorithmus in C/C++ ersetzt werden.

Der in iGesture verfügbare Algorithmus für die 3D-Gestenerkennung ist Rubine (siehe Abschnitt 7.5.3). Dieser hat eine hohe Berechnungskomplexität. Dies könnte bei einer zu großen Anzahl an Gesten im Alphabet dazu führen, dass die Latenz zu sehr erhöht wird. Aus diesem Grund sollte das Gestenalphabet zu Beginn klein gehalten werden.

Mit *tuiokinect*[3] ist eine alternative Implementierung verfügbar, die anstatt Skelettmodelierung – wie bei OpenNI – eine Verfolgung der Hände mit einem Kalman-Filter bietet. Zur Kommunikation mit der Kamera kommt libfreenect zum Einsatz. Dies ist eine weitere Alternative, falls sich die Verwendung von OpenNI als zu langsam herausstellen sollte. Und sobald libfreenect das PrimeSensor-System unterstützt.

Sowohl die Integration in die Living Place Infrastruktur durch ein Ethernet-Netzwerk, als auch die Verwendung des ActiveMQ-Systems bergen das Risiko, die Latenz der Erkennung zu erhöhen. Gründe hierfür sind die Latenzen durch die Netzwerk-Stacks[4] sowie der Durchlauf der generierten Nachricht durch das ActiveMQ-System.

Clients

Derzeit sind keine Multimedia-Clients im Living Place verfügbar. Es besteht das Risiko, dass sich dieser Zustand sich nicht schnell ändert. Ein Multimedia-Client (Mediacenter) ist nötig zur Messung der Gesamtlatenz im realen System.

Stabilität

Der Quellcode der Erweiterung für die Skelettmodellierung von OpenNI (NITE) ist derzeit nicht verfügbar. Auftretende Fehler können also nicht selber behoben werden. Über die Stabilität der anderen Komponenten kann derzeit noch keine Aussage getroffen werden. Die Kombination von ASUS Xtion Pro und OpenNI (ohne NITE) hat sich während der Latenzmessungen als stabil erwiesen.

[3] http://code.google.com/p/tuiokinect/

[4] Vom Hostsystem zum ActiveMQ-Server; Vom ActiveMQ-Server zum Mediaplayer.

10.2 Erfüllung der Kriterien

In Bezug auf die in Abschnitt 9.1 beschrieben Kriterien ergibt sich folgendes Bild:

Die Bestimmung der Leistungsdaten kann erst nach einem prototypischen Aufbau erfolgen. Als Schnittstellen kommen Ethernet und ActiveMQ zum Einsatz. Der Datenschutz ist grundlegend beim Design berücksichtigt worden, die Schnittstelle gibt nur aggregierte Daten nach außen, eine Identifikation einzelner Personen ist aus diesen Informationen nicht möglich. Es erfolgt keine Aufnahme oder Speicherung von Videodaten. Die Transparenz ist mit Einschränkung gegeben, da die Kamera über eine LED verfügt. Die LED unterliegt allerdings der Steuerung durch den Treiber und ist somit abschaltbar.

Zur Situation und Beleuchtung sind Untersuchungen erforderlich, ebenso zu den Kriterien der klassischen Usability. Das Gestenalphabet scheint geeignet, eine hohe Erkennungsrate zu liefern. Da das Anwendungsszenario keine ständige Interaktion mit dem System erfordert, ist eine Ermüdung unwahrscheinlich. Das zu Grunde liegende Steuerungsmodell sollte dem Anwender mit einem Trainingsprogramm oder einer einfachen Anleitung vermittelt werden. Die Kosten sind zu vernachlässigen, da es sich um einen prototypischen Aufbau mit bereits vorhandener Technik handelt.

Ein an das Hostsystem angeschlossener Monitor (LCD-TV) dient als Schnittstelle für visuelles Feedback an den Benutzer.

10.3 Fazit

Die gewählten Komponenten sind prinzipiell geeignet, ein Testsystem für Gestenerkennung im Living Place Hamburg aufzubauen. Das größte Risiko birgt die derzeit nicht wirklich abschätzbare Gesamtlatenz, insbesondere die Skelettmodelierung mit NITE dürfte hier für eine Erhöhung sorgen. Diese Gesamtlatenz muss durch Messungen bestimmt werden. Die in 9.1 zusammengestellten Kriterien wurden, soweit derzeit feststellbar, erfüllt. Im Zuge der Nutzung für weitere Untersuchungen (siehe 9.3) wird auch die Überprüfung der anderen Kriterien und somit eine Beurteilung möglich.

11 Zusammenfassung und Ausblick

Dieses Kapitel bildet den Abschluss der Masterarbeit und fasst die Arbeit und die gewonnenen Erkenntnisse zusammen. Zusätzlich bietet es einen Ausblick auf zukünftige Forschungen und Anwendungen.

11.1 Zusammenfassung

Zu Beginn stand die Frage nach konkreten Anwendungsbeispielen für die Benutzung von dreidimensionalen Gesten im Living Place: Spiele, Vorträge, die Steuerung von Mediaplayern und neuartigen Musikinstrumenten. Basierend auf diesen Anwendungsbeispielen erfolgte eine Analyse der sich daraus ergebenden Anforderungen. Dabei sind die Themenbereiche Smart Home, Usability und Datenschutz einer näheren Betrachtung unterzogen worden. Insbesondere die Frage der *Responsiveness*, also der Latenz bei der Reaktion des Systems, wurde intensiv betrachtet. Eine Reaktion in Echtzeit gibt dem Benutzer das Gefühl, die Maschine reagiere unmittelbar auf seine Eingaben. Eine Voraussetzung für die Gestenerkennung in Echtzeit ist eine geringe Latenz der verwendeten 3D-Kameras. Die im Living Place verfügbaren Kamerasysteme wurden deshalb auf ihre Latenzzeiten untersucht.

Eine Definition der Klassifizierung von Gesten, gefolgt von einer Übersicht über Gestenalphabete, unterschiedliche Algorithmen zur ihrer Erkennung und die vorhandenen Schnittstellen führten als Grundlage zu einem ersten Entwurf eines Systems zur Gestenerkennung im Living Place.

Die Zusammenfassung der erarbeiteten Anforderungen aus den verschiedenen Themengebieten, ebenso wie die bei der Recherche aufgefallen offenen Fragen waren die Basis für den Ausblick auf weitere Forschungsansätze.

11.1.1 Was wurde erreicht?

Wie in der Aufgabenstellung vorgegeben, wurden die Anforderungen, die sich aus der alltäglichen Benutzung von Gestenerkennung ergeben, untersucht. Die Anforderungen an die Software, Hardware und Architektur, hauptsächlich aus den Bereichen Usability und Datenschutz,

sowie praktische Anforderungen an die Kamerasysteme wurden dargestellt. Es wurden Anwendungsbeispiele vorgestellt, die sich für die Nutzung im Living Place eignen.

Die Eignung von verschiedenen Kamerasystemen wurde sowohl theoretisch als auch praktisch – mit Latenzmessungen – untersucht.

Die erarbeiteten Kriterien zur Beurteilung von Systemen zur Gestenerkennung wurden zusammengefasst; sie können für die weitere Evaluierung von Systemen verwendet werden.

Bestehende Lücken in der Literatur wurden identifiziert und Vorschläge zu ihrer Beseitigung gegeben. Auf Grundlage dieses Wissens wurde eine Designskizze für die Integration von dreidimensionaler Gestenerkennung in den Living Place vorgestellt.

11.1.2 Das unentdeckte Land

Obwohl es schon seit mehr als zwei Jahrzehnten Forschungen im Bereich der Gestenerkennung gibt, scheint man manchmal unentdecktes Land zu betreten. Es gibt weder allgemeinene Standards für Gestenalphabete oder Schnittstellen zwischen den einzelnen Geräten noch zur Frage des Schutzes der anfallenden Daten.

Trotz einiger Vorschläge in den einzelnen Bereichen hat sich bisher keiner durchgesetzt. Dies wird bei kommerzieller Verfügbarkeit im Massenmarkt allerdings schnell geschehen, wahrscheinlich werden zunächst mehrere Standards parallel existieren.

11.2 Fazit

Ob die in der Einleitung verwendete Vision von *Sal* – die Integration von Gestenerkennung in alle Lebensbereiche – jemals Realität wird, ist derzeit offen. Die im Kontext des Living Place Hamburg in dieser Arbeit aufgeführten Anwendungsbeispiele sind für den Anfang realistischer: Der Einsatz von Gestenerkennung in einigen klar definierten Einsatzgebieten.

Die bisherigen Arbeiten in diesem Bereich waren hauptsächlich akademische Studien, mit Studenten oder Mitarbeitern als Probanden und einer Laborumgebung als Umfeld. Der Living Place bietet eine geeignete Ausstattung und die Möglichkeit, das Folgende in einer realitätsnahen Umgebung zu untersuchen:

Ob die zur Verfügung stehenden Kameras, und insbesondere das im Design vorgeschlagenen ASUS Xtion Pro-System den Anforderungen an die Latenz in einem echtzeitfähigen Gesamtsystem genügt, ist nicht klar. Obwohl das Kamerasystem von den Latenzen selber deutlich

unter der Grenze von 150 Millisekunden liegt, besteht die Gefahr, diese Grenze bei der Verwendung eines Skelettmodells zu überschreiten. Ob dem so ist, muss an einem Komplettsystem untersucht werden.

Abgesehen davon bieten 3D-Kameras eine gute Basis für die Erkennung von räumlichen Gesten im Smart Home, da sie auf jeden Fall das Problem der Segmentierung zufriedenstellend lösen können.

Räumliche Gesten werden herkömmliche Eingabemodulatäten wohl nicht *ersetzen*, aber sie können sie gut *ergänzen*. Insbesondere eignet sich die Konzentration auf eine Modalität in dieser neuen Technik dazu, diese zu perfektionieren. Das Ziel, eine menschengerechtere Kommunikation mit der Maschine, und somit die Integration weiterer Modalitäten wie Sprache und Mimik, sollte man dabei nicht komplett aus den Augen verlieren.

Die Erkennung räumlicher Gesten ist ein Schritt weiter in eine Richtung, in der diese Integration wahrscheinlicher wird. Ob dies wünschenswert ist, bleibt den (späteren) potenziellen Benutzern überlassen. Ihre Sicht auf die Technik entscheidet, ob sie ein Spielzeug, eine Kommunikationsmöglichkeit zu ihrem Computer oder eine Überwachungstechnik wie aus *1984*[1] vor sich haben. Den Datenschutz im Blick zu behalten, kann allerdings nicht schaden.

Insgesamt besteht die Gefahr, dass die bestehende Risiken im Hinblick auf Überwachungsszenarien erst erforscht werden, wenn die Technik schon massiv im Markt etabliert wurde. Eine frühzeitige Betrachtung der Chancen und der Risiken wäre angebracht. Die Integration von kamerabasierten Systemen zur Gestenerkennung hat das Potential, die Kommunikation von Menschen und Maschinen zu vereinfachen. Allerdings sollte man sich bewusst sein, dass der Preis dafür hoch sein kann. Vorhandene Kameras wecken die Begierde, sie auch zu anderen Zwecken zu benutzen. Es ist Aufgabe der Entwickler solcher Systeme, die Risiken im Blick zu behalten und durch ein gutes Design möglichst im Vorwege auszuschließen. Es ist Aufgabe der Gesellschaft, die notwendige Debatte über diese Risiken zu führen.

11.3 Ausblick

Eine Möglichkeit für weitere interdisziplinäre Forschungen ist die Verwendung der Technik *off limits*, also für andere Zwecke als die Gestenerkennung. Beispielsweise werden in der Sportwissenschaft derzeit Tracking-Mechanismen mit passiven Infrarotmarkern eingesetzt. Diese durch eine einfache, robuste und vor allem mobile Technik zu ersetzen, wäre ein lohnendes Ziel, da es die Forschungen in diesem Bereich vereinfachen würde.

Dreidimensionale Kameras lassen sich auch für den Versuch nutzen, die Intentionen von Menschen anhand ihrer Bewegungen und Aktionen zu interpretieren. Beispielsweise kann man

[1] *1984* von George Orwell

durch die Position in der Wohnung Rückschlüsse auf kommende Aktionen ziehen: ein Mensch der sich aus dem Bad kommend der Kaffeemaschine nähert und vor dieser stehen bleibt, möchte vielleicht wirklich einen Kaffee. Ob die Intention immer klar aus den Bewegungen ableitbar ist, werden weitere Forschungen auf dem Gebiet zeigen müssen.

Ein Ausblick auf mögliche Forschungen im Kontext des Living Place findet sich bereits in Abschnitt 9.3. Ausgehend von diesen Untersuchungen sollte das Thema multimodale Interaktion weiter betrachtet werden: Die Integration von Gesten im Kontext von multimodaler, situationsabhängiger Kommunikation zwischen Mensch und Maschine ist ein Gebiet, auf dem noch viel Arbeit nötig ist. Dies bietet Raum für umfangreiche Forschung.

Die in dieser Arbeit behandelten Problemfelder beziehen auf die *Kinderkrankheiten* einer neuen Technik. Sobald diese zuverlässig funktioniert, und die Syntax der Kommunikation klar erkannt wird, steht schon die nächste Herausforderung bereit: Die zuverlässige Erkennung der Semantik.

Abbildungsverzeichnis

Literaturverzeichnis

[SR4] *SR4000 User Manual*. : *SR4000 User Manual*, http://www.mesa-imaging.ch/dlm.php?fname=customer/Customer_CD/SR4000_Manual.pdf

[mey 1981] *Meyers Großes Taschenlexikon*. Bd. Band 3. Meyer, 1981. – ISBN 3411–019239

[802.3ab 1999] 802.3AB, IEEE: IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements. Supplement to Carrier Sense Multiple Access With Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications - Physical Layer Parameters and Specifications for 1000 Mb/s Operation Over 4-Pair of Category 5 Balanced Copper Cabling, Type 1000BASE-T. In: *IEEE Std 802.3ab-1999* (1999), S. i. http://dx.doi.org/10.1109/IEEESTD.1999.90568. – DOI 10.1109/IEEESTD.1999.90568

[ADFC Wedel 2010] ADFC WEDEL: *Betellampel*. http://www.adfc-wedel.de/benutzer/arne/bilder/bettelampel01.jpg. Version: 2010

[Aggarwal u. Wang 1988] In: AGGARWAL, J. K. ; WANG, Y. F.: *Inference of object surface structure from structured lighting - an overview*. San Diego, CA, USA : Academic Press Professional, Inc., 1988. – ISBN 0–12–266720–4, 193–220

[Amazon Co. Uk 2011] AMAZON CO. UK: *kinect*. http://www.amazon.co.uk/. Version: 2011

[ASUSTeK COMPUTER INC. 2011] ASUSTEK COMPUTER INC.: *Xtion PRO*. http://www.asus.de/Multimedia/Motion_Sensor/Xtion_PRO/. Version: 2011

[Athitsos u. a. 2010] ATHITSOS, Vassilis ; WANG, Haijing ; STEFAN, Alexandra: A database-based framework for gesture recognition. In: *Personal Ubiquitous Comput.* 14 (2010), September, 511–526. http://dx.doi.org/http://dx.doi.org/10.1007/s00779-009-0276-x. – DOI http://dx.doi.org/10.1007/s00779–009–0276–x. – ISSN 1617–4909

[Ballmer 2010] BALLMER, Steve: *CES 2010: A Transforming Trend – The Natural User Interface*. http://www.huffingtonpost.com/steve-ballmer/ces-2010-a-transforming-t_b_416598.html. Version: 2010

[Barnard u. Thompson 1980] BARNARD, D. T. ; THOMPSON, William B.: Disparity Analysis of Images. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-2 (1980), S. 333–340. http://dx.doi.org/10.1109/TPAMI.1980.4767032. – DOI 10.1109/TPAMI.1980.4767032

[Baum u. Petrie 1966] BAUM, Leonard E. ; PETRIE, Ted: Statistical Inference for Probabilistic Functions of Finite State Markov Chains. In: *The Annals of Mathematical* (1966)

[Van den Bergh u. Van Gool 2011] BERGH, Michael Van d. ; VAN GOOL, Luc: Combining RGB and ToF cameras for real-time 3D hand gesture interaction. In: *Applications of Computer Vision (WACV), 2011 IEEE Workshop on* (2011), S. 66 –72. http://dx.doi.org/10.1109/WACV.2011.5711485. – DOI 10.1109/WACV.2011.5711485. – ISSN 1550–5790

[Bergmann 2009] BERGMANN, Nicole: *Volkszählung und Datenschutz - Proteste zur Volkszählung 1983 und 1987 in der Bundesrepublik Deutschland*. Diplomica Verlag, 2009. – ISBN 9783836673884

[Bernin 2009] BERNIN, Arne: Räumliche Segmentierung mit Differenzbildern. / HAW Hamburg. Version: 2009. http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2009-proj/bernin.pdf. 2009. – Forschungsbericht

[Bernin 2011a] BERNIN, Arne: *S-Bahn Berührungsschalter*. 2011

[Bernin 2011b] BERNIN, Arne: *S-Bahn Druckschalter*. 2011

[Blagojevic u. a. 2010] BLAGOJEVIC, Rachel ; CHANG, Samuel Hsiao-Heng ; PLIMMER, Beryl: The power of automatic feature selection: Rubine on steroids. In: *Proceedings of the Seventh Sketch-Based Interfaces and Modeling Symposium*. Aire-la-Ville, Switzerland, Switzerland : Eurographics Association, 2010 (SBIM '10). – ISBN 978–3–905674–25–5, 79–86

[Boetzer 2008a] BOETZER, Joachim: *Bewegungs- und gestenbasierte Applikationssteuerung auf Basis eines Motion Trackers*. Bachelorarbeit. http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/bachelor/boetzer.pdf. Version: 08 2008

[Boetzer 2008b] BOETZER, Vogt-Wendt v. Rahimi: Gestenbasierte Interaktion mit Hilfe von Multitouch und Motiontracking. In: CLEVE, Jürgen (Hrsg.) ; Hochschule Wismar (Veranst.): *Proceedings WIWITA 2008* Hochschule Wismar, 2008, S. 38–45

[Bolt 1980] BOLT, Richard A.: Put-that-there: Voice and gesture at the graphics interface. In: *Proceedings of the 7th annual conference on Computer graphics and interactive techniques*. New York, NY, USA : ACM, 1980 (SIGGRAPH '80). – ISBN 0–89791–021–4, 262–270

[Booth 1980] BOOTH, Paul A.: *An introduction to human-computer interaction*. 1980

[Boyer u. Kak 1987] BOYER, K. L. ; KAK, A. C.: Color-encoded structured light for rapid active ranging. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 9 (1987), January, 14–28. http://dx.doi.org/10.1109/TPAMI.1987.4767869. – DOI 10.1109/TPAMI.1987.4767869. – ISSN 0162–8828

[Bragg 1997] BRAGG, Lois: Visual-Kinetic Communication in Europe Before 1600: A Survey of Sign Lexicons and Finger Alphabets Prior to the Rise of Deaf Education. In: *Journal of Deaf Studies and Deaf Education* 2 (1997), Nr. 1, 1-25. http://jdsde.oxfordjournals.org/content/2/1/1.abstract

[Bærentsen 2002] BÆRENTSEN, Klaus B.: *INTUITIVE USER INTERFACES*. 2002

[Card u. a. 1983] CARD, Stuart K. ; NEWELL, Allen ; MORAN, Thomas P.: *The Psychology of Human-Computer Interaction*. Hillsdale, NJ, USA : L. Erlbaum Associates Inc., 1983. – ISBN 0898592437

[Carroll 1990] CARROLL, John M.: *The Nurnberg funnel: designing minimalist instruction for practical computer skill*. Cambridge, MA, USA : MIT Press, 1990. – ISBN 0–262–0316390

[Cassell 1998] CASSELL, Justine: A Framework For Gesture Generation And Interpretation. In: *Computer Vision in Human-Machine Interaction*, Cambridge University Press, 1998, 191–215

[Castellano u. a. 2007] CASTELLANO, Ginevra ; BRESIN, Roberto ; CAMURRI, Antonio ; VOLPE, Gualtiero: Expressive control of music and visual media by full-body movement. In: *Proceedings of the 7th international conference on New interfaces for musical expression*. New York, NY, USA : ACM, 2007 (NIME '07), 390–391

[Castellucci u. Mackenzie] CASTELLUCCI, Steven J. ; MACKENZIE, I. S.: Unigest: Text entry using three degrees of motion, Extended. In: *Abstracts of the ACM Conference on Human Factors in Computing Systems - CHI 2008*, ACM, S. 3549–3554

[Covey u. Chen 2011] COVEY, John ; CHEN, Ray: Kinect safety / Nanophotonics Research Group, University of Texas at Austin. Version: 2011. http://laserpointerforums.com/f53/kinect-safety-60186.html. 2011. – Forschungsbericht

[Dabrowski u. Munson 2001] DABROWSKI, James R. ; MUNSON, Ethan V.: Is 100 Milliseconds Too Fast? In: *CHI '01 extended abstracts on Human factors in computing systems*. New York, NY, USA : ACM, 2001 (CHI EA '01). – ISBN 1–58113–340–5, 317–318

[Daniel Reetz 2011a] DANIEL REETZ, Matti K.: *Kinect Hacking 103: Looking at Kinect IR Patterns.* Verified, 29.6.2011. http://www.futurepicture.org/?p=116. Version: 2011

[Daniel Reetz 2011b] DANIEL REETZ, Matti K.: *Kinect Hacking 104: Is the Kinect IR Projector Modulated or Synced?* Verified, 29.6.2011. http://www.futurepicture.org/?p=124. Version: 2011

[Darrell u. Pentland 1993] DARRELL, T. ; PENTLAND, A.: Space-time gestures. In: *Computer Vision and Pattern Recognition, 1993. Proceedings CVPR '93., 1993 IEEE Computer Society Conference on*, 1993. – ISSN 1063–6919, S. 335 –340

[Decker 2009] DECKER, Samantha: *Parc Monceau*. http://www.flickr.com/photos/sammers05/3576237827/sizes/l/in/photostream/. Version: 05 2009

[Dey u. a. 2002] DEY, Anind ; LEDERER, Scott ; ; MANKOFF, Jennifer ; LEDERER, Scott ; DEY, Anind K. ; MANKOFF, Jennifer ; MANKOFF, Jennifer: A Conceptual Model and Metaphor of Everyday Privacy in Ubiquitous Computing / in Ubiquitous Computing. Intel Research. Version: 2002. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.5.2673&rep=rep1&type=pdf. 2002. – Forschungsbericht

[Downey 2011] DOWNEY, Matthew: *Ipad2*. Verified, 29.6.2011. http://upload.wikimedia.org/wikipedia/commons/thumb/b/b7/IPad_2.jpg/800px-IPad_2.jpg. Version: 2011

[Edwards u. Grinter 2001] EDWARDS, W. ; GRINTER, Rebecca: At Home with Ubiquitous Computing: Seven Challenges. Version: 2001. http://dx.doi.org/10.1007/3-540-45427-6_22. In: ABOWD, Gregory (Hrsg.) ; BRUMITT, Barry (Hrsg.) ; SHAFER, Steven (Hrsg.): *Ubicomp 2001: Ubiquitous Computing* Bd. 2201. Springer Berlin / Heidelberg, 2001, 256-272. – 10.1007/3-540-45427-6$_2$2

[Fang 2009] FANG, Chunsheng: *From Dynamic Time Warping (DTW) to Hidden Markov Model (HMM)*. 03 2009. – University of Cincinnati

[Ferri 2008] FERRI, Fernando ; FERRI, Fernando (Hrsg.): *Visual languages for interactive computing: definitions and formalizations*. Information Science Reference, 2008, 2008

[Fischer 2011] FISCHER, Jonas: *re:publica XI: Politische Klicks*. Verified, 24.7.2011. http://www.flickr.com/photos/re-publica/5618666577/. Version: 2011

[Focault 1976] FOCAULT, Michel: *Überwachen und Strafen: Die Geburt des Gefängnisses*. Frankfurt am Main : Suhrkamp Verlag, 1976

[Ganapathi u. a. 2010] GANAPATHI, Varun ; PLAGEMANN, Christian ; KOLLER, Daphne ; THRUN, Sebastian: Real time motion capture using a single time-of-flight camera. In: *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on* 0 (2010), S. 755–762. http://dx.doi.org/http://doi.ieeecomputersociety.org/10.1109/CVPR.2010.5540141. – DOI http://doi.ieeecomputersociety.org/10.1109/CVPR.2010.5540141. ISBN 978–1–4244–6984–0

[Glinsky u. Moog 2005] GLINSKY, A. ; MOOG, R.: *Theremin: Ether Music and Espionage*. University of Illinois Press, 2005 (Music in American Life). http://books.google.com/books?id=tXOCAAAACAAJ. – ISBN 9780252072758

[Goldberg u. Richardson 1993] GOLDBERG, David ; RICHARDSON, Cate: Touch-typing with a stylus. In: *Proceedings of the INTERACT '93 and CHI '93 conference on Human factors in computing systems*. New York, NY, USA : ACM, 1993 (CHI '93). – ISBN 0–89791–575–5, 80–87

[Guo u. a. 2010] GUO, Bin ; ZHANG, Daqing ; IMAI, Michita: Enabling user-oriented management for ubiquitous computing: The meta-design approach. In: *Comput. Netw.* 54 (2010), November, 2840–2855. http://dx.doi.org/http://dx.doi.org/10.1016/j.comnet.2010.07.016. – DOI http://dx.doi.org/10.1016/j.comnet.2010.07.016. – ISSN 1389–1286

[Guðmundsson u. a. 2010] GUÐMUNDSSON, Sigurjón ; SVEINSSON, Jóhannes ; PARDÀS, Montse ; AANÆS, Henrik ; LARSEN, Rasmus: Model-Based Hand Gesture Tracking in ToF Image Sequences. Version: 2010. http://dx.doi.org/10.1007/978-3-642-14061-7_12. In: PERALES, Francisco (Hrsg.) ; FISHER, Robert (Hrsg.): *Articulated Motion and Deformable Objects* Bd. 6169. Springer Berlin / Heidelberg, 2010. – ISBN 978–3–642–14060–0, 118-127. – 10.1007/978-3-642-14061-7_12

[von Hardenberg u. Berard 2001] HARDENBERG, Christian von ; BERARD, Francois: Bare-hand human-computer interaction. In: *PUI '01: Proceedings of the 2001 workshop on Perceptive user interfaces*. New York, NY, USA : ACM, 2001, 1–8

[Hassanpour u. a. 2008] HASSANPOUR, Reza ; WONG, Stephan ; SHAHBAHRAMI, Asadollah: VisionBased Hand Gesture Recognition for Human Computer Interaction: A Review. In: *IADIS International Conference Interfaces and Human Computer Interaction 2008*, 2008

[HAW Hamburg] HAW HAMBURG: *Living Place Hamburg*. http://livingplace.informatik.haw-hamburg.de/content/LivingPlaceHamburg_en.pdf

[Heitsch 2008] HEITSCH, Johann: *Ein Framework zur Erkennung von dreidimensionalen Gesten*. Bachelorarbeit HAW Hamburg. http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/bachelor/heitsch.pdf. Version: 08 2008

[Henjes u. a. 2007] HENJES, Robert ; SCHLOSSER, Daniel ; MENTH, Michael ; HIMMLER, Valentin: Throughput Performance of the ActiveMQ JMS Server. Version: 2007. http://dx.doi.org/10.1007/978-3-540-69962-0_10. In: BRAUER, W. (Hrsg.) ; (Hrsg.) ; BRAUN, Torsten (Hrsg.) ; CARLE, Georg (Hrsg.) ; STILLER, Burkhard (Hrsg.): *Kommunikation in Verteilten Systemen (KiVS)*. Springer Berlin Heidelberg, 2007 (Informatik aktuell). – ISBN 978–3–540–69962–0, 113-124. – 10.1007/978-3-540-69962-0_10

[Hoffman u. a. 2010] HOFFMAN, M. ; VARCHOLIK, P. ; LAVIOLA, J.J.: Breaking the status quo: Improving 3D gesture recognition with spatially convenient input devices. In: *Virtual Reality Conference (VR), 2010 IEEE*, 2010. – ISSN 1087–8270, S. 59 –66

[Hollatz 2008] HOLLATZ, Dennis: *Managing Information - Personal Information Environments auf der Basis von iROS*. Verified, 14.8.2011. http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master07-08/hollatz/bericht.pdf. Version: 2008

[Hollatz 2010] HOLLATZ, Dennis: *Entwicklung einer nachrichtenbasierten Architektur für Smart Homes*, Faculty of Engineering and Computer Science Department of Computer Science, Diplomarbeit, 2010. http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/master/hollatz.pdf

[Hossain u. a. 2009] HOSSAIN, M. ; PARRA, Jorge ; ATREY, Pradeep ; EL SADDIK, Abdulmotaleb: A framework for human-centered provisioning of ambient media services. In: *Multimedia Tools and Applications* 44 (2009), 407-431. http://dx.doi.org/10.1007/s11042-009-0285-9. – ISSN 1380–7501. – 10.1007/s11042-009-0285-9

[Huang u. a. 2008] HUANG, Chung-Ming ; LIN, Ming-Sian ; WONG, Hon-Long: A ubiquitous IAs access platform (UIAP) over UPnP. In: *Softw. Pract. Exper.* 38 (2008), September, 1127–1147. http://dx.doi.org/10.1002/spe.v38:11. – DOI 10.1002/spe.v38:11. – ISSN 0038–0644

[Huang u. a. 2009] HUANG, Hsuan-Yu ; TENG, Wei-Chung ; CHUNG, Sheng-Luen: Smart home at a finger tip: OSGi-based MyHome. In: *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, 2009. – ISSN 1062–922X, S. 4467 –4472

[Hummels u. Stappers 1998] HUMMELS, C. ; STAPPERS, P.J.: Meaningful Gestures for Human Computer Interaction: Beyond Hand Postures. In: *Automatic Face and Gesture Recognition, IEEE International Conference on* 0 (1998), S. 591. http://dx.doi.org/http://doi.ieeecomputersociety.org/10.1109/AFGR.1998.671012. – DOI http://doi.ieeecomputersociety.org/10.1109/AFGR.1998.671012. ISBN 0–8186–8344–9

[Hutschenreuther 2005] HUTSCHENREUTHER, Bernd: *Etherwave Thermin Kit*. Verified, 24.7.2011. http://en.wikipedia.org/wiki/File:Etherwave_Theremin_Kit.jpg. Version: 2005

[IEEE 802.3u 1995] IEEE 802.3U: IEEE Standards for Local and Metropolitan Area Networks: Supplement to Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications Media Access Control (MAC) Parameters, Physical Layer, Medium Attachment Units, and Repeater for 100 Mb/s Operation, Type 100BASE-T (Clauses 21-30). In: *IEEE Std 802.3u-1995 (Supplement to ISO/IEC 8802-3: 1993; ANSI/IEEE Std 802.3, 1993 Edition)* (1995), S. $0_1 - -398$

[Ijsselsteijn u. a. 2007] IJSSELSTEIJN, Wijnand ; NAP, Henk H. ; KORT, Yvonne de ; POELS, Karolien: Digital game design for elderly users. In: *Proceedings of the 2007 conference on Future Play*. New York, NY, USA : ACM, 2007 (Future Play '07). – ISBN 978–1–59593–943–2, 17–22

[intertek.com 2003] INTERTEK.COM: *Smart home definition*. http://www.housingcare.org/downloads/kbase/2545.pdf. Version: 2003

[Jacob 1990] JACOB, Robert J. K.: What you look at is what you get: eye movement-based interaction techniques. In: *Proceedings of the SIGCHI conference on Human factors in computing systems: Empowering people*. New York, NY, USA : ACM, 1990 (CHI '90). – ISBN 0–201–50932–6, 11–18

[Jaimes u. Sebe 2007] JAIMES, Alejandro ; SEBE, Nicu: Multimodal human-computer interaction: A survey. In: *Comput. Vis. Image Underst.* 108 (2007), October, 116–134. http://dx.doi.org/10.1016/j.cviu.2006.10.019. – DOI 10.1016/j.cviu.2006.10.019. – ISSN 1077–3142

[Javier Garcia] JAVIER GARCIA, Zalevsky H.: *Range mapping using speckle decorralation*. http://www.freepatentsonline.com/7433024.pdf

[Kakumanu u. a. 2007] KAKUMANU, P. ; MAKROGIANNIS, S. ; BOURBAKIS, N.: A survey of skin-color modeling and detection methods. In: *Pattern Recogn.* 40 (2007), March, 1106–1122. http://dx.doi.org/http://dx.doi.org/10.1016/j.patcog.2006.06.010. – DOI http://dx.doi.org/10.1016/j.patcog.2006.06.010. – ISSN 0031–3203

[Kaltenbrunner u. Bencina 2007] KALTENBRUNNER, Martin ; BENCINA, Ross: reacTIVision: a computer-vision framework for table-based tangible interaction. In: *Proceedings of the 1st international conference on Tangible and embedded interaction*. New York, NY, USA : ACM, 2007 (TEI '07). – ISBN 978–1–59593–619–6, 69–74

[Kaltenbrunner u. a. 2005] KALTENBRUNNER, Martin ; BOVERMANN, Till ; BENCINA, Ross ; COSTANZA, Enrico: TUIO: A Protocol for Table-Top Tangible User Interfaces. In: *In Proceedings of the 2 nd Interactive Sonification Workshop*, 2005

[Karam 2006] KARAM, Maria: *PhD Thesis: A framework for research and design of gesture-based human-computer interactions*, University of Southampton, Diss., October 2006. http://eprints.ecs.soton.ac.uk/13149/

[Kendo 1988] KENDO, A.: How gestures can become like words. In: *Crosscultural Perspectives in Nonverbal Communication* (1988), S. 131–141

[Kilan 2011] KILAN, Dennis: *Beispielfeatures des Algorithmus von Rubine*. Verified, 24.7.2011. http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master10-11-aw1/kilan/bericht.pdf. Version: 2011

[Kirishima u. a. 2005] KIRISHIMA, T. ; SATO, K. ; CHIHARA, K.: Real-time gesture recognition by learning and selective control of visual interest points. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 27 (2005), march, Nr. 3, S. 351 –364. http://dx.doi.org/10.1109/TPAMI.2005.61. – DOI 10.1109/TPAMI.2005.61. – ISSN 0162–8828

[Kjell Otto 2010] KJELL OTTO, Sören V.: Entwicklung einer Architektur für den Living Place Hamburg / HAW Hamburg. 2010. – Forschungsbericht

[Klaas u. a. 2006] KLAAS, Mike ; BRIERS, Mark ; FREITAS, Nando de ; DOUCET, Arnaud ; MASKELL, Simon ; LANG, Dustin: Fast particle smoothing: if I had a million particles. In: *Proceedings of the 23rd international conference on Machine learning*. New York, NY, USA : ACM, 2006 (ICML '06). – ISBN 1–59593–383–2, 481–488

[Koskela 2000] KOSKELA, Hille: 'The gaze without eyes': video-surveillance and the changing nature of urban space. In: *Progress in Human Geography* 24.2 (2000), S. 243–265. http://dx.doi.org/doi:10.1191/030913200668791096. – DOI doi: 10.1191/030913200668791096

[Kug] KUG: *Gesetz betreffend das Urheberrecht an Werken der bildenden Künste und der Photographie, §22*. Verified, 24.7.2011. http://www.gesetze-im-internet.de/kunsturhg/__22.html

[Kusznir u. Cook 2010] KUSZNIR, J. ; COOK, D.J.: Designing Lightweight Software Architectures for Smart Environments. In: *Intelligent Environments (IE), 2010 Sixth International Conference on*, 2010, S. 220 –224

[Lange 2000] LANGE, Robert: *3D Time-of-Flight Distance Measurement with Custom Solid-State Image Sensors in CMOS/CCD-Technology*, DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE AT UNIVERSITY OF SIEGEN, Diss., 2000. http://dokumentix.ub.uni-siegen.de/opus/volltexte/2006/178/pdf/lange.pdf

[Langheinrich 2005] LANGHEINRICH, Marc: *Personal Privacy in Ubiquitous Computing – Tools and System Support*. Zurich, Switzerland, ETH Zurich, Diss., Mai 2005

[Levine 1998] LEVINE, T.: Computer use, confidence, attitudes, and knowledge: A causal analysis. In: *Computers in Human Behavior* 14 (1998), Januar, Nr. 1, 125–146. http://dx.doi.org/10.1016/S0747-5632(97)00036-8. – DOI 10.1016/S0747–5632(97)00036–8. – ISSN 07475632

[Library of Congress 2009] LIBRARY OF CONGRESS: *Midas gold*. Verified, 14.8.2011. http://upload.wikimedia.org/wikipedia/commons/d/d6/Midas_gold2.jpg. Version: 2009

[Liu u. a. 2008] LIU, Yun ; GAN, Zhijie ; SUN, Yu: Static Hand Gesture Recognition and its Application based on Support Vector Machines. In: *Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, ACIS International Conference on* 0 (2008), S. 517–521. http://dx.doi.org/http://doi.ieeecomputersociety.org/10.1109/SNPD.2008.144. – DOI http://doi.ieeecomputersociety.org/10.1109/SNPD.2008.144. ISBN 978–0–7695–3263–9

[Lorenzen 2005] LORENZEN, Jonas: *Konturbasierte Gestenerkennung mit Hilfe der Dynamischen Programmierung*, HAW Hamburg, Diplomarbeit, 2005. http://users.informatik.haw-hamburg.de/projects/robotvision/Wissenschaftliche%20Dokumente/Abgeschlossene%20Abschlussarbeiten/JonasLorenzen_Diplomarbeit.pdf

[Lu u. a. 2011] LU, Ching-Hu ; WU, Chao-Lin ; FU, Li-Chen: A Reciprocal and Extensible Architecture for Multiple-Target Tracking in a Smart Home. In: *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* 41 (2011), jan., Nr. 1, S. 120 – 129. http://dx.doi.org/10.1109/TSMCC.2010.2051026. – DOI 10.1109/TSMCC.2010.2051026. – ISSN 1094–6977

[MacKenzie u. Ware 1993] MACKENZIE, I. S. ; WARE, Colin: Lag as a determinant of human performance in interactive systems. In: *Proceedings of the INTERACT '93 and CHI '93 conference on Human factors in computing systems*. New York, NY, USA : ACM, 1993 (CHI '93). – ISBN 0–89791–575–5, 488–493

[MacWilliams u. Sloane 1976] MACWILLIAMS, F.J. ; SLOANE, N.J.A.: Pseudo-random sequences and arrays. In: *Proceedings of the IEEE* 64 (1976), dec., Nr. 12, S. 1715 – 1729. http://dx.doi.org/10.1109/PROC.1976.10411. – DOI 10.1109/PROC.1976.10411. – ISSN 0018–9219

[Mattern u. Langheinrich 2001] MATTERN, Friedemann ; LANGHEINRICH, Marc: Allgegenwärtigkeit des Computers – Datenschutz in einer Welt intelligenter Alltagsdinge. In: MÜLLER, G. (Hrsg.) ; REICHENBACH, M. (Hrsg.): *Sicherheitskonzepte für das Internet, Springer-Verlag*. 2001, S. 7–26

[McNeill 1992] MCNEILL, David: *Hand and mind: What gestures reveal about thought*. University of Chicago Press, 1992

[Mesa Imaging AG 2010] MESA IMAGING AG: *SR4000*. http://www.mesa-imaging.ch/img/SR4000.png. Version: 2010

[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)), 267–277

[MobileRobots Inc 2011] MOBILEROBOTS INC: *MobileRobot C3D*. http://www.mobilerobots.com/ResearchRobots/Accessories/MobileRangerC3D.aspx. Version: 2011

[Moll 2011] MOLL, Wolfgang: *Mangelnder Schallschutz im Wohnungsbau*. Verified 28.8.2011. http://www.berliner-mieterverein.de/presse/sonstigesarchiv/urania08-schallschutz/moll.pdf. Version: 2011

[Morano u. a. 1998] MORANO, Raymond A. ; OZTURK, Cengizhan ; CONN, Robert ; DUBIN, Stephen ; ZIETZ, Stanley ; NISSANOV, Jonathan: Structured Light Using Pseudorandom Codes. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20 (1998), S. 322–327. http://dx.doi.org/http://doi.ieeecomputersociety.org/10.1109/34.667888. – DOI http://doi.ieeecomputersociety.org/10.1109/34.667888. – ISSN 0162–8828

[Morency u. a. 2003] MORENCY, Louis-Philippe ; RAHIMI, Ali ; DARRELL, Trevor: Adaptive View-Based Appearance Models. In: *In Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, 2003, S. 803–810

[Murakami u. Taguchi 1991] MURAKAMI, Kouichi ; TAGUCHI, Hitomi: Gesture recognition using recurrent neural networks. In: *Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology*. New York, NY, USA : ACM, 1991 (CHI '91). – ISBN 0–89791–383–3, 237–242

[Myers 1998] MYERS, Brad A.: A brief history of human-computer interaction technology. In: *interactions* 5 (1998), March, 44–54. http://dx.doi.org/http://doi.acm.org/10.1145/274430.274436. – DOI http://doi.acm.org/10.1145/274430.274436. – ISSN 1072–5520

[Neßelrath u. a. 2011] NESSELRATH, Robert ; LU, Chensheng ; SCHULZ, Christian H. ; FREY, Jochen ; ALEXANDERSSON, Jan: A Gesture Based System for Context-Sensitive Interaction with Smart Homes. In: *Advanced Technologies and Societal Change* (2011)

[Nielsen u. Pernice 2002] NIELSEN, Jacob ; PERNICE, Kara: *Usability for Senior Citizens*. http://www.useit.com/alertbox/seniors.html. Version: 2002

[Nielsen 1994] NIELSEN, Jakob: *Usability Engineering*. San Francisco, California : Morgan Kaufmann Publishers, 1994. – ISBN 0125184069

[Norman 2010] NORMAN, Donald A.: Natural user interfaces are not natural. In: *interactions* 17 (2010), May, 6–10. http://dx.doi.org/http://doi.acm.org/10.1145/1744161.1744163. – DOI http://doi.acm.org/10.1145/1744161.1744163. – ISSN 1072–5520

[Norman u. Nielsen 2010] NORMAN, Donald A. ; NIELSEN, Jakob: Gestural interfaces: a step backward in usability. In: *interactions* 17 (2010), September, 46–49. http://dx.doi.org/http://doi.acm.org/10.1145/1836216.1836228. – DOI http://doi.acm.org/10.1145/1836216.1836228. – ISSN 1072–5520

[(NREL) 1999] (NREL), National Renewable Energy L.: *Solar Spectral Irradiance: ASTM G-173*. checked 22.7.2011. http://rredc.nrel.gov/solar/spectra/am1.5/. Version: 1999

[Ogris u. a. 2005] OGRIS, G. ; STIEFMEIER, T. ; JUNKER, H. ; LUKOWICZ, P. ; TROSTER, G.: Using ultrasonic hand tracking to augment motion analysis based recognition of manipulative gestures. In: *Wearable Computers, 2005. Proceedings. Ninth IEEE International Symposium on*, 2005, S. 152 – 159

[OpenNI] OPENNI: *OpenNI UserGuide*. V3. OpenNI

[Patel 1995] PATEL, S.: A lower-complexity Viterbi algorithm. In: *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on* Bd. 1, 1995. – ISSN 1520–6149, S. 592 –595 vol.1

[Paul 2010] PAUL, Ryan: *FBI accused of planting backdoor in OpenBSD IPSEC stack*. Verified, 11.8.2011. http://arstechnica.com/open-source/news/2010/12/fbi-accused-of-planting-backdoor-in-openbsd-ipsec-stack.ars. Version: 2010

[Pavlovic u. a. 1997] PAVLOVIC, Vladimir I. ; SHARMA, Rajeev ; HUANG, Thomas S.: Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19 (1997), S. 677–695

[pcgames] PCGAMES: *Kinect: Microsoft verkauft 8 Millionen Xbox-360-Kameras in den ersten 60 Tagen*. Verified, 9.8.2011. http://www.pcgames.de/Kinect-Input-Device-Consoles-232538/News/Kinect-Microsoft-verkauft-8-Millionen-Xbox-360-Kameras-in-den-ersten-60-Tagen-806375/

[Plagemann u. a. 2010] PLAGEMANN, C. ; GANAPATHI, V. ; KOLLER, D. ; THRUN, S.: Real-time identification and localization of body parts from depth images. In: *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, 2010. – ISSN 1050–4729, S. 3108 –3113

[Point Grey Research Inc 2011] POINT GREY RESEARCH INC: *Bumlebee 2 Stereo Vision*. http://www.ptgrey.com/products/stereo.asp. Version: 2011

[Potratz 2011] POTRATZ, Olaf: *Ein System zur physikbasierten Interpretation von Gesten im 3D-Raum*. Bachelorarbeit HAW Hamburg. http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/bachelor/potratz.pdf. Version: 04 2011

[PrimeSense Ltd 2010] PRIMESENSE LTD: *PrimeSensor Block Diagram*. http://www.primesense.com/?p=514. Version: 2010

[Public record office, London 1943] PUBLIC RECORD OFFICE, LONDON: *Colossus Mark II*. http://de.wikipedia.org/w/index.php?title=Datei:Colossus.jpg&filetimestamp=20090301192445. Version: 1943

[Rahman u. a. 2009] RAHMAN, ASM M. ; HOSSAIN, M. A. ; PARRA, Jorge ; EL SADDIK, Abdulmotaleb: Motion-path based gesture interaction with smart home services. In: *Proceedings of the 17th ACM international conference on Multimedia*. New York, NY, USA : ACM, 2009 (MM '09). – ISBN 978–1–60558–608–3, 761–764

[Raskin 1994] RASKIN, Jef: Viewpoint: Intuitive equals familiar. In: *Commun. ACM* 37 (1994), September, 17–18. http://dx.doi.org/http://doi.acm.org/10.1145/182987.584629. – DOI http://doi.acm.org/10.1145/182987.584629. – ISSN 0001–0782

[Rauterberg u. Steiger 1996] RAUTERBERG, Matthias ; STEIGER, Patrick: Pattern Recognition as a Key Technology for the Next Generation of User Interfaces. In: *In Proc. of IEEE International Conference on Systems, Man and Cybernetics–SMC'96 (Vol. 4, IEEE Catalog Number: 96CH35929*, Piscataway: IEEE, 1996, S. 2805–2810

[Rehm u. a. 2008] REHM, Matthias ; BEE, Nikolaus ; ANDRÉ, Elisabeth: Wave Like an Egyptian — Accelerometer Based Gesture Recognition for Culture Specific Interactions. In: *Procedings of HCI 2008 Culture, Creativity, Interaction*, 2008

[Reichinger 2011] REICHINGER, Andreas: *kinect pattern uncoverd*. Verified, 29.6.2011. http://azttm.wordpress.com/2011/04/03/kinect-pattern-uncovered/. Version: 2011

[Research 2011] RESEARCH, Microsoft: *Skeletal Viewer Walkthrough: C++ and C#*. Microsoft Corporation, 2011

[Rheingold 1991] RHEINGOLD, Howard: *Virtual reality*. Secker & Warburg, London, 1991. – 415 s S.

[Rocchini u. a. 2001] ROCCHINI, C. ; CIGNONI, P. ; MONTANI, C. ; PINGI, P. ; SCOPIGNO, R.: A low cost 3D scanner based on structured light. In: *Computer Graphics Forum* 20 (2001), Nr. 3, 299–308. http://dx.doi.org/10.1111/1467-8659.00522. – DOI 10.1111/1467–8659.00522. – ISSN 1467–8659

[Rohde 2007] ROHDE, Robert A.: *Solar Spectrum*. Verified, 29.6.2011. http://en.wikipedia.org/wiki/File:Solar_Spectrum.png. Version: 2007

[Rubine 1991] RUBINE, Dean: Specifying gestures by example. In: *Proceedings of the 18th annual conference on Computer graphics and interactive techniques*. New York, NY, USA : ACM, 1991 (SIGGRAPH '91). – ISBN 0–89791–436–8, 329–337

[Rödiger 2010] RÖDIGER, Marcus: *Interaktive Steuerung von Computersystemen mittels Erkennung von Körpergesten*, HAW Hamburg, Diplomarbeit, 2010

[Saffer 2010] SAFFER, Dan: *Why You Want (But Won't Like) a Minority Report-style Interface*. Verified 20.8.2011. http://www.kickerstudio.com/blog/2010/11/why-you-want-but-wont-like-a-minority-report-style-interface/. Version: 2010

[Sakoe 1978] SAKOE, Hiroaki: Dynamic programming algorithm optimization for spoken word recognition. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 26 (1978), S. 43–49

[Salvadore u. Chan 2004] SALVADORE, S. ; CHAN, P.: FastDTW: Toward accurate dynamic time warping in linear time and space. In: *3rd Workshop on Mining Temporal and Sequential Data*, 2004

[Schaar 2007] SCHAAR, Peter: *Das Ende der Privatsphäre: Der Weg in die Überwachungsgesellschaft*. C. Bertelsmann, 2007

[Schmallenberg 2006] SCHMALLENBERG, Stefan D.: *Yamaha DJX*. Verified, 24.7.2011. http://upload.wikimedia.org/wikipedia/commons/d/db/Yamaha_DJX.jpg. Version: 2006

[Schmidt 1988] SCHMIDT, Richard A.: *Motor Control and Learning: A Behavioral Emphasis*. Human Kinetics (Champaign, Ill.), 1988

[Seligmann 2003] SELIGMANN, Jacob: *IWC GST ref. 3707 - dial*. Verified, 24.7.2011. http://upload.wikimedia.org/wikipedia/commons/0/02/IWC_GST_ref._3707_-_dial.jpg. Version: 2003

[Senior u. a. 2005] SENIOR, A. ; PANKANTI, S. ; HAMPAPUR, A. ; BROWN, L. ; TIAN, Ying-Li ; EKIN, A. ; CONNELL, J. ; SHU, Chiao F. ; LU, M.: Enabling video privacy through computer vision. In: *Security Privacy, IEEE* 3 (2005), may-june, Nr. 3, S. 50 – 57. http://dx.doi.org/10.1109/MSP.2005.65. – DOI 10.1109/MSP.2005.65. – ISSN 1540–7993

[Senkbeil 2005] SENKBEIL, Martin: *Entwicklung eines Systems zur Programmsteuerung mit Hilfe von Interpretation visueller Gesten*, HAW Hamburg, Diplomarbeit, 2005. http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/diplom/senkbeil.pdf

[Shpunt u. a. 2010] SHPUNT, Alexander ; FREEDMAN, Barak ; MEIR, Machline ; ARIELI, Youel: *Depth Mapping using projected patterns*. http://www.freepatentsonline.com/20100118123.pdf. Version: May 2010

[Shpunt 2009] SHPUNT, IL) Alexander (Petach-Tikva: *OPTICAL DESIGNS FOR ZERO ORDER REDUCTION*. http://www.freepatentsonline.com/y2009/0185274.html. Version: July 2009

[Signer u. a. 2007] SIGNER, B. ; KURMANN, U. ; NORRIE, M.C.: iGesture: A General Gesture Recognition Framework. In: *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on* Bd. 2, 2007. – ISSN 1520–5363, S. 954 –958

[Smith u. Milberg 1996] SMITH, H. J. ; MILBERG, Sandra J.: Information privacy: measuring individuals' concerns about organizational practices. In: *MIS Q.* 20 (1996), June, 167–196. http://dx.doi.org/10.2307/249477. – DOI 10.2307/249477. – ISSN 0276–7783

[Spiekermann u. Cranor 2009] SPIEKERMANN, S. ; CRANOR, L.F.: Engineering Privacy. In: *Software Engineering, IEEE Transactions on* 35 (2009), jan.-feb., Nr. 1, S. 67 –82. http://dx.doi.org/10.1109/TSE.2008.88. – DOI 10.1109/TSE.2008.88. – ISSN 0098–5589

[Stahlkocher 2006] STAHLKOCHER: *Not-aus Betätiger*. Verified, 24.7.2011. http://upload.wikimedia.org/wikipedia/commons/f/f9/Not-Aus_Bet%C3%A4tiger.jpg. Version: 2006

[Stark u. a. 1995] STARK, M. ; KOHLER, M. ; ZYKLOP, P. G.: Video Based Gesture Recognition for Human Computer Interaction / Informatik VII, University of Dortmund. 1995. – Forschungsbericht

[StGB a] STGB: *StGB §201*. Verified, 24.7.2011. http://dejure.org/gesetze/StGB/201.html

[StGB b] STGB: *StGB §201a*. Verified, 24.7.2011. http://www.gesetze-im-internet.de/stgb/__201a.html

[Störing 2011] STÖRING, Marc: *Smart Home - Rechtskonforme Gestaltung eines intelligent vernetzten Haushalts*. http://www.smartlife2011.de/vortraege/symposium2011stoering.pdf. Version: 4 2011

[Suma u. a. 2011] SUMA, E.A. ; LANGE, B. ; RIZZO, A. ; KRUM, D.M. ; BOLAS, M.: FAAST: The Flexible Action and Articulated Skeleton Toolkit. In: *Virtual Reality Conference (VR), 2011 IEEE*, 2011. – ISSN 1087–8270, S. 247 –248

[Sutherland 1964] SUTHERLAND, Ivan E.: Sketch pad a man-machine graphical communication system. In: *Proceedings of the SHARE design automation workshop*. New York, NY, USA : ACM, 1964 (DAC '64), 6.329–6.346

[Tamura u. Kawasaki 1988] TAMURA, S. ; KAWASAKI, S.: Recognition of sign language motion images. In: *Pattern Recogn.* 21 (1988), July, 343–353. http://dx.doi.org/10.1016/0031-3203(88)90048-9. – DOI 10.1016/0031–3203(88)90048–9. – ISSN 0031–3203

[Taylor u. a. 2001] TAYLOR, Russell M. II ; HUDSON, Thomas C. ; SEEGER, Adam ; WEBER, Hans ; JULIANO, Jeffrey ; HELSER, Aron T.: VRPN: a device-independent, network-transparent VR peripheral system. In: *Proceedings of the ACM symposium on Virtual reality software and technology*. New York, NY, USA : ACM, 2001 (VRST '01). – ISBN 1–58113–427–4, 55–61

[Tsalakanidou u. a. 2005] TSALAKANIDOU, Filareti ; FORSTER, Frank ; MALASSIOTIS, Sotiris ; STRINTZIS, Michael G.: Real-time acquisition of depth and color images using structured light and its application to 3D face recognition. In: *Real-Time Imaging* 11 (2005), October, 358–369. http://dx.doi.org/http://dx.doi.org/10.1016/j.rti.2005.06.006. – DOI http://dx.doi.org/10.1016/j.rti.2005.06.006. – ISSN 1077–2014

[User:Bumm13 2006] USER:BUMM13: *Pong*. Verified, 24.7.2011. http://upload.wikimedia.org/wikipedia/commons/f/f8/Pong.png. Version: 2006

[User:Steevven1 2007] USER:STEEVVEN1: *OK-sign*. Verified, 24.7.2011. http://upload.wikimedia.org/wikipedia/en/1/19/OK_Sign.jpg. Version: 2007

[Viterbi 1967] VITERBI, A.: Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. In: *Information Theory, IEEE Transactions on* 13 (1967), apr, Nr. 2, S. 260 – 269. http://dx.doi.org/10.1109/TIT.1967.1054010. – DOI 10.1109/TIT.1967.1054010. – ISSN 0018–9448

[Wachs u. a. 2011] WACHS, Juan P. ; KÖLSCH, Mathias ; STERN, Helman ; EDAN, Yael: Vision-based hand-gesture applications. In: *Commun. ACM* 54 (2011), February, 60–71. http://dx.doi.org/http://doi.acm.org/10.1145/1897816.1897838. – DOI http://doi.acm.org/10.1145/1897816.1897838. – ISSN 0001–0782

[Weik 2006] WEIK, Martin: *Computer Science and Communications Dictionary*. Secaucus, NJ, USA : Springer-Verlag New York, Inc., 2006. – ISBN 0387335560

[Weiser 1995] WEISER, Mark: Human-computer interaction. Version: 1995. http://wiki.daimi.au.dk/pca/_files/weiser-orig.pdf. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 1995. – ISBN 1–55860–246–1, Kapitel The computer for the 21st century, 933–940

[Wobbrock u. a. 2007] WOBBROCK, Jacob O. ; WILSON, Andrew D. ; LI, Yang: Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes. In: *Proceedings of the 20th annual ACM symposium on User interface software and technology*. New York, NY, USA : ACM, 2007 (UIST '07). – ISBN 978–1–59593–679–0, 159–168

[Wu u. Huang 1999] WU, Ying ; HUANG, Thomas S.: Vision-Based Gesture Recognition: A Review. In: *Proceedings of the International Gesture Workshop on Gesture-Based Communication in Human-Computer Interaction*. London, UK : Springer-Verlag, 1999 (GW '99). – ISBN 3–540–66935–3, 103–115

[Wydra 2008] WYDRA, Thomas: *fernbedienung*. Verified, 28.7.2011. http://upload.wikimedia.org/wikipedia/de/5/5b/IR_Remote_control.jpg. Version: 2008

[Wöllmer u. a. 2009] WÖLLMER, Martin ; AL-HAMES, Marc ; EYBEN, Florian ; SCHULLER, Björn ; RIGOLL, Gerhard: A multidimensional dynamic time warping algorithm for efficient multimodal fusion of asynchronous data streams. In: *Neurocomput.* 73 (2009), December, 366–380. http://dx.doi.org/10.1016/j.neucom.2009.08.005. – DOI 10.1016/j.neucom.2009.08.005. – ISSN 0925–2312

[Xiaphias 2007] XIAPHIAS: *Bowlerbowling*. Verified, 24.7.2011. http://upload.wikimedia.org/wikipedia/commons/4/4b/Bowlerbowling.JPG. Version: 2007

[Xu u. a. 2009] XU, Yishen ; GU, Jihua ; TAO, Zhi ; WU, Di: Bare Hand Gesture Recognition with a Single Color Camera. In: *Image and Signal Processing, 2009. CISP '09. 2nd International Congress on*, 2009, S. 1 –4

Glossar

Cloud Cloud bzw. Cloud Computing (Rechnerwolke) beschreibt den Ansatz, Rechnerkapazität oder Datenspeicher dynamisch über ein Netzwerk zur Verfügung zu stellen. Dem Benutzer ist dabei nicht klar, wo sich seine Daten befinden, aus seiner Sicht undurchsichtig in einer 'Wolke'.

FPGA Field Programmable Gate Array - ist ein Integrierter Schaltkreis (IC) der Digitaltechnik, in den ein Alogrithmus in eine logische Schaltung programmiert werden kann. Dies ermöglicht die Bearbeitung dieser Algorithmen in Hardware und entsprechende Geschwindigkeitsvorteile.

Immersion Immersion beschreibt den Zustand des "Eintauchens" in eine virtuelle Realität (Spiel, Film), die dazu führt, dass eine Person denkt, sie wäre Teil des Geschehens.

iPhone Das iPhone ist ein Smartphone der Firma Apple.

Iso Standard 24752 Der Iso-Standard 24752 definiert Steuerung von Diensten und elektronischen Geräten durch Fernsteuerung, alternative Interfaces und intelligente Agenten.

Lightpen Lightpen, im deutschen 'Lichtgriffel' genannt, ist ein Eingabegerät, dass aus einem Stift mit einer Photodiode am Ende besteht, und damit das Auftreffen des Kathodenstrahls eines Röhrenmonitors registrtiert. Dadurch lässt sich die Position auf dem Bildschirm bestimmen.

Modalität Eine Modalität ist ein ein Eingabekanal in der HCI, also beispielsweise Sprache, Gestik, Tastatur, Maus.

Reverse Engineering Reverse Engineering bezeichnet den Vorgang, aus einem bestehenden, fertigen System durch Untersuchung der Strukturen, Zustände und Verhaltensweisen, die Konstruktionselemente zu extrahieren.

Smartphone Ein Smartphone ist ein Mobiltelefon, das eher die Funktionalität eines Computers als nur eines Telefons bietet.

SOC System on a Chip - versteht man die Integration aller oder eines großen Teils der Funktionen eines Systems auf einem Chip.

tty *text terminal type*, ein textbasiertes Interface zur Kommunikation mit dem Computer.

Ubiquitous Computing Ubiquitous Computing (auch ubicomp) bzw. Rechnerallgegenwart, gelegentlich allgegenwärtiges (ubiquitäres) Rechnen, bezeichnet die Allgegenwärtigkeit (Ubiquität, engl. ubiquity) der rechnergestützten Informationsverarbeitung.