



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Joachim Boetzer

Bewegungs- und gestenbasierte
Applikationssteuerung auf Basis eines Motion
Trackers

Joachim Boetzer

Bewegungs- und gestenbasierte
Applikationssteuerung auf Basis eines Motion
Trackers

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang Technische 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 v. Luck

Abgegeben am 11. August 2008

Joachim Boetzer

Thema der Bachelorarbeit

Bewegungs- und gestenbasierte Applikationssteuerung auf Basis eines Motion-Trackers

Stichworte

Motion Tracker, Motion Capture, Gestenerkennung, Bewegungsverfolgung, Powerwall

Kurzzusammenfassung

Im Kontext von Mensch Maschine Interaktion (HCI) und nahtloser Interaktion (Seamless Interaktion) wird untersucht, inwieweit sich verschiedene Arten von Anwendungen durch Gesten im Raum steuern lassen. Schwerpunkt sind hierbei Anwendungen, die auf hochauflösenden Visualisierungssystemen von mehreren Personen kollaborativ genutzt werden können und welche Arten von Gesten sich zur Steuerung solcher Anwendungen eignen. Im Rahmen dieser Arbeit wurde eine Gestenerkennung auf Basis eines Motion Trackers implementiert mit der sich eine solche Steuerung umsetzen lässt, und verschiedene Arten von Gesten auf ihre Nutzbarkeit hin untersucht.

Joachim Boetzer

Title of the Bachelor Thesis

Motion and gesture based application control through a motion tracker

Keywords

Motion Tracker, Motion Capture, Gesture Recognition, Motion tracking, Powerwall

Abstract

This paper deals with the research investigation in how far different kinds of applications can be controlled through gestures in an certain area within the context of human computer interaction (HCI) and seamless interaction. Emphasis in this context is laid on applications which can be used collaboratively by a number of persons with high-resolution visualizing systems and which kinds of gestures of such applications are suitable for the given purpose. In this paper gesture recognition was implemented on the basis of a motion tracker by which such a controlling can be realized and also different kinds of gestures were examined on their usability.

Inhaltsverzeichnis

1	Einführung	7
1.1	Motivation	7
1.2	Gliederung der Arbeit	9
2	Grundlagen	10
2.1	Mensch-Maschine-Interaktion (HCI)	10
2.1.1	Nahtlose Interaktion (Seamless Interaction)	11
2.2	Powerwall	11
2.3	Motion Tracking	12
2.3.1	Technologien	12
	Elektromagnetisch	12
	Elektromechanisch	13
	Akustisch	13
	Inertial	13
	Optisch	14
2.3.2	Einsatzgebiete	14
2.4	Anwendungssteuerung mit Motion Tracking	15
2.4.1	Einsatzgebiete	16
	Konstruktions- und Designstudios	16
	Spiele- und Entertainmentindustrie	17
3	Analyse	19
3.1	Szenarien: Anwendungssteuerung mit Gesten und Bewegungen	20
3.1.1	Szenario I: Spiel	21
3.1.2	Szenario II: Applikation	21
3.2	Funktionale Anforderungen	22
3.2.1	Anwendungsfälle	22
	Spiel	22
	Modell	23
3.2.2	Interaktionstechniken	24
3.3	Nicht-funktionale Anforderungen	25
3.3.1	Intuitive Bedienbarkeit	25
3.3.2	Performance	25

3.3.3	Übertragbarkeit	25
3.3.4	Mehrbenutzerfähigkeit	26
3.4	Fazit	26
4	Gesten	27
4.1	Begriffsklärung	27
4.2	Bestehende Klassifizierung von Gesten	28
4.3	Kategorisierung von manipulativen Gesten	29
4.3.1	Bewegungsverfolgung	29
4.3.2	Kontinuierliche Gesten	30
4.3.3	Symbolisch-manipulative Gesten	31
4.4	Anfang und Ende von Gesten	31
4.5	Gesten im Anwendungskontext	32
4.6	Gesten im dreidimensionalen Raum	32
5	Design	34
5.1	Der Interaktive Raum	34
5.1.1	Powerwall	34
5.1.2	Motiontracker	35
5.2	Beispielanwendungen	36
5.2.1	Kommunikation Tracker - Software	37
5.2.2	Spiel: Squash	38
5.2.3	Anwendungen: Google Earth und Google Maps	40
Google Earth	40	
Google Maps	41	
5.2.4	Softwarearchitektur (MVC)	42
5.2.5	Gestenerkennung	43
6	Realisierung	46
6.1	Tools und Frameworks	46
6.1.1	Java	46
Processing	46	
JACOB (Java COM-Bridge)	47	
6.1.2	JSON (JavaScript Object Notation)	47
6.2	Technische Umsetzung	47
6.2.1	Trackeransteuerung	48
6.2.2	Gestenerkennung	48
6.2.3	Spiel: Squash	50
6.2.4	Google Applikationen (Google Earth und Google Maps)	52
Google Earth Anbindung	54	
Google Maps Anbindung	55	

6.3 Anmerkungen zur Umsetzung	55
7 Zusammenfassung	57
7.1 Fazit	58
7.2 Ausblick	58
Literaturverzeichnis	60

1 Einführung

Inspiziert durch Mark Weiser ([Weiser \(1991\)](#)) wird seit 2002 auch an der HAW Hamburg über das Thema ubiquitous computing nachgedacht. Eines der Leitthemen sind „handbuchfreie Computer“ als konkrete Interpretation von disappearing computing. In diesem Zusammenhang ist auch die intuitive Bedienbarkeit von Technik, die auch als seamless interaction (siehe u.a. [Ishii u. a. \(1994\)](#)) bekannt ist, von großen Interesse. Die spezifischen Eigenschaften von Eingabegeräten haben dabei einen entscheidenden Einfluss auf die Entwicklung von Interaktionstechniken für Computersysteme.

Dabei ist auch zu beachten, dass mittlerweile ein einzelner Nutzer über mehrere Computersysteme, an seiner Arbeitsstelle und privat, verfügt. [Pierce und Nichols](#) sprechen hier von der Entwicklung weg vom *Personal Computer*, hin zum *Personal Information Environment*. Dies beschreibt eine Informationsumgebung, bei der die Daten des Anwenders über eine heterogene Gerätemenge verteilt sind.

Analog zu dieser Entwicklung sind in den letzten Jahren gemeinschaftlich genutzte, kollaborative Informationsumgebungen, so genannte Collaborative Workspaces ins Zentrum der Forschung gerückt. Diese neuartigen Arbeitsräume erlauben die Zusammenarbeit mehrerer Personen an unterschiedlichen Computersystemen.

Im diesem Kontext ist es sinnvoll, neue Technologien auf Ihre Eignung als Eingabemedium für Computersysteme hin zu untersuchen die diesen neuen Anforderungen gerecht werden können. Hierbei sind Eingabegeräte, die in der Lage sind natürliche, aus der realen Welt bekannte, Interaktionstechniken zu adaptieren, von besonderem Interesse.

1.1 Motivation

Mit dem Aufkommen von neuen intuitiven Eingabemöglichkeiten fernab der traditionellen Maus und Tastatur wie z. B. Multitouch Oberflächen oder der Nintendo Wii sind die Möglichkeiten der Interaktion mit Computersystemen deutlich gewachsen.

Diese neuen Entwicklungen haben auch Hollywood motiviert sich mit Wissenschaftlern aus unterschiedlichen Forschungsgebieten zusammenzuarbeiten, um realistische Visionen des Alltags der Zukunft zu entwickeln. Durch diese Zusammenarbeit entstehen dann Werke wie



Abbildung 1.1: Interface der Zukunft in Minority Report (Twentieth Century-Fox)

z. B. *Minority Report*, von Regisseur Steven Spielberg in Zusammenarbeit mit dem MIT¹, der eine Zukunftsvision zeigt, in der unter anderem die Steuerung von Computersystemen über Gesten angenommen wird (siehe Abb. 1.1). Und so wie Hollywood sich vom aktuellen Stand der technischen Entwicklung inspirieren lässt, so nutzt auch umgekehrt die Forschung die bildgewaltigen Werke aus Hollywood zur Anregung und stellt Überlegungen an, wie sich diese Visionen umsetzen lassen.

Allerdings fehlen in der Realität noch ausgereifte Konzepte, die eine bequeme, intuitive Bedienung über mehrere Anwendungen hinweg möglich machen. Speziell im Bereich der Steuerung durch Gesten und Bewegungen bestehen zwar einige Ansätze in der Industrie und der Entertainment Branche, die aber sehr speziell für einzelne Anwendungsgebiete konzipiert sind und sich nur sehr schwer verallgemeinern lassen.

Gerade in Anwendungsgebieten, in denen eine große Datenmenge visualisiert und mit dieser Visualisierung interagiert werden soll, wie z. B. in den Konstruktionsbüros der Auto- und Flugzeugindustrie, fehlen diese Konzepte zur Interaktion mehrerer kollaborativ arbeitender Ingenieure mit einem hochauflösenden Anzeigemedium.

Deshalb sind grundlegende Überlegungen notwendig, die sich damit beschäftigen, was für Bewegungen und Gesten sich für eine Steuerung von Computersystemen eignen und wie diese Gesten performant von bestehender Technologie erkannt werden können.

In diesem Zusammenhang wurde eine Gesten- und Bewegungssteuerung für Applikationen entwickelt, die komplexe Datenbestände auf einem skalierbaren Visualisierungssystem darstellen und diese manipulieren kann.

¹<http://web.mit.edu/newsoffice/2002/underkoffler-0717.html>

1.2 Gliederung der Arbeit

Im Anschluss an dieses Kapitel werden im Kapitel 2 zuerst einige grundsätzliche Überlegungen zur Interaktion von Mensch und Maschine angestellt. Danach werden kurz die verfügbaren und verwendeten Technologien erläutert und im Anschluss daran werden exemplarisch einige bestehende Steuerungsansätze auf der Basis von Motion Tracking vorgestellt.

Im Analyse Kapitel (3) werden zuerst zwei mögliche Szenarien für eine gestenbasierte Steuerung vorgestellt und daraus dann verschiedene Interaktionsmöglichkeiten entwickelt. Danach werden einige funktionale und nicht-funktionale Anforderungen definiert die eine Steuerung erfüllen muss um sowohl einen sinnvollen Funktionsumfang zu bieten als auch leicht bedienbar zu sein.

Im Kapitel 4 wird zuerst erläutert, was eine Geste eigentlich ist und welche grundlegenden Probleme bei der Interpretation von Gesten auftreten können. Danach werden dann einige bestehenden Ansätze zur Einteilung von Gesten erläutert und diese dann um einige eigene Überlegungen erweitert. Danach werden einige Probleme bei der computergestützten Erkennung von Gesten angesprochen und die Ausprägung von Gesten im dreidimensionalen Raum untersucht.

Im Design Kapitel (5) werden am Anfang die Rahmenbedingungen der bestehenden Laborumgebung besprochen und danach auf die benötigten Module zur Umsetzung der Anforderungen in der Analyse im Detail eingegangen. Hierbei wird zuerst die Anbindung an den Motion Tracker behandelt und danach auf die Einzelanwendungen und deren Spezialitäten eingegangen. Hier ist vor allem die gewählte Softwarearchitektur und die geplante Umsetzung der Gestenerkennung von Interesse. Desweiteren wird die Anbindung an die einzelnen externen Anwendungen erwähnt.

In der Realisierung (Kapitel 6) werden die genutzten Tools vorgestellt und dann die technische Umsetzung im Detail erläutert. Auch hier ist die realisierte Umsetzung der Gestenerkennung von besonderem Interesse. Desweiteren werden die Besonderheiten der einzelnen Umsetzungen für die verschiedenen Anwendungen erklärt und im Abschluss auf die Vorteile und Beschränkungen der gewählten Umsetzung eingegangen.

Im letzten Kapitel (7) werden die Erfahrungen mit der erstellten Ansteuerung zusammengefasst und ein kurzer Ausblick auf Verbesserungsmöglichkeiten und die vermutete Entwicklung dieser Eingabemethode gegeben.

2 Grundlagen

In diesem Kapitel werden kurz die Grundlagen vorgestellt, die für das Verständnis dieser Arbeit notwendig sind. Zunächst werden im Kapitel 2.1 die Grundideen der Forschungsgebiete zu optimalen Mensch-Maschine-Interaktionen erläutert. Dann in Kapitel 2.2 wird eine kurze Begriffsklärung vorgenommen, was eine Powerwall ist und für welche Einsatzgebiete diese Technologie geeignet ist. In Kapitel 2.3 werden bestehende Technologien beschrieben, die in der Lage sind Gesten und Bewegungen zu erkennen und in Kapitel 2.4 werden einige Bereiche genannt, die diese Technologien bereits erfolgreich einsetzen.

2.1 Mensch-Maschine-Interaktion (HCI)

Leider existiert im Moment keine allgemein gültige Definition was Mensch-Maschine-Interaktion alles umfasst. Um trotzdem eine Arbeitsgrundlage zu haben, bedienen wir uns hier der vorläufigen Definition der [ACM SIGCHI](#):

Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them.

Aus Sicht der Informatik liegt der Schwerpunkt auf der Interaktion von einem oder mehreren Benutzern mit einem oder mehreren Computersystemen. Wobei Computersystem alles von einem kleinen eingebetteten System in der Armbanduhr bis zum Supercomputer in einem Forschungszentrum umfasst. Ein Kernpunkt der Forschung¹²³ im Bereich HCI ist das Entwickeln und Verbessern von Computereingabeschnittstellen, um die Bedienung für den Menschen einfacher und intuitiver zu gestalten. Ein wichtiges Stichwort in diesem Zusammenhang ist die *seamless interaktion* also die barrierefreie Interaktion mit einem beliebigen Computersystem.

¹<http://www.hcii.cmu.edu/>

²<http://hci.stanford.edu/>

³<http://web.media.mit.edu/>

2.1.1 Nahtlose Interaktion (Seamless Interaction)

Nahtlose Interaktion beschreibt den möglichst einfachen und intuitiven Umgang mit Computersystemen auch über mehrere unterschiedliche Systeme hinweg. Geringer Lernaufwand (Handbuchfreies Arbeiten) und an bereits Bekanntes angepasste Arbeitsabläufe sollen das Arbeiten an Computersystemen und das Wechseln zwischen verschiedenen Systemen erleichtern.

Nahtlosigkeit besteht im wesentlichen aus zwei Bestandteilen. Zum einen aus sogenannten „smooth transitions“, welche für einen nahtlosen Übergang zwischen unterschiedlichen Arbeitsumfeldern stehen, die bei kollaborativen Umgebungen häufig zu finden sind. Die Nahtlose Interaktion entlastet den Benutzer, indem es ihm ermöglicht wird, ohne sich an eine andere Eingabephilosophie anpassen zu müssen, dynamisch zwischen Arbeitsumgebungen zu wechseln.

Zum anderen besteht die Nahtlosigkeit aus der „continuity“, die für eine gleichbleibende Benutzung aller eingesetzten Arbeitsräume steht. Menschen entwickeln ihre eigenen Arbeitspraktiken bei der Benutzung von vielen unterschiedlichen Tools während sie mit anderen Menschen interagieren. Also ist es wichtig, die Kontinuität von bestehenden Praktiken und Fertigkeiten des Alltags zu wahren (Ishii u. a. (1994)).

2.2 Powerwall

In vielen Anwendungsbereichen aus Industrie und Wissenschaft ist die dreidimensionale Visualisierung von großen Informationsmengen notwendig, die einen hohen Detaillierungsgrad erfordern. Beispiele dafür finden sich unter anderem in der Konstruktion und Planung von großen Gebäudeanlagen (z. B. Fabriken) oder in der Konstruktion von Autos oder Flugzeugen.

Für diese Ansprüche an die Auflösung der Darstellung sind normale Anzeigemedien wie Monitore oder Beamer häufig nicht ausreichend. Aus diesem Grund werden für diesen Zweck skalierbare Visualisierungssysteme (Robert (2006)) eingesetzt. Eine Powerwall ist ein solches skalierbares Visualisierungssystem.

Powerwalls bestehen in der Regel aus einem Cluster von Rechnern in dem jeder Knoten (Rechner + Ausgabemedium) für die Visualisierung eines Teilbereichs der darzustellenden Bildpunkte zuständig ist. Auf diese Weise können mehrere Ausgabemedien zu einem gemeinsamen, hochauflösenden Ausgabemedium zusammengeschlossen werden. Während die Realisierung von hochauflösenden Powerwalls vor einigen Jahren aufgrund spezieller Hardwareanforderungen noch sehr kostenintensiv war, ermöglichen heutzutage softwarebasierte skalierbare Visualisierungsstrategien eine verhältnismäßig günstige Umsetzung mit Standardkomponenten (Bezerianos und Balakrishnan (2005)). Dies führte dazu, dass in den



Abbildung 2.1: 5x4 Monitor Powerwall ([University of Alaska Fairbanks](#))

letzten Jahren der Einsatz von Powerwalls verstärkt ins Interesse der Forschung gerückt ist.⁴

2.3 Motion Tracking

Motion Tracking ist eine Technik, mit der man reale Bewegungen erfassen und in eine digitale Repräsentation umwandeln kann.

Motion Tracking is the process of recording a live motion event and translating it into usable mathematical terms by tracking a number of key points in space over time ([Menache \(2000\)](#)).

Motion Tracking ist mit einer Vielzahl unterschiedlicher Technologien möglich.

2.3.1 Technologien

Elektromagnetisch

Elektromagnetische Systeme (siehe [Abb. 2.2](#)) strahlen Elektromagnetische Felder niedriger Frequenz aus, die von Sensoren, die die zu verfolgende Person trägt, gemessen werden.

⁴siehe auch [Napitupulu \(2008\)](#)

Diese Daten werden dann an einen Rechner gesendet, der aus den Feldstärken die Position und Ausrichtung der Sensoren errechnet. Diese Technik erlaubt eine Erfassung nahezu in Echtzeit von Bewegungen und ist unempfindlich gegen Verdeckungen der Sensoren. Allerdings können schon sehr geringe Magnetfelder die Erfassung der Daten erheblich stören.

Elektromechanisch

Elektromechanische Systeme gleichen großen Außenskeletten aus Plastik und Metall. (siehe Abb. 2.3) Dieses System ist Echtzeitfähig und ist sehr robust gegen elektrische oder optische Störungen. Allerdings ist es sehr unflexibel und schwer und schränkt deshalb den Nutzer in seinen motorischen Möglichkeiten ein.



Abbildung 2.2: [Ascension](#) MotionStar Electromagnetic System



Abbildung 2.3: [Meta Motion](#) Gypsy Electromechanic System

Akustisch

Bei einem akustischen System emittiert ein Sender Ultraschallwellen und die Empfänger am Körper des Nutzers messen entweder die Phasenkohärenz oder die Zeit, die der Schall zum Zurücklegen der Strecke zum Empfänger gebraucht hat. Dieses System ist relativ preiswert, allerdings ist es empfindlich gegenüber Hintergrundgeräuschen und Reflektionen. Außerdem werden für ein genaues Ergebnis relativ viele Sender und Empfänger benötigt.

Inertial

Bei inertial Systemen messen Beschleunigungssensoren und Gyroskope die Bewegungsrichtung und Stärke. Ein Rechner addiert die einzelnen Bewegungsvektoren auf und ermittelt daraus die Position im Raum. Dieses System ist unempfindlich gegenüber Störungen von

außen, allerdings ergibt sich durch das Aufsummieren der Daten ein langsam immer größer werdender Fehler.



Abbildung 2.4: Die Wii Remote nutzt unter anderem Bewegungssensoren

Optisch

Optische Systeme verwenden zwei oder mehr Kameras, mit denen der Benutzer gefilmt wird. Der Benutzer trägt dabei reflektierende Marker, anhand derer die Kamera Punkte auf einer zweidimensionalen Ebene zuordnen kann. Aus den Bildern aller Kameras zusammen errechnet dann eine Software die Position und Ausrichtung der einzelnen Markierungen. Bei Infrarot Systemen senden die Kameras zusätzlich noch Infrarotimpulse aus die von den Markern reflektiert werden. Solche Systeme können echtzeitfähig sein und besitzen eine sehr hohe Präzision, allerdings ist der Rechenaufwand zur Bestimmung der Position recht hoch und das System ist anfällig für Verdeckungen der Marker.

2.3.2 Einsatzgebiete

Motion Tracking wird für eine Vielzahl von Anwendungen genutzt:

- Biomedizinische Forschung über Bewegungsabläufe
- Bewegungsanalyse im professionellen Sport, um z.B. den Hürdensprung oder den Golfschlag zu optimieren
- In der Filmindustrie, um Computer generierten Figuren realistische Bewegungen zu verschaffen (s.a. Abb. 2.7)



Abbildung 2.5: Vicon Infrarot Tracking Kamera



Abbildung 2.6: ART (2008) Motion Tracking Targets

- Interaktive virtuelle Simulationen z. B. beim [Virtual Soldier Research \(2004\)](#).
- Computerspiele



Abbildung 2.7: Scene from Beowulf which uses Imagemotion, a motion capture system developed at Sony. [New York Times \(2007\)](#)

2.4 Anwendungssteuerung mit Motion Tracking

Im Zuge der Suche nach neuen effektiveren Methoden der Steuerung von Applikationen abseits der traditionellen Maus und Tastatur wird unter anderem Motion Tracking als alternative Eingabemöglichkeit genutzt. Hier steht vor allem die Interaktion mit großen Anzeigeflächen oder virtuellen Räumen im Vordergrund.

Die Steuerung kann hier durch Bewegungsverfolgung oder durch vorher vereinbarte Gesten (siehe Kapitel 4) erfolgen. Einige Systeme arbeiten auch multimodal, d.h. sie nutzen noch andere Techniken wie Spracherkennung oder Eye Tracking zusammen mit dem Motion Tracking System.

2.4.1 Einsatzgebiete

Im folgenden wird eine Auswahl von Anwendungsgebieten vorgestellt, in denen Motion Tracking bereits erfolgreich eingesetzt wird.

Konstruktions- und Designstudios

Ein großes Arbeitsfeld, in dem Motion Tracking eingesetzt wird, ist das Konstruktions- oder Designstudio. Hier kann das zu entwickelnde Produkt auf einer hochauflösenden Anzeige vor dem Erstellen eines realen Prototypen als virtuelles Modell betrachtet werden. Man kann sich hier das Modell von allen Seiten im Detail und als Gesamtbild ansehen und in einigen Anwendungen sogar einzelne Teile oder ganze Baugruppen verändern.

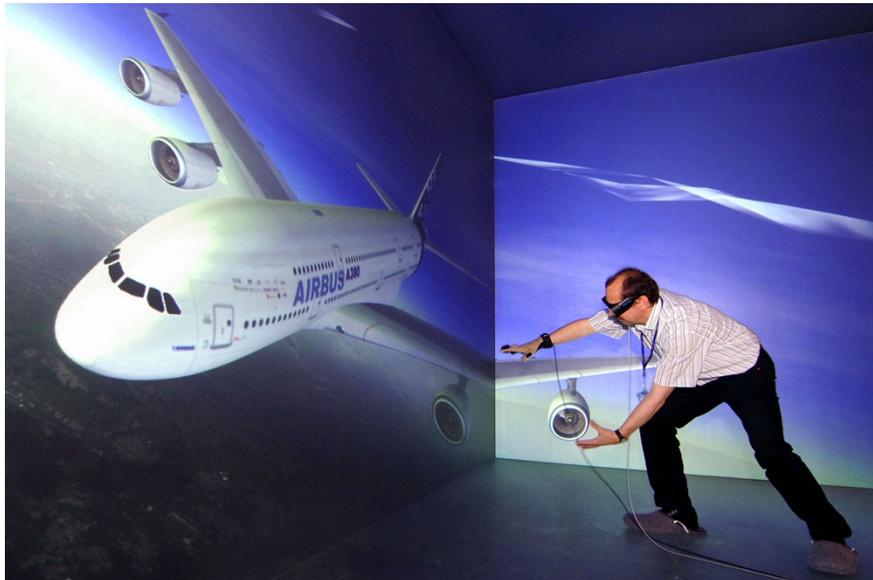


Abbildung 2.8: Airbus Stereo Cave in Hamburg

Spiele- und Entertainmentindustrie

Ein anderes Feld, in der sich die Steuerung mit Hilfe von Motion Tracking etabliert hat, ist die Entertainment Industrie. Der hohe Grad an Interaktivität und Immersion, sowie die intuitive Erlernbarkeit haben hier zu einer schnell steigenden Popularität geführt.

Das wohl bekannteste Beispiel ist die Nintendo Wii⁵, die ein Gyroskop und einen Infrarottracker in Ihrem System kombiniert, um ein völlig neues Spielerlebnis zu schaffen (siehe Abb. 2.9).

Aber auch für die Steuerung etablierter Home-Entertainmentsysteme wird diese Form der Steuerung langsam adaptiert. So hat z. B. Kodak für September 2008 einen HD Video Player angekündigt, der über eine der Nintendo Wii ähnliche Gestensteuerung verfügen soll (siehe Abb. 2.10).

Auch für Künstler hat Motion Tracking interessante Perspektiven zu bieten. So wird z. B. sowohl vom Fraunhofer Institut⁶ als auch von InnoTeamS⁷ eine *Virtual Graffiti* Installation angeboten in der man mit der Hilfe von Motion Tracking virtuelle Bilder mit einer markierten Sprühdose erstellen kann (siehe Abb. 2.11).



Abbildung 2.9: Nintendo Wii



Abbildung 2.10: Kodak HD Player zur Fernbedienung Gestensteuerung

⁵Nintendo

⁶<http://a4www.igd.fraunhofer.de/products/9/>

⁷http://www.innoteams.de/P_VirtualGraffiti.htm



Abbildung 2.11: Virtual Graffiti (InnoTeamS)

3 Analyse

In der heutigen Arbeitswelt oder auch in heutigen Spielsystemen suchen immer mehr Menschen nach Alternativen zu den herkömmlichen Eingabemethoden mit Maus und Tastatur. Sei es, um neue intuitivere Interaktionstechniken zu ermöglichen, oder für Anwendungen die für diese althergebrachten Eingabemöglichkeiten einfach nicht geeignet sind.

Auch wenn Prognosen wie jene von Gartner Research¹, in denen einige Analysten davon ausgehen, dass die Maus in den nächsten 3-5 Jahren obsolet sein wird² wohl ein wenig zu optimistisch scheinen, haben doch mit Technologien wie Multitouch in Apples iPhone³ oder Motion Tracking mit der Nintendo Wii⁴ alternative Eingabemethoden bereits den Massenmarkt erreicht.

Auch wenn diese Entwicklung sich erst in ihrem Anfangsstadium befindet (oder gerade deswegen), scheint es sinnvoll, sich damit zu beschäftigen wie zukünftige Eingabe- und Interaktionsmöglichkeiten in heutigen und zukünftigen Anwendungen aussehen könnten.

Gerade in Anwendungsgebieten, die mit großen Anzeigeflächen oder komplett virtuellen Räumen (Caves (Cruz-Neira u. a. (1992))) arbeiten, erkennt man schnell die Grenzen der tradierten Eingabemethoden. Beispiele hierfür sind z. B. Konstruktionsbüros der Auto- und Flugzeugindustrie, Trainingsaufbauten des Militärs (z. B. 3.1) oder Visualisierungen von künftigen oder vergangenen Bauwerken. Auch für die Spieleindustrie ist das Interagieren mit sehr großen Anzeigeflächen natürlich interessant.

In diesem Szenario, in der der Benutzer frei vor einer großen Anzeigefläche steht, scheint das Steuern mittels Gesten und Bewegungen analog zu den rudimentären Eingabemöglichkeiten der Nintendo Wii als besonders lohnenswert. Das Ziel dieser Arbeit ist deshalb das Erstellen und Evaluieren prototypischer Interaktionsmethoden mit grafischen Anwendungen auf Basis eines Motion Trackers mit Gesten und Bewegungen.

Die Anforderungen in diesem Kapitel orientieren sich an bestehenden Einsatzgebieten in Entwicklungs- und Konstruktionsbüros oder der Steuerung von Einzel- oder Multiplayerspielen auf Großleinwänden oder in Caves.

¹<http://www.gartner.com/>

²<http://news.bbc.co.uk/2/hi/technology/7508842.stm>

³<http://www.apple.com/de/iphone/>

⁴http://www.nintendo.de/NOE/de_DE/systems/ueber_wii_1069.html



Abbildung 3.1: Ultra-high resolution multi-projector VR-System at Nellis Air Force Base in Nevada

Für die Anforderungsanalyse in diesem Kapitel werden im folgenden zwei unterschiedliche Beispielszenarien vorgestellt, mit denen die unterschiedlichen Anforderungen an eine Steuerung über Bewegung und Gesten untersucht werden kann. Aus diesen Szenarien werden dann einige Funktionale (3.2) und Nicht-Funktionale (3.3) Anforderungen abgeleitet. Am Ende wird kurz auf die Möglichkeiten und zu erwartenden Schwierigkeiten der zu erstellenden Umsetzungen eingegangen.(3.4)

3.1 Szenarien: Anwendungssteuerung mit Gesten und Bewegungen

Die Szenarien beschreiben zwei grundsätzlich unterschiedliche Anwendungsgebiete, um eine große Abdeckung möglicher Anforderungen zu erhalten. Auf der einen Seite wird am Beispiel eines Spiels evaluiert, in wie weit man den natürlichen Spielablauf eines bekannten Spiels aus der realen Welt mit einem Motion Tracker umsetzen kann, ohne dass der Nutzer eine neue Steuerung erlernen muss.

Im zweiten Szenario werden die Interaktionsmöglichkeiten mit einer eher traditionellen grafischen Anwendung untersucht, wie sie in den oben genannten professionellen Anwendungsgebieten vorkommt und geprüft, inwieweit eine auf Gesten- und Bewegungen basierende Steuerung in der Lage ist, traditionelle Eingabegeräte zu ergänzen oder zu ersetzen.

3.1.1 Szenario I: Spiel

Im Szenario 'Spiel' sollen anhand eines Spiels Einzel- und Mehrspielersituationen simuliert werden. Schwerpunkt liegt hier auf der Evaluierung der Möglichkeiten, die eine direkte Umsetzung von Bewegungen in eine Spielsteuerung bietet (Analog z. B. zu der Nintendo Wii⁵). Auch können hier die möglichen Umsetzungen der Bedienung von traditionellen Schaltflä-



Abbildung 3.2: Wii Sports:Tennis (<http://www.gamesoul.com>)

chen wie z. B. bei der Spielmodusauswahl oder zum Eingeben des Namens in einer Highscoreliste untersucht werden.

Zu diesem Zweck wird ein simples Spiel entwickelt, das ausschließlich mit Gesten und Bewegungen gesteuert werden kann, anhand dessen diese Problemstellung konkret untersucht werden kann. Hierbei liegt der Schwerpunkt auf einer intuitiven „handbuchfreien“ Benutzbarkeit des Spiels und der Evaluation der möglichen unterschiedlichen Gestenarten, um dieses Ziel zu erreichen.

3.1.2 Szenario II: Applikation

Im Szenario 'Applikation' wird die Interaktion mit einer hochauflösenden grafischen Anwendung dargestellt. Dies soll die Arbeit in einem Konstruktions- oder Designbüro simulieren wie sie z. B. in der Automobilindustrie stattfindet. Hierbei wird auf der einen Seite dargestellt wie sich eine auf Maussteuerung ausgelegte traditionelle Anwendung sinnvoll mit Gesten steuern lässt und auf der anderen Seite wird gezeigt, inwieweit eine Anwendung, die sich nur schwierig mit traditionellen Eingabegeräten steuern lässt, mit Gesten gesteuert werden

⁵http://www.nintendo.de/NOE/de_DE/games/wii/wii_sports_2781.html



Abbildung 3.3: Audi Virtual Reality Studio in Ingolstadt (Audi (2008))

kann und welche Vorteile dies bietet.

Um eine möglichst hohe Vergleichbarkeit mit in der Industrie genutzten Anwendungen zu erreichen, sollte die für diese Arbeit gewählte Anwendung die Möglichkeit bieten, mit einem hochauflösenden texturierten Modell interagieren zu können. Außerdem sollte die gewählte Anwendung für mehrere Plattformen zur Verfügung stehen und eine Schnittstelle zur externen Steuerung besitzen, um die Gesten- und Bewegungssteuerung möglichst einfach anbinden zu können.

3.2 Funktionale Anforderungen

In diesem Abschnitt werden zunächst beispielhaft mehrere Anwendungsfälle beschrieben, die bei der Nutzung der oben beschriebenen Szenarien auftreten können. Aus den Anwendungsfällen werden anschließend konkrete Interaktionstechniken und Anforderungen abgeleitet, mit denen die Steuerung der verschiedenen Anwendungen sinnvoll möglich ist.

3.2.1 Anwendungsfälle

Spiel

Um ein intuitives, befriedigendes Spielerlebnis zu erzeugen, ist es notwendig, dass für den Benutzer eindeutig erkennbar ist, welche Aktionen seinerseits zu welchen Reaktionen des

Spielgeschehens führen. Auch ist es dafür notwendig, dass diese Reaktion möglichst verzögerungsfrei geschieht.

Die Spielsteuerung selbst sollte sich an dem Bewegungsablauf der realen Vorlage orientieren, d.h. im optimalen Fall sollte derselbe Bewegungsablauf, den der Benutzer in der realen Welt ausführen würde, auch in der Spielumgebung zum selben Ergebnis führen.

Zusätzlich zur Steuerung des eigentlichen Spiels ist es notwendig, dass der Benutzer verschiedene Grundeinstellungen des Spiels ändern kann, z. B. die Auswahl der Spieleranzahl oder den Schwierigkeitsgrad des Spiels. Um eine Highscoreliste mit den Namen der besten Spieler erstellen zu können, muss es außerdem eine Möglichkeit zur Eingabe von Text geben, die ebenfalls über Bewegungen ausgeführt werden kann.

Aus diesen Voraussetzungen ergeben sich die folgenden zwei Anwendungsfälle:

- Steuerung des Spiels durch einen der Realität entsprechenden Bewegungsablauf
- Möglichkeit zur Eingabe, z. B. über Schaltelemente (Buttons)

Modell

Bei der Benutzung von Konstruktionsmodellen auf einem skalierbaren Visualisierungssystem ist einer der Vorteile die Fähigkeit, sowohl das Modell als ganzes (oder einen großen Teil davon) aber eben auch kleinste Details, erkennen zu können. Zur effizienten Nutzung eines solchen Systems ist es deshalb wünschenswert, in der Lage zu sein, die Sicht auf das Modell schnell und einfach an die aktuellen Erfordernisse anzupassen.

Deshalb liegt bei der Anwendung, die der Darstellung eines Konstruktionsmodells entspricht, das Hauptaugenmerk auf der mühelosen Veränderbarkeit der augenblicklichen Sicht auf das Modell, um den Nutzern den von Ihnen gewünschten Detaillierungsgrad zu ermöglichen.

Um dieses Ziel erreichen zu können, ist es sinnvoll die Sicht auf das Modell möglichst frei verändern zu können. Dabei wurden vier grundlegende Arten zum Verändern der Sicht identifiziert:

- Verschieben der Ansicht in jede beliebige Richtung auf der von der X und Y-Achse aufgespannten Ebene
- Rotieren der Ansicht
- Hinein- oder Herauszoomen aus der Ansicht
- Neigen der Ansicht

Mit diesen vier Methoden zur Veränderung der Sicht ist es möglich, jeden beliebigen Punkt eines dreidimensionalen Modells auf jeden beliebigen Ort auf dem Anzeigemedium zu bringen und beliebig hinein- oder herauszuzoomen.

3.2.2 Interaktionstechniken

Aus diesen im Vorfeld geforderten Möglichkeiten der Interaktion lassen sich nun verschiedene Interaktionstechniken entwickeln, die geeignet sind, diese Anforderungen zu erfüllen.

Da für die Spielsteuerung eine möglichst realitätsnahe Steuerung gewünscht ist, scheint es am sinnvollsten die Bewegungen des Benutzers direkt an das Spiel weiterzugeben und als Eingabe umzusetzen. Im Detail sollen hier die Position und Ausrichtung der Marker, die die Hände darstellen, direkt als Eingabe genutzt werden. In dieser Art soll auch das Verschieben der Ansicht beim Konstruktionsmodell umgesetzt werden, um dem Benutzer eine möglichst direkte Steuerung der Sicht zu ermöglichen. Diese direkte Koppelung der Steuerung an die Handbewegung verläuft analog zu der Steuerung mit der Nintendo Wii oder auch mit einer traditionellen Maus (Direkte Koppelung Maus <-> Mauszeiger).

Das Wechseln der Spieleinstellungen und das Eingeben von Text kann als ähnliche Interaktion betrachtet werden. Dazu werden die unterschiedlichen Auswahlmöglichkeiten oder Buchstaben als einzelne Schaltelemente (Buttons) realisiert, die über das gleiche Bewegungsmuster betätigt werden können. Um die Eignung verschiedener Bewegungsmuster für das Bedienen von Steuerelementen evaluieren zu können, ist es sinnvoll mehrere alternative Muster zu implementieren, die dann auf Ihre Akzeptanz beim Nutzer hin getestet werden können. Zu diesem Zweck sollten mehrere unterschiedliche Gesten realisiert werden. Für das Rotieren, Skalieren und Neigen der Ansicht sind unterschiedliche Bewegungsmuster vonnöten. Diese sollten sich möglichst an Aktionen orientieren, die eine Person auch in der realen Welt nutzen würde, wie z. B. das Drehen und Neigen eines Blatts Papiers, um eine möglichst intuitive Bedienung zu gewährleisten.

Zusammenfassend lassen sich die folgenden fünf Interaktionstechniken unterscheiden:

1. Direktes Koppeln eines Objekts an die Handbewegung
2. Auslösen eines Steuerelements (Button)
3. Rotieren
4. Skalieren
5. Neigen

Mit diesen fünf Interaktionstechniken lassen sich alle gewünschten Aktionen zur Steuerung der zu erstellenden Anwendungen realisieren.

3.3 Nicht-funktionale Anforderungen

Neben den eben genannten funktionalen Anforderungen gibt es noch mehrere nicht-funktionale Anforderungen, die zu beachten sind, um den Nutzern einen möglichst hohen Bedienkomfort zu bieten und so die Akzeptanz dieser Art der Steuerung zu erhöhen.

3.3.1 Intuitive Bedienbarkeit

Die Intuitivität und leichte Erlernbarkeit (Gemäß dem Ideal der „handbuchfreien Bedienung“) ist eine der wichtigsten Aufgaben die eine Steuerung durch Gesten erfüllen muss, wenn sie sich gegen die tradierten Eingabemedien durchsetzen will. Zu diesem Zwecke sollten die gewählten Gesten möglichst wenig komplex und natürlichen Interaktionsmustern in der realen Welt entlehnt sein.

Um dieses Ziel zu erreichen, ist es auch notwendig, Gesten anwendungsübergreifend zu konzipieren, also sicherzustellen, dass ähnliche Aufgaben in unterschiedlichen Anwendungen sich durch ähnliche (am besten gleiche) Gesten bedienen lassen.

3.3.2 Performance

Eine hohe Performance ist nötig, um dem Nutzer eine Reaktion des Systems auf seine Eingaben in Echtzeit bieten zu können. Nur wenn es keine Verzögerung zwischen Aktion und Reaktion gibt, hat der Nutzer das Gefühl einer direkten Kontrolle.

Zum Erreichen dieses Ziels ist es wünschenswert Gesten schon erkennen zu können, bevor der Nutzer sie vollständig ausgeführt hat. Durch dieses frühe Erkennen erhöht sich die gefühlte Responsivität der Anwendung für den Nutzer erheblich.

3.3.3 Übertragbarkeit

Da in den heutigen Arbeitsumgebungen eine immer inhomogenere Systemlandschaft anzutreffen ist, ist es wünschenswert, dass sich eine Steuerung, die in der Lage sein soll, verschiedene Anwendungen zu steuern, auch auf möglichst vielen Betriebssystemen einsetzen lässt (Allein im Ambient Labor der HAW Hamburg sind drei unterschiedliche Betriebssysteme im Einsatz). Zu diesem Zwecke ist es deshalb sinnvoll eine Entwicklungsumgebung und Implementationsform zu wählen, die möglichst systemunabhängig ist.

3.3.4 Mehrbenutzerfähigkeit

Sowohl in Konstruktionsbüros als auch in heutigen Spielen hat die Kommunikation und die Zusammenarbeit mehrerer Personen einen immer höheren Stellenwert. Gerade bei Anwendungen, die auf skalierbaren Visualisierungssystemen dargestellt werden, ist es wünschenswert, dass mehrere Personen gleichzeitig mit der Anwendung interagieren können. Aus diesem Grund sollte eine Gestensteuerung nicht an einzelne Personen gebunden sein, deren Gesten es erlernt hat, sondern möglichst ohne Einarbeitungszeit von unterschiedlichen Personen genutzt werden können.

3.4 Fazit

Aus den in den vorherigen Unterkapiteln aufgezeigten Anforderungen lässt sich ableiten, dass ein optimales System in der Lage sein sollte, alle benötigten Funktionen einer Anwendung mit intuitiven Gesten von mehreren Benutzern verzögerungsfrei ansprechen zu können. Der wichtigste Punkt ist aber die intuitive Bedienbarkeit des Systems, um der Anforderung die Weiser an eine Maschinensteuerung stellt

Machines that fit the human environment, instead of forcing humans to enter theirs⁶

gerecht werden zu können.

⁶Weiser (1991)

4 Gesten

In diesem Kapitel wird zuerst eine kurze Begriffsklärung vorgenommen, was Gesten eigentlich sind (4.1) und dann im folgenden auf die Komplexität des Erkennens von Gestenbedeutungen und was daraus folgt eingegangen. Danach wird die in der Literatur vorgenommene Kategorisierung von Gestentypen vorgestellt (4.2) und deren Mängel im Bezug auf eine computergestützte Gestenerkennung dargelegt. Aufbauend darauf wird im folgenden Abschnitt eine Erweiterung des bestehenden Modells vorgenommen (4.3), um es an unsere Anforderungen anzupassen.

Im Unterkapitel 4.4 wird auf das Problem der Erkennung von Anfang und Ende einer Geste eingegangen, um dann im letzten Abschnitt (4.6) auf die Eigenheiten der Erkennung von Gesten speziell im dreidimensionalen Raum einzugehen.

4.1 Begriffsklärung

Gesten sind im Grunde zeichenhafte Bewegungen bestimmter Körperteile (z. B. der Hand), um nonverbal zu kommunizieren. Anders als Gesichtsausdrücke sind Gesten nicht angeboren, sondern erlernt und daher von gesellschaftlichen und kulturellen Gegebenheiten abhängig.

Ein gutes Beispiel hierfür ist das Daumen hoch Zeichen (siehe Abb. 4.1) das in Deutschland eine positive Bedeutung hat (Alles ok, gut gelaufen) aber diese Geste ist im Mittleren Osten und Russland eine obszöne Beleidigung.

Wenn man eine Steuerung von Computersystemen mit Hilfe von Gesten entwickeln will, muss man sich also über die verschiedenen Arten von Gesten, und welche davon für eine Steuerung geeignet sind, Gedanken machen. Erste Überlegungen zum Einordnen von Gesten aus linguistischer Sicht wurden von McNeill basierend auf den früheren Arbeiten von Kendon erstellt. Anhand dieser Arbeiten lässt sich erkennen, dass eine allgemeine Erkennung von Gesten durch die hohe Anzahl und Komplexität möglicher Gesten nicht sinnvoll ist. Daraus ergibt sich, dass man aus dem Gesamtbereich des menschlichen Gestenrepertoire nur einen Teilbereich erkennen und für eine Steuerung nutzbar machen kann. Deshalb ist es notwendig Gesten zu klassifizieren und geeignete Gesten für eine Steuerung zu identifizieren.



Abbildung 4.1: Daumen hoch Foto: Jens Schierenbeck/dpa/gms

4.2 Bestehende Klassifizierung von Gesten

Im Umfeld der Forschung zu Mensch Maschine Interaktionen (siehe 2.1) versuchte [Queck u. a. \(1995\)](#), aus linguistischer Sicht, eine erste Kategorisierung von Gestenarten. Auf dieser aufbauend entwickelte [Pavlovic u. a. \(1997\)](#) eine erste Gesten Taxonomie, in der er Gesten in bewusste und unbewusste Gesten unterteilt und dann die bewussten Gesten weiter aufsplittet (siehe Abb. 4.2).

Dabei werden die Gesten in manipulative und kommunikative Gesten unterteilt und die kom-

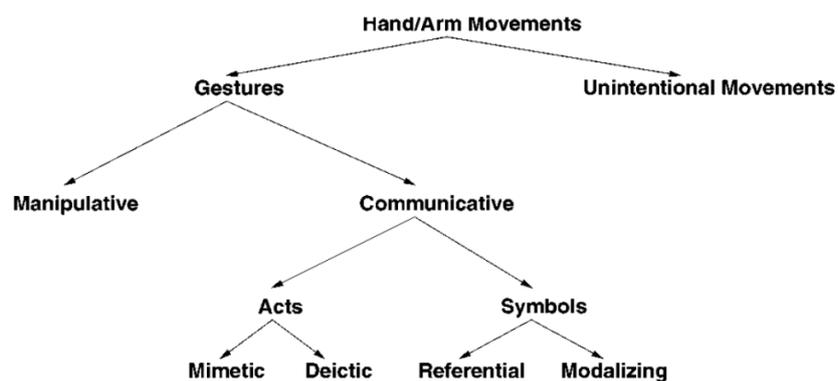


Abbildung 4.2: Taxonomie von Gesten nach [Pavlovic u. a. \(1997\)](#)

munikativen Gesten weiter untersucht, aber auf die manipulativen nicht weiter eingegangen. Auch wenn dies aus einem linguistischen Ansatz heraus sicherlich sinnvoll ist, so scheinen

doch die manipulativen Gesten für eine Steuerung von Computersystemen mindestens eben so interessant wie die kommunikativen.

Ohne tiefer darauf eingehen zu wollen, was Kommunikation im Detail ist, so lässt sich doch sagen, dass eine Kommunikation immer in mindestens zwei Richtungen stattfindet. Denn selbst wenn ein Gesprächspartner stumm bleibt, so kommuniziert er doch durch Körperhaltung und Gesichtsausdruck gemäß dem Motto

Man kann nicht nicht Kommunizieren¹.

Ein Computersystem hingegen ist in seinen Kommunikationsformen im Moment noch sehr begrenzt und kann deshalb auf Interaktionen nur eingeschränkt reagieren. Manipulative Gesten, d.h. die Interaktion mit physikalischen Objekten, scheinen als Metapher für eine Computereingabe geeigneter. Deshalb konzentrieren sich unsere weiteren Überlegungen an den manipulativen Gesten, so dass eine Erweiterung der Taxonomie von Pavlovic für diese Arbeit notwendig wurde.

4.3 Kategorisierung von manipulativen Gesten

Im Rahmen einer Arbeit für die [WIWITA \(2008\)](#) wurde eine erste Kategorisierung von manipulativen Gesten aufgestellt ([Boetzer u. a. \(2008\)](#)). Im Folgenden wird eine Kategorisierung der manipulativen Gesten vorgestellt, die auf diesen Überlegungen aufbaut und im Ambient Labor der HAW-Hamburg entwickelt wurde².

4.3.1 Bewegungsverfolgung

Die Bewegungsverfolgung bezeichnet die direkte Manipulation von Objekten, auf die das manipulierte Objekt sofort in realistischer Weise reagiert. Die Bewegungsverfolgung entspricht am ehesten den Effekten von ähnlichen Handlungen der Objektmanipulation in der realen Welt. In diese Klasse fallen z. B. das Umherschieben oder das Rotieren von Objekten (siehe [Abb. 4.3](#)). Diese Gesten sind äußerst intuitiv, da sie den Erfahrungen des Nutzers aus der realen Welt am nächsten kommt. Die meisten Gesten, die sich aus der realen Welt direkt ableiten lassen, gehören in diese Klasse.

¹[Watzlawick \(1996\)](#)

²siehe auch [Rahimi und Vogt \(2008\)](#)

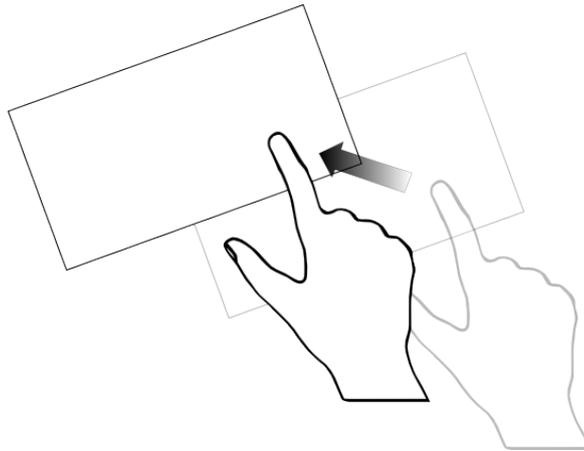


Abbildung 4.3: Verschieben eines Objekts (Rahimi und Vogt (2008))

4.3.2 Kontinuierliche Gesten

Kontinuierliche Gesten sind das Bindeglied zwischen Bewegungsverfolgung und symbolisch-manipulativen Gesten. In diese Klasse fallen jene Gesten, die zwar keine direkte Entsprechung in der Realität haben, aber sich durch einfache Metaphern schnell erlernen lassen. So kann man z. B. das Vergrößern und Verkleinern von Objekten durch einen Gummibandefekt vorstellbar machen. Auch Gesten, die durch ihren hohen Bekanntheitsgrad den meisten Benutzern ein Begriff sind, können in diese Kategorie fallen. Ein gutes Beispiel hierfür sind die Gesten die Apple³ für ihre Touch Oberfläche eingeführt hat (siehe Abb. 4.4).

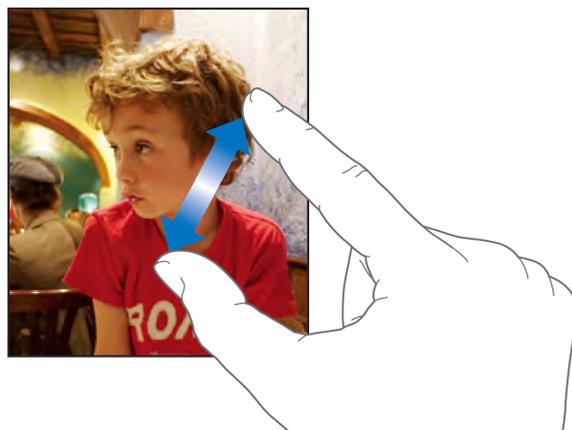


Abbildung 4.4: Zoomgeste bei Apples iPhone

³<http://www.apple.com/de/>

4.3.3 Symbolisch-manipulative Gesten

Symbolisch-manipulative Gesten sind Gesten, die nur eine indirekte Entsprechung in der natürlichen Welt haben und deshalb einen gewissen Grad von Abstraktion erfordern. Aus diesem Grund können sie vom Benutzer einen gewissen Lernaufwand erfordern. Dies ist z. B. der Fall für Manipulationen, die nur in Computerprogrammen möglich sind und so in der realen Welt nicht existieren (Dies könnte z. B. die Funktion sein, dass aktuell gewählte Objekt aus der Vogelperspektive zu betrachten). Trotzdem können diese Gesten das Repertoire der möglichen Befehle sinnvoll erweitern, da es so möglich wird, auch komplexe Manipulationen mit Gesten zu verbinden. Zu diesen Manipulationen gehört z. B. das Stapeln und Verstreuen von Objekten wie sie bei der *BumpTop* Anwendung von [Agarawala und Balakrishnan \(2006\)](#) verwendet werden.

Es können sehr unterschiedliche Gesten an eine solche Aktion gebunden werden, da für diese Art von Manipulationen keine spezifische Erwartungshaltung beim Benutzer besteht. Selbst in diesem Fall muss jedoch eine Metapher existieren, mit der der Nutzer die Geste mit der auszuführenden Aktion in Verbindung bringen kann. Deshalb muss hier im Vorfeld evaluiert werden, welche Gesten in Bezug auf einfache Erlernbarkeit und harmonisches Einfügen in den Arbeitsablauf und die Vorstellungswelt des Nutzers am besten eignet, um das Bedienerlebnis möglichst intuitiv zu gestalten.

Hierfür sind gesonderte Usability Untersuchungen notwendig, die die Akzeptanz von möglichen Gesten bei der Zielgruppe der zu steuernden Applikation evaluieren.

4.4 Anfang und Ende von Gesten

Eins der großen Problemfelder bei der Erkennung von Gesten durch Computersysteme ist der Zeitpunkt, an dem eine Geste beginnt, bzw. beendet wird. Da ein Mensch niemals völlig bewegungslos ist, ist es notwendig vorab Vereinbarungen zu treffen, wie man einer Anwendung die Erkennung des Anfangs einer Geste erleichtern kann.

Eine der genutzten Möglichkeiten ist die Festlegung einer Ruheposition. Wird diese verlassen, so beginnt eine neue Geste. Dies könnte z. B. bei Touch Anwendungen die Vereinbarung sein, dass ein Berühren der Touchoberfläche die Geste beginnt und ein Loslassen die Geste beendet. Analog dazu könnte man für Motion Tracking Anwendungen die Vereinbarung treffen, dass eine bestimmte Handhaltung (z. B. Hände unter der Gürtellinie) die Ruheposition darstellt.

Alternativ dazu wäre es eine Möglichkeit das Problem mit multimodalen Eingabemethoden zu lösen, also z. B. die Nutzung von Sprachbefehlen um dem Computer zu signalisieren, dass eine neue Geste begonnen wird.

Dabei ist anzumerken, dass jede dieser Vereinbarungen natürlich eine Beschränkung für

den Benutzer darstellt. Deshalb sollten die Start/Stop Vereinbarungen so ausgewählt werden, dass sie den Nutzer in seinem natürlichen Bewegungsablauf nicht behindern.

Zu den Stop Bedingungen ist abschließend zu sagen, dass es sinnvoll ist, die Geste erkennen zu können, bevor die Geste vollständig ausgeführt wurde. Also bei einer Geste, bei der die Hand einen Kreis beschreiben soll, die Geste bereits zu erkennen, wenn erst ein Teil der Kreisstrecke zurückgelegt wurde. Diese Maßnahme erhöht die gefühlte Responsivität der Anwendung und erhöht die Akzeptanz beim Anwender, da sich die Bedienung dadurch „flüssiger“ anfühlt. Zu diesem Zwecke müssen sich die einzelnen Gesten natürlich signifikant voneinander unterscheiden, um eine sichere Erkennung gewährleisten zu können.

4.5 Gesten im Anwendungskontext

Der Anwendungskontext in dem sich der Nutzer bewegt hat starken Einfluss auf die Bedeutung von Gesten. Die Navigation durch einen dreidimensionalen Raum kann eine andere Form von Gesten erfordern, als auf einer planen zweidimensionalen Fläche. Durch solche Rahmenbedingungen können völlig andere Gesten für ähnliche Interaktionen sinnvoll sein. Zwar ist eine gleiche Belegung von Gesten in unterschiedlichen Anwendungskontexten erstrebenswert, um die Bedienbarkeit möglichst einheitlich und den Lernaufwand niedrig zu halten. Allerdings erhöht dies die Schwierigkeit bei der Wahl von sinnvoll einsetzbaren Gesten erheblich und kann sogar unlösbar sein.

Ziel sollte deshalb sein, Gesten im Bereich der Bewegungsverfolgung und der kontinuierlichen Gesten zu suchen, um die Intuitivität der Anwendung möglichst hoch zu halten. Durch die beschriebenen Probleme, unterschiedliche Anwendungskontexte abzubilden, wird es aber kaum zu vermeiden sein, auch Gesten aus der Klasse der symbolisch-manipulativen Gesten zu verwenden.

4.6 Gesten im dreidimensionalen Raum

Dreidimensionale Gesten definieren sich über die Veränderung Ihrer Koordinaten im Raum über die Zeit. Da sich der Benutzer frei im Raum bewegen kann, sollten auf jeden Fall zumindest Startvereinbarungen (siehe 4.4) getroffen werden die den Anfang einer gewollten Geste markieren.

Auch wenn Gesten unter der Ausnutzung aller Freiheitsgrade möglich sind, ist das Erlernen und präzise Ausführen dieser Gesten mit einiger Körperbeherrschung des Nutzers verbunden. Deshalb sollte man Gesten, die sich entlang einer Ebene bewegen, bevorzugt verwenden, da diese wesentlich einfacher zu erlernen und auszuführen sind.

Da unterschiedliche Nutzer Gesten im dreidimensionalen Raum sehr unterschiedlich ausführen (allein schon aus Gründen unterschiedlicher Körperproportionen) ist es notwendig bei der Gestenerkennung eine hohe Toleranz einzuplanen. Welche Verfahren sich hier am besten eignen, um Gesten für Computersysteme zu beschreiben, muss noch durch weiterführende Forschung untersucht werden. (siehe [Heitsch \(2008\)](#))

5 Design

In diesem Kapitel wird auf Basis der formulierten Anforderungen aus der Analyse das Design von drei exemplarischen Beispielanwendungen, die eine Applikationssteuerung beispielhaft realisieren, entwickelt.

Zunächst wird im Unterkapitel 5.1 der im Labor vorhandene Aufbau vorgestellt und die technischen Randbedingungen der vorhandenen Installation erläutert.

In Unterkapitel 5.2 werden dann das Design der zu erstellenden Anwendungen im Detail dargestellt. Unterkapitel 5.2.1 erklärt die Kommunikation zwischen Trackerhardware und Applikation. Auf Basis dieser Überlegungen werden dann in Unterkapitel 5.2.2 und 5.2.3 die Details der Planung für die Entwicklung der einzelnen Anwendungen erläutert und die Überlegungen wie die Anforderungen aus der Analyse erfüllt werden können, präsentiert.

Im Anschluss daran (Unterkapitel 5.2.4) wird die Wahl des Entwurfsmusters erklärt und das gewählte Entwurfsmuster vorgestellt. Unterkapitel 5.2.5) behandelt schließlich die geplante Architektur zur Erkennung der zur Steuerung nötigen Gesten.

5.1 Der Interaktive Raum

Der Raum, in dem die Bewegungsverfolgung und Gestenerkennung stattfindet, hat eine Größe von 4x5 Metern. An seiner Stirnseite befindet sich ein großes hochauflösendes skalierbares Visualisierungssystem (Powerwall) aus neun 30“ Monitoren und der Tracker mit vier Infrarotkameras von ART (2008) (siehe Abb. 5.1).

5.1.1 Powerwall

Die Powerwall im Ambient Labor der HAW Hamburg besteht aus einer Matrix von neun Bildschirmen (Anordnung 3x3), die von einem Cluster aus insgesamt sechs Rechnern betrieben wird. Auf einem dieser Rechner läuft die darzustellende Anwendung zusammen mit der Applikation zum Verteilen der Anzeige. Hier wird als Softwarebasis *Chromium* eingesetzt. Chromium ist ein leicht erweiterbares System zum interaktiven, verteilten rendering von auf OpenGL¹ basierenden Oberflächen. Der Rechner teilt das virtuelle hochauflösende Bild in

¹<http://www.opengl.org/>



Abbildung 5.1: Powerwall mit Tracker im Ambient-Lab

mehrere Teile, die dann an die für die Anzeige zuständigen Rechner gesendet werden, welche dann das reale Teilbild rendern. Die Powerwall besitzt eine maximale Auflösung von 7500x4800 Punkten. Dadurch ist es möglich, hochdetaillierte Grafiken und Modelle darzustellen und, in Verbindung mit dem Motion Tracker, mit ihnen zu interagieren.

5.1.2 Motiontracker

Ein Motiontracker (oder kurz Tracker) ist ein Gerät oder System, das dazu in der Lage ist, die Position von ausgewählten Objekten und/oder Personen in einem vorgegebenen Raum zu finden und zu verfolgen. Sowohl die Position als auch die Ausrichtung der verfolgten Objekte im Raum kann gemessen werden. Der im Labor eingesetzte Tracker der Firma [ART \(2008\)](#) gehört zur Gruppe der kontaktlosen optischen Tracker. Er arbeitet mit vier Infrarotkameras, die periodisch Infrarotsignale aussenden und dann das reflektierte Infrarotlicht aufnehmen und in dem aufgenommenen Bild die Objektpositionen berechnen (Zweidimensional). Diese zweidimensionalen Koordinaten werden dann an einen Rechner zur Auswertung weitergeleitet. Das zu trackende „Objekt“ wird zu diesem Zweck mit speziellen Markern versehen (passive Infrarotlicht reflektierende Marker oder aktive Infrarotlicht aussendende Marker). Der Rechner berechnet aus diesen vier zweidimensionalen Bildern der korrespondierenden

Kameras die Position und Ausrichtung im dreidimensionalen Raum des markierten Objekts (siehe Abb. 5.2).

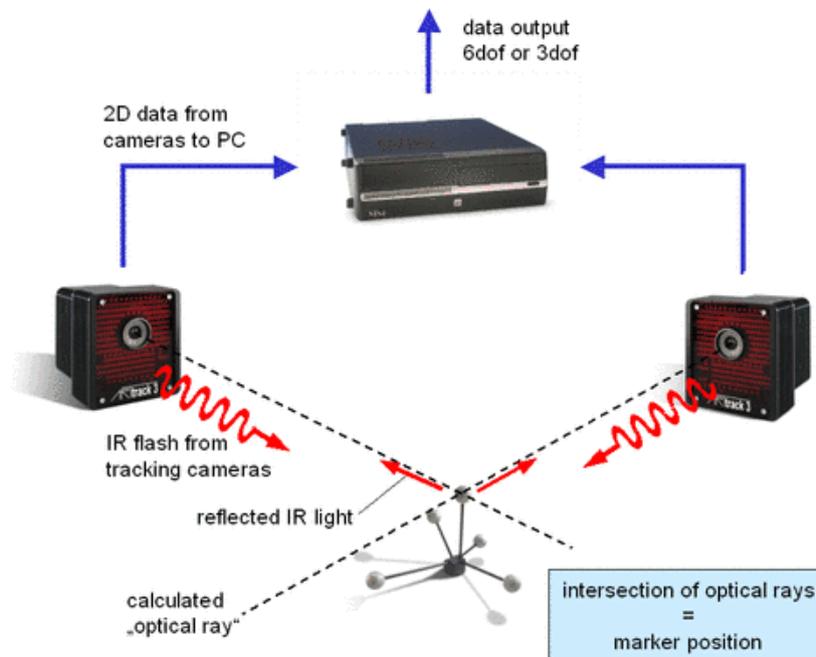


Abbildung 5.2: Funktionsweise Tracker

5.2 Beispielanwendungen

Bei der Entwicklung der exemplarischen Anwendungen fällt auf, dass bestimmte Funktionalitäten von allen Anwendungen benötigt werden. Diese trifft sowohl auf die Anbindung zum Tracker als auch auf die grundlegenden Funktionalitäten der Gestenerkennung zu. Deshalb bietet sich an, die Anbindung an den Tracker (siehe 5.2.1), die Gestenerkennung und die visuelle Darstellung unabhängig voneinander zu halten. Dies entspricht auch dem Prinzip der Trennung von Verantwortlichkeiten zum besseren Verständnis und Wartbarkeit von Computerprogrammen (siehe [Dijkstra \(1982\)](#)).

Zum ersten Verständnis des geplanten Programmablaufs siehe Abbildung 5.3.

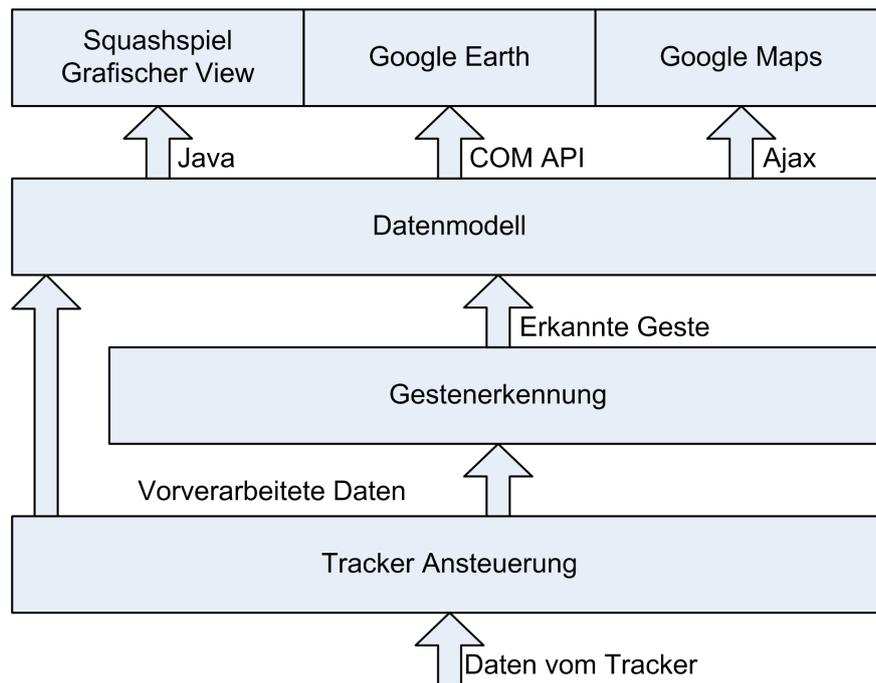


Abbildung 5.3: Schema des Programmaufbaus

5.2.1 Kommunikation Tracker - Software

Aufgabe der Tracker Anbindung ist es, die Daten, die vom Tracker gesendet werden, zu empfangen, in ein geeignetes Objekt zu verpacken und dieses dann den Anwendungen zur Verfügung zu stellen. Dabei soll es die gesamte Netzwerkanbindung und deren Details vor den restlichen Komponenten verbergen.

Die Trackersoftware von [ART \(2008\)](#) sendet die Positionsdaten der bekannten Objekte 60 mal in der Sekunde als UDP Paket im Single- oder Multicast verfahren an alle Clients. Damit mehrere Anwendungen gleichzeitig auf diese Daten zugreifen können und um gleichzeitig die Netzlast gering zu halten, wurde das Multicast-Verfahren gewählt (siehe auch: [Network Working Group \(2001\)](#)).

Diese Pakete werden dann von diesem Modul ausgelesen, die enthaltenen Daten mit regulären Ausdrücken extrahiert und in ein geeignetes Objekt verpackt, das alle für die Gestenerkennung relevanten Informationen enthält. Dieses Objekt wird dann den einzelnen Applikationen zur Verfügung gestellt.

5.2.2 Spiel: Squash

Als exemplarische Beispiel für eine Spielsteuerung soll ein kleines Spiel entwickelt werden. Um den Anforderungen aus der Analyse zu entsprechen, muss es sowohl eine Bewegungsverfolgungskomponente zur Steuerung des eigentlichen Spieles aufweisen als auch über klassische Eingabemöglichkeiten über Gesten verfügen. Als einfaches Spiel wurde ein simplifizierte Variante von Squash² gewählt.

Für das Schlagen des Balls gegen die Wand ist eine direkte Bewegungsverfolgung vorge-



Abbildung 5.4: Ein realer Squashplatz (Wikipedia commons)

sehen. Ein beliebiges Markerobjekt repräsentiert den Squashschläger mit dem der Benutzer mit dem Spiel interagieren kann. Hierbei soll nur die aktuelle Position des Markerobjekts direkt an die grafische Oberfläche weitergegeben werden. Eventuell muss allerdings die Bewegungsstrecke skaliert werden, um eine realistische Schlägerbewegung auf dem Anzeigemedium zu erzeugen, die der Größe der Repräsentation des Spielers in der grafischen Darstellung entspricht.

Desweiteren soll die Anwendung einen 1- und einen 2-Spieler-Modus aufweisen und es soll eine Highscoreliste vorhanden sein. Diese soll über einfache, intuitive Gesten bedienbar sein. Die Eingabe soll über einfache Schaltelemente (Buttons) realisiert werden, wie man sie auch aus traditionellen Anwendungen kennt. Das Auslösen soll dann über eine Geste erreicht werden.

²für eine Übersicht siehe: <http://de.wikipedia.org/wiki/Squash>

Die Gesten sollten intuitiv oder zumindest sehr einfach erlernbar sein und relativ robust gegenüber Falscherkennung sein, d.h. dass sie eindeutig genug sind, um sie von einer unbewussten Bewegung unterscheiden zu können.

Um verschiedene Gesten auf ihre Eignung für die Steuerung von Applikationen untersuchen zu können, sollen mehrere unterschiedliche Gesten implementiert werden. Dabei wird sowohl auf Vorbilder in der realen Welt, als auch auf bestehende Klick-Alternativen und Überlegungen aus anderen (Computer-) Anwendungsgebieten zurückgegriffen³.

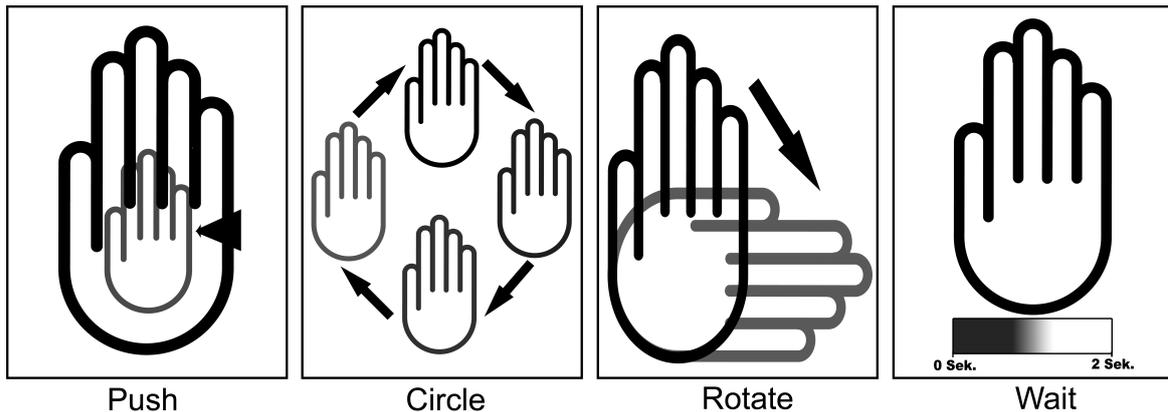


Abbildung 5.5: Gesten zur Betätigung eines Schaltelements

1. **Push** Dies soll simulieren, wie man ein großen realen Knopf drücken würde, nämlich durch eine Vorwärtsbewegung der Hand auf der Z-Achse vom Körper weg auf den Knopf (Anzeigefläche) zu. Dies entspricht am ehesten dem natürlichen Betätigen eines Knopfs in der realen Welt, allerdings gibt es keinen Widerstand oder eine physische Rückmeldung, da man den Knopf nur virtuell sieht, ihn aber nicht haptisch erfassen kann.
2. **Circle** Das Umkreisen des Schaltelements entspricht dem Einkreisen eines wichtigen Elements auf einem Blatt Papier. Auch wenn die Geste nicht vollständig intuitiv ist, so ist sie doch sehr einfach zu erlernen und auszuführen. Desweiteren ist eine saubere Kreisbewegung mit einer Hand eine Geste, die man sonst im normalen Bewegungsablauf nicht so häufig verwendet und damit ist es relativ unwahrscheinlich, dass man sie versehentlich ausführt.
3. **Rotate** Die Rotate Geste wurde einem Drehschalter nachempfunden. Hier soll eine Rotation der Hand um 90° nach rechts das Schaltelement auslösen. Damit der Benutzer die Rotation verfolgen kann, soll sich der Zeiger, der die Position der Hand auf der Anzeigefläche darstellt, mit der Hand gleichmäßig mitdrehen.

³siehe auch: <http://dontclick.it/>

4. **Wait** Das Verharren mit der Hand über der Schaltfläche für zwei Sekunden soll das Schaltelement auslösen. Damit der Benutzer problemlos erkennen kann, welchen Schalter er gerade betätigt und wie lange er noch warten muss bis der Schaltvorgang erfolgt, soll der Wartevorgang optisch durch einen Fortschrittsbalken sichtbar gemacht werden.

5.2.3 Anwendungen: Google Earth und Google Maps

Als exemplarisches Beispiel für eine professionelle Anwendung sollte ein frei verfügbares Programm genutzt werden, das eine ähnlich komplexe visuelle Darstellung ermöglicht wie z. B. eine texturierte Konstruktionszeichnung in einem CAD-Modelviewer. Eine Schnittstelle zur Ansteuerung grundlegender Funktionen durch eine eigene Software ist ebenfalls notwendig. Desweiteren sollte sie unter Linux und Windows lauffähig sein und eine OpenGL Schnittstelle besitzen, um sie auf der Powerwall darstellen zu können.

Google Earth



Abbildung 5.6: Google Earth auf der Powerwall

Aus diesen Rahmenbedingungen heraus wurde zum einen [Google Earth](http://earth.google.com/comapi/) ausgewählt, da es mit seinem hochauflösenden Modell von der Erde einem Konstruktionsmodell recht nahe kommt. Es ist unter Linux, MacOS X und Windows XP lauffähig und besitzt eine OpenGL Engine. Desweiteren besitzt es eine dokumentierte Schnittstelle⁴⁵ mit der es angesteuert werden kann. Die grundlegenden Interaktionsmöglichkeiten, die Google Earth mit der Weltkugel bietet, sollen nun per Gestensteuerung umgesetzt werden. Diese Interaktionsmöglichkeiten sind im einzelnen:

⁴<http://earth.google.com/comapi/>

⁵unter Windows XP und eingeschränkt unter MacOS X. Unix in Vorbereitung

- **Pan** Freies bewegen der Ansicht in eine beliebige Richtung.
- **Zoom** Stufenloses Hinein- oder Herauszoomen aus der Ansicht.
- **Rotate** Das Drehen der Ansicht.
- **Tilt** Das Horizontale Neigen der Ansicht.

Diesen Interaktionsmodi sollen nun Gesten zugeordnet werden. Um verschiedene Herangehensweisen evaluieren zu können, wurde geplant jeweils zwei verschiedene Gestentypen zu implementieren. Sowohl jeweils eine mit einer Hand auszuführende Geste als auch eine die beide Hände nutzt. Die den einzelnen Interaktionsmodi zugeordneten Gesten sind im einzelnen:

- **Pan** Das freie Bewegen der Ansicht wurde nur mit einer Hand umgesetzt, da es kaum sinnvoll erschien hier mit zwei Händen zu arbeiten. Der Pan selbst soll hier als direkte Umsetzung der Handbewegung durch den Benutzer realisiert werden.
- **Zoom** *1 Hand*: Hier wird das Zoomen direkt durch das Heran- bzw. Hinwegbewegen von der Anzeigefläche gesteuert. Die zurückgelegte Entfernung bestimmt dann wie weit hinein- oder herausgezoomt wird.
2 Hände: Hier soll der Zoom durch das Zusammen- oder Auseinanderbewegen beider Hände realisiert werden. Die Verringerung/Vergrößerung des Abstands der beiden Hände bestimmt wie weit hinein- oder hinausgezoomt wird.
- **Rotate** *1 Hand*: Das Drehen der Hand um die Z-Achse wird direkt als Rotation auf die Anzeige umgesetzt.
2 Hände: Hier wird analog zu einer Lenkrad Bewegung mit beiden Händen in die Richtung gedreht, in der die Ansicht gedreht werden soll.
- **Tilt** *1 Hand*: Das drehen der Hand um die Y-Achse soll hier direkt als Neigung der Anzeige umgesetzt werden.
2 Hände: Es wird eine Hand ruhig gehalten und die andere Hand entlang der Z-Achse bewegt. Eine Bewegung zum Körper hin hebt und vom Körper weg neigt die Ansicht.

Google Maps

Als zweite Möglichkeit soll die Anwendbarkeit der Steuerung auf Google Maps⁶ evaluiert werden. Google Maps ist als Browseranwendung auf allen großen Betriebssystemen verfügbar und ist über eine Technik namens AJAX⁷ ansteuerbar. Leider ist Google Maps als Browseranwendung nicht OpenGL basiert und deshalb nur mit Performanceabstrichen auf

⁶<http://maps.google.de/>

⁷<http://www.adaptivepath.com/ideas/essays/archives/000385.php>

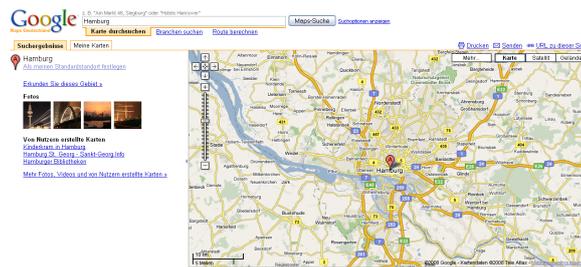


Abbildung 5.7: Google Maps

der Powerwall verwendbar. Aufgrund der allgemeinen Verfügbarkeit und des hohen Bekanntheitsgrads wurde dies aber als tolerabel eingeschätzt.

Als Browseranwendung unterstützt Google Maps nur ein Subset der Interaktionsmodi des großen Bruders Google Earth. Diese sind im einzelnen:

- Pan
- Zoom

Für diese beiden Interaktionsmodi sollen die gleichen Gesten genutzt werden können, wie für die entsprechenden Interaktionsmodi bei Google Earth.

5.2.4 Softwarearchitektur (MVC)

Aufgrund dieser Aufteilung bot sich als Softwarearchitektur das MVC Entwurfsmuster ([XEROX PARC \(1978–79\)](#)) an, da es ebenfalls eine strikte Trennung von Daten, Logik und Präsentation zum Ziel hat. Auch ist die einfache Austauschbarkeit der einzelnen Komponenten und damit eine höhere Flexibilität der Entwicklung Teil des Entwurfsmusters. (s.a. [Buschmann u. a. \(1996\)](#)) Das MVC Entwurfsmuster teilt ein System in drei Hauptkomponenten ein:

- **Model** Die Model-Komponente verwaltet die Daten der Anwendung und stellt diese über eine definierte Schnittstelle zur Verfügung. Ferner nimmt sie aufgrund von Instruktionen, die ihr von der Controller-Komponente gesendet werden, Änderungen an den Daten vor.
- **View** Die View-Komponente ist für die grafische Präsentation der Anwendungsdaten zuständig, die von der Model-Komponente verwaltet werden. Sie leitet Benutzereingaben an die Controller-Komponente weiter und reagiert auf Zustandsänderungen, die ihr von der Controller-Komponente übermittelt werden.

- **Controller** Die Controller-Komponente interpretiert Interaktionen des Anwenders, die ihr mit Hilfe von Eingabemedien, wie hier der Gestenerkennung, von der View-Komponente gesendet wurden. Wenn Anwendungsdaten oder deren Darstellung sich aufgrund von Benutzereingaben ändern sollen, informiert der Controller die Model- und View-Komponente.

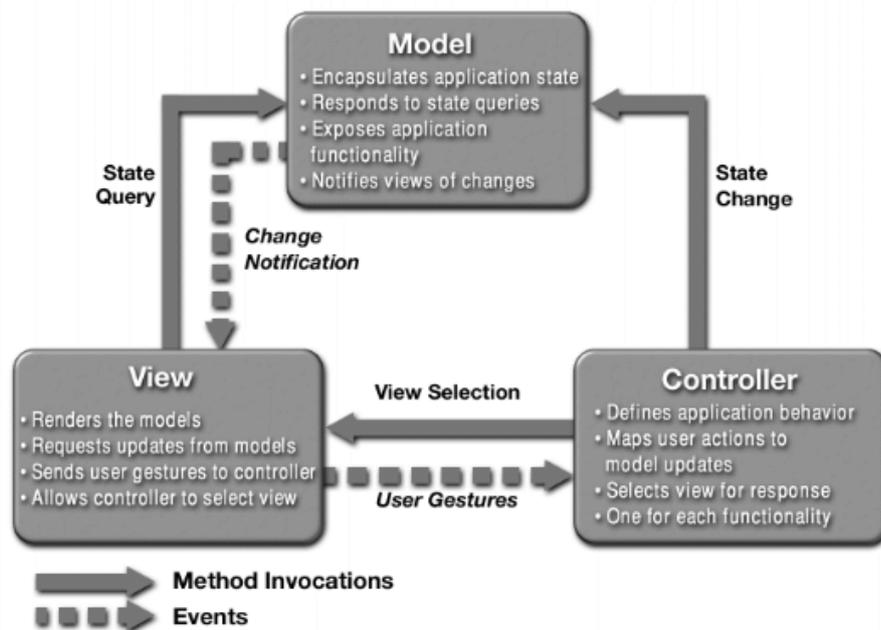


Abbildung 5.8: MVC Entwurfsmuster [Ramachandran \(2002\)](#)

Üblicherweise spielen eine Control- und eine View-Komponente paarweise zusammen. Sie „kennen“ sich gegenseitig und beziehen sich auf dasselbe Modell. Dieser Ansatz ermöglicht es unter anderem mehrere Views an ein Modell zu binden, um verschiedene Präsentationen des selben Datenmodells zu erstellen, oder den View komplett durch ein Neues zu Ersetzen, ohne das Datenmodell ändern zu müssen.

5.2.5 Gestenerkennung

Die Gestenerkennung ist der wichtigste Teil der Steuerung. Hier wird aus den einzelnen Positionen, die vom Tracker über einen bestimmten Zeitraum geliefert werden, Bewegungsmuster extrahiert und diese mit gespeicherten Gestenformen verglichen. (Für einen Überblick des zeitlichen Ablaufs der Gestenerkennung siehe [Abbildung 5.9](#))

Um gewollte Bewegungen von ungewollten unterscheiden zu können, wurde eine Startbedingung (siehe [Unterkapitel 4.4](#)) für die Steuerung der Applikationen von Google vereinbart.

Nur wenn die Handinnenseite der Anzeigefläche zugewandt ist ($\pm 10^\circ$) wird die Gestenerkennung aktiviert.

Nun wird geprüft, ob sich eine signifikante Anzahl von Punkten in der Bewegungshistorie befindet, um eine Geste erkennen zu können. Um ein Bewegungsmuster sicher erkennen zu können, werden mindestens 30 unterschiedliche Positionsdaten benötigt. Dies entspricht einer Bewegungsdauer von einer halben Sekunde.

Da Gesten auch bei langsamen Bewegungen im Regelfall kürzer als zwei Sekunden andauern, werden nun alle Positionsdaten, die älter als zwei Sekunden sind, gelöscht, um die zu verarbeitende Punktemenge möglichst gering zu halten.

Danach wird durch ein mathematisches Verfahren⁸ versucht, aus der zeitlichen Abfolge der Positionen Linien oder Kreise zu erkennen (Raum-Zeit Linearisierung) und die Bewegungsrichtung festzulegen.

Nachdem die Punktemenge auf diese Weise geglättet wurde, wird der Raum den die Positionsdaten aufspannen, nun auf eine einheitliche Größe normiert, um sie mit den vorgegebenen Gesten vergleichen zu können. Jetzt können die vorgegebenen Gesten mit dem aufgezeichneten normierten Bewegungsablauf verglichen werden. Wenn weniger als 5% der Punkte außerhalb des eines Musters liegen, gilt die Geste als erkannt.

Die Bewegungshistorie wird nun gelöscht und die Information, welche Geste erkannt wurde, den Applikationen zur Verfügung gestellt, damit diese die an die Geste gebundene Aktion ausführen können.

⁸Hough Transform s.a. <http://homepages.inf.ed.ac.uk/rbf/HIPR2/hough.htm> oder Chau. a. (2006)

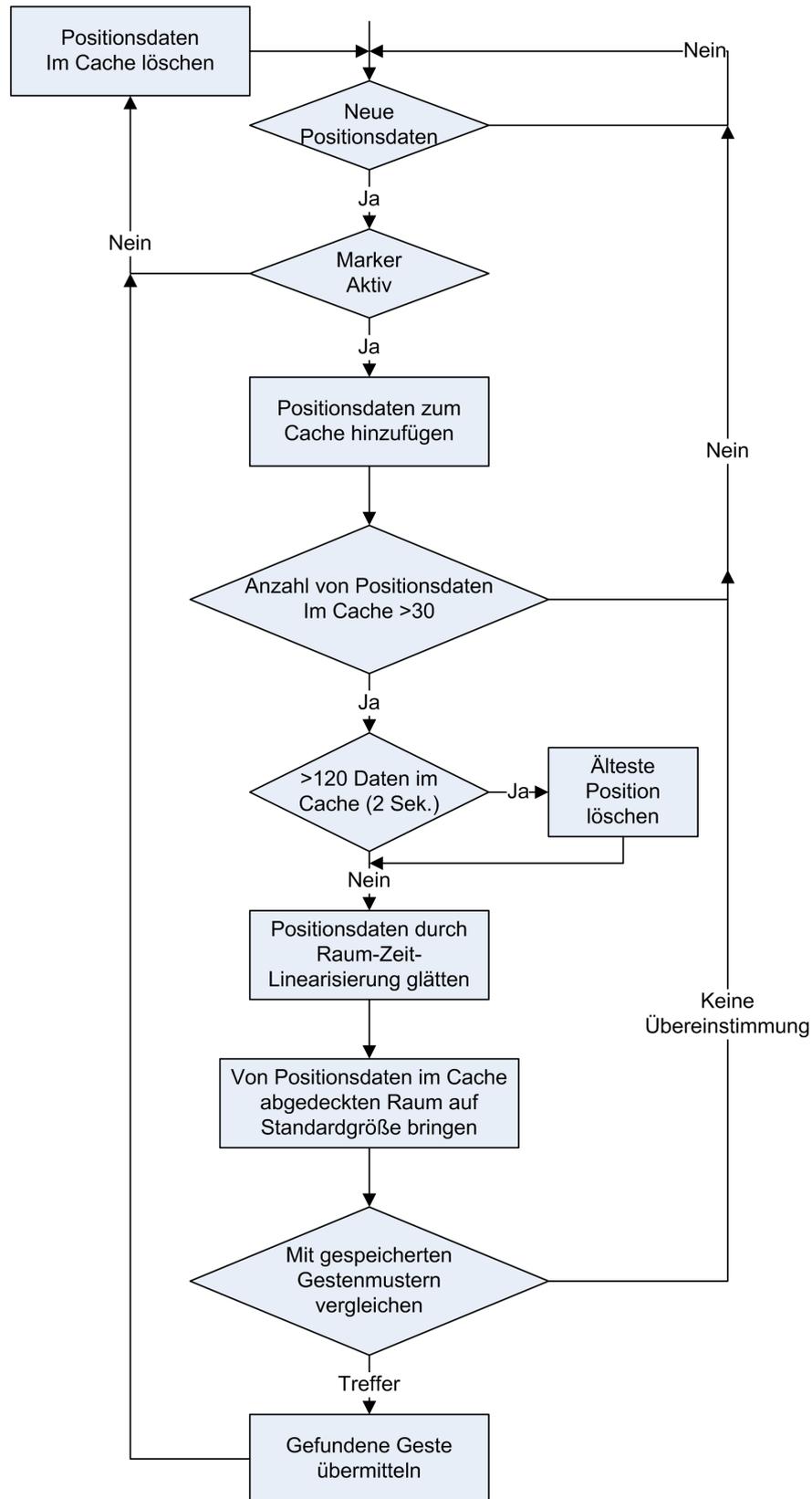


Abbildung 5.9: Zeitlicher Ablauf der Gestenerkennung

6 Realisierung

6.1 Tools und Frameworks

Da die Beispielanwendungen unter verschiedenen Betriebssystemen lauffähig sein sollten (MacOS X/Linux für die Powerwall und WinXP), wurde als Betriebssystem übergreifende Programmiersprache Java verwendet, um den Portierungsaufwand möglichst gering zu halten. Auch waren für Java einige Frameworks vorhanden, die eine schnellere Umsetzung der Programme ermöglichten, als es z. B. mit C++ möglich gewesen wäre.

6.1.1 Java

Java ist eine objektorientierte Programmiersprache entwickelt von [Sun Microsystems](#). Um es mit den Worten einiger Ihrer Entwickler zu sagen „The Java programming language is a general-purpose concurrent class-based object-oriented programming language, specifically designed to have as few implementation dependencies as possible.“ [Gosling u. a. \(1996\)](#) Java Programme werden in Bytecode übersetzt, die in einer speziellen Laufzeitumgebung (*Virtual Machine VM*) ausgeführt werden. Damit ist ein Java Programm im Prinzip auf jedem Betriebssystem lauffähig, für das es ein VM gibt. Da für alle drei verwendeten Betriebssysteme eine VM existiert, konnten die Programme mit minimalen Aufwand portiert werden.

Processing

Für die Spiel-Beispielanwendung war es nötig, selbst eine Grafische Oberfläche zu erstellen. Da die Powerwall (siehe [5.1.1](#)) auf der die Anwendungen dargestellt werden, OpenGL als Grafik-API verwendet, wurde ein OpenGL Framework benötigt mit dem die Anforderungen umgesetzt werden können. Dieses Framework sollte sowohl in der Lage sein, dass Spiel optisch ansprechend und performant umzusetzen, als auch eine geringe Einarbeitungszeit zu benötigen, um den Entwicklungsprozess nicht unnötig in die Länge zu ziehen. Hier fiel die Wahl auf das [Processing Framework](#). Processing ist ein auf Java basierender OpenGL Wrapper, der als digitales Sketchbuch hilft grafische Ideen umzusetzen. Es eignet sich aufgrund

seiner sehr leicht erlernbaren Programmierschnittstelle hervorragend dafür, kleine grafische Anwendungen in kurzer Zeit umzusetzen.

JACOB (Java COM-Bridge)

Für die Ansteuerung von Google Earth stellt Google unter Windows eine COM-Schnittstelle¹ zur Verfügung. Um diese Schnittstelle unter Java ansprechen zu können, benötigt man einen COM-Wrapper der die C++ Schnittstelle unter Java zugänglich macht. Die Wahl fiel hier auf **JACOB**. JACOB ist eine Open Source JAVA COM-Bridge, die es ermöglicht COM Funktionalitäten von Java aus zu nutzen. Durch die Möglichkeit den Sourcecode einsehen und selbst verändern zu können, war es leicht möglich, JACOB an die gegebenen Anforderungen anzupassen.

6.1.2 JSON (JavaScript Object Notation)

Im Laufe der Entwicklung der Google Earth Ansteuerung wurde parallel dazu auch eine Anbindung an Google Maps erstellt, um zu evaluieren, inwieweit die Steuerung auf andere Anwendungen übertragbar ist. Für die Ansteuerung von Maps war ein Containerformat notwendig, dass die Daten von der Gestenerkennung an die HTML Seite transportieren kann und sich dann per JavaScript einfach auslesen lässt. Hierfür standen zwei Alternativen zur Verfügung: XML (**W3C**) und **JSON**. Kurze Tests ergaben, dass JSON hier performanter ist², da es deutlich weniger Overhead als XML besitzt und es sich deshalb für den reinen Datenaustausch nahe Echtzeit besser eignet.

6.2 Technische Umsetzung

Für die technische Umsetzung sind einige Konditionen zum besseren Verständnis zu beachten. Der dreidimensionale Raum vor der Anzeigefläche ist so ausgerichtet, dass die X-Achse in der Mitte des Raumes vertikal von links nach rechts verläuft; die Y-Achse hat Ihren Nullpunkt am Boden und verläuft horizontal nach oben und die Z-Achse verläuft in der Mitte des Raumes horizontal, von der Anzeigefläche aus, in den Raum hinein.

Dieses Koordinatensystem wird im Folgenden für alle Objekte im Raum angenommen.

¹<http://earth.google.com/comapi/>

²siehe auch <http://blogs.msdn.com/mikechampion/archive/2006/12/21/the-json-vs-xml-debate-begins-in-earnest.aspx>

6.2.1 Trackeransteuerung

Für das Empfangen und Vorverarbeiten der Trackerdaten wurde ein unabhängiges Modul entwickelt, das von allen Anwendungen, die im weiteren vorgestellt werden unverändert genutzt werden kann.

Die Trackerdaten werden per Multicast UDP jede 1/60 Sekunde als ASCII Pakete von der Trackersoftware (D-Track) von ART versandt.

Die einzelnen Pakete enthalten eine Framenummer(fr), einen Timestamp(ts), die Anzahl der aktuell getrackten Standard-Körper(6d), eine ID(id), eine Angabe über die Qualität der erkannten Position (qu, ungenutzt), Position im Raum(sx-sz) und Ausrichtung in Grad im Raum(ax-az) und eine Rotationsmatrix.

```
fr integer
ts double
6d integer
[id qu][sx sy sz ax ay az][b0 b1 b2 b3 b4 b5 b6 b7 b8]
```

Diese Pakete werden mit einem regulären Ausdruck ausgewertet und die Positionsdaten, wenn gewünscht, auf ein beliebiges Zahlenintervall gemappt. Diese Normierung ist z. B. notwendig wenn gewünscht ist, dass die Position exakt am Rand des trackbaren Raumes genau den Rändern der Anzeigefläche entspricht. Alternativ könnte gewünscht sein, dass eine Person die in der Mitte des Raumes steht, die Ränder der Anzeigefläche nur durch Ausstrecken der Arme erreichen können soll.

Diese Daten werden nun in ein Objekt verpackt und in eine Queue eingefügt, die über eine Funktion nach aussen sichtbar ist.

6.2.2 Gestenerkennung

Die so vorbereiteten Daten werden von der Gestenerkennung weiter verarbeitet. Auch die Gestenerkennung ist ein eigenständiges Modul, das von allen Anwendungen unverändert genutzt werden kann.

Der Aufbau der Gestenerkennung orientiert sich an dem im Design Kapitel vorgestellten Ablauf (siehe Abb. 5.9). Hier ist insbesondere das Finden von Bewegungsrichtungen und die Erkennung von Bewegungsmustern von Interesse. Dazu gehört auch die Art und Weise wie man ein Gestenmuster überhaupt speichert.

Am Anfang hat man eine Menge von 30-120 Punkten die sich frei verteilt im dreidimensionalen Raum befinden, jedoch besitzt jeder Punkt eine Zeitinformation, so dass ein zeitlicher Ablauf der Bewegung nachvollziehbar ist (Siehe Abb. 6.1). Nun wird durch eine erweiterte

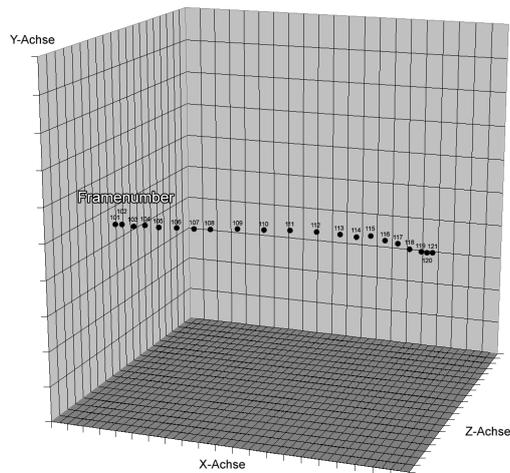


Abbildung 6.1: Punkte wie sie vom Tracker zur Verfügung gestellt werden

Hough Transformation³ versucht, Linien und Kreise innerhalb dieser Punkteansammlung zu erkennen. An diese gefundenen Formen wird dann mit Hilfe der Zeitinformation eine Bewegungsrichtung gebunden, so dass Form und Richtung der Bewegung bekannt sind (Siehe Abb. 6.2).

Die bekannten Gesten sind als unscharfe dreidimensionale Form gespeichert. Am Einfachs-

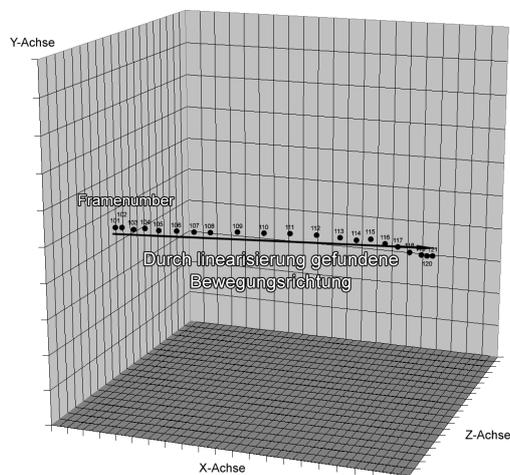


Abbildung 6.2: Durch Linearisierung gefundene Bewegungsvektoren

ten kann man sich diese Gesten als dreidimensionalen Gummischlauch mit großem Querschnitt vorstellen der die zu erkennende Geste umhüllt. Wenn mehr als 95% der Punkte des

³Cha u. a. (2006)

geglätteten Bewegungsmusters innerhalb dieses Schlauchs liegen, gilt die Geste als erkannt (Siehe Abb. 6.3).

Damit Gesten erkannt werden können, bevor die Bewegung abgeschlossen wurde, und um

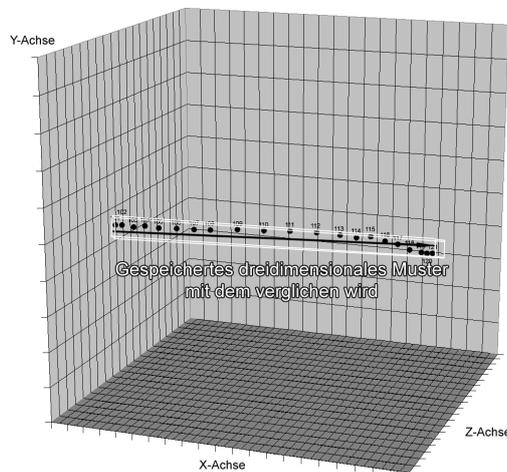


Abbildung 6.3: Vergleichen der Geste mit dem realen Bewegungsmuster

eine möglichst zeitnahe Reaktion des Programms herzustellen und das Gefühl einer nahtlosen Steuerung zu bieten, sind die Gesten in der Gestenbibliothek nicht vollständig. Dass heißt z. B. bei der Kreisgeste, dass das abgelegte Muster keinem Vollkreis entspricht sondern nur einer Abdeckung von 270° .

Bei längeren Bewegungspausen (d.h. Bewegungen unter 1 cm auf allen Achsen) wird davon ausgegangen, dass beim Wiederaufnehmen der Bewegung eine neue Geste beginnt und der Positionsdatencache wird gelöscht. Der Cache wird auch gelöscht, sobald das Markerobjekt die für Gesten spezifizierte Ausrichtung verlässt.

6.2.3 Spiel: Squash

Das Squashspiel wurde als Java-Applikation auf Basis des *Processing Frameworks*⁴ erstellt. Das Ziel war es sowohl ein intuitives Spielerlebnis auf Basis des direkten Abgreifens der Markerposition zu erreichen, als auch verschiedene Möglichkeiten des Auslösens von Standard Schaltelementen zu evaluieren.

Es wurde sowohl ein 1- als auch ein 2-Spieler Modus erstellt, um einerseits die Fehleranfälligkeit des Trackers durch Überdeckungen zu testen (ein Spieler könnte den anderen aus Kamerasicht verdecken), als auch die Performance der erstellten Softwarearchitektur bei mehreren aktiven Objekten zu evaluieren.

⁴[Processing](#)

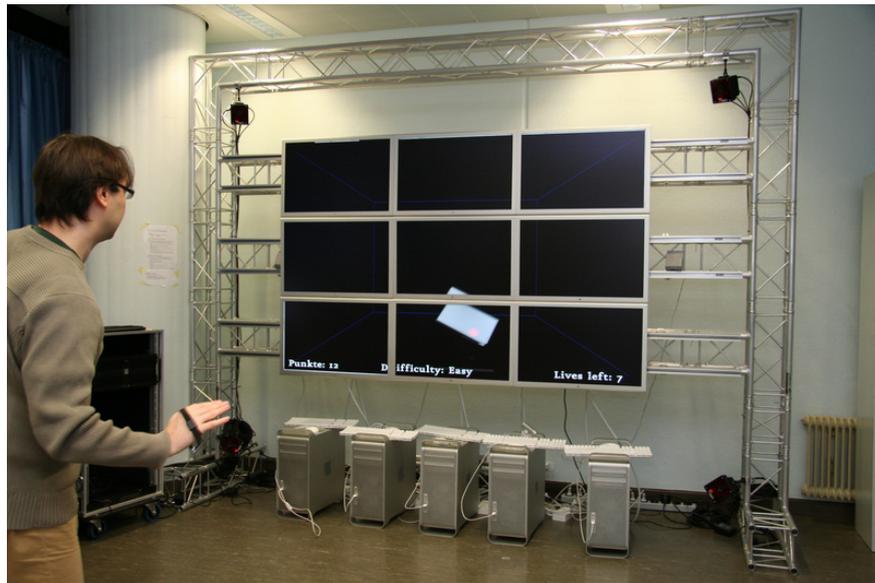


Abbildung 6.4: Das Squashspiel

Beim Spiel selbst repräsentiert ein beliebiger Marker den Squashschläger mit dem der Ball an die gegenüberliegende Wand geschlagen werden muss. Für das Spiel wurde nur eine Schlägerbewegung auf der Ebene, die von der X und Y-Achse aufgespannt wird, realisiert. Die Z-Achse wird nicht ausgewertet, so dass der Schläger nur auf einer zweidimensionalen Ebene bewegt werden kann. Dies wurde so implementiert, da kurze Tests mit Laborbesuchern ergaben, dass es schon so schwer genug ist, die Tiefenposition des Balles genau abzuschätzen. Die zusätzliche Komplexität, auch den Schläger auf der Z-Achse bewegen zu können, hätte das Spiel zu schwer und unintuitiv gemacht.

Es wurden drei verschiedene Schwierigkeitsgrade implementiert, die die Ballgeschwindigkeit verändern. Dadurch konnte untersucht werden, wie schnell ein Spieler auf den Ball reagieren und ob der Tracker diese Bewegungen schnell genug verfolgen kann.

Ebenfalls wurde eine Highscoreliste (siehe Abb. 6.5) implementiert, bei der der Name des Spielers eingegeben werden kann. Hierdurch konnte die Eingabe von kurzen Texten (bis zu 20 Zeichen) mittels Motion Tracking untersucht werden. Für diese Eingabe von Text und die Auswahl der Spielmodi wurden die vier im Design Kapitel vorgeschlagenen Gesten (siehe Abb. 5.5) realisiert (Für einen Überblick über den Ablauf des Spielverlaufs siehe Abb. 6.6).

Es wurden im Labor einige informative und nicht repräsentative Untersuchungen mit einigen Gruppen von insgesamt ca. 50 Schülern und Studenten durchgeführt, in der die Umsetzung getestet wurde.

Hierbei konnte festgestellt werden, dass die Steuerung praktisch keine Verschlechterung der Reaktionszeit gegenüber einem echten Squashspiel erzeugt und auch bei zwei Spielern so



Abbildung 6.5: Eingabebildschirm für den Highscore

performant ist, dass keine Verzögerung zwischen Bewegung und Computerdarstellung zu erkennen ist.

Bei der Evaluation der Gesten konnte festgestellt werden, dass es sehr unterschiedliche Präferenzen im Bezug auf die Gesten gab, allerdings lassen sich einige klare Trends herausarbeiten.

Die *Push* Geste die dem natürlichen Betätigen eines Knopfes am nächsten kommt erwies sich als die unbeliebteste. Das Ausstrecken des Armes im freien Raum ohne einen physikalischen Widerstand wurde von den meisten Probanden auf Dauer (bei dem Eingaben von Text) als zu anstrengend empfunden.

Die *Wait* und die *Rotate* Geste waren die beliebtesten. Ihre sehr einfache Anwendung, und bei der *Rotate* Geste die besonders schnelle Ausführbarkeit, führte zu einer hohen Akzeptanz.

Die *Circle* Geste befand sich im Mittelfeld, da sie zwar als nicht so anstrengend wie die *Push* Geste empfunden wurde, aber einige der Probanden mehr als einen Versuch brauchten bis ihre Kreisbewegung erkannt wurde.

6.2.4 Google Applikationen (Google Earth und Google Maps)

Die Anbindung an die Google Applikationen wurde als unabhängiges Modul konzipiert, um den Zugriff auf die externen Google Applikationen vom Rest der Anwendung zu kapseln. So konnte erreicht werden, dass die beiden Google Applikationen beliebig austauschbar sind, ohne die darunterliegenden Komponenten anpassen zu müssen (Auch könnte man im Prinzip mit jeder anderen extern steuerbaren Applikation, für die Gesten Sinn machen, nur durch erstellen einer solchen Anbindung interagieren).

Die Steuerung kann mit ein oder zwei Händen erfolgen. Je nachdem, ob eine oder zwei

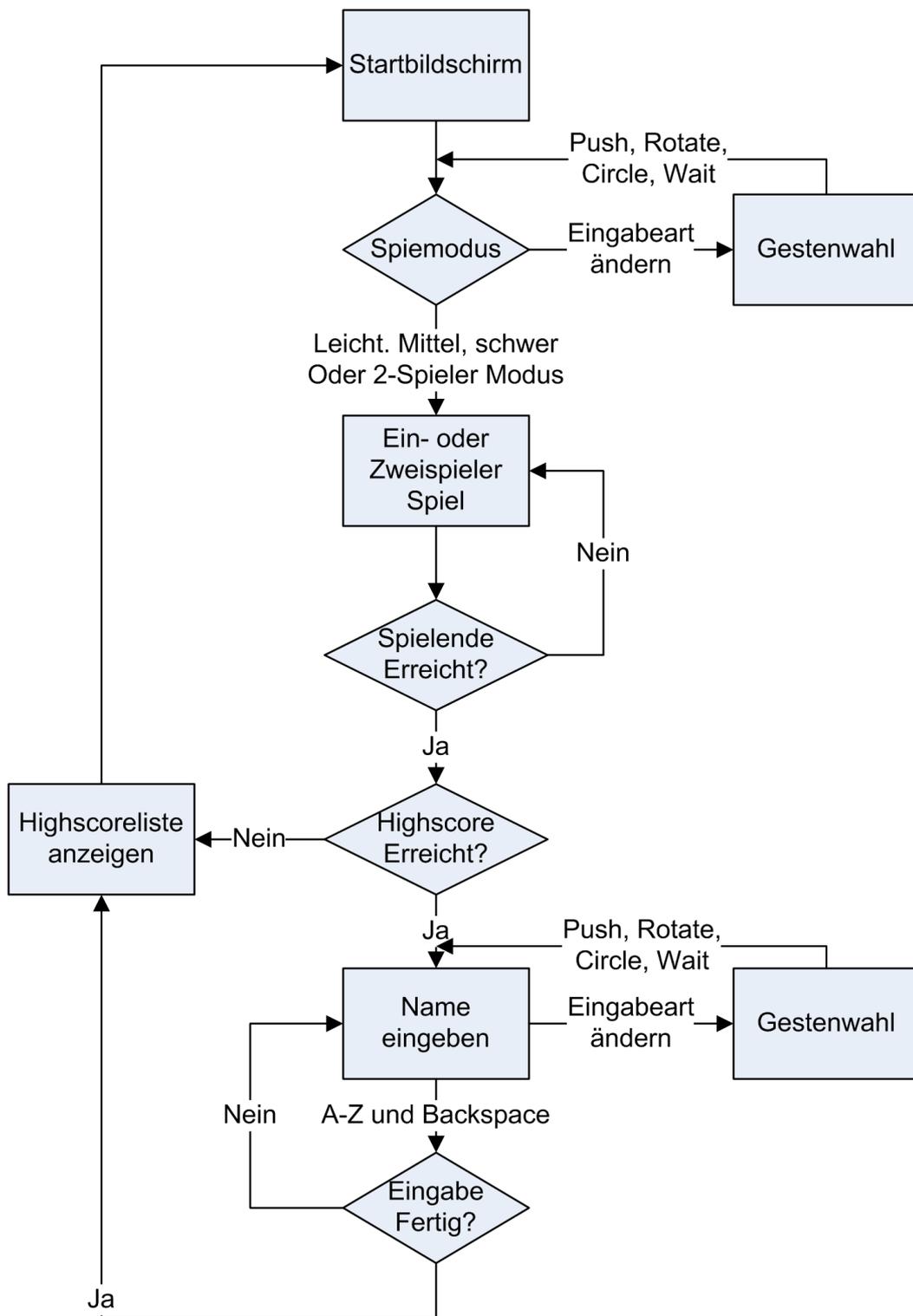


Abbildung 6.6: Spielablauf beim Squashspiel

Hände „aktiv“ sind, wird in den jeweiligen Modus gewechselt. Als Gesten wurden die im Designkapitel vorgestellten vier Gesten implementiert.

Im Folgenden werden die Details, in denen sich die beiden Google Applikationen unterscheiden, näher erläutert.

Google Earth Anbindung



Abbildung 6.7: Google Earth Steuerung mit Tree Marker

Um Google Earth extern steuern zu können, wurde mit der JACOB Java COM Bridge eine Anbindung an die COM API⁵ von Google Earth erstellt, die die komplette Steuerung des Globus zur Verfügung stellt. Leider stellte sich erst im Laufe der Arbeit heraus, dass zwar eine ähnliche Anbindung unter Linux von Google geplant ist, diese aber noch nicht realisiert wurde. Aus diesem Grunde konnte zwar die Steuerung unter Windowssystemen implementiert werden, es war aber nicht möglich, die Steuerung auf der Powerwall zu zeigen, da die Anbindung an die Powerwall nur unter Linux verfügbar ist.

Es wäre zwar theoretisch möglich, den Tastaturreiber unter der X-Oberfläche so zu verändern, dass er von der Gestensteuerung angesprochen werden kann und man somit Google Earth über Tastaturbefehle steuern kann, aber dieser Ansatz wurde aus Zeitgründen nicht weiter verfolgt.

Die Ansteuerung der COM-Schnittstelle wird durch ein Interface zur Verfügung gestellt, das die Details der COM-Schnittstelle verbirgt. Dieses Interface stellt nur die benötigten Funktionen in sehr abstrakter Form zu Verfügung, so dass die Befehle zur Steuerung des Globus

⁵<http://earth.google.com/comapi/>

äußerst einfach aufzurufen sind. Zum Verändern der aktuellen Kameraposition genügt deshalb z. B. folgender Aufruf:

```
moveCamera(googleEarth, camera, CAMSPEED);
```

wobei `googleEarth` das Interface zur GoogleEarth Anbindung ist, `camera` die aktuelle Kameraposition und `CAMSPEED` die Geschwindigkeit, mit der die Kamera sich an die neue Position bewegen soll.

Durch diese Art der Implementierung ist nicht nur die Steuerung der Ansicht äußerst simpel, sondern auch die Lesbarkeit des Quellcodes höher.⁶

Die Trackeranbindung und die Gestenerkennung konnten ohne Veränderung für die Steuerung verwendet werden, allerdings mit einem auf Google Earth zugeschnittenen Gestenrepertoire.

Google Maps Anbindung

Als Alternative zur Google Earth Anbindung und zur Untersuchung der Universalität der Ansteuerung wurde parallel eine Ansteuerung von Google Maps entwickelt. Diese Ansteuerung ist bis auf die Kommunikation mit Google Maps 100% identisch mit der Google Earth anbindung.

Die Anbindung an Google Maps wird mit AJAX⁷ (Asynchronous JavaScript and XML) realisiert. Im Prinzip liest eine Webseite, die Google Maps darstellt, eine strukturierte Textdatei (z. B. XML) in regelmäßigen Zeitintervallen ein, wertet diese aus, und verändert die Maps Ansicht entsprechend.

Um eine möglichst hohe Performanz bei dieser Anbindung zu erreichen, wurde statt XML JSON eingesetzt (siehe 6.1.2). Mit diesem Containerformat war es möglich, ein sehr kurzes Zeitintervall (<10 msek.) für die Aktualisierung der Webseite zu realisieren und so eine gefühlte Reaktion der Anwendung in Echtzeit zu erreichen.

6.3 Anmerkungen zur Umsetzung

Als erstes ist festzustellen, dass die Sprache Java zwar eine sehr schnelle und robuste Umsetzung der Aufgabenstellung ermöglichte und durch seine große Plattformunabhängigkeit

⁶Im Rahmen der Arbeit im Ambient Lab der HAW-Hamburg konnte diese Anbindung an Google Earth teilweise mit einer Multitouch Steuerung (Rahimi und Vogt (2008)) in wenigen Minuten erfolgreich kombiniert werden.

⁷<http://www.adaptivepath.com/ideas/essays/archives/000385.php>

⁸<http://code.google.com/apis/maps/documentation/index.html>

auch sehr leicht zu portieren war, aber das die Geschwindigkeit bei Anwendungen, die nahe Echtzeit reagieren sollen, nicht ausreichend ist.

Speziell bei der Darstellung komplexer grafischer Objekte wurden schnell die Grenzen des auf Java basierenden Processing Framework erreicht. Tests mit einer auf C++ basierenden Engine (Irrlicht⁹) zeigten, dass hier weit mehr möglich wäre.

Auch die Java COM-Bridge erwies sich leider als zu langsam. Schnell aufeinanderfolgende Befehle führten zu einer deutlichen Verzögerung bei der Ausführung. Eine testweise rudimentäre Implementation einer Ansteuerung der COM-Schnittstelle in C++ zeigten das dies nicht an der COM-Schnittstelle selbst liegt.

Für weiterführende Untersuchungen wäre deshalb zu empfehlen, das gesamte Konzept in einer performanteren Sprache zu implementieren. Dadurch würde die Portierung auf andere Systeme zwar erheblich erschwert, aber um komplexere Aufgabenstellungen nahe Echtzeit zu bewältigen, gibt es keine vernünftige Alternative.

Trotz dieser Einschränkungen waren die erstellten Anwendungen voll funktionsfähig und ausreichend performant, um erste Erkenntnisse über die Möglichkeiten der Anwendungssteuerung durch Gesten zu gewinnen.

⁹<http://irrlicht.sourceforge.net/>

7 Zusammenfassung

Diese Arbeit hatte zum Ziel eine Bewegungs- und Gestensteuerung zu realisieren, die in verschiedenen Anwendungen eingesetzt werden kann. Diese Gestensteuerung sollte intuitiv bedienbar sein und möglichst keine Einarbeitungszeit erfordern.

Im Rahmen dieser Arbeit wurde deshalb eine Gestensteuerung auf Basis eines Motion Trackers entwickelt, mit deren Hilfe sich drei unterschiedliche Anwendungen steuern ließen. Diese Anwendungen decken sowohl den professionellen als auch den Entertainment Bereich ab.

Zu diesem Zwecke wurde in Kapitel 4 erörtert, was Gesten eigentlich sind und welche Problemstellung sich generell bei der Erkennung von Gesten ergeben. Anschließend wurden dann die Speziellen Probleme, die bei der Gestenerkennung durch Computersysteme auftreten können, dargelegt. Zur Lösung dieser Probleme wird eine Kategorisierung von Gesten vorgenommen und einige Gestentypen werden definiert, die sich für eine Steuerung von Computersystemen eignen.

Es wurden in Kapitel 3 zwei Szenarien aufgestellt, in denen eine Gestenerkennung eingesetzt werden könnte und analysiert, was für Aktionen notwendig sind, um den Anforderungen, die die beiden Szenarien stellen, gerecht zu werden. Hierzu werden einige funktionale und nicht-funktionale Anforderungen aufgestellt und erläutert.

Anhand dieser Anforderungen wurde dann in Kapitel 5 eine Softwarearchitektur entwickelt mit der die Gestensteuerung umgesetzt, und die Anwendungen angesteuert werden können. Hierfür wurden zuerst die technischen Gegebenheiten des genutzten Geräts erläutert und dann auf die einzelnen Anwendungen im Detail eingegangen. Hier wurden auch die Gesten definiert mit denen mit den Anwendungen interagiert werden kann und es wird festgelegt, auf welche Weise die Gesten erkannt werden können.

Zum Schluss in Kapitel 6 wurde die Realisierung vorgestellt und Ihre Besonderheiten und Grenzen erläutert. Zu diesem Zweck wurden zuerst die genutzten Frameworks vorgestellt und dann auf die technische Umsetzung der einzelnen Komponenten eingegangen. Am Ende werden dann noch einige Auffälligkeiten der Umsetzung diskutiert.

7.1 Fazit

Eine der Hauptideen aus dieser Arbeit ist sicherlich, dass eine Gestenerkennung nur dann von den Nutzern akzeptiert werden wird, wenn sie einfach und ohne große Eingewöhnungsphase zu bedienen ist. Hier hat sich durch die Erfahrungen mit Nutzern in unserem Labor gezeigt, dass nicht nur die Intuitivität einer Geste im dreidimensionalen Raum entscheidend ist, sondern auch wie motorisch aufwendig die Geste auszuführen ist (siehe z. B. 6.2.3).

In diesem Feld sind deshalb noch weitreichende Forschungen mit verschiedenen Nutzergruppen notwendig, um feststellen zu können, was für Gesten für bestimmte Aktionen geeignet sind.

Auch wenn sich die gewählte Form der Gestenerkennung als recht robust und fehlertolerant erwiesen hat, so hat diese einfache Form der Mustererkennung doch Ihre Grenzen. Je komplizierter die zu findende Geste ist, desto höher wird die Fehleranfälligkeit. In diesem Bereich muss deshalb weiter untersucht werden, wie sich die Gestenerkennung weiter verbessern lässt. Hier könnte untersucht werden, inwieweit adaptive Systeme für die Gestenerkennung geeignet sind¹.

Allerdings konnte schon mit dieser exemplarischen Implementation gezeigt werden, dass sich Anwendungen mit Bewegungen und Gesten steuern lassen können und das mit geeigneten Gesten eine hohe Akzeptanz beim Nutzer erreicht werden kann. Speziell beim Spiel konnte man sehen, was für eine hohe Faszination diese direkten Interaktionsmöglichkeit bei unterschiedlichsten Nutzergruppen auslöst. Wenn geeignete Gesten gefunden werden ist die Gestensteuerung auf jeden Fall als Ergänzung oder Ersatz für tradierte Eingabegeräte geeignet.

7.2 Ausblick

Durch die schnell fortschreitende Entwicklung bei der Motion Track fähigen Hardware und die Faszination und Intuitivität, die die Gestensteuerung gewährleisten kann, ist damit zu rechnen, dass sich diese Technik in nicht allzu ferner Zukunft für eine Vielzahl von Anwendungstypen etablieren kann.

Die schnell fortschreitende Entwicklung der Anwendung von Gestenbasierten Steuerungssystemen mit den unterschiedlichsten Hardwareplattformen wie z. B. Multitouch Anwendungen zeigt, welches Potential in der Erkennung von Gesten liegt.

Da eine neue Form der Steuerung von Computersystemen immer auch neue Arten von

¹Für weitere Überlegungen in diese Richtung siehe [Heitsch \(2008\)](#)

Anwendungstypen ermöglicht (wie verschiedene neue Spielsysteme bereits gezeigt haben) sind in Zukunft einige völlig neue Anwendungstypen zu erwarten die von dieser neuen Form der Steuerung Gebrauch machen werden.

Literaturverzeichnis

- [ACM SIGCHI] ACM SIGCHI: *Definition and Overview of Human-Computer Interaction*.
– URL <http://sigchi.org/cdg/cdg2.html>. – Zuletzt besucht: 23.07.2008
- [Agarawala und Balakrishnan 2006] AGARAWALA, Anand ; BALAKRISHNAN, Ravin: Keepin' it real: pushing the desktop metaphor with physics, piles and the pen. In: *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*. New York, NY, USA : ACM, 2006, S. 1283–1292. – ISBN 1-59593-372-7
- [Airbus] AIRBUS: *Airbus in Germany*. – URL www.airbus.com/en/worldwide/airbus_in_germany.html. – Zuletzt besucht: 24.07.2008
- [ART 2008] ART: *advanced realtime tracking GmbH*. 2008. – URL <http://www.ar-tracking.de/>
- [Ascension] ASCENSION: *Ascension Technology Corporation*. – URL <http://www.ascension-tech.com/>. – Zuletzt besucht: 23.07.2008
- [Audi 2008] AUDI: *Audi News*. Januar 2008. – URL http://www.audi.com.au/audi/au/en2/about_audi_ag/audi_news/audi_adapts_the_silver.html. – Zuletzt besucht: 22.07.2008
- [Bezerianos und Balakrishnan 2005] BEZERIANOS, Anastasia ; BALAKRISHNAN, Ravin: View and Space Management on Large Displays. In: *IEEE Computer Graphics and Applications* 25 (2005), Nr. 4, S. 34–43. – ISSN 0272-1716
- [Boetzer u. a. 2008] BOETZER, J. ; RAHIMI, M. ; VOGT, M. ; WENDT, P. ; K.V.LUCK: Gestenbasierte Interaktion mit Hilfe von Multitouch und Motiontracking. In: CLEVE, Jürgen (Hrsg.): *Proceedings WIWITA 2008* Hochschule Wismar (Veranst.), Hochschule Wismar, 2008, S. 38–49. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/papers/WIWITA2008.pdf>. – Zuletzt besucht: 24.08.2008
- [Buschmann u. a. 1996] BUSCHMANN, Frank ; MEUNIER, Regine ; ROHNERT, Hans ; SOMMERLAD, Peter ; STAL, Michael: *Pattern-Oriented Software Architecture: A System of Patterns*. Bd. 1. Wiley, 1996. – ISBN: 978-0-471-95869-7

- [Cha u. a. 2006] CHA, J. ; COFER, R. H. ; KOZAITIS, S. P.: Extended Hough transform for linear feature detection. In: *Pattern Recogn.* 39 (2006), Nr. 6, S. 1034–1043. – ISSN 0031-3203
- [Chromium] CHROMIUM: *Chromium: A system for interactive rendering on clusters of graphics workstations.* – URL <http://chromium.sourceforge.net/>
- [Cruz-Neira u. a. 1992] CRUZ-NEIRA, Carolina ; SANDIN, Daniel J. ; DEFANTI, Thomas A. ; KENYON, Robert V. ; HART, John C.: The CAVE: audio visual experience automatic virtual environment. In: *Commun. ACM* 35 (1992), Nr. 6, S. 64–72. – ISSN 0001-0782
- [Dijkstra 1982] DIJKSTRA, Edsger W.: *Selected writings on computing: a personal perspective.* New York, NY, USA : Springer-Verlag New York, Inc., 1982. – 60–66 S. – ISBN 0-387-90652-5
- [Google Earth] GOOGLE EARTH: *Google Earth.* – URL <http://earth.google.de/>. – Zuletzt besucht: 17.07.2008
- [Gosling u. a. 1996] GOSLING, James ; JOY, Bill ; STEELE, Guy: *The Java Language Specification.* First Edition. Addison Wesley, 1996. – 825 S. – URL http://java.sun.com/docs/books/jls/first_edition/html/index.html. – ISBN 978-0201634518
- [Heitsch 2008] HEITSCH, Johann: Ein Framework zur Erkennung von dreidimensionalen Gesten / Hochschule für Angewandte Wissenschaften Hamburg. 2008. – Bachelorarbeit
- [Ishii u. a. 1994] ISHII, Hiroshi ; KOBAYASHI, Minoru ; ARITA, Kazuho: Iterative design of seamless collaboration media. In: *Commun. ACM* 37 (1994), Nr. 8, S. 83–97. – ISSN 0001-0782
- [JACOB] JACOB. – URL <http://sourceforge.net/projects/jacob-project/>. – Zuletzt besucht: 21.07.2008
- [JSON] JSON. – URL <http://www.json.org/>. – Zuletzt besucht: 21.07.2008
- [Kendon] KENDON, Adam: *Agenda for Gesture Studies.* – URL <http://www.univie.ac.at/wissenschaftstheorie/srb/srb/gesture.html>. – Zuletzt besucht: 24.07.2008
- [Kodak] KODAK: *Kodak Theatre HD Player.* – URL http://www.kodak.com/eknec/PageQuerier.jhtml?pq-path=13112&pq-locale=en_US. – Zuletzt besucht: 24.07.2008
- [McNeill] MCNEILL, David: *McNeill Lab: Center for gesture and speech research.* – URL <http://mcneilllab.uchicago.edu/>. – Zuletzt besucht: 24.07.2008

- [Menache 2000] MENACHE, Alberto: *Understanding Motion Capture for Computer Animation and Video Games*. Morgan Kaufmann Publishers, 2000
- [Meta Motion] META MOTION: *Gypsy motion capture system*. – URL <http://www.metamotion.com/gypsy/gypsy-motion-capture-system.htm>. – Zuletzt besucht: 23.07.2008
- [Napitupulu 2008] NAPITUPULU, Jan: *Ein System mit skalierbarer Visualisierung zur Entwicklung kollaborativer Serious Games*, Hochschule für Angewandte Wissenschaften Hamburg, Diplomarbeit, 2008. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/master/napitupulu.pdf>
- [Network Working Group 2001] NETWORK WORKING GROUP: *RFC3170 - IP Multicast Applications: Challenges and Solutions*. September 2001. – URL <http://www.faqs.org/rfcs/rfc3170.html>. – Zuletzt aufgerufen: 16.07.2008
- [New York Times 2007] NEW YORK TIMES: *Beowulf and Motion Capture*. November 2007. – URL <http://www.nytimes.com/2007/11/18/movies/18kehr.html>. – Zuletzt besucht: 23.07.2008
- [Nintendo] NINTENDO: *Nintendo Wii*. – URL http://www.nintendo.de/NOE/de_DE/wii_54.html. – Zuletzt besucht: 24.07.2008
- [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 Trans. Pattern Anal. Mach. Intell.* 19 (1997), Nr. 7, S. 677–695. – ISSN 0162-8828
- [Pierce und Nichols] PIERCE, Jeff ; NICHOLS, Jeff: *Personal Information Environments*. – URL <http://www.almaden.ibm.com/cs/projects/pie/>. – Zuletzt besucht: 30.07.2008
- [Processing] PROCESSING. – URL <http://processing.org/>. – Zuletzt besucht: 21.07.2008
- [Queck u. a. 1995] QUECK, K. H. ; MYSLIWIEC, T. ; ZHAO, M.: Finger mouse: A free-hand pointing interface. In: *Proc. of Internat. Workshop on Automatic Face and Gesture Recognition*. Zürich, Switzerland, 1995, S. 372–377
- [Rahimi und Vogt 2008] RAHIMI, Mohammadali ; VOGT, Matthias: *Gestenbasierte Computerinteraktion auf Basis von Multitouch-Technologie / Hochschule für Angewandte Wissenschaften Hamburg*. 2008. – Bachelorarbeit
- [Ramachandran 2002] RAMACHANDRAN, Vijay: *Design Patterns for Building Flexible and Maintainable J2EE Applications*. Januar 2002. – URL <http://java.sun.com/developer/technicalArticles/J2EE/despat/>. – Zuletzt besucht: 17.07.2008

- [Robert 2006] ROBERT, Philippe C. D.: Methoden zur skalierbaren Visualisierung / Universität Bern. URL http://cgg.unibe.ch/teaching/lectures/ss06/seminar-computergeometrie-und-grafik/hmm/philippe_robert_skalierbare_visualisierung.pdf, 2006. – CGG Seminar SS06. Zuletzt besucht: 24.07.2008
- [Sun Microsystems] SUN MICROSYSTEMS: *Developer Resources for Java Technology*. – URL <http://java.sun.com/>. – Zuletzt besucht: 21.07.2008
- [University of Alaska Fairbanks] UNIVERSITY OF ALASKA FAIRBANKS: *UAF CS Bioinformatics Powerwall*. – URL <http://www.cs.uaf.edu/2007/powerwall/>. – Zuletzt besucht: 24.07.2008
- [Vicon] VICON: *Motion Capture Systems*. – URL <http://www.vicon.com/index.html>. – Zuletzt besucht: 23.07.2008
- [Virtual Soldier Research 2004] VIRTUAL SOLDIER RESEARCH: *Virtual Soldier Research End Of Year Report 2004*. 2004. – URL <http://www.digital-humans.org/Report2004/>. – Zuletzt besucht: 24.07.2008
- [W3C] W3C: *XML*. – URL <http://www.w3.org/XML/>. – Zuletzt besucht: 21.07.2008
- [Watzlawick 1996] WATZLAWICK, Paul: *Menschliche Kommunikation. Formen, Störungen, Paradoxien*. 9. Hans Huber, 1996. – ISBN-13: 978-3456828251
- [Weiser 1991] WEISER, Mark: The Computer for the 21st Century. In: *Scientific American* (1991), Februar. – URL <http://www.ubiq.com/hypertext/weiser/SciAmDraft3.html>
- [WIWITA 2008] WIWITA: *6. Wismarer Wirtschaftsinformatik-Tage*. 2008. – URL <http://www.wi.hs-wismar.de/~cleve/wiwita.html>. – Zuletzt besucht: 24.07.2008
- [XEROX PARC 1978–79] XEROX PARC: *MVC*. 1978-79. – URL <http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html>. – Zuletzt besucht: 17.07.2008

Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §24(5) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 11. August 2008

Ort, Datum

Unterschrift