



Hochschule für Angewandte Wissenschaften Hamburg  
*Hamburg University of Applied Sciences*

# Bachelorarbeit

Mohammadali Rahimi, Matthias Vogt

Gestenbasierte Computerinteraktion auf Basis  
von Multitouch-Technologie

Mohammadali Rahimi, Matthias Vogt  
Gestenbasierte Computerinteraktion auf Basis von  
Multitouch-Technologie

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

Abgegeben am 18. August 2008

**Mohammadali Rahimi, Matthias Vogt**

**Thema der Bachelorarbeit**

Gestenbasierte Computerinteraktion auf Basis von Multitouch-Technologie

**Stichworte**

Multitouch, Gesten, Computerinteraktion, Seamless Interaction, Intuitive Bedienung

**Kurzzusammenfassung**

In der heutigen Zeit der allgegenwärtigen Computer und des „Information Overload“, spielt die intuitive und handbuchfreie Bedienung von Computern eine immer wichtigere Rolle. Im Rahmen dieser Bachelorarbeit wurden zwei Systeme entwickelt, welche diese neuen Bedienkonzepte mittels Multitouch-Technologie umsetzen. Hierzu wurde eine Reihe von Gesten eingeführt mit denen die entwickelte Multitouch-Wall und das TouchTable bedient werden können.

**Mohammadali Rahimi, Matthias Vogt**

**Title of the paper**

Gesture based computer interaction based on Multitouch-Technology

**Keywords**

Multitouch, Gestures, Human-Computer-Interaction, Seamless-Interaction, Intuitive manipulation

**Abstract**

In times of ubiquitous computing and information overload, an intuitive and guide-free use of computers plays an increasingly important role. In the course of this thesis, two systems were developed realizing these new operating techniques by multi-touch technology. Therefore a series of gestures was installed in order to use the Multi-Touch Wall and the Touch-Table System.

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>7</b>
<b>2</b>	<b>Grundlagen</b>	<b>10</b>
2.1	Mensch-Computer-Interaktion . . . . .	10
2.1.1	Co-located Computer-Supported Cooperative Work (CCW) . . . . .	11
	Ubiquitous Computing . . . . .	12
	Vom Personal Information Environment (PIE) zum Group Information Environment (GIE) . . . . .	12
	Context-Aware Applications . . . . .	13
2.2	Seamless Interaction . . . . .	14
2.2.1	Mental Models . . . . .	15
2.3	Techniken (Hardware) . . . . .	15
2.3.1	Touch Technologien . . . . .	15
2.3.2	Von Single- zu Multi-Touch . . . . .	22
2.3.3	Erhältliche Touch Hardware . . . . .	23
2.4	Fazit . . . . .	30
<b>3</b>	<b>Analyse</b>	<b>31</b>
3.1	Szenarien . . . . .	31
3.1.1	Einsatzleitzentrale . . . . .	31
3.1.2	Messe Szenario . . . . .	32
3.1.3	Vergleich der Szenarien . . . . .	34
3.2	Interaktion mit Multitouch Systemen . . . . .	36
3.3	Anforderungsanalyse . . . . .	37
3.3.1	Funktionale Anforderungen . . . . .	37
	Anwendungsfälle . . . . .	37
	Interaktion mit Information . . . . .	38
	Benötigte Benutzeraktionen zum Informationsgewinn . . . . .	39
	Sinnvolle Gesten zu den Aktionen . . . . .	40
	Resultierende Interaktion . . . . .	41
	Bedienbarkeit . . . . .	42
	Performance . . . . .	43
	Benutzeranzahl . . . . .	43

---

3.3.2	Nichtfunktionale Anforderungen	44
	Zuverlässigkeit	44
	Portierbarkeit	44
	Wartbarkeit	45
3.4	Fazit	45
<b>4</b>	<b>Gesten</b>	<b>46</b>
4.1	Gesten Taxonomie	47
4.2	Kategorisierung	48
4.2.1	Bewegungsverfolgung	48
4.2.2	Kontinuierliche Gesten	49
4.2.3	Symbolisch-manipulative Gesten	49
4.3	Start- Stoppbedingungen von Gesten	51
4.4	Gesten in Anwendungskontexten	51
4.5	Gesten in Multitouch Systemen	52
4.6	Fazit	54
<b>5</b>	<b>Design und Realisierung</b>	<b>55</b>
5.1	Anwendungskontext	55
5.1.1	Gewählte Hardware	57
5.2	Grundlegende Funktionalitäten von Multitouch Wand und -Tisch	59
5.2.1	Bedienoberfläche der Multitouch Wand	61
5.2.2	Funktionalitäten der Multitouch Wand	62
5.2.3	Bedienoberfläche des Multitouch-Tisches	64
5.2.4	Funktionalität des Multitouch-Tisches	65
5.3	Architektur	66
5.3.1	MVC	67
5.3.2	MVC in der Multitouch Architektur	68
	View	69
	Model	70
	Controller	72
5.4	Technische Umsetzung	82
5.4.1	Oberflächengestaltung	82
5.4.2	Verwendete und entwickelte Komponenten	84
5.4.3	Kompensation von fehlenden Informationen der Hardware	88
5.4.4	Kommunikation innerhalb der Anwendung	89
5.5	Evaluation	90
5.5.1	Durchführung der Evaluation	91
5.5.2	Erkenntnisse aus der Evaluation	91
	Grundlegende Funktionen	91
	Erlernbarkeit der Funktionen	92

---

Mitgebrachtes Wissen über Gesten und Interaktion . . . . .	92
Ergonomie der Gesten . . . . .	93
5.6 Bewertung der Umsetzung . . . . .	94
<b>6 Schluss</b>	<b>95</b>
6.1 Zusammenfassung . . . . .	95
6.2 Ausblick . . . . .	96
<b>Literaturverzeichnis</b>	<b>100</b>
<b>Glossar</b>	<b>106</b>

# 1 Einführung

Mark Weiser sagte in einem Interview ([Telepolis \(1999\)](#)) über Ubiquitous Computing:

„[...] Es geht um das unsichtbare Interface. Es ist eine Art von Benutzerschnittstelle - aber eine, zu der man eine andere Beziehung hat, weil man eigentlich nicht weiß, daß sie da ist. Ubiquitous Computing basiert auf der Idee, daß wir uns - um die Menschen in eine vertrautere Beziehung mit der Technologie zu bringen - an den Dingen orientieren müssen, mit denen die Leute schon vertraut sind. Mit Dingen aus ihrem Leben, ihrer Umgebung, ihrer Zimmereinrichtung, ihren Bücherregalen. Also Dingen, die Nutzer normalerweise nicht mit „Computer“ verbinden. Das große Problem mit dem PC ist, daß alles über diesen Tunnel, diesen engen Kanal zwischen Bildschirm, Tastatur und Maus läuft. Das ist sehr unnatürlich, und es muß nicht so sein. Man kann Informationen aus seiner ganzen Umgebung beziehen, auch digitale Informationen. Buchdeckel, Bildschirme oder Uhren - alle Gegenstände können uns aktiv mit Informationen versorgen, zum Beispiel über intelligente Software-Agenten.“

Technologien, welche heute zum Einsatz kommen, um dieses von Weiser aufgestellte Ziel zu erreichen, sind unter anderem Sprachsteuerungen, Motion Tracking oder Multitouch. Durch technologische Fortschritte wird zurzeit weltweit in verschiedenen Laboren an diesen Technologien im Einzelnen und in Kombination gearbeitet, um Computer in den Alltag zu integrieren.

Die Herausforderung ist nicht nur, das Verständnis für die Technologien zu schaffen, sondern auch die Art der Bedienung an die neuen Möglichkeiten anzupassen. Den „Tunnel“, den Weiser beschreibt, schränkt die Bedienung erheblich ein und erfordert die Anpassung des Menschen an die Maschine. Ziel sollte eine Anpassung der Computerumgebung und deren Bedienung an den Menschen sein.

Das Gebiet der Seamless Interaction ([Hiroshi Ishii \(2008\)](#)) nimmt sich der Herausforderung an, genau diese einfache Mensch-Computer-Interaktion umzusetzen. Die nahtlose Interaktion ist eine Voraussetzung für das Gelingen der Interaktion mit allgegenwärtigen Computersystemen. Mit dem Ziel die zwischenmenschliche Kommunikation zu unterstützen und nicht die Bedienung der Systeme in den Vordergrund zu stellen, gewinnt der Begriff



Abbildung 1.1: iRoom Stanford (von [Stanford \(2008a\)](#))

„Seamless Interaction“ eine große Bedeutung im Rahmen der Mensch-Computer-Interaktion und dem Ubiquitous Computing.

Der Einsatz von natürlicher Interaktion wird in Bereichen von kollaborativen Arbeitsumgebungen wie sie [Greif](#), [Winograd](#) (s. Abb.1.1) und [Tandler](#) beschreiben außerordentlich wichtig. Die eigentliche Arbeit mit Computern und vor allem untereinander muss im Vordergrund stehen und darf nicht aufgrund komplizierter Bedienung in den Hintergrund rücken. Um dies zu gewährleisten sind neue kooperative Interaktionsmöglichkeiten von Nöten.

Die Erfolge des iPhone, oder anderen intuitiv bedienbaren Produkten, zeigt die Notwendigkeit der genauen Betrachtung dieser Konzepte der natürlichen Bedienung unter Zuhilfenahme neuer Technologien. Es sind mittlerweile viele dieser Technologien weit genug entwickelt, dass intuitive Interaktionen damit realisiert werden können. Eine dieser Technologien in diesem Bereich ist die Touch-Technologie, die das direkte Anfassen ermöglicht und somit Indirektionen auflöst. Durch die Möglichkeit mehrere Berührungspunkte zu verwenden und die dadurch entstehende Vielfalt an Variationen macht die Multitouch Technologie zu einem Gegenstand dieser Arbeit. Viele Ansätze zur Realisierung wurden in den letzten Jahren vorgestellt und haben einen ausreichenden Entwicklungsstatus erreicht. Unter anderen trug Jeff Y. Han mit seinem Multitouch Ansatz dazu bei die Visionen von Weiser, Winograd und anderen näher zu kommen. Ziel dieser Arbeit ist es, Multitouch Systeme auf ihre natürliche Bedienung hin zu untersuchen und den Einsatz in verschiedenen Umgebungen zu ermöglichen. Mentale Modelle und die daraus resultierenden Gesten dienen als Grundlage zur Interaktion mit diesem System.



## Gliederung der Arbeit

In Kapitel 2 werden die notwendigen Grundlagen erläutert, zu denen sowohl konzeptionelle Ideen und Theorien, aber auch die verwendete Hardware gehören. Darunter fallen die Mensch-Computer-Interaktion (HCI) und deren Ausprägungen, eine Einführung in Computer-Supported Cooperative Work oder auch Personal Information Enviroments. Der Überblick über die verschiedenen Hardware Lösungen zeigt ihre Funktionsweise und ihr Eigenschaften. Am Ende des Kapitels findet sich eine Übersicht über heute Markt erhältlichen Produkte, die Multitouch einsetzen.

In der Analyse (s. Kapitel 3) werden anhand von Szenarien die Anforderungen an eine Interaktion mit Multitouch Systemen definiert. Es werden Aktionen erläutert, die benötigt werden, um mit Computersystemen zu arbeiten und anschließend einen Zusammenhang mit Gesten gestellt.

Um ein tieferes Verständnis für Gesten und deren Bedeutung zu schaffen, wird in Kapitel 4 speziell auf Gesten eingegangen. Es wird eine Einordnung von Gesten vorgenommen, um eine Abgrenzung und eine Idee für die Gesten und den entsprechend zugeordneten Aktionen zu schaffen.

In Teil 5 der Arbeit wird auf Grundlage der Analyse und Gesten ein mögliches Design vorgestellt, welches den Einsatz von Multitouch in Computersystem ermöglichen soll. Es wird eine Einordnung in einem Anwendungskontext vorgenommen und es werden die Funktionalitäten von zwei Anwendungen vorgestellt, welche im Rahmen dieser Arbeit entwickelt wurden. Danach wird speziell auf das entworfene Design eingegangen und in Einzelheiten vorgestellt. Es folgt die technische Umsetzung der Anwendungen, welche sich an dem aufgestellten Design orientieren. Anschließend wird über vorgenommene Evaluationen berichtet, welche bezüglich der implementierten Anwendungen durchgeführt wurden.

Zum Schluss wird in Kapitel 6 ein Fazit gezogen. Es enthält eine Zusammenfassung und zeigt mögliche zukünftige Entwicklungen.

## 2 Grundlagen

Die Grundlagen zum besseren Verständnis dieser Ausarbeitung werden hier kurz angeführt. Sie erheben nicht den Anspruch vollständig zu sein, sie dienen als Grundstock-Informationen, die bei Bedarf weiter vertieft werden können. Einen Einstieg hierfür bietet das große Gebiet der Mensch-Computer-Interaktion (2.1). Darauf folgt eine Einführung in die nahtlose Interaktion (2.2), welche eine wichtige Grundlage für diese Arbeit darstellt. Für die Realisierung sind die technischen Gegebenheiten interessant, welche in dem Kapitel Techniken (2.3.1) erörtert werden. Dazu gehören auch die konkreten und bisherigen Umsetzungen dieser Technologien (2.3.3).

### 2.1 Mensch-Computer-Interaktion

Es gibt keine allgemein gültigen und beschlossenen Definitionen der vielen Themen, welche das Gebiet der interdisziplinären Mensch-Computer-Interaktion oder Human-Computer-Interaction (HCI) beschreiben. Trotzdem benötigt man einen Ansatz, um einen Einstieg in dieses Thema zu erleichtern. So bezeichnen viele HCI als eine Disziplin, welche sich mit dem Design, der Evaluation und Implementierung von interaktiven Computersystemen für den menschlichen Gebrauch und der Erforschung verwandter Themen beschäftigt. Aus Sicht der Informatik liegt der Fokus auf der Interaktion und speziell auf der Interaktion zwischen einem oder mehreren Menschen mit einem oder mehreren Computern. Für diese Arbeit wichtig sind hierbei die Themen: Co-located Collaborative Work, Ubiquitous Computing, Personal Information Environment, Context-Awareness und Seamless Interaction.

Eine klassische Situation ist bei HCI ein Mensch, der am PC interaktiv mit einem Programm mit grafischer Oberfläche arbeitet (s. Abb. 2.1). Es wird hierbei sofort klar, dass bei variierendem Verständnis für Interaktion, Mensch und Maschine eine große Fülle von möglichen, zu betrachtenden Themengebieten zu beleuchten sind (ACM (2008)). Bei HCI geht es also um die vereinte Leistung von Mensch und Computer und die Fähigkeit des Menschen die Leistung die der Computer erbringt, effektiv zu nutzen. Dies wird erreicht, indem eine einfache Erlernbarkeit der Interaktion mit dem Computersystem gewährleistet wird (2.2).

Geprägt wurde die Forschung an HCI durch Hiroshi Ishii (Ishii u. a. (1994)), welcher diesen Bereich in Kombination mit Computer-Supported Cooperative Work (2.1.1) durch seine Veröffentlichungen stark beeinflusst hat (Hiroshi Ishii (2008)).

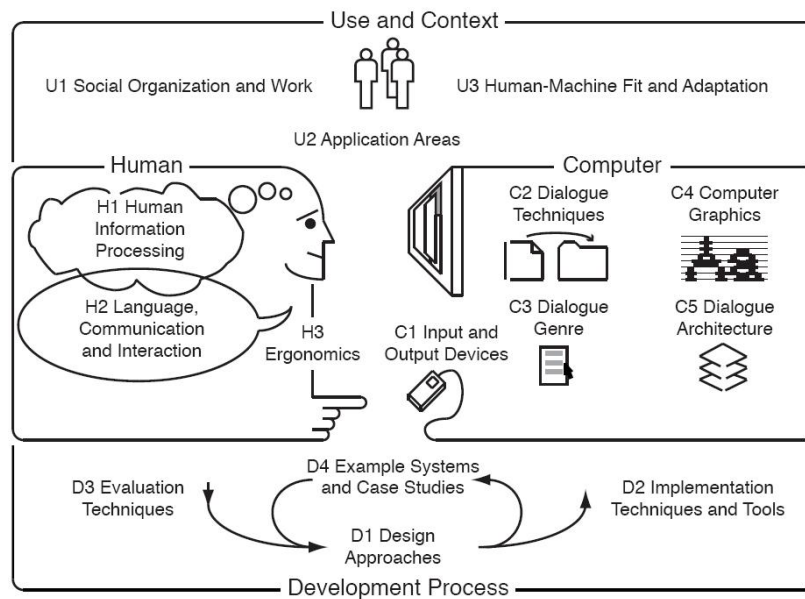


Abbildung 2.1: HCI (2.1) grafische Darstellung aus ACM (2008)

Auch bekannt für ihre Ergebnisse sind Forscher von der Carnegie Mellon University (Carnegie Mellon (2008)) und aus Stanford (Stanford (2008b)).

### 2.1.1 Co-located Computer-Supported Cooperative Work (CCW)

CCW ist die Verbindung der Fachgebiete Informatik, Gesellschaftswissenschaften, Psychologie und weiteren Disziplinen, es geht also darum, wie Computersysteme kollaborative Arbeiten unter Menschen und deren Koordination unterstützen können. Entstanden ist es im Rahmen eines Workshops von Irene Greif und Paul M. Cashman im Jahr 1985 (Brown u. a. (1985)). In dieser Arbeit wird davon ausgegangen, dass sich alle beteiligten Personen zur gleichen Zeit im gleichen Raum befinden, daher die zusätzliche Notation der Co-location (s. Abb. 2.2). Die zusammenarbeitenden Teams im CCW befinden sich im Group Information Environment (2.1.1), da sie sich die auf den Systemen vorhandenen Informationen teilen und zusammen erarbeiten. Computersysteme solcher Umgebungen gehören zur Ausstattung und aufgrund der Allgegenwärtigkeit dieser Systeme spricht man vom Ubiquitous Computing (s. 2.1.1).

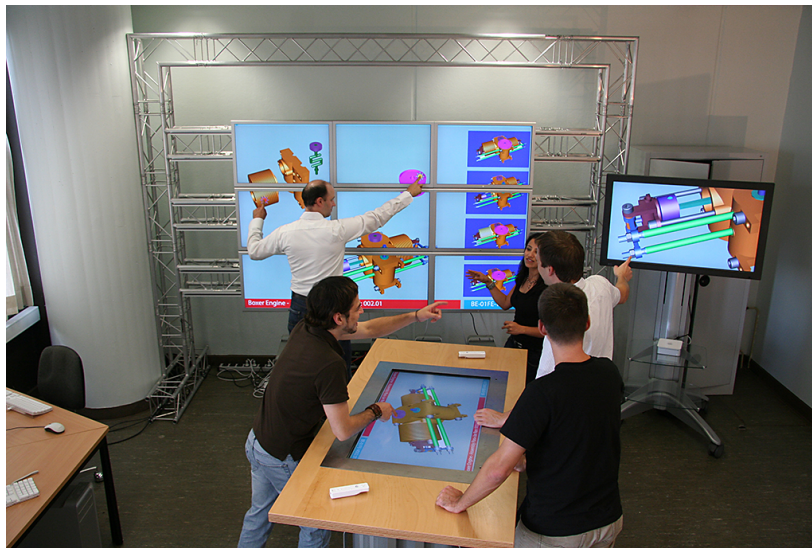


Abbildung 2.2: Beispielhaftes (2.1.1) CCW an der HAW

### Ubiquitous Computing

Ubiquitous Computing ist die dritte Ära der Computer, in der ein Mensch viele verschiedene Computersysteme zur Verfügung hat. Es ist die Zeit der sogenannten „calm technology“, in der die Technologie in den Hintergrund des alltäglichen Lebens tritt (s. Abb. 2.3).

Begonnen hat das Computer-Zeitalter mit den Mainframes, die von vielen Benutzern geteilt wurden. Darauf folgte die Zeit des Personal Computers(PC), in der ein Mensch einen Computer bedient. Heutzutage befinden wir uns in der Zeit zwischen dem PC und Ubiquitous Computing, in der Personal Information Environments eine Rolle spielen (2.1.1). Beschrieben wurde dieses Phänomen erstmals von Mark Weiser in seinem Paper „The Computer for the 21st Century“ von 1991(Weiser (1991)). Die übergangslose oder nahtlose Interaktion mit allen Geräten bezeichnet man als Seamless Interaction (s. 2.2).

### Vom Personal Information Environment (PIE) zum Group Information Environment (GIE)

Heutige Arbeitsplätze sind sogenannte PIEs, nach Jeff Pierce(Jeff Pierce (2008)). Die steigende Leistung und die sinkenden Preise von Computersystemen führen dazu, dass ein Benutzer mehrere dieser Systeme besitzt und mit dieser heterogenen Sammlung von Geräten (z.B. Desktops, Laptops, Tablets, PDAs, Handys und andere) interagiert (s. Abb. 2.4).

Um eine Transparenz zwischen all diesen Geräten zu erhalten, bieten viele größere Hersteller von Betriebssystemen Lösungen an, die es dem Benutzer ermöglichen, auf allen

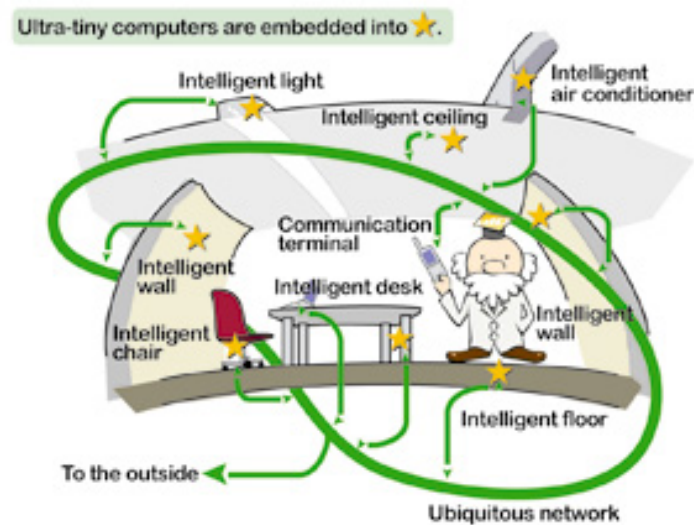


Abbildung 2.3: Beispielhaftes (2.1.1) Schema für Ubiquitous Computing (Ubiquitous ID Center (2008))

eingesetzten Geräten einen konsistenten Status zu halten, die Geräte haben somit eine Art Context-Awareness (2.1.1). Von Apple wird hierzu das serverbasierte System MobileMe (Apple Inc. (2008c)) angeboten, welches zusätzlichen Webspace bietet. Microsoft entwickelt zur Zeit LiveMesh (Microsoft Corporation (2008)), welches ähnliche Funktionalitäten bereit stellt.

Es wird an dieser Stelle der Begriff des GIE (Group Information Environment) eingeführt, da die Informationen in der Arbeitsumgebung von Teams verteilt vorliegen. In CCWs werden die Informationen auf den allgegenwärtigen Computersystemen in dieser Umgebung für alle Beteiligten bereitgestellt und sie können somit in einem Teamarbeitsprozess zusammen betrachtet und manipuliert werden. Computersysteme, die bei diesen Arbeitsprozessen assistierend wirken, müssen die Fähigkeit besitzen, sich auf ihre aktuelle Umgebung einzustellen, um nahtlose Interaktion zu ermöglichen.

### Context-Aware Applications

Bei Context-Aware Applications (engl. für kontextsensitive Applikationen), werden Informationen aus der Umgebung eingesetzt, die das Verhalten des Systems auf den Kontext anpassen (Schilit u. a. (1994)). Diese Systeme können durch Sensoren ihre Umgebung wahrnehmen und auf Veränderungen in dieser reagieren. Häufig werden mehrere Typen unterschiedlicher Sensoren eingesetzt, welche dann mittels Sensor-Fusion, also die Zusammenführung der Informationen dieser Sensoren, vereint werden.



Abbildung 2.4: Beispielhaftes PIE (2.1.1) von [Jeff Pierce \(2008\)](#)

## 2.2 Seamless Interaction

Seamlessness besteht im Wesentlichen aus zwei Bestandteilen.

Zum einen die sogenannte „smooth transition“, welche für einen nahtlosen Übergang unterschiedlicher Arbeitsumfelder steht, denn bei kollaborativen Umgebungen findet häufig ein schneller Wechsel zwischen verschiedenen Funktionsräumen und -modi statt. Somit entlastet die Seamless Interaction den kognitiven Aufwand des Benutzers, indem es ihm ermöglicht wird, dynamisch zwischen diesen Funktionsräumen zu wechseln.

Zum anderen besteht die Seamlessness aus der „continuity“, die für eine gleichbleibende Bedienung aller eingesetzten Arbeitsumfelder 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. \(1994\)](#)). Um nahtlose Interaktion zu ermöglichen, ist es nötig zu wissen, wie sich Menschen unbekannte Gegenstände erklären. Einen Hinweis darauf bieten die Mental Models.

### 2.2.1 Mental Models

Die Bestandteile der Seamless Interaktion berufen sich auf die sogenannten Mental Models. Mental Models ([Craik \(1943\)](#)) drücken die Erwartungshaltung eines Benutzers aus, wenn dieser vor einem unbekanntem System steht. Über diese Mental Models erklären sich Menschen die Funktionsweise von allen unbekanntem Gegenständen der Realität. Die gewählten Metaphern bei der Erstellung eines Systems, müssen nahe genug an der Realität sein, so dass das Mental Model eines Benutzers eine maximal große Überschneidung mit der Funktionsweise des Systems hat. Auf der Basis von passend gewählten Metaphern für die System-Funktionsweise, lässt sich so eine intuitive Bedienung realisieren.

Die Verwendung direkter Eingabe-Technologien ermöglicht eine Verringerung der Diskrepanz zwischen Metapher und Realität. Aus diesem Grund werden nachfolgend einige Touch Technologien vorgestellt, welche es dem Benutzer ermöglichen, mittels Eingabe durch Finger, auf direktem Wege mit dem Computersystem zu interagieren.

## 2.3 Techniken (Hardware)

Es gibt unterschiedliche Techniken zur Realisierung von Touch-Technologie. Auf Basis dieser grundlegenden Technologien ([2.3.1](#)) und der in [2.3.3](#) beschriebenen auf dem Markt erhältlichen Hardware, wird in der Analyse die Entscheidung getroffen, welche Hardware für das im Rahmen dieser Arbeit entwickelte System am Besten geeignet ist.

Es wird im Folgenden kurz dargestellt, welche Techniken sich zur Zeit durchgesetzt haben und schon heute auf dem freien Markt erhältlich sind.

### 2.3.1 Touch Technologien

Die Daten, zu den hier beschriebenen Touch-Technologien, außer der FTIR Technik, können unter [Touchscreenguide \(2008\)](#) genauer nachgelesen werden.

**Resistive** Touchscreens bestehen aus zwei Layern, die jeweils mit einer leitfähigen Substanz auf deren Innenseiten beschichtet sind. Diese leitenden Layer sind durch kleine, gleichmäßig verteilte Gummi-Noppen getrennt. Durch das Auflegen eines Objektes und der Ausübung eines gewissen Druckes, wird ein interner elektrischer Kontakt am angefassten Punkt erzeugt. Dieser versorgt den Controller-Baustein mit vertikalen und horizontalen Daten, welche dann in die entsprechende Bildschirmposition übersetzt werden.

Es gibt 4-, 5-, 7- und 8-Draht resistive Touchscreens, abhängig von der Anzahl der

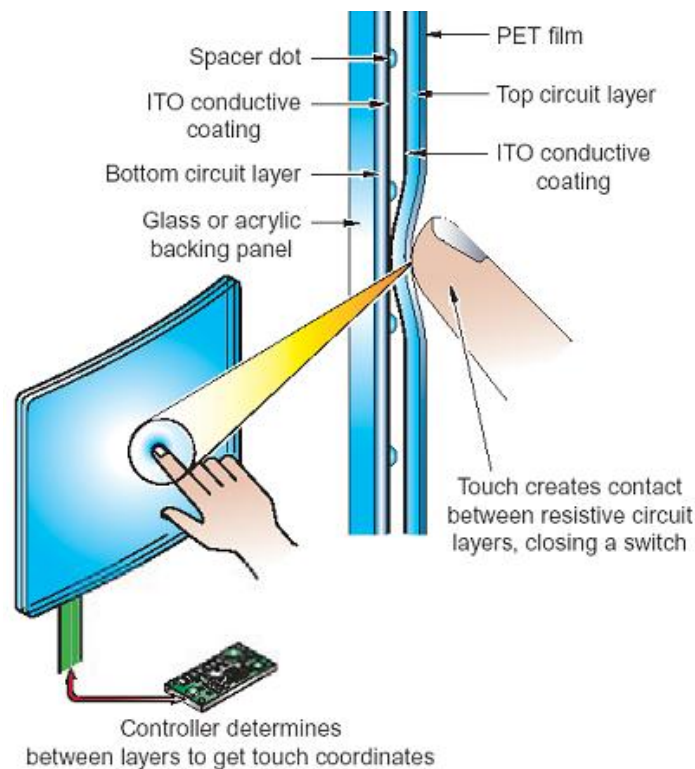


Abbildung 2.5: Funktionsweise eines resistiven Touch-Sensors (2.3.1) von [Touchscreenguide \(2008\)](#)

Drähte, welche vom Sensor zum Controller geführt werden (s. Abb. 2.5).

Der 5-Draht Touch ist der, aufgrund des guten Preis/Leistung Verhältnisses, am häufigsten eingesetzte. Diese Technologie wird in anspruchsvollen Umgebungen eingesetzt, die sich durch einen rauen Umgang mit den Geräten auszeichnen. Sie sind relativ resistent gegen Flüssigkeiten, die in diesem Umfeld vorkommen, wie z.B. Kaffee, Essig, Reinigungsflüssigkeiten und Benzin.

**Kapazitive** Touchscreen Sensoren haben im Vergleich zu den resistiven Sensoren eine klarere Durchsicht, sind widerstandsfähiger und haltbarer. Sie bestehen aus einem Overlay aus Glas, welches mit einem elektrischer Ladung speichernden Material beschichtet ist. Auf jede der vier Ecken des Touch-Sensors wird eine sich verändernde elektrische Spannung angelegt. Wenn der Sensor nun angefasst wird, entlädt sich die Ladung an der Oberfläche und es fließt ein bestimmter Strom zu den Ecken. Die Stärke dieses Stromes an den Ecken wird genutzt um die Position des Fingers zu ermitteln (s. Abb. 2.6).

Der Nachteil dieser Technik ist, dass der Sensor nur durch einen Finger oder mit einem leitenden Stylus bedient werden kann.



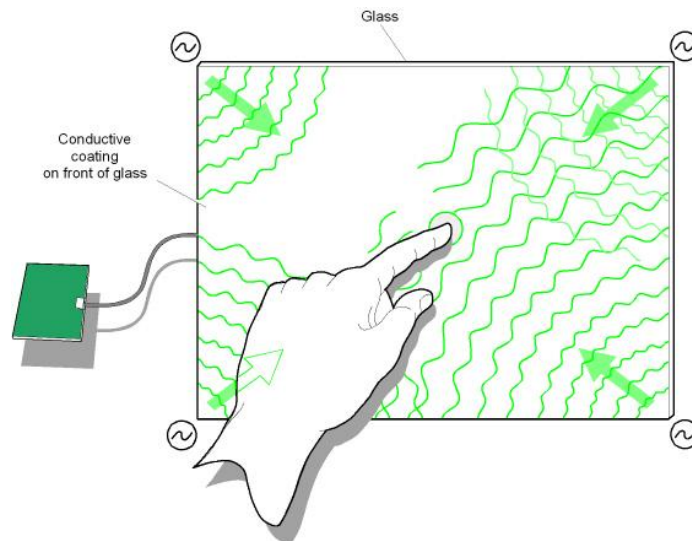


Abbildung 2.6: Funktionsweise eines kapazitiven Touch-Sensors (2.3.1) von [Touchscreen-guide \(2008\)](#)

**Near Field Imaging (NFI)** Technologie Touchscreens bestehen aus zwei laminierten Glasscheiben, mit einer transparenten und gemusterten Metalloxidschicht. Durch das Anlegen einer Wechselspannung an die leitende Metalloxid-Beschichtung wird ein elektrostatisches Feld erzeugt. Wird nun ein leitender Gegenstand, wie z.B. ein Finger, auf den Sensor gehalten, wird das Feld gestört und somit ein Punkt registriert (s. Abb. 2.7).

Die Haltbarkeit dieser Technik ist vergleichbar mit der der kapazitiven Screens, allerdings sehr viel aufwändiger in der Herstellung.

**Surface Acoustic Wave (SAW)** Technologie Touchscreens nutzen eine massive Glasscheibe als Touch-Sensor. Schallwellen für vertikale und horizontale Erkennung werden, unhörbar für das menschliche Ohr, über die Oberfläche des Glases übertragen. Die Wellen, die sich über den Schirm verbreiten, werden durch akustische Reflektoren an den Seiten der Glasscheibe reflektiert und an Empfangsmodulen detektiert. Beim Anlegen eines Fingers absorbiert dieser die Wellen und die akustischen Empfänger können eine Änderung feststellen (s. Abb. 2.8).

Ein großer Nachteil dieser Technik ist, dass ein Objekt welches detektiert werden soll, eine gewisse Absorptionsfähigkeit akustischer Wellen haben muss.

**Infrarot** Touchscreens werden durch einen Rahmen von Infrarotlicht ausstrahlenden LEDs und diametral gegenübergestellten Phototransistoren betrieben. Ein Controller-Baustein leitet eine Folge von Impulsen auf die LEDs, wodurch ein Gitter von unsichtbaren Infrarotstrahlen direkt vor dem Bildschirm erzeugt wird. Die Controller-Einheit

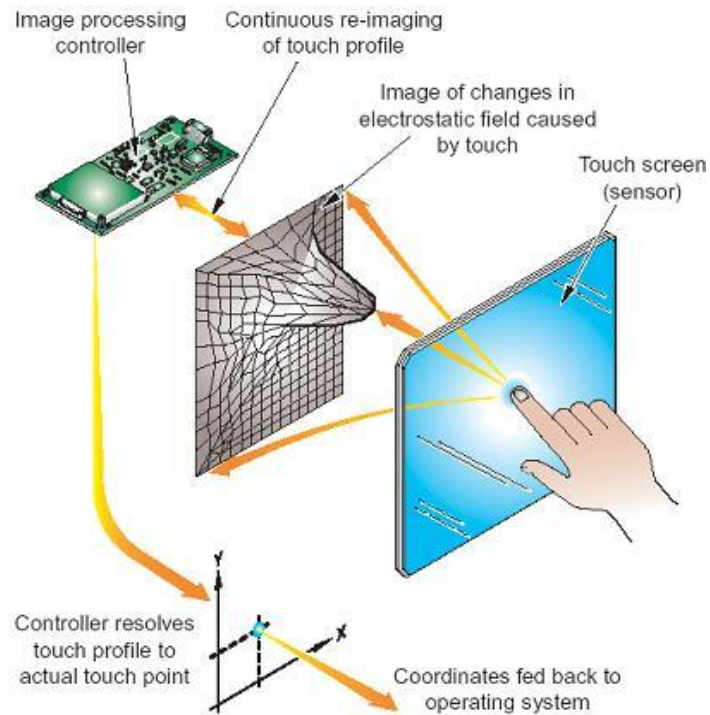


Abbildung 2.7: Funktionsweise eines NFI Touch-Sensors (2.3.1) von [Touchscreenguide \(2008\)](#)

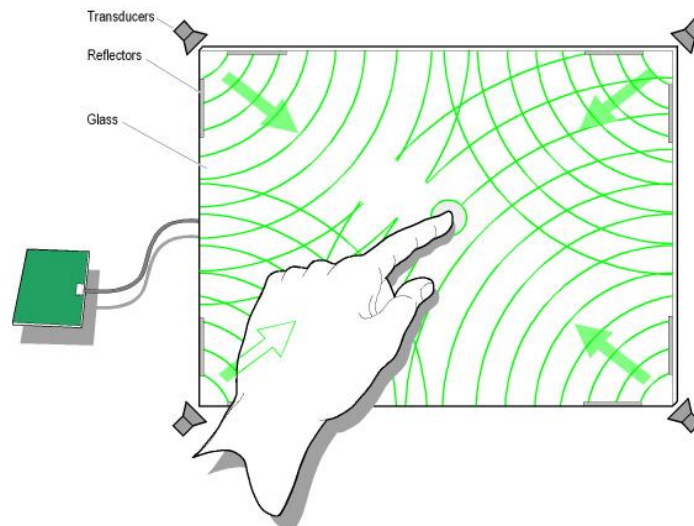


Abbildung 2.8: Funktionsweise eines SAW Touch-Sensors (2.3.1) von [Touchscreenguide \(2008\)](#)

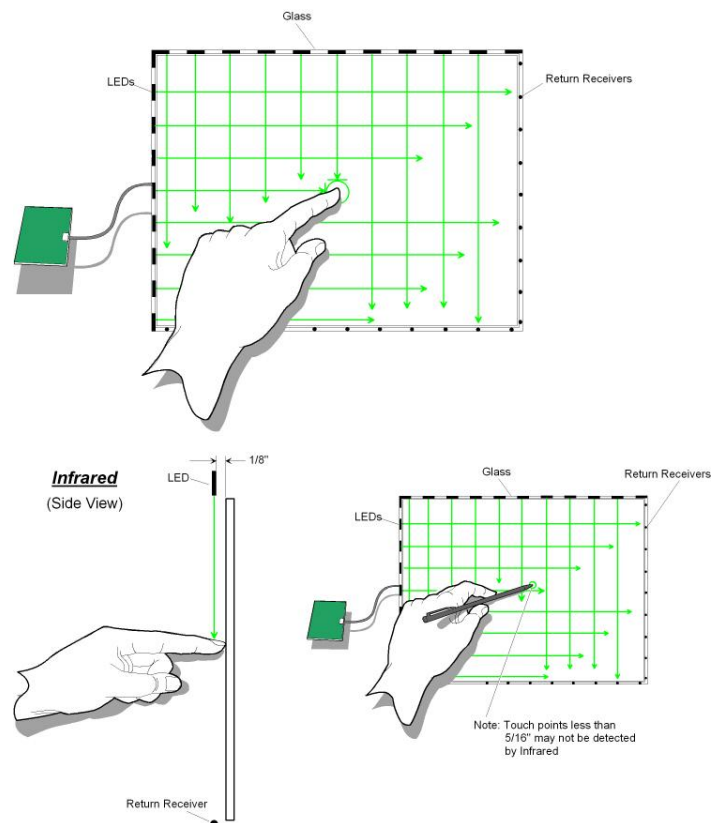


Abbildung 2.9: Funktionsweise eines IR Touch-Sensors (2.3.1) von Touchscreenguide (2008)

kann dann Eingaben an den Positionen erkennen, an denen die Lichtstrahlen unterbrochen werden. Diese Technologie erfordert vom Benutzer keinen Druck bei der Eingabe, welche von einem Finger oder einem beliebigen, undurchsichtigem Objekt stammen kann (s. Abb. 2.9).

Im Gegensatz zu den anderen Technologien ist nur eine einmalige Kalibrierung des Systemes nötig, da die LEDs nur langsam altern und dabei eine relativ konstante Lichtstärke haben.

Ein Nachteil ist, dass keine Berührung zur Aktivierung des Sensors notwendig ist. Dadurch ist dieses System anfällig für unbeabsichtigte Aktivierungen durch einfaches Schweben oberhalb des Bildschirms. Alternativ zu den Phototransistoren können auch Kameras eingesetzt werden (s. Abb. 2.10), die sich an den Ecken des IR-Rahmens befinden (NextWindow Limited (2008)).

**Camera-based Infrared (FTIR/DI)** Touchscreens nutzen im Falle der Frustrated Total Internal Reflexion (FTIR) ein bekanntes Phänomen aus der Optik.

Unter Totalreflexion (Total Internal Reflection: TIR) versteht man eine verlustfreie Re-

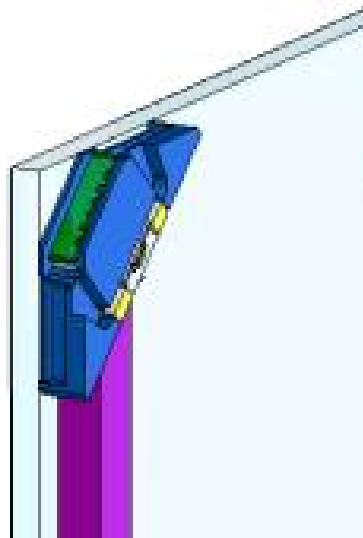


Abbildung 2.10: Funktionsweise eines Kamera basierten IR Touch-Sensors (2.3.1) von [Next-Window Limited \(2008\)](#)

flexion von Licht am Übergang des Mediums, in dem es sich befindet, zu einem mit höherem Brechungsfaktor. Dies wird bei der Datenübertragung mit Lichtleitern schon sehr lange genutzt, um weite Strecken zu überwinden, ohne Signalverstärker verwenden zu müssen.

Wenn nun allerdings am Wellenleiter ein anderes Material anliegt, wird die Totalreflexion verhindert (Im englischen „frustrated“, daher FTIR) und das Licht verlässt an dieser Stelle den Leiter. Um dieses Phänomen für Touchscreens einsetzen zu können, wird eine Acrylplatte als Leiter für Licht aus Infrarot LEDs verwendet, die an einem Rahmen um die Acrylplatte angebracht sind. Orthogonal zu diesem Aufbau wird eine Kamera mit Infrarotlicht-Filter eingesetzt, welche die vorher beschriebenen verhinderten Strahlen filmt und so z.B. einen Finger erkennt ([Han \(2005\)](#)). Auf die Acrylplatte werden häufig Gummischichten aufgebracht, um eine bessere Haptik zu erzeugen (s. Abb. 2.11).

Bei der Diffused Illumination (DI) wird Infrarotlicht auf die Ober- oder Unterseite der Acrylplatte, welche mit einem Diffusor Material belegt ist, bestrahlt. Wenn ein Objekt auf die Oberfläche gelegt wird, reflektiert es das Licht und kann von einer Kamera erkannt werden, welche unterhalb der Fläche angebracht ist (s. Abb. 2.12).

Zu beachten ist, dass hier nur Technologien vorgestellt wurden, die auch Multitouch-fähig sind, also Sensoren besitzen, die auch mehrere Unterbrechungen zur gleichen Zeit angeben können. Hierzu werden im folgenden Teil kurz die Unterschiede dargestellt.

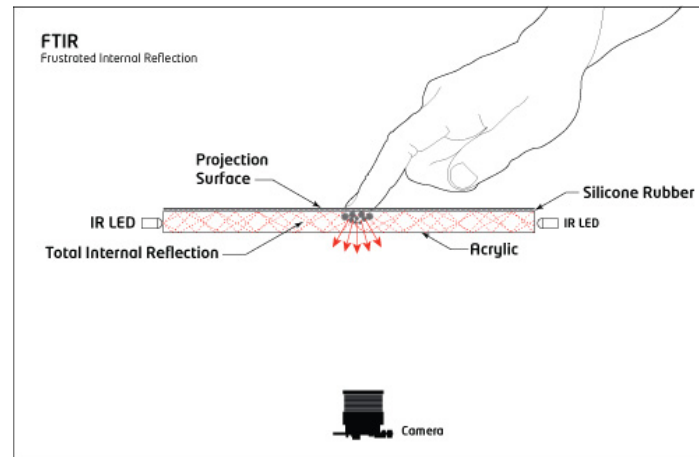


Abbildung 2.11: Funktionsweise eines Kamera basierten IR Touch-Sensors (2.3.1) von [Tim Roth \(2008\)](#)

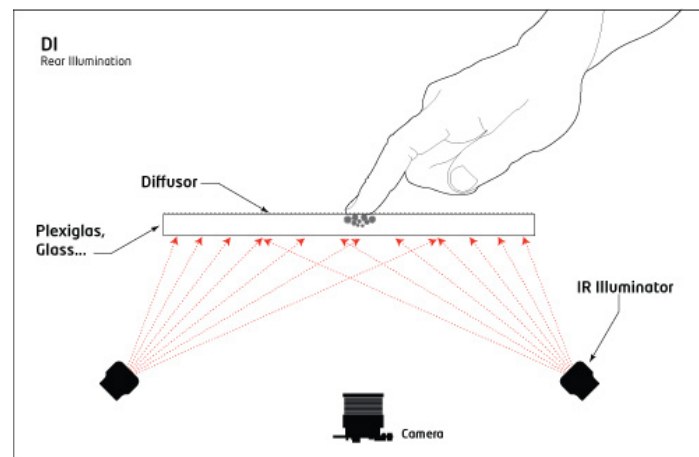


Abbildung 2.12: Funktionsweise eines Kamera basierten IR Touch-Sensors (2.3.1) von [Tim Roth \(2008\)](#)

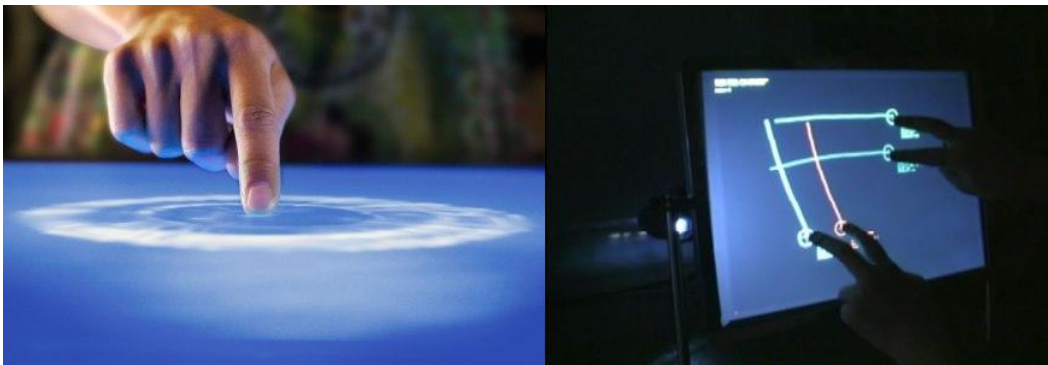


Abbildung 2.13: Links ein Bild zu Single-Touch (2.3.2) und rechts eines zu Multi-Touch ([Perceptive Pixel \(2007\)](#))

### 2.3.2 Von Single- zu Multi-Touch

Die Seamless Interaktion, wie oben geschildert, besitzt in dieser Arbeit eine tragende Rolle. Um die notwendigen Bestandteile für die Seamless Interaktion umsetzen zu können, bieten sich unterschiedliche Technologien an. Zwar beschränkt sich diese Arbeit auf Multitouch-Technologien, um eine Abgrenzung zu schaffen, werden hier kurz Unterschiede zu Single-Touch Systemen aufgeführt.

Bei Single-Touch Systemen wird genau ein Berührungspunkt zu einer bestimmten Zeit angegeben. Dies ist für den gewöhnlichen Einsatz bei heutigen Betriebssystemen ausreichend, allerdings für die Entwicklung von intuitiven Bedienformen eher hinderlich.

Aus diesem Grunde wird für diese Arbeit der Einsatz von Multitouch Technologie in Betracht gezogen. Diese Systeme können mehrere Berührungspunkte auf dem Sensor wahrnehmen und weitergeben, so dass der Entwickler bei der Erstellung der Applikationen größere Freiheiten hat. Auch die Bedienung durch mehrere Benutzer zur gleichen Zeit ist mit derartigen Technologien realisierbar. So ermöglicht eine Erkennung von mehreren Unterbrechungen auf dem Touch-Sensor eine Erweiterung in der Interaktionsvielfalt. Während bei Single-Touch Systemen viele Funktionalitäten in Menüs untergebracht werden müssen, können bei Multitouch-Systemen viele Funktionen durch die gleichzeitige Bewegung mehrerer Finger auf dem Sensor realisiert werden.

Im Rahmen von GIEs, in denen das verteilte Wissen kommuniziert werden soll, ist der Einsatz von Multitouch-Technologien zur Realisierung von Mehrbenutzeranwendungen sinnvoll.

### 2.3.3 Erhältliche Touch Hardware

Auf dem Markt sind einige der o.g. Technologien in Endgeräten verfügbar. Hier wird kurz dargestellt, welches diese sind, und deren Vor- und Nachteile aufgelistet. Dies ist nötig, um ein Verständnis für die zu treffende Entscheidung zu erleichtern, welche Hardware verwendet wird.

**Microsoft's Surface (Microsoft Corporation (2007))** ist eine Implementation von Multi-touch Technologie. Diese ist von Microsoft<sup>1</sup> auf Bestellung zu erwerben.

Verwendet wird eine kamerabasierte Technologie, ähnlich der Diffused Illumination(2.3.1). Das ganze System ist ein Tisch mit halbdurchsichtiger Auflageplatte, welche von unten von einem Beamer angestrahlt wird (s. Abb. 2.14). Weiterhin befinden sich unterhalb der Tischplatte Infrarot-LEDs, welche die Fläche anstrahlen und Kameras, welche die Unterseite der Oberfläche abfilmen und so Objekte auf der Oberfläche erkennen können. Das System wird mit einem PC im Tisch geliefert, mit installiertem Windows Vista und der entsprechenden Auswertungssoftware für die Objekterkennung.

Microsoft hat für dieses System spezielle Multitouch Applikationen entwickelt, die Multitouch-Fähigkeit des Tisches lässt sich also nicht auf das Betriebssystem übertragen.

Es wird bereits an zwei Stellen produktiv eingesetzt. Eines dieser Systeme hat AT&T (AT&T Inc. (2008)) gekauft, ihnen dient es als Vorführgerät in einem Mobiltelefon Shop, um neue Telefone interaktiv und ansprechend zu präsentieren. Andere wurden von Harrahs Entertainment (Harrah's Entertainment (2008)) für ihre Kasinos gekauft, dort werden über diese Tische Getränke bestellt und über Chat-Programme können sich an unterschiedlichen Tischen Menschen unterhalten.

Vorteile:

- gute Erkennungs-Rate von Objekten
- beliebige Objekte sind erkennbar
- mehrere Benutzer gleichzeitig möglich
- innovatives Design

Nachteile:

- Hitzeprobleme durch den Einsatz eines Beamers

---

<sup>1</sup><http://www.microsoft.com/surface/>



Abbildung 2.14: Microsofts Surface ([Microsoft Corporation \(2007\)](#))(2.3.3)

- aufgrund der verwendeten Technologie nur als Tisch verfügbar und nicht beliebig in der Größe skalierbar
- Durch den Einsatz eines Beamers ergibt sich eine relativ hohe Einbautiefe.
- zur Zeit nur auf Bestellung für Hotels, Restaurants und andere erhältlich
- proprietäres System von Microsoft, es werden nur Applikationen erlaubt, die Microsoft auf dem System installieren möchte.
- nur Windows Applikationen möglich, da Windows als Betriebssystem eingesetzt wird

**Apple iPhone ([Apple Inc. \(2007\)](#))** ist ein mobiles Telefon mit Touchscreen, welches über Multitouch-Fähigkeiten verfügt (s. Abb. 2.16). Es ist mittlerweile in der zweiten Generation verfügbar und nur mit einer Vertragsbindung an einen Mobilfunkanbieter von Apple zu erwerben.

Das iPhone arbeitet unter anderem mit der NFI(2.3.1)- und der kapazitiven (2.3.1) Technologie und kann so mehrere Finger oder andere Objekte mit ähnlichem elektrischem Feld erkennen (s. Abb. 2.15). Es verfügt auch über einen Beschleunigungssensor, welcher es dem Gerät ermöglicht, die Orientierung, in der es gehalten wird, festzustellen und das Bild anzupassen. Auf diesem Gerät laufen nur speziell hierfür



entwickelte Applikationen, welche das SDK<sup>2</sup> von Apple benutzen müssen. Zudem muss ein Entwickler bei Apple angemeldet sein, um die entwickelten Applikationen auf dem Gerät installieren zu dürfen.

Vorteile:

- gute Erkennungs-Rate von Fingern
- tragbares Gerät
- nahezu weltweit verfügbar
- innovatives Design
- durch GPS(Global Positioning System), Lichtsensor und Beschleunigungssensor Context-Aware (2.1.1) Applikationen möglich
- Applikationen können in Apple AppStore<sup>3</sup> online verfügbar gemacht werden.

Nachteile:

- Gebunden an einen Mobilfunkvertrag.
- Zu kleines Display für interessante Applikationen, daher auch nicht Mehrbenutzer-Fähig.
- Proprietäres System von Apple.
- Nur speziell für das iPhone geschriebene Applikationen möglich.
- Nur Finger und andere Objekte mit ähnlichem elektrischem Feld erkennbar.

**Perceptive Pixel** ist eine Firma von Jeff Y. Han. Nach sehr erfolgreichen ersten Videos<sup>4</sup> von seiner Entwicklung, gründete er diese Firma<sup>5</sup> und vertreibt nun Multitouch-Wände an sein Kunden(s. Abb. 2.17). Er hat mit einem Paper (Han (2005)) von 2005 das Thema Multitouch wieder publik gemacht.

Verwendet wird hierbei seine selbst entwickelte Technologie des FTIR (2.3.1). Diese Wände können für alle Zwecke der Visualisierung und Manipulation eingesetzt werden. Allerdings müssen die Applikation auch hier speziell dafür geschrieben sein.

Vorteile:

- gute Erkennungs-Rate von Fingern

---

<sup>2</sup><http://developer.apple.com/iphone/>

<sup>3</sup><http://www.apple.com/iphone/features/appstore.html>

<sup>4</sup>[http://www.youtube.com/results?search\\_query=jeff+han](http://www.youtube.com/results?search_query=jeff+han)

<sup>5</sup><http://www.perceptivepixel.com>

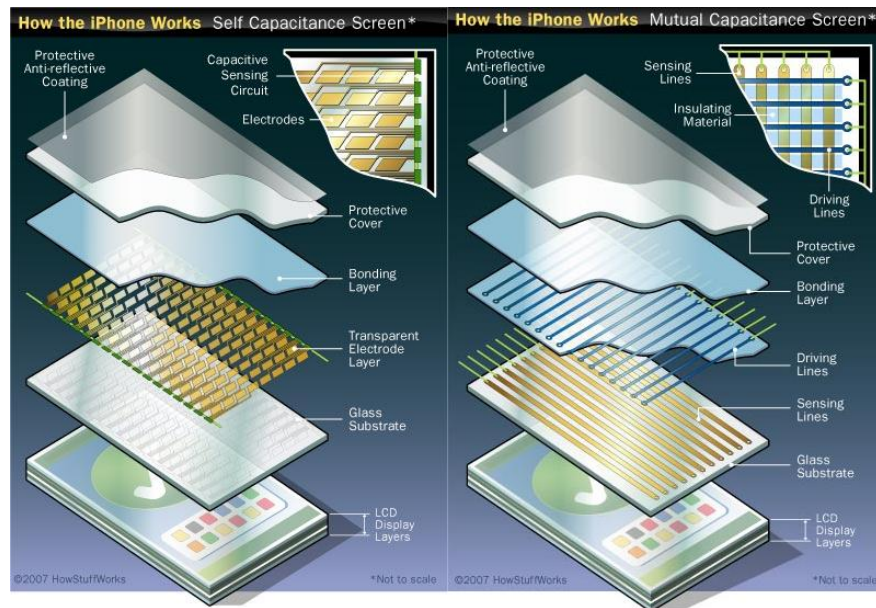


Abbildung 2.15: Apple iPhone Technologie (HowStuffWorks (2008))(2.3.3)



Abbildung 2.16: Apple iPhone (Apple Inc. (2007))(2.3.3)



Abbildung 2.17: Jeff Han vor seiner Multitouch-Wall (2.3.3)

- nahezu beliebig in der Größe skalierbar
- durch eingesetzte Technologie Mehrbenutzer fähig
- als Tisch und Wand umsetzbar

Nachteile:

- nur auf Bestellung zu erwerben
- bisher keine Angabe zu Entwicklungsumgebungen oder verwendetem Betriebssystem
- nur von Jeff Han entwickelte Applikationen verfügbar, daher hohe Abhängigkeit vom Entwickler
- proprietäres System von Perceptive Pixel
- nur in abgedunkelten Räumen einsetzbar, da einfallendes Licht das System stark in der Erkennung beeinträchtigt

**JazzMutant's Lemur und Dexter** sind Kontrollgeräte für Medien-Applikationen ([JazzMutant Lemur \(2008\)](#)). Sie nutzen NFI- oder kapazitive Technologien, um das Display durch

OpenSoundControl<sup>6</sup> entwickelte Programme, auf spezielle Gegebenheiten anzupassen<sup>7</sup>.

Vorteile:

- nahezu weltweit verfügbar
- gute Erkennungs-Rate von Fingern
- durch eingesetzte Technologie Mehrbenutzer fähig
- als Tisch und Wand umsetzbar
- entwicklungs Umgebung bekannt und durch große Community gepflegt

Nachteile:

- durch eingesetzte Technologie nur begrenzt skalierbar
- jedes Gerät hat ein etwa 12 Zoll großes Display, daher keine großen Applikationen möglich
- proprietäres System von JazzMutant
- stark eingeschränktes Einsatzgebiet durch verwendete Programmierschnittstelle

**IRTouch** bietet Infrarot-Touch Rahmen (2.3.1) an<sup>8</sup>, die in speziellen Versionen Multitouch-Fähig sind. Die Geräte sind relativ robust, da sie für den Außeneinsatz ausgelegt und in vielen Größen verfügbar sind. Das Angebot beinhaltet nur den Touchrahmen, etwaige Displays müssen zusätzlich angeschafft werden. Weitere Details in 5.

Vorteile:

- weltweit verfügbar
- unabhängig von der Anzeige
- relativ robust, da für den Außeneinsatz vorgesehen
- gute Erkennungs-Rate von Objekten
- durch eingesetzte Technologie eingeschränkt Mehrbenutzer fähig
- als Tisch und Wand umsetzbar
- nahezu beliebig in der Größe skalierbar, Rahmen mit bis zu 150“ Diagonale

---

<sup>6</sup><http://opensoundcontrol.org/>

<sup>7</sup><http://www.jazzmutant.com/index.php>

<sup>8</sup><http://www.irtouch.com/>

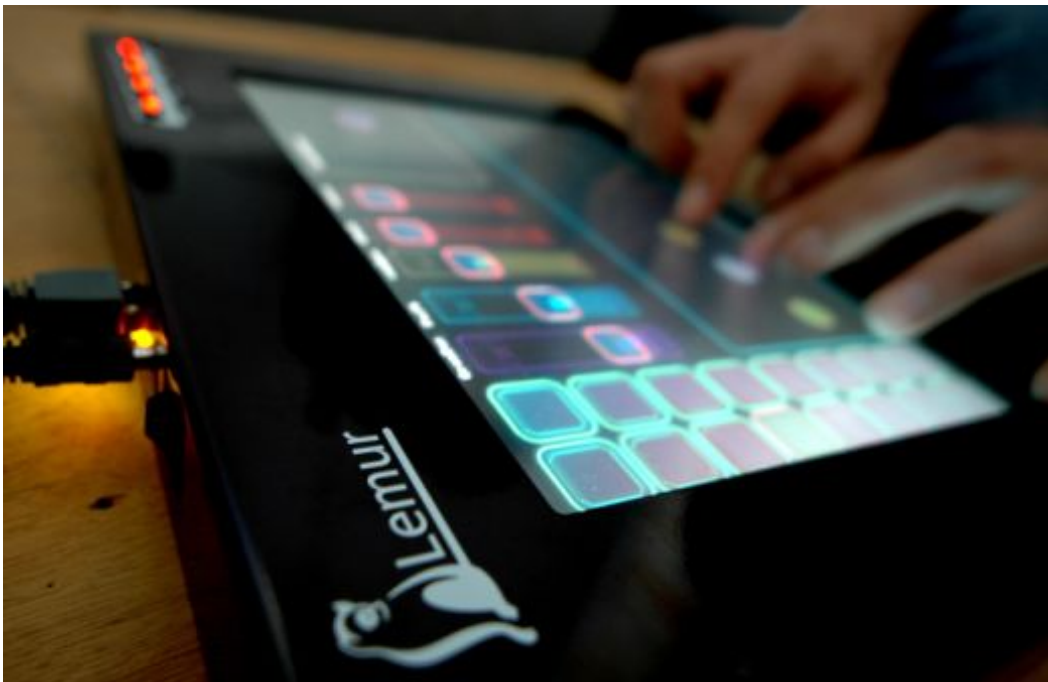


Abbildung 2.18: JazzMutant's Lemur (2.3.3)

- betriebssystem unabhängig, da über USB oder RS232 anschließbar

Nachteile:

- Multitouch-Fähigkeit nur eingeschränkt, aufgrund von so genannten „Ghost Points“ (5), welche Hardware-bedingt entstehen
- es sind keine beliebigen Objekte erkennbar
- größenabhängige Eigenschaftsänderung

## 2.4 Fazit

Die Grundlagen haben vielfältige Möglichkeiten für die Analyse und technische Umsetzung aufgezeigt. Daraus wird ersichtlich, wie weit die technischen Gegebenheiten bereits fortgeschritten sind, sodass sich in der Analyse klare Anforderungen ableiten lassen.

Die theoretischen Grundlagen, wie die Seamless Interaction von Hiroshi Ishii sind solide genug, dass eine Umsetzung der Ideen möglich ist.

Die rasante Entwicklung und eine zügige Marktreife von neuen Technologien ermöglicht es, Menschen aus Berufsfeldern außerhalb der Informatik, wie etwa Geisteswissenschaftlern und Designern, unter Beihilfe der Informatik neue Ideen und Konzepte vorzustellen.

# 3 Analyse

Ziel dieser Analyse ist es die Interaktion mit Multitouch-Systemen zu untersuchen und Anforderungen für eine Benutzung zu identifizieren. Es werden zunächst Beispielszenarien 3.1 vorgestellt und anschließend in Abschnitt 3.2 die eigentliche Interaktion untersucht. Eines der Hauptziele ist dabei das von Weiser formulierte Ziel:

[...]Machines that fit the human environment, instead of forcing humans to enter theirs[...] (aus [Weiser \(1991\)](#))

## 3.1 Szenarien

Im folgenden werden zwei Szenarien vorgestellt, um den Einsatz eines intuitiv bedienbaren Systems zu erläutern und zu motivieren.

### 3.1.1 Einsatzleitzentrale

Das Szenario der Einsatzleitzentrale ([Piening \(2007\)](#)) beschreibt ein oder mehrere eng beieinander liegende Räume, die mit einigen technischen Geräten ausgestattet werden (s. [Abb. 3.1](#)). Den professionellen Fachkräften soll durch diese technische Unterstützung ein Hilfsmittel an die Hand gegeben werden, mit dem sie einfach und schnell Informationen austauschen und diskutieren können.

Es handelt sich hierbei um den Raum des CCW ([2.1.1](#)), wobei hier speziell die räumliche und zeitliche Koexistenz aller Beteiligten vorausgesetzt wird. Es wird davon ausgegangen, dass alle Beteiligten in einer Teamstruktur zusammenarbeiten. Hierbei werden sowohl Einzelarbeitsplätze, als auch eine Gruppenarbeit von der räumlichen Aufteilung berücksichtigt. Einerseits soll verteiltes Wissen in dieser Teamstruktur untereinander ausgetauscht werden, andererseits aber auch das komplette Know How des Teams in die aktuelle Entwicklung integriert werden. An dieser Stelle spielen die Grundlagen des Group Information Environments eine wesentliche Rolle.

Unter professionell ausgebildeten Menschen kann es nicht eine Person geben, die alles Wissen verinnerlicht hat. Viel wichtiger ist das, durch die in dieser Arbeit entwickelten Systeme unterstützte, Sammeln und Bewerten unterschiedlicher Informationsquellen, die dann auf eine intuitive Art und Weise kommuniziert werden können. Hierzu ist es nötig, einfach begreifbare und natürliche Eingabemethoden zu schaffen, so dass die Kommunikation und nicht die Handhabung der technischen Geräte im Vordergrund steht. Das Einsatzpersonal besteht aus Fachkräften, bei denen nur ein geringer Lernaufwand erforderlich ist, da sich durch intuitive Technik-Unterstützung die bestehenden Arbeitsprozesse vereinfachen lassen.

Aufgrund der hohen Informationsdichte bei heutigen Einsätzen werden oft wichtige Details aus den Augen verloren. Es werden gewisse Methoden zur einheitlichen Navigation durch diese Informationsbestände nötig. Sinnvoll ist es auch den Einsatzkräften zu ermöglichen, unterschiedliche Informationsbestände in verschiedenen Detailstufen zu betrachten und zu manipulieren. Manipulation wird in dieser Arbeit für die händische (lat. Manus) Bearbeitung von Informationen genutzt. Hieraus ergibt sich als logische Konsequenz der Einsatz von mehreren Bildschirmen in unterschiedlichen Größen und Detailschärfen.

Hierzu eignet sich der Einsatz von Powerwalls ([Napitupulu \(2008\)](#)), die ein Betrachten des gesamten Bildes, aber auch, aufgrund der hohen Auflösung, ein Wahrnehmen von Details ermöglichen. Um ein direktes Anfassen von Objekten auf dem Bildschirm zu ermöglichen und zum Anzeigen von Ausschnitten und Manipulation von diesen, eignen sich touchfähige mittelgroße Bildschirme. Bekannt ist diese Art des Einsatzes von Computersystemen aus Tangible User Interfaces (TUI [Ishii und Ullmer \(1997\)](#)).

Bildschirme sollten als Tische und Wände zur Verfügung stehen, um unterschiedliche Interaktionsmethoden zu unterstützen. Desweiteren könnte man sich HUDs vorstellen, die jeder Einsatzkraft unmittelbar wichtige oder kritische Informationen direkt vor das Auge projizieren.

Da es sich bei diesen Büroräumen um definierte Umgebungen handelt, sollten alle Geräte, welche sich im Hintergrund befinden, durch Software virtuell nachgebildet werden, um so eine nahtlose Interaktion zwischen den einzelnen Geräten zu ermöglichen. Die Position dieser virtualisierten Geräte ist relativ, kann also verändert werden, je nachdem wo sich ein Gerät befindet ([Köckritz \(2007\)](#)).

### 3.1.2 Messe Szenario

Das Messe Szenario beschreibt im Gegensatz zur Einsatzleitzentrale den Anwendungsbereich eines intuitiven Systems im Consumer-Bereich. Das Multitouch-System soll hier beliebige Informationen, als virtuelle Objekte auf dem Bildschirm ansprechend präsentieren. Es soll als ein Eye-Catcher, in einer sonst sehr überfüllten, Licht und Ton überfluteten Umgebung, die Aufmerksamkeit der Messebesucher erregen.





Abbildung 3.1: Einsatzleitzentrale der Internationalen Raumstation ISS ([Deutsches Zentrum für Luft- und Raumfahrt \(2008\)](#))

Der Kunde hat durch die Touch-Funktionalität die Möglichkeit, interaktiv die gewünschten Informationen direkt anzufassen, zu betrachten, durch viele unterschiedliche Informationen zu navigieren und diejenigen von Interesse zu mitzunehmen.

Die Interaktion mit dem System ist hier hauptsächlich durch die relativ kurze Interaktionsdauer geprägt. Daher ist es nötig, sehr intuitive Interaktionsmöglichkeiten anzubieten. Das Augenmerk der Kundschaft soll hier gezielt auf wesentliche Produkte gelenkt werden. Da davon auszugehen ist, dass der Passant eine sehr geringe Frustrationsgrenze besitzt, ist es fundamental eine handbuchfreie Interaktion zu ermöglichen.

Hierzu bietet sich ein Multitouch-System an, da es über einfache Gesten mit mehreren Fingern, Produktdetails, Spezifikationen und Preise zu den Produkten auf dem Bildschirm anzeigen kann. Eine Mitnahme der interessanten Produktinformationen, sollte durch direktes Ausdrucken auf einem angeschlossenen Drucker, oder durch das Speichern auf einem tragbaren Speichermedium möglich sein.

Das Erkennen von Besuchern, welche in der Nähe des Tisches stehen ermöglicht Context-Aware Applications ([2.1.1](#)), die im Bereich des Ambient-Assisted-Living Anwendung finden.

### 3.1.3 Vergleich der Szenarien

Es wurden zwei Szenarien gewählt, um die unterschiedlichen Anwendungsbereiche darzustellen, in denen das System verwendet werden kann. Auch wenn diese nicht alle Anwendungsgebiete abdecken, lassen sich hierdurch exemplarisch realistische Einsatzgebiete aufzeigen.

Der Hauptunterschied in den gewählten Szenarien liegt im skizzierten Einsatzbereich. Bei der Einsatzleitzentrale handelt es sich um den Professionell-Bereich und bei dem Messe-Szenario um den Consumer-Bereich, wodurch sich unterschiedliche Voraussetzungen ergeben, die sich wiederum auf das Lernen der Interaktionsmethoden auswirken.

Zudem wird die Anbringung und Aufstellung des Systems in 3.1.3 gesondert betrachtet, weil hierdurch wiederum andere Erwartungshaltungen (2.2.1) impliziert werden, die in neuen Möglichkeiten und auch Einschränkungen resultieren.

Sinnvoll ist die Unterscheidung auch aufgrund der unterschiedlichen Interaktionsdauer. In der Einsatzleitzentrale ist davon auszugehen, dass es lange und wiederholende Interaktionsphasen gibt, während diese innerhalb einer Messe nur relativ kurz sind.

#### Horizontale vs Vertikale Ausrichtung

Wie oben festgestellt, ergeben sich aufgrund der Anbringung bzw. Aufstellung des Systems verschiedene Erwartungshaltungen. Zum Einen könnte man sich eine vertikale Anbringung in Form einer Wand, forthin als Multitouch-Wall (Abb. 3.2) bezeichnet, vorstellen. Um sich ein besseres Bild machen zu können, wird bei der Multitouch-Wall eine Pinnwandmetapher zu Hilfe genommen. Man kann sich diese also als eine Pinnwand vorstellen, auf der virtuelle Objekte in Form von Bildern angeheftet werden können.

Zum Anderen könnte man sich aber auch eine horizontale Anbringung als Tisch, welcher im weiteren Text als TouchTable (Abb. 3.3) bezeichnet wird, vorstellen. Bei dem TouchTable wird die Geländemetapher zu Hilfe genommen, die sich an die physikalische Realität anlehnt. Virtuelle Objekte auf dem virtuellen Tischgelände verhalten sich also physikalisch korrekt, haben ein Gewicht, eine Reibung und es herrscht eine globale Anziehungskraft, welche die Objekte auf dem Tisch hält.

Diese vorgestellten Anwendungsbeispiele werden als Mental Models(2.2.1) bezeichnet. Die Objekte bei der Multitouch-Wall können, nach Anwendung der Pinnwandmetapher, verschoben, neue können hinzugefügt, alte ersetzt und mit Informationen versehen werden. Beim TouchTable ergeben sich Möglichkeiten, wie z.B. das Gegeneinanderstoßen von unterschiedlichen Objekten, dabei wird dann je nach Masse, also Priorität des Objektes, die

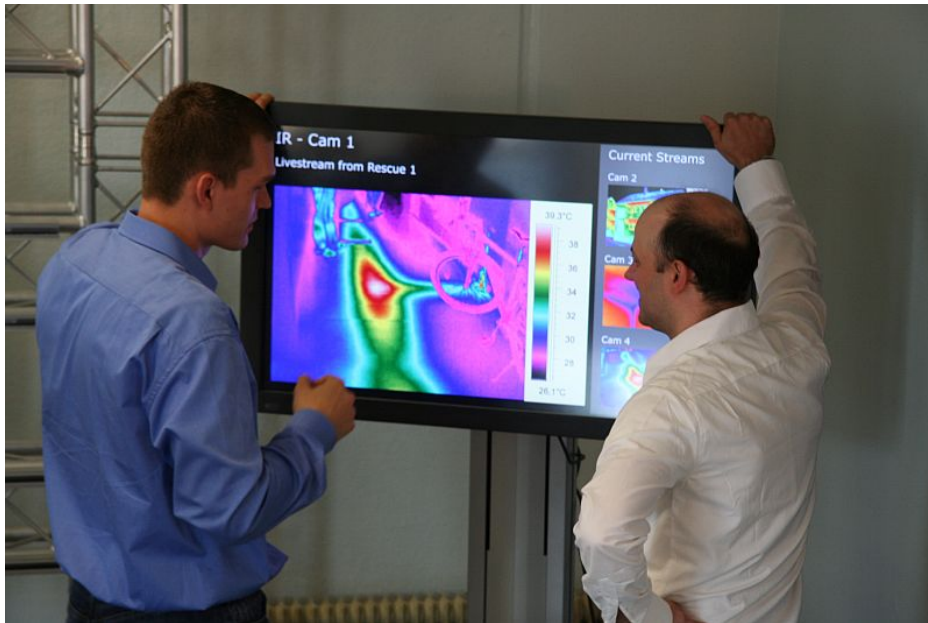


Abbildung 3.2: Multitouch-Wall



Abbildung 3.3: TouchTable

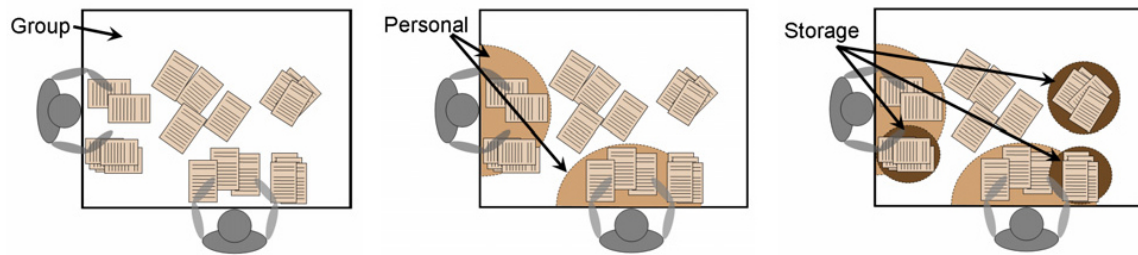


Abbildung 3.4: Territorialtypen bei kollaborativer Arbeit an Tischen (aus [Scott \(2005\)](#) )

Richtung, in die das Objekt geworfen wurde, passend zum Winkel verändert.

Da sich beim TouchTable die Personen frei um den Bildschirm positionieren können, fehlt anders als bei der Pinnwand, ein definiertes Oben oder Unten. Es stellt sich also die Frage, wie das System einem Nutzer Daten präsentiert.

Bei Einzelnutzung kann durch Betrachtung der Interaktionsfläche eine Start-Ausrichtung von Informationen ermittelt werden. Bei mehr als einem Benutzer im Bereich des CCW, muss ermittelt werden, wie Informationen ausgerichtet werden müssen, um für den Nutzer lesbar zu sein. Diese Problematik wurde von Philipp Roßberger ([Roßberger \(2007\)](#)) genauer betrachtet. Aufgrund von Territorialtypen bei kollaborativer Arbeit an Tischen (Abb. 3.4), die bei Versuchen von [Scott \(2005\)](#) ermittelt wurden, muss die Ausrichtung genau betrachtet und angepasst werden. So könnte eine Information sich automatisch zum Nutzer drehen, sobald sie in den „Personal“ Bereich gelangt.

## 3.2 Interaktion mit Multitouch Systemen

Es wurde nun dargestellt, dass die Erwartungshaltung des Benutzers eine ausschlaggebende Rolle spielt, wenn die natürliche Bedienung im Vordergrund steht. Daher ist die Anlehnung an die physikalische Realität für neue Bedienkonzepte von Bedeutung. Als Konsequenz daraus ergibt sich die Anwendung von Touch-Technologie, die das direkte Berühren von Objekten auf dem Bildschirm unterstützt. Werden statt Singletouch, Multitouch-Technologien eingesetzt, kann der Benutzer Applikationen mit mehreren Fingern bedienen. Daraus ergeben sich neue Interaktionsmöglichkeiten, die eher einer Manipulation von Objekten in der physikalischen Realität entsprechen und somit intuitive Bedienkonzepte besser unterstützen. Die positiven Konsequenzen aus der Verwendung von Multitouch-Technologie für die natürliche Bedienbarkeit dieser Systeme sind groß genug, dass sich der Zeit- und Kostenaufwand für die Einrichtung solcher Systeme lohnt. Die Konsequenz aus natürlicher Interaktion mit Computersystemen, auf Basis von Touch, ist somit die Multitouch-Technologie.

## 3.3 Anforderungsanalyse

Die verwendeten Szenarien und die unterschiedlichen Ausprägungen führen zu einigen wichtigen Anforderungen an derartige Systeme. Es werden erst eine Reihe von funktionalen Anforderungen formuliert, welche nötig sind, um die gewünschten intuitiven Bedienkonzepte umzusetzen. Darauf folgen einige nichtfunktionale Anforderungen an das zu entwickelnde System, welche hinlängliche Eigenschaften beschreiben.

Aufgrund der Anwendungsszenarien werden zwei Typen von Stakeholdern identifiziert. Zum einen diejenigen im Consumer-Bereich, welche eine interaktive Produktpräsentation wünschen (digital signage<sup>1</sup>). Die zweite Gruppe ist im Professional-Bereich anzutreffen, welche nach dauerhaft einsetzbaren integrierten Lösungen sucht.

In der Anforderungsanalyse wird von einer gemeinsamen Basis der unterschiedlichen Anwendungskontexte ausgegangen, signifikante Unterschiede werden gesondert dargestellt.

### 3.3.1 Funktionale Anforderungen

Das hier vorgestellte System muss einige essentielle Funktionen bereitstellen, um den folgenden Anforderungen zu genügen. Diese funktionalen Anforderungen bestehen hauptsächlich aus Funktionen, die die Handhabung von Informationen betreffen. Hierbei ist besonders die Interaktion mit Informationen wichtig.

Die Erlernbarkeit sollte nicht aus den Augen verloren werden, denn auch komplexere Funktionen sollten mit dem System möglich sein, die unter Umständen nicht sofort intuitiv sind.

#### Anwendungsfälle

Eine allgemeine Übersicht über mögliche Anwendungsfälle bietet das Use Case [3.5](#).

Ausgehend von den vorliegenden Szenarien, dem Anwendungsfallüberblick und der mentalen Modelle, wird festgestellt, dass es stets um Betrachtung, Verarbeitung, Austausch und Schaffung von Informationen geht. Was nötig ist, um diese Ziele zu erreichen, wird nachfolgend genauer betrachtet.

---

<sup>1</sup><http://www.digitalsignerresource.com/>

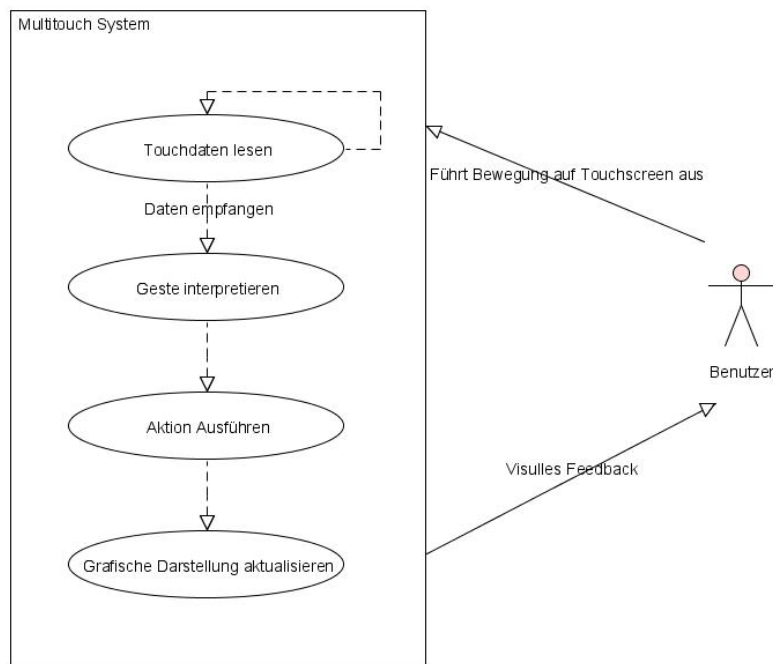


Abbildung 3.5: Schema eines Anwendungsfalles

### Interaktion mit Information

Die Anwendungsfälle sind nur realistisch, wenn eine Präsentation der Information vorhanden ist. Ausgehend davon, dass Information Gegenstand jeder Bearbeitung sind, kann der Benutzer nur annehmbar die Funktionen ausführen, wenn auch eine Präsentation der durchgeführten Aktionen (3.3.1) möglich ist.

Die Darstellung von Informationen ist Grundlage jedes Nutzens von diesen. Wenn die Informationen oder deren Ergebnisse aus der Bearbeitung nicht dargestellt werden können, sind die Informationen nutzlos. Präsentation von Informationen ist sehr vielfältig, da Informationen immer an bestimmte Kontexte gebunden sind. Die Informationsträger können Texte sein, genauso wie z.B. auch Videos, Musik und Grafiken Informationen enthalten können. Diese können direkt im Inhalt stecken oder auch als Meta-Informationen, nicht direkt ersichtlich, vorliegen. Also ist es wichtig, unter der Prämisse der natürlichen Bedienbarkeit des zu entwickelnden Systems, für jede Art des Inhaltes passende virtuelle Repräsentationen zu finden.

Bei heutigen Computern wird die Darstellung vom Betriebssystem oder speziellen Programmen ermöglicht, die allerdings nicht erweiterbar sind, da sie zu speziell, wie im Falle von besonderen Darstellungsprogrammen, oder zu allgemein, wie im Falle der Betriebssysteme

sind. Als Navigationsinstrumente haben sich bisher nur die Maus, die Tastatur und Derivate von diesen durchgesetzt. Dies hat klare Vorteile, wie z.B. die allgemeine Anwendbarkeit auf unterschiedlichste Programme, allerdings auch schwerwiegende Nachteile.

Bei der Benutzung von Mäusen bildet sich eine Abstraktionsebene zwischen Bewegung und Darstellung, die der Benutzer beherrschen muss, um diese Zeigergeräte nutzen zu können. Hierbei wird eine horizontale Fläche, nämlich die Tischfläche auf der sich die Maus bewegt, auf eine vertikale Fläche, die Fläche der Darstellung auf dem Bildschirm, abgebildet. Erschwerend kommt hinzu, dass die Mausbewegungen nicht absolut erfolgen, sondern nur relativ zur aktuellen Position des Mauszeigers, welches ein Tribut an größere Bildschirme und Auflösungen ist. So kommt es dazu, dass der Benutzer mehrfach die gleiche Bewegung ausführen muss, um mit dem Zeiger ein bestimmtes Ziel zu erreichen.

Diese Hürden müssen verstanden und erlernt werden, um ein Arbeiten zu ermöglichen. Des Weiteren stellen sie einen dauerhaften kognitiven Aufwand dar, da neben der Verarbeitung der visuellen Eindrücke auch ein verarbeiten dieser Abstraktionsebene stattfinden muss. Dieser Aufwand ist bei großen Mengen sowie komplexer und schnelllebiger Information nicht mehr tragbar.

Resultierend aus den dargestellten Missständen bei den heutigen Bedienkonzepten und den Vorschlägen von Ishii zur Seamless Interaction, ergibt sich die Anforderung natürlicher Interaktion mit Computer-Systemen. Ein erster Schritt in diese Richtung wäre das Ermöglichen des direkten Berührens der präsentierten Objekte auf dem Bildschirm. So ist es möglich, diese Vorgänge zu vereinfachen, sodass auch komplexe Tastatureingabekombinationen durch einfach begreifbare Bewegungen aufgelöst werden. Die benötigten Benutzeraktionen hierzu gilt es nun ausfindig zu machen.

### **Benötigte Benutzeraktionen zum Informationsgewinn**

Benutzeraktionen sind vom System ausführbare Aktionen, die eine Interpretation von manipulativen Handlungen, auf Basis von Informationen, darstellen.

Werden die Informationen an auf dem Bildschirm darstellbare Objekte gebunden, ist es meistens nicht ausreichend, diese nur betrachten zu können. Vielmehr sollte es möglich sein, die Informationsträger zu manipulieren, um einen Wissensgewinn zu erzielen.

So ist es bei der Darstellung von Produkten von Interesse, weitergehende Informationen zu diesen zu erhalten. Bei Videos sollte es möglich sein, diese anzuhalten, neuzustarten oder einen Vor- und Rücklauf anzubieten, um bestimmte Szenen genauer oder erneut betrachten zu können. Bei Gebäude-Modellen sind Ansichten aus unterschiedlichen Winkeln und Detailansichten interessant.

Im Arbeitsalltag wird der Benutzer häufig mit vielen Informationsträgern konfrontiert, die sich zur gleichen Zeit auf der Präsentationsfläche befinden, welches als Information Overload bezeichnet wird. Diese haben aber unterschiedliche Prioritäten, somit ist es sinnvoll eine Navigationsmöglichkeit bereitzustellen, um die Informationsflut beherrschen zu können. Hierzu gehört auch das Sortieren der Informationen durch das Bewegen einzelner oder in Gruppen vereinigter Objekte.

Die Manipulationen von Informationsträgern zum Wissensgewinn sind also Aktionen. Häufig benutzte Aktionen umfassen z.B. das Öffnen einer Datei, sowie auch das Verschieben, Vergrößern oder Auswählen. Die einfachste und am häufigsten benutzte Aktion ist dabei das Auswählen.

Die Aktionen sollten möglichst atomare Bausteine sein, so dass komplexere Aktionen aus diesen zusammengesetzt werden können. Als Beispiel für eine Aktion stelle man sich ein Produktblatt vor, welches auf der Vorderseite ein Produktbild und auf der Rückseite dessen Spezifikationen enthält. Um an die Spezifikationen zu gelangen, braucht der Benutzer eine Aktion zum Umdrehen des Blattes, welches somit die Spezifikationen freigibt.

Desweiteren sollte es möglich sein, Annotationen, wie z.B. Notizen, auf dem dargestellten Objekt hinterlegen zu können. Diese können dann im weiteren Gebrauch wieder aufgerufen werden und erleichtern hiermit das oft schwierige Auswerten von Informationen. Eine passende Aktion wäre hierfür das Schreiben auf eine dafür vorgesehene Stelle auf dem Objekt, wobei zu klären ist, ob hier eine Stifteingabe oder das bloße Ziehen des Fingers auf der Fläche erfolgreicher ist. Auf diese Weise lassen sich stark verschachtelte Menüs mit viel Informationsinhalt vermeiden. All diese Aktionen sind Teil der Anforderungen an ein System, welches der Seamless Interaction gerecht werden soll.

Im Falle der Multitouch-Wall geht es bei den Aktionen darum, die vorhandenen Informationen mittelfristig analysieren zu können, während beim TouchTable eher die kurzfristige Bewertung und Weitergabe der Informationen im Vordergrund steht.

Es müssen nun sinnvolle Gesten gefunden werden, die es dem Benutzer ermöglichen, dem System zu vermitteln, welche Aktion ausgeführt werden soll.

### **Sinnvolle Gesten zu den Aktionen**

Seamless Interaktion wird bei den diskutierten Systemen erreicht, indem Gesten eingeführt werden, um eine realitätsnahe und intuitive Bedienung zu verwirklichen. Hierzu werden die oben genannten Aktionen an Gesten gebunden, die es dem Benutzer ermöglichen, bei erfolgreicher Durchführung der Geste auf einem Objekt, die Aktionen auszuführen.

Um den Zusammenhang zwischen Aktion und Geste nicht zu verlieren, ist insbesondere darauf zu achten, dass die Gesten, unter Betrachtung der Mental Models, eine deutliche



Verbindung zur gewollten Aktion besitzen. Da dieser Zusammenhang für eine natürliche Interaktion so wichtig ist, wird dieses Thema noch einmal gesondert im Gesten Kapitel 4 betrachtet.

Eine erste Evaluation von einfachen Gesten für Dualtouch Systeme wird in der Masterarbeit von Stephan Gehn (Gehn (2008)) erarbeitet.

Aus den vorhandenen, an Gesten gebundenen Aktionen (siehe Gesten Kapitel 4) resultieren nun die essentiellen Interaktionen mit dem System.

### Resultierende Interaktion

Interaktionen sind die an Gesten gebundenen, ausführbaren Aktionen, die ein Resultat für den Benutzer liefern. Der gesamte Vorgang der Interaktion stellt sich wie folgt dar:

Möchte ein Benutzer eine Manipulation vornehmen, vollführt dieser eine Geste, passend zu der gewünschten Manipulation, welche dann vom Computersystem Interpretiert wird. Das System entscheidet dann, welche Aktion an die vollführte Geste gebunden ist und führt diese aus. Diese Aktion bewirkt, als Feedback für den Benutzer, eine Veränderung in der grafischen Oberfläche.

So ergeben sich, je nach Geste und zugehöriger Aktion, Veränderungen der im System befindlichen virtuellen Modelle in Bezug auf deren Orientierung, der Navigation im Raum, der Maskengestaltung und Meldungen, also System-Reaktionen im Allgemeinen. Die Grundlegenden atomaren Interaktionen, welche benötigt werden, um mit dem System zu interagieren, lauten wie folgt:

„**Objekt bewegen**“ beschreibt das Bewegen eines virtuellen Datenobjektes auf der dargestellten Fläche.

Dies wird vom Anwender realisiert, indem er eine durchgehende Bewegung, mit einem Finger über dem zu bewegendem Objekt, ausführt. Die Reaktion des Datenobjektes auf die Bewegung ist die Verschiebung desselben parallel zur, mit dem Finger gezogenen, Linie. Der Zielpunkt des Objektes ist der Punkt der Linie, der durch ein Abheben des Fingers von der Touch-Fläche gekennzeichnet wird.

„**Objekt rotieren**“ beschreibt eine Rotation des Datenobjektes um die X-Achse des dargestellten Raumes.

Diese Interaktion wird durch das Ablegen von zwei Fingern auf dem Datenobjekt und einer kreisförmigen Bewegung realisiert. Der Winkel, um den das Datenobjekt zu drehen ist, wird von dem Winkel des durch die Finger gezeichneten Kreises bestimmt.

Das Ende der Bewegung wird durch das Abheben eines oder beider Finger gekennzeichnet.

„**Objekt skalieren**“ beschreibt die Größenveränderung eines Datenobjektes.

Der Anwender platziert hierfür zwei Finger auf dem Objekt und spreizt diese oder führt sie zusammen, welches in einer entsprechenden linearen Größenveränderung resultiert. Diese wird durch die Veränderung im Abstand der Finger gekennzeichnet. Die Bewegung wird mit dem Abheben eines oder beider Finger beendet.

„**Objekt flip**“ beschreibt die Rotation eines Datenobjektes um die Y-Achse des dargestellten Raumes.

Dies wird anders als die anderen Interaktionen, mit drei Berührungspunkten realisiert. Hierzu positioniert der Anwender drei Finger über einem Objekt und bewegt diese horizontal über das Objekt, resultierend in einer Transition des Objektes um die Y-Achse.

„**Physikbasierte**“ Interaktion ist im Vergleich zu den anderen Interaktionsmodi eine gesondert zu betrachtende Interaktionmöglichkeit.

Diese wurde in der Masterarbeit von Philipp Roßberger (Roßberger (2007)) evaluiert und in einer Testumgebung auch realisiert. Diese Realisierung dient dieser Arbeit als Basis, die physikbasierte Interaktion in Verbindung mit Multitouch Systemen zu erforschen. Die im Messe-Szenario bereits genannten Bedingungen, wie Gravitation, Gewicht und Reibung, die eine physikbasierte Umgebung schafft, erzeugen neue Interaktionsmöglichkeiten, die eher der Realität entsprechen und der Erwartungshaltung des Anwenders genügen sollten.

Die Umsetzbarkeit der Interaktionen ist allerdings abhängig davon, wie gut die Bedienbarkeit tatsächlich ist. Zudem ist auch die Performance ausschlaggebend für ein intuitive Bedienung, denn bei der angestrebten Mehrbenutzer-Fähigkeit muss das System Resultate flüssig darstellen.

Nachfolgend werden diese Funktionalitäten kurz beschrieben.

### **Bedienbarkeit**

Die Bedienbarkeit ist ausschlaggebend für den Erfolg des hier diskutierten Systems, da der Einsatz von Multitouch-Technologie stark davon abhängt, ob es sich gegen Maus und Tastatur durchsetzen kann.

Es ist nicht erforderlich, die bestehenden Eingabegeräte komplett zu ersetzen, allerdings sollten bei den in den Szenarien beschriebenen Situationen diese Technologie als Hilfsmittel hinzugezogen werden. Dabei ist zu beachten, dass eine einheitliche Handhabung systemübergreifend eingehalten wird, um den Benutzer nicht zu verwirren. Es ist ebenfalls sinnvoll,

wenn das Aussehen des dargestellten virtuellen Raumes die meiste Zeit der Bedienung gleichbleibend ist. Damit wird eine bessere Orientierung des Benutzers ermöglicht, so dass dieser bei der Benutzung geringerem Stress ausgeliefert ist.

Ein Negativbeispiel hierfür ist das unterschiedliche Verhalten einiger iPhone Applikationen im Anfangsstadium, die die durchgeführten Gesten, entgegengesetzt zur vorher angelern-ten Erwartungshaltung, unterschiedlich interpretierten.

### **Performance**

Das System muss die oben spezifizierten, atomaren Aktionen ohne Verzögerung ausführen, da es sich bei der Interaktion um eine direkte Manipulation der dargestellten Objekte handelt. Also ist es unabdingbar, dass das System in Echtzeit reagiert.

Es müssen genügend Ressourcen vorhanden sein, um auch mehrere komplexe Objekte, wie z.B. Gebäudemodelle und Videos, flüssig darzustellen und ruckelfrei manipulieren zu können.

Die Performance des Systems wirkt sich ausschlaggebend auf die Usability aus.

### **Benutzeranzahl**

Die Mehrbenutzer-Fähigkeit ist wichtig, da es sich in diesem Kontext um Collaborative Workplace handelt und hier, wie im Szenario beschrieben, mehrere Personen gleichzeitig an einem Ort, an einem Thema arbeiten. Des Weiteren erschließt dies den Benutzern neue Möglichkeiten der Interaktion und auch die „Mensch zu Mensch Kommunikation“ wird gefördert. Ein Beispiel für Mehrbenutzer an einer Multitouch Wand zeigte Perceptive Pixel ([Perceptive Pixel \(2007\)](#)). In [Abb. 3.6](#) wird diese simultane Interaktion gezeigt, hierbei ermöglicht die Aufteilung in unterschiedliche Arbeitsbereiche die simultane Arbeit.

In der Arbeit von Dennis Hollatz ([Hollatz \(2007\)](#)) geht er beispielhaft auf die Mehrbenutzer-Interaktion mit Wand- und Tischsystemen ein. So nutzt er eine Wand als moderiertes Brainstorming wobei eine Person die Funktion des Moderators einnimmt und andere Personen als Beobachter agieren und entsprechende Clients zur Verfügung haben, um entsprechende Inhalte für die Wand vorzuschlagen. Der Moderator hat dann die Möglichkeit, diese zuzulassen oder abzulehnen. Die Interaktion des Moderators, welcher keinen Laptop zur Verfügung hat, geschieht über direkte Eingaben an der Wand. Er hat die Möglichkeiten Inhalte zu sortieren, positionieren, skalieren und auf andere Arbeitsbereiche zu verschieben.

Die Benutzeranzahl von horizontal ausgerichteten Systemen, wie z.B. bei einem Tisch, führt zu Überlegungen hinsichtlich der Ausrichtung der dargestellten Datenobjekte, wie in [3.1.3](#) beschrieben.

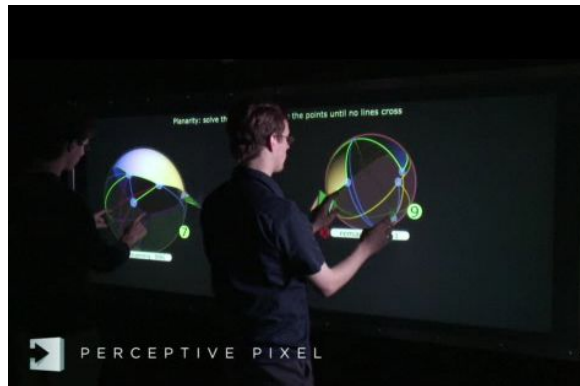


Abbildung 3.6: Simultane Interaktion aus (Perceptive Pixel (2007))

### 3.3.2 Nichtfunktionale Anforderungen

Die nichtfunktionalen Anforderungen an das System ergeben sich hauptsächlich aus dem Ziel dieser Arbeit, die natürliche und intuitive Bedienung von Computersystemen zu realisieren. Hieraus ergeben sich die folgenden Anforderungen, welche nötig sind, um die Bedienbarkeit möglichst optimal umzusetzen.

Internationalisierung wird hier nicht explizit aufgeführt, da eine gewisse internationale Verständlichkeit durch die Gesten gegeben ist.

#### Zuverlässigkeit

Die Zuverlässigkeit des Systems sollte Grundvoraussetzung sein.

Der Mensch hat eine gewisse Fehlertoleranz, die sich zeigt, wenn sich ein System nicht erwartungsgemäß verhält. Es wird dann versucht mit einer gewissen Justierung die Bewegungen in den Bereich der für das System verständlichen Aktion zu bringen. Der Mensch hat also eine gewisse Anpassungsfähigkeit. Dies ist jedoch nicht überzustrapazieren. Eine Überschreitung dieser menschlichen Toleranzschwelle endet sehr schnell in Frustration, welche oft in eine Meidung des Systems übergeht.

Das Verhalten des Systems sollte im Betrieb fortwährend einheitlich sein, um eine gewisse Zuverlässigkeit zu zeigen.

#### Portierbarkeit

Eine native Unterstützung vom Betriebssystem wäre wünschenswert, um die gewählten Anforderungen portierbar umsetzen zu können. Eine einfache Portierbarkeit trägt dazu bei,

dass der Benutzer nahtlos seine Ergebnisse auf anderen, ähnlichen Systemen betrachten und manipulieren kann.

### **Wartbarkeit**

Das System sollte aufgrund der Zuverlässigkeit selten in einen irregulären Zustand geraten. Ist dies trotzdem der Fall, muss es möglich sein, dass der Benutzer das System wieder in einen regulären Zustand versetzen kann.

Durch die Verwendung von Standardkomponenten kann ermöglicht werden, ein einfach zu wartendes System zu erstellen.

## **3.4 Fazit**

Es wurden anhand der Szenarien zwei unterschiedliche Systeme ausgemacht. Diese unterschiedlichen Ausprägungen als Wand und Tisch bringen zwar verschiedene Anforderungen mit sich, allerdings sind die Anforderung an solche Systeme hauptsächlich geprägt durch die Seamless Interaction, da das gewünschte System intuitiv und handbuchfrei zu bedienen sein soll.

Aus den Grundlagen ist hervorgegangen, dass die benötigte Hardware bereits erhältlich und eine Umsetzung prinzipiell machbar ist.

Durch die Anwendungsfälle ergaben sich bestimmte Voraussetzungen an die Interaktion mit diesen Systemen, wie z.B. die Auswirkungen der Darstellung von Informationen auf die Verwendbarkeit. Aus den diversen Voraussetzungen wurden einige atomare Aktionen entwickelt, welche in Kombination mit den Gesten zu den geforderten Interaktionen führten. Ein ideales System wäre also eines, das die atomaren Aktionen mithilfe von Gesten umsetzt und dabei stets performant reagiert, sodass beim Benutzer keine Verunsicherung über die Zuverlässigkeit des Systems entsteht.

Die Natürlichkeit der Gesten ist hierbei essentiell, um diese Anforderungen einhalten zu können. Daher werden Gesten nun separat behandelt.

## 4 Gesten

Gesten sind Teil des alltäglichen Ausdrucks jeglicher Interaktion und Kommunikation. Menschen können sich verständigen ohne jegliche verbale Kommunikation. Taube Menschen unterhalten sich über die Gebärdensprache und auch unterbewusst spielen Gesten eine große Rolle. Gesichtsausdrücke, Bewegung mit den Armen und die Art der Haltung können Aussagen in ihrer Bedeutung verändern oder ihnen andere Ausdruckskraft verleihen. Deutlich wird die Ausdruckskraft von Gesten bei dem Pantomimen Marcel Marceau ([Marceau Foundation \(2008\)](#)), welcher allein durch Mimik und Bewegung den Sinn seiner Darbietung deutlich machen konnte.

Interessant ist auch das kulturelle Verständnis von Gesten. So können gleiche Finger bzw. Handstellungen regional unterschiedliche Bedeutungen haben. Die Geste in [Abbildung 4.1](#) hat im europäischen Raum überwiegend die Bedeutung der numerischen Zahl zwei, in China wird es hingegen als numerische Zahl acht interpretiert.

Die hohe Komplexität von Gesten macht eine genauere Betrachtung dringend erforderlich. Eine Klassifizierung von Gesten fand im linguistischen Bereich durch McNeill ([McNeill \(2008\)](#)) statt. Strebt man eine Steuerung von Computersystem über Gesten an, ist eine allgemeine Betrachtung nicht tragbar. Ein Computersystem, welches komplett den Mensch mit Gestik und Sprache versteht, ist sicherlich eines der Ziele von HCI. Eine solche komplette Erkennung ist zwar ein erstrebenswertes Ziel, doch bisher konnten nur Teilbereiche der



Abbildung 4.1: Regional abhängige Bedeutung von Handgesten (Bild aus [Wikipedia \(2008\)](#))

Erkennung realisiert werden. Gesamtlösungen, welche alle Bereiche abdecken, konnten bis heute nicht erreicht werden.

## 4.1 Gesten Taxonomie

Im HCI Umfeld entwickelte Queck einen ersten Ansatz, aus linguistischer Sicht zur Kategorisierung von Gesten. Auf dessen Basis entwickelten Vladimir Pavlovic, Rajeev Sharma und Thomas Huang ([Pavlovic u. a. \(1997\)](#)) eine Erweiterung. Ausgehend von Hand- und Armbewegungen unterteilen Pavlovic et. al. in Gesten und ungewollte Bewegungen, auf die sie dann aber nicht weiter eingehen. Im Bereich der gewollten Bewegung, also den eigentlichen Gesten, unterteilt er abermals in kommunikative und manipulative Gesten. Pavlovics Überlegung sind im Bereich der kommunikativen Gesten ausgeprägt, die hier dargestellten Überlegungen hingegen bewegen sich im Raum der manipulativen Gesten, so dass eine Erweiterung im Rahmen dieser Arbeit erfolgt.

Es wird nachfolgend davon ausgegangen, dass Computerinteraktionen manipulative Handlungen sind, also keine kommunikativen Eigenschaften haben.

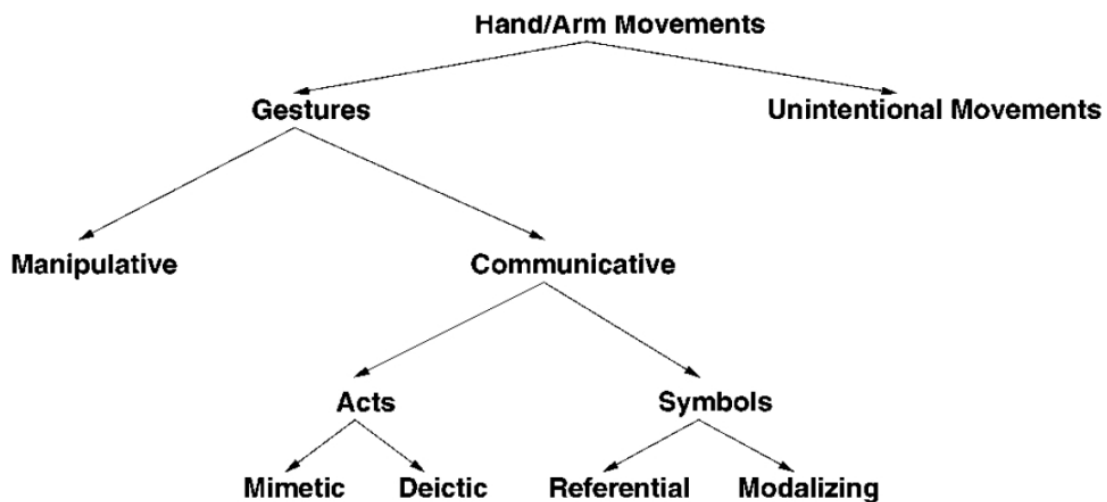


Abbildung 4.2: Taxonomie of Handgestures for HCI (aus [Pavlovic u. a., 1997](#))

## 4.2 Kategorisierung

Bei den laufenden Projekten wurden verschiedene Klassen von Gesten identifiziert, die den manipulativen Teilbaum von Pavlovics Kategorisierung (s. Abb. 4.2) erweitern. Es werden drei Bereiche eingeführt, die unterschiedliche Gesten abdecken.

### 4.2.1 Bewegungsverfolgung

Die Kategorie der Bewegungsverfolgung entspricht am ehesten einer Entsprechung in der physikalischen Welt. Sie zeichnet sich durch direkten Einfluss auf Position und Ausrichtung aus. Das Schieben eines Objektes (Abb. 4.3) oder das Rotieren fallen in diese Klasse von manipulativen Gesten. In dieser Kategorie finden sich die am meisten benutzten intuitiven Gesten wieder, welche sich aus der physikalischen Welt ableiten lassen.

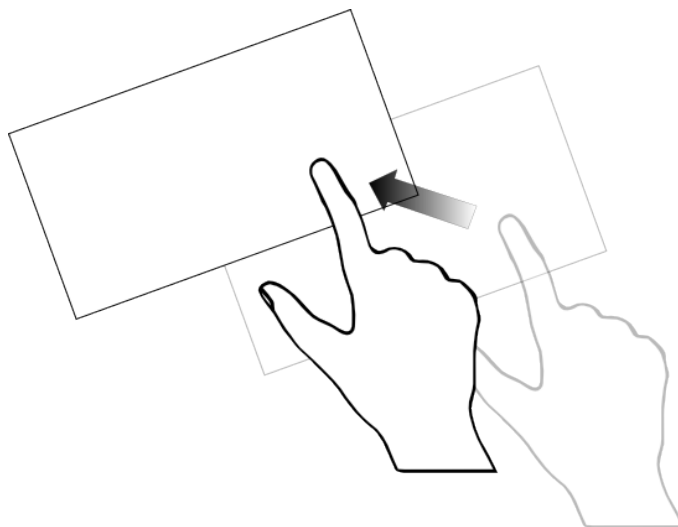


Abbildung 4.3: Verschieben eines Objektes

Bewegungsverfolgung zeichnet sich ebenfalls dadurch aus, dass eine Reaktion des Objektes sofort ersichtlich ist. Zu große zeitliche Verzögerungen bis zur Aktion stellen entweder einen Fehler dar oder es fällt in eine andere Kategorie von Gesten.



## 4.2.2 Kontinuierliche Gesten

Kontinuierliche Gesten bilden das Bindeglied zwischen Bewegungsverfolgung und symbolisch-manipulativen Gesten. Es gibt kaum reale Entsprechung in der physikalischen Welt, aber sie lassen sich anhand von Metaphern und mentalen Modellen leicht erklären und erlernen. So kann man das Vergrößern und Verkleinern von Objekten anhand eines Gummieffektes erklären. Auch massive Werbemaßnahmen, wie im Falle von Apple für das iPhone können dazu beitragen, dass bestimmte Gesten einem mentalen Model entsprechen und klassifiziert werden. Eine Abgrenzung zu der Bewegungsverfolgung ist der Vergleich mit Bewegungen eines Objektes im Raum oder auf der Ebene und einem Ausrichtungswechsel (4.3) eines Objektes (Pan).

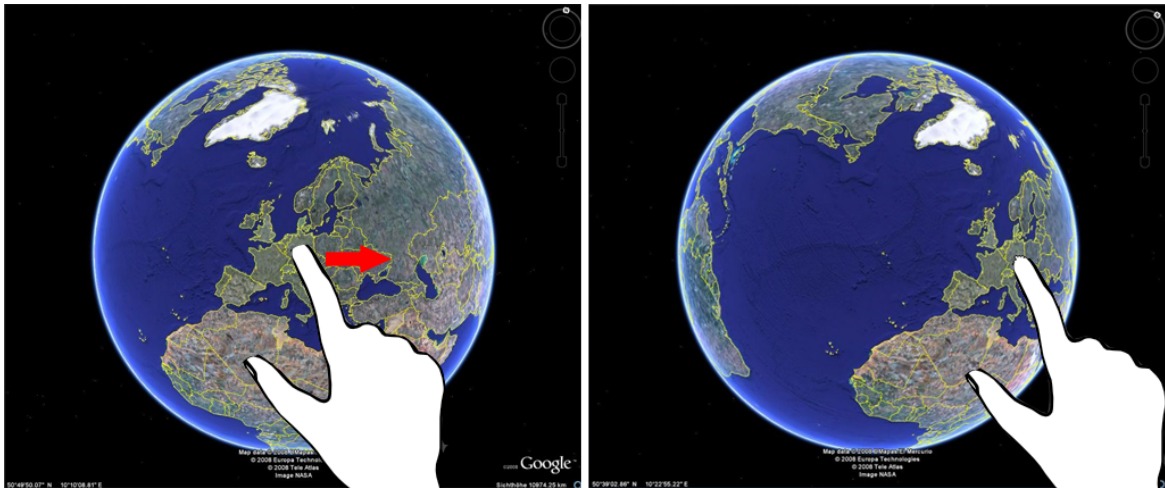


Abbildung 4.4: Pan eines Globus

Wie in der Abbildung 4.4 zu sehen ist, wird durch die Bewegung der Globus gedreht. Was in der Realität dem Drehen eines frei schwebenden Globus entsprechen würde. Beim Verschieben des kompletten Globus würde man sich bei der Bewegungsverfolgung wiederfinden.

## 4.2.3 Symbolisch-manipulative Gesten

Symbolisch-manipulative Gesten sind oftmals mit einem gewissen Lernaufwand bzw. mit einem sehr abstrakten mentalen Model verbunden. Diese Kategorie der Gesten sind weniger intuitiv, da keine, eine sehr schwer vorstellbare oder sehr abstrakte reale Entsprechung existiert. Allerdings erweitert sie die Gruppe der nutzbaren Gesten, mit dem Einsatz von

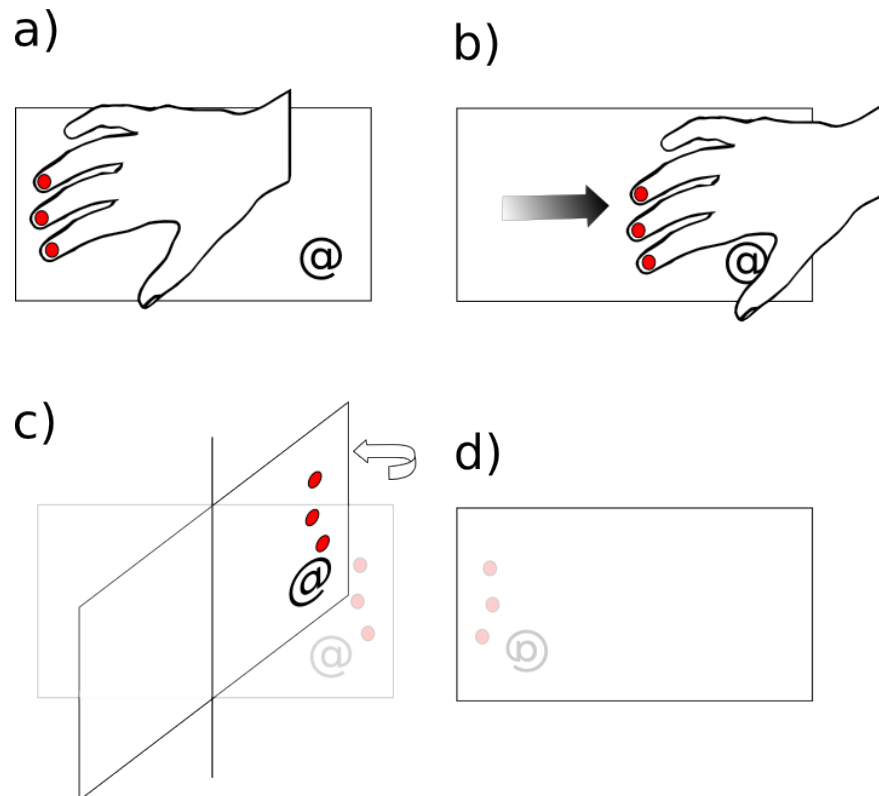


Abbildung 4.5: Flip eines Bildes mit drei Fingern

komplizierteren mentalen Modellen auf eine große Anzahl von Gesten. Stellt man sich das Umblättern einer Seite in einem Buch vor bei dem mit mehreren Fingern eine Seite einfach beiseite geschoben wird, so lässt sich das Umblättern oder Umdrehen mit dieser Geste verbinden. Als Beispiel kann man hier ein Drehen um die X oder Y Achse im Raum (Flip s. Abb. 4.5) nennen welches in einem Prototypen mit einem Ziehen mit mehreren Fingern über das Bild erreicht wurde. Ein weiteres Beispiel ist das Stapeln und systematische Verstreuung von Objekten wie in der Anwendung von Bumptop ([Agarwala und Balakrishnan \(2006\)](#)) zu sehen ist. Eine Entsprechung in der Realität ist oftmals nur schwer findbar. Die Bindung einer Semantik an eine Geste kann also fast beliebig stattfinden, da Erwartungshaltungen zwar eine Rolle spielen, allerdings hier ein gewisses Maß Lernaufwand in Kauf genommen wird. Diese Gruppe der Gesten bietet interessante und faszinierende Möglichkeiten, allerdings muss durch die weniger intuitive Semantik Zuschreibung der Gesten und ihr Einsatz in Applikationen genau geprüft und validiert werden. Ein Auswahlverfahren für Steuerungsaufgaben stellt hier eine gesonderte Herausforderung dar.

### 4.3 Start- Stoppbedingungen von Gesten

Das Erkennen von Anfang und Ende einer Geste stellt eine weitere Herausforderung dar. Übergänge von Bewegungsverfolgung in eine symbolisch-manipulative Geste sind schwer bis gar nicht zu erkennen. Durch Reglementierung, wann Gesten eingesetzt werden können und dürfen, ließe sich das Problem zwar eingrenzen, würde allerdings wieder zu einer erhöhten Komplexität und Reglementierung der Anwendungen führen.

Wann eine Geste beginnt und wann sie endet, hängt oftmals mit dem Anfassen und Loslassen der Touch-sensitiven Oberfläche zusammen. Dies kann man als Anhaltspunkte für eine Geste benutzen. Auch das Starten oder Beenden einer Bewegung kann ein Indikator sein. Eine genauere Betrachtung über Start-Stoppbedingungen im 3D Raum beschreibt Joachim Boetzer ([Boetzer \(2008\)](#)), die beschriebenen Ansätze lassen sich auf den 2D Raum reduzieren und sind somit auch für den Multitouch Bereich denkbar.

### 4.4 Gesten in Anwendungskontexten

Neben der Ausrichtung der Hardware hat auch die Lage der virtuell aufgebauten Welt einen Einfluss auf die Bedeutung der Gesten. Eine Navigation durch 3D-Gebäudemodelle kann eine andere semantische Belegung von Gesten erfordern, als auf einer planen 2D Umgebung. Durch solche Faktoren kann die Bedienung komplizierter werden. Um das Verständnis zu fördern, sollte eine gesonderte Betrachtung je nach Anwendung durchgeführt werden. Eine gleiche Belegung von Gesten in unterschiedlichen virtuellen Räumen ist erstrebenswert, um die Bedienbarkeit möglichst einheitlich und einfach zu halten. Dies stellt allerdings einen erheblichen Aufwand in der Überlegung von sinnvoll einsetzbaren Gesten dar. Daher wird es für sinnvoll erachtet, im Bereich der Bewegungsverfolgung und der kontinuierlichen Gesten zu bleiben. Jedoch sind Gesten, durch die beschriebenen Probleme die unterschiedlichen Welten abzubilden, schnell in der Gruppe der symbolisch-manipulativen Gesten wiederzufinden.

Analog zur vorgenommenen Kategorisierung, werden folgende Gesten, als Anforderungen an ein gestenbasiertes System eingeführt.

- Zweidimensionale Gesten definieren sich über die X-Y-Veränderungen von Punkten über die Zeit. Die Start- und Endpunktproblematik von Gesten kann in diesem Fall vereinfacht betrachtet werden. Es besteht zwar nach wie vor die Problematik beim Erkennen, wenn Gesten ineinander übergehen, allerdings findet durch Auflegen des Fingers und somit der Generierung eines neuen Berührungspunktes und dem Abnehmen eines Fingers und dem somit deutlich erkannten Ende einer Interaktion ein zusätzlicher Informationsgewinn statt. Dieser kann zur zusätzlichen Interpretation

genutzt werden und kann das Erkennen von Gesten und die Steuerung von Applikationen erleichtern und zusätzliche Interaktionsmöglichkeiten bieten <sup>1</sup>.

- Gesten in Physikbasierten Umgebungen sind Manipulationen in der realen Welt sehr ähnlich und erfordern somit wenig Lernaufwand für den Benutzer. Besonders in Verbindung mit Touch-Technologien kommen die Interaktionen den Mental Models (2.2.1) sehr nahe. Zudem wird durch die Darstellung der Zustände des Objektes während der Manipulation immer ein direktes Feedback über den Status des Objektes an den Benutzer geliefert. Dies kommt der intuitiven Bedienbarkeit unmittelbar zugute. Die Interaktion wird in physikbasierten Umgebungen mittels Joints realisiert. Joints sind in der Darstellung nicht sichtbare, federartige Strukturen, welche zwei Objekte mit physikalischen Eigenschaften miteinander verbinden. Wird eines der Objekte bewegt, so wirkt eine Zugkraft, je nach Stärke der Feder, auf das verbundene Objekt (siehe Abb. 4.6). Die aktuelle Position des Fingers wird auf ein unsichtbares Objekt abgebildet, welches mit einem darunterliegenden Objekt mittels eines Joints verbunden wird. Wenn der Finger nun bewegt wird, spannt sich die Feder bis zu einem bestimmten voreingestellten Grad, so dass die Zugkraft größer wird als die entgegenwirkenden Kräfte, wie Gewicht und Anziehungskraft in Verbindung mit Reibung und Oberflächenbeschaffenheit, des zu bewegenden Objektes (Roßberger (2007)). Auf diese Weise lassen sich schon mit dem Einsatz eines Fingers sinnvolle Gesten umsetzen. Unter anderem gehören hierzu das Verschieben und Rotieren von Objekten, wie in Abb. 4.7 zu sehen ist. Auch zu sehen ist dort der Einsatz einer Zwei-Finger-Geste zum Skalieren der Größe.

Es bleibt zu überprüfen inwieweit symbolisch-manipulative Gesten in physikbasierten Umgebungen die Realität nachbilden oder nachbilden können.

- Dreidimensionale Gesten, wie sie zum Beispiel bei Motion Tracking verwendet werden spielen in dieser Arbeit nur eine untergeordnete Rolle. Eine genauere Betrachtung findet in Arbeiten von Boetzer (2008) und Heitsch (2008) statt.

## 4.5 Gesten in Multitouch Systemen

Gesten in Multitouch Systemen zeichnen sich hauptsächlich durch das direkte Anfassen der virtuellen Datenobjekte aus. Der Einsatz des gesamten Körpers bleibt hierbei aus, da die Interaktion mit den Fingern und Händen stattfindet. Dies ist keine Einschränkung, wie es

---

<sup>1</sup>Das iPhone benutzt das Loslassen über einem Buchstaben als Bestätigung bei der Tastatureingabe

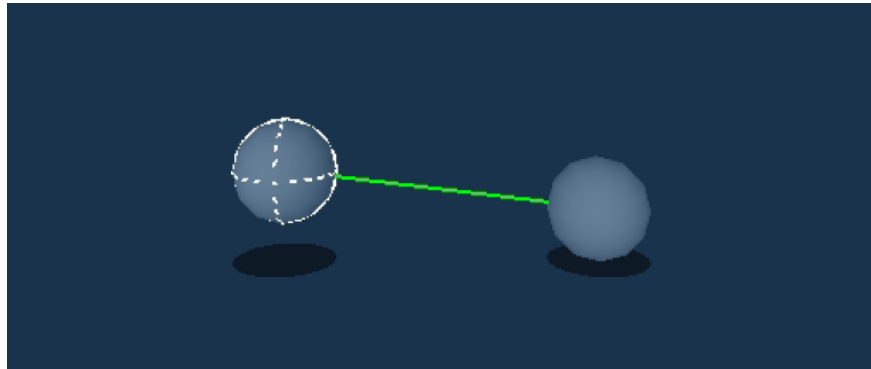


Abbildung 4.6: Die Verbindung zwischen zwei Objekten mit physikalischen Eigenschaften mittels Joint aus [Roßberger \(2007\)](#)

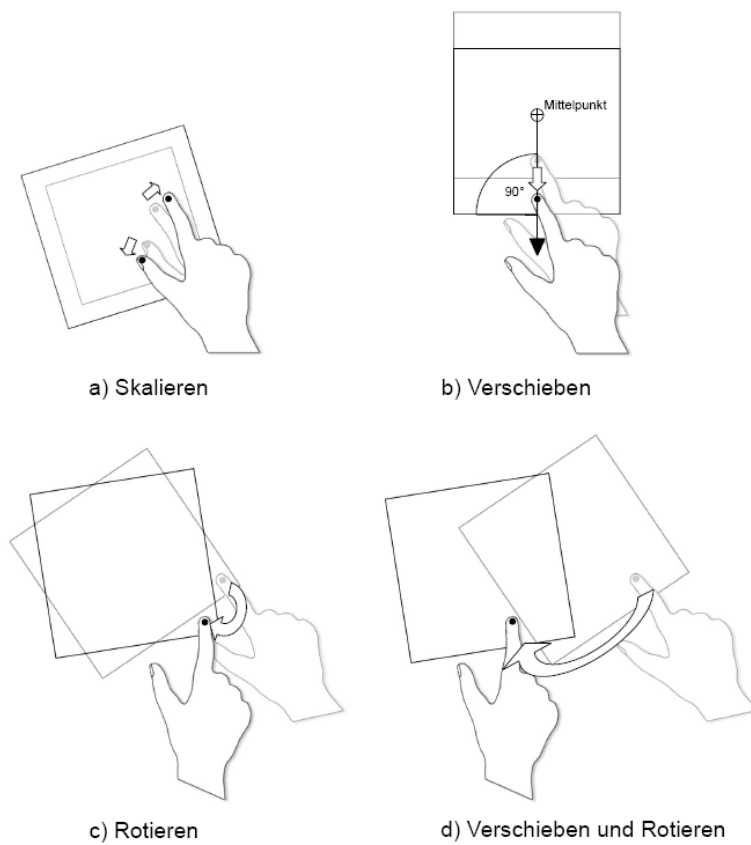


Abbildung 4.7: Mögliche Interaktionstechniken aus [Roßberger \(2007\)](#)

vielleicht zu erwarten wäre. Laut Apple Patent ([Apple Inc. Multi-touch Gesture Dictionary Patent Application \(2007\)](#)) ergeben sich pro Hand etwa 300 mögliche Gesten.

## 4.6 Fazit

Gesten sind ein umfangreiches Thema, welches im Rahmen dieser Arbeit Disziplinen wie Linguistik und Informatik zusammenbringt. Es ist viel Kommunikation zwischen diesen Bereichen nötig um optimale Ergebnisse zu erzielen, da diese Disziplinen sehr artfremd sind.

Aufgrund den Erfahrungen aus der Analyse ist klar geworden, dass Gesten in Zusammenhang mit Computersystemen, gerade im Bereich der Multitouch Anwendungen, ein noch recht frisches Forschungsgebiet darstellen, welches noch nicht durch starke Konventionen eingeschränkt ist. Dies bedeutet allerdings auch, dass es keine Richtwerte gibt, welche einzuhalten sind, um ein erfolgreiches System zu garantieren. Daraus resultieren Freiheiten in der Entwicklung solcher Systeme, aber auch Unsicherheiten, da es kein wirkliches „Richtig“ oder „Falsch“ gibt.

Unter Berücksichtigung der Vorarbeiten, welche von HCI (2.1) Begründern geleistet wurden, konnte im Rahmen dieser Arbeit ein Gestenrepertoire aufgebaut werden, welches es nun gilt in einem Design einzubetten und in einer Realisierung umzusetzen.

# 5 Design und Realisierung

Aufgrund der vorliegenden Analyse und den Erwartungen, welche an solch ein System gestellt werden, wurden Applikationen entwickelt, woran die analysierten Konzepte evaluiert werden können und der Einsatz von Multitouch mittels von uns entwickelter Software in Applikationen ermöglicht werden soll.

Inspiziert durch die verschiedensten existierenden Anwendungen, wie in den Grundlagen schon vorgestellt (2.3.3) wurde an der Multitouch-Wall Anzeige und Interaktionsmöglichkeit für Multimediale Inhalte geschaffen. Bei dem Multitouch-Tisch wurde basierend auf der Arbeit von Roßberger (2007) um Multitouch-Funktionalitäten erweitert. Berücksichtigt wurde bei diesen Systemen eine mögliche Einbettung in das Ambient Labor der HAW Hamburg, um eine Portierbarkeit der Systeme in andere CCW-Umgebungen zu ermöglichen.

Die Entwicklung dieser Systeme und erhaltene Rückmeldungen von Besuchern des Labors schufen ein tieferes Verständnis für diese Technologie und deren Anforderungen auf der Design- und Realisierungsebene (5.5). Die hier vorgestellten Systeme orientieren sich an dem Folgenden visionären Design.

Um das Verständnis für die geführte Diskussion um Design und Realisierung zu stärken, werden die erstellten Systeme hier kurz vorgestellt und dann das Design tiefergehend erläutert, welches Applikationen ermöglichen soll, um Multitouch als Form der Eingabe und Interaktion nutzbar zu machen. Darauf wird die tatsächliche technische Umsetzung, welche unter Berücksichtigung der Anforderungen und des Design erstellt wurde, unter 5.4 vorgestellt.

Zunächst wird ein kurzer Einblick in den Anwendungskontext gegeben, in dem sich diese Systeme befinden, um eine Abgrenzung zu anderen Geräten im Ambient Labor zu schaffen.

## 5.1 Anwendungskontext

Die Multitouch-Wand und -Tisch werden im Ambient Labor der HAW Hamburg betrieben. Das Labor ist im Sinne des CCWs aufgebaut und ermöglicht es so, unterschiedlichste Geräte miteinander zu verbinden.

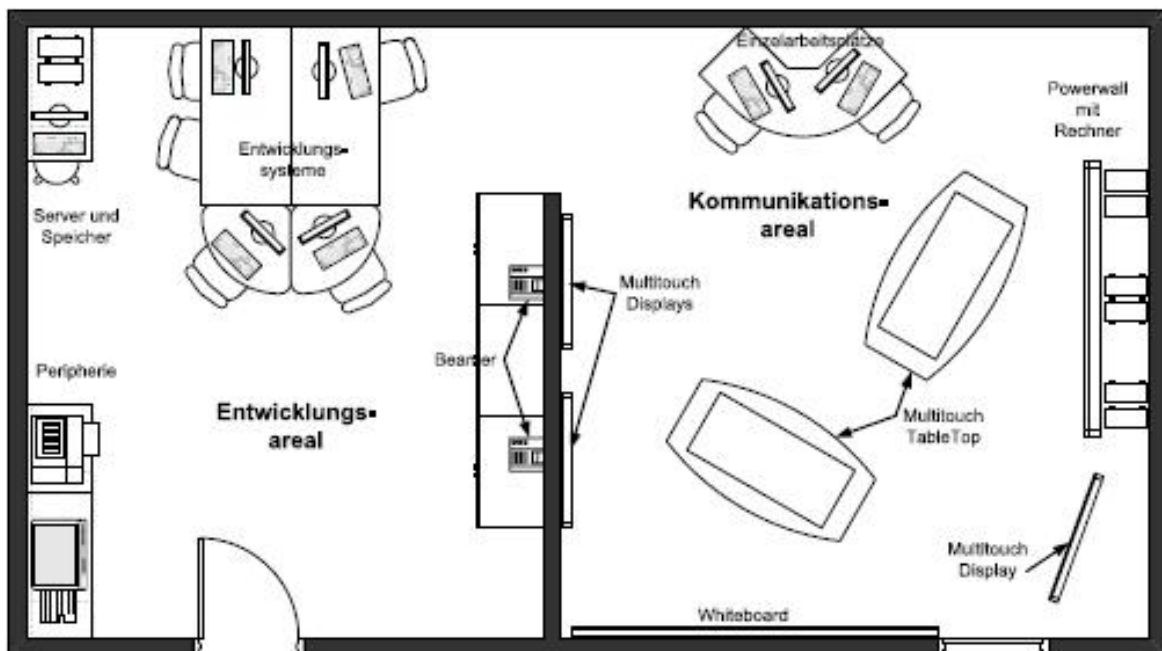


Abbildung 5.1: Schematischer Lageplan der Geräte des Ambient Lab aus [Fischer \(2007\)](#)

Hierzu gehören unter anderem Computer und deren Anzeigen, sowie eine Speicherarchitektur und die Netzwerkanbindung zueinander. Ein schematischer Lageplan der Geräte des Ambient Labors ist in Abb. 5.1 zu sehen. Hieraus wird ersichtlich, dass die berücksichtigten Interaktionsflächen von Multitouch-Wand und -Tisch im Vordergrund stehen, da sie eine signifikante Rolle in der computergestützten Zusammenarbeit von Menschen spielen.

Als Referenzbeispiel für ein solches Labor dient der IRoom aus Stanford ([Winograd \(2003\)](#)), welcher dort im Rahmen des Interactive Workspace Projekt entwickelt und eingesetzt wird. Dort wurde unter anderem auch das iTable entwickelt, welches der Erforschung neuer Interaktionstechniken dient ([Stanford HCI Group's iTable \(2008\)](#)).

Auch das Fraunhofer Institut für integrierte Publikations- und Informationsdienste hat Geräte für CCW im Rahmen von „AMBIENTE - Arbeitswelten der Zukunft“ entwickelt<sup>1</sup>. Sie fassen unter dem Begriff Roomware alle ihre Entwicklungen, wie z.B. Wände, Türen und Tische, welche mit Technik ausgestattet sind, zusammen. Unter anderem gehören dazu auch die DynaWall, eine große Display-Wand, auf der Informationen dargestellt und manipuliert werden können, und der InteracTable, ein Tisch mit eingelassenem Display auf dem Informationen angezeigt und manipuliert werden können.

<sup>1</sup><http://www.ipsi.fraunhofer.de/>



### 5.1.1 Gewählte Hardware

In dieser Arbeit dient ein Infrarot-Touch Rahmen (2.3.1) als Hardware für die Eingabe von Single- und Multitouch Benutzer-Aktionen. Dieser Rahmen hat aufgrund der verbauten Technologie die Eigenschaft, mehrere Unterbrechungen auf den Sensoren zur gleichen Zeit anzugeben. Auf diese Weise, erhält man eine Hardware die es ermöglicht, ein Multitouch-System aufzubauen.

Diese Hardware wurde gewählt, da sie besonders gut in der Größe skalierbar ist und sich durch einen nahezu wartungsfreien Betrieb auszeichnet. Es sind Rahmen mit bis zu 150" Diagonale verfügbar. Auch die Geschwindigkeit, mit der Datenpakete über erkannte Unterbrechungen vom Rahmen geliefert werden, ist für das hier zu entwickelnde System ausreichend.

Die Ausrichtung des Rahmens ist beliebig, welches es ermöglicht die Hardware sowohl für die Wand als auch den Tisch zu nutzen. Dies ist ein Grund, warum eine gemeinsame Basis-Architektur beider Systeme nötig ist. Nachteilig ist diese, da sehr viel Entwicklungsaufwand betrieben werden muss, um sie so allgemein wie möglich zu halten. Ist dieses Ziel der Portierbarkeit allerdings erreicht, bietet diese Art der Architektur sehr viele Vorteile. Unter anderem ließe sich z.B. ein komplett unabhängiges und leicht modifizierbares Controller Modul (5.3.2) erzeugen, welches dann als API für andere Multitouch-Systeme und Applikationen dienen kann.

Um die Eigenschaften der Hardware besser kennenzulernen, wurden im ersten Schritt kleinere Testapplikationen entwickelt, die zur ersten Evaluation der genutzten Hardware dienen. Hierbei stellten sich spezielle Eigenschaften heraus, die aufgrund der verbauten Technologie inhärent vorhanden sind.

#### Eigenschaften der Hardware

Der Rahmen wird über eine serielle Schnittstelle betrieben und hat eine mögliche maximale Übertragungsrate, die im Bereich von 38400 Baud liegt. Dies ergibt bei variierender Paketgröße eine gemessene Rate von etwa 26 Paketen pro Sekunde für einen 42"-Rahmen. Diese Paktrate ist bei kleineren Rahmen zwar höher, pendelt sich aber bei größeren Rahmen bei dem gemessenen Wert ein. Verglichen mit Mäusen ist der Rahmen also etwa um den Faktor 6 langsamer in der Abtastrate.

Eine Beeinträchtigung in der Entwicklung stellt dies nicht dar, allerdings wurde durch einen Rahmen in einer neueren Revision mit höherer Datenrate festgestellt, dass höhere Übertragungsraten eine natürlichere Bedienung ermöglichen. Begründet liegt dies in der Verfolgbarkeit von Bewegungen auf dem Schirm, denn während bei schnellen Bewegungen und Datenraten die angegebenen Punkte nur kleine Sprünge machen, werden bei schnellen

Bewegungen und langsamen Datenraten die Sprünge zu groß, um eine zuverlässige Verfolgung zu erlauben.

Die Pakete werden Byte-weise verschickt und beginnen immer mit einer bestimmten Startsequenz. Sie enthalten als Information die Anzahl der erkannten Unterbrechungen auf X- und Y-Achse des Rahmens, deren Werte sowie Breite und Höhe der Unterbrechung. Auffällig ist hierbei, dass die Werte für die X- und Y-Unterbrechungen im Paket nach Reihenfolge der LEDs im Rahmen sortiert sind. Aus diesem Grund lassen sich Zuordnungen von einem bestimmten X zu dem passenden Y nicht vornehmen, da auch keine weiteren Informationen, denkbar wären Zeitstempel, geliefert werden.

Die Konsequenz aus der verlorenen Zuordnung für X- und Y-Werte ist ein Phänomen, was üblicherweise als „Ghost-Points“ bezeichnet wird. In der grafischen Darstellung 5.2 wird ersichtlich, dass beim Auftreten von zwei Unterbrechungen, mit jeweils X- und Y-Koordinate, vier mögliche Punkte entstehen. Es lässt sich mit Bestimmtheit sagen, dass die diagonal zueinander stehenden Punkte zusammengehören, welche der beiden Punktepaare aber nun die „echten“ sind, lässt sich nicht ermitteln. Die Entstehung von Ghost-Points wird bei der Betrachtung der Abb. 5.3 sehr deutlich. Hier ist zu sehen, wie die Infrarotstrahlen verlaufen, und dass sich die Unterbrechungen der aufgesetzten Objekte so auf die Erkennung auswirkt, dass nicht mehr klar ist, welche der Punktepaare den aufgesetzten Objekten entsprechen.

Auch dies stellt keine Beeinträchtigung dar, solange eine akkurate Verfolgung der Objekte nicht notwendig ist, wie z.B. bei einer Vergrößerungsgeste, bei der zwei Finger diagonal über ein virtuelles Datenobjekt gezogen werden.

Eine weitere Schwierigkeit ergibt sich aus der Überschattung von mehreren horizontal nebeneinander oder vertikal über- oder untereinander liegenden Unterbrechungen. Hierbei wird eine Unterbrechung von der neben ihr liegenden so überschattet, dass nur noch ein X- oder ein Y-Wert vom Rahmen ausgegeben wird, obwohl es eigentlich zu jeder Unterbrechung jeweils einen geben müsste. In Abb. 5.4 ist zu sehen, wie bei zwei Unterbrechungen nur drei Koordinaten entstehen, da die etwas größere Unterbrechung den X-Wert der kleineren überschattet.

Dies stellt ein Problem dar, wenn ein kontinuierliches Objekt-Tracking realisiert werden soll oder ein Mehrbenutzerbetrieb in Betracht gezogen wird. Denn für diese Anwendungen muss es möglich sein, verlässliche Werte für die Unterbrechungen zu erhalten.

Durch den im Rahmen verwendeten Abtastalgorithmus für die IR-Empfangsmodule, tritt eine mögliche Verzögerung oder gar ein Ausfall der Erkennung von einer oder mehreren Unterbrechungen ein. Da der Status der Module der Reihe nach ermittelt wird, können bei nur kurzzeitig aufgetretenen Unterbrechungen unter Umständen nicht alle Unterbrechungen rechtzeitig aufgenommen werden. Dies äußert sich in fehlenden Koordinaten, welche im Datenpaket schlichtweg fehlen und nicht nachgeliefert werden. Für genauere Informationen siehe [Gehn \(2007b\)](#).

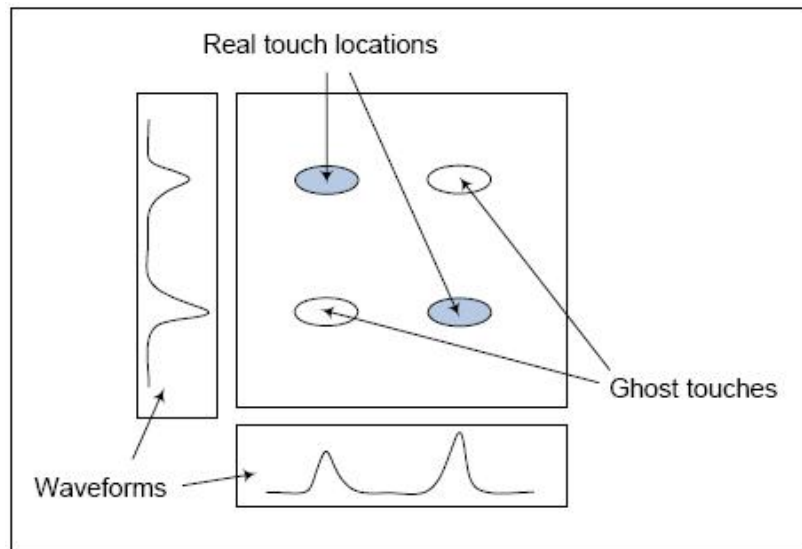


Abbildung 5.2: Grafische Darstellung von Ghost Points aus [NextWindow Limited, Multitouch Whitepaper \(2008\)](#)

Unter Berücksichtigung dieser Eigenschaften wurden die Wand und Tisch Applikationen entwickelt, deren Design nun beschrieben werden soll.

## 5.2 Grundlegende Funktionalitäten von Multitouch Wand und -Tisch

Im Rahmen vom Ambient Lab der HAW werden Applikationen entwickelt und auf der gewählten Hardware (siehe [5.1.1](#)) in Betrieb genommen. Die Applikationen ermöglichen es den Einsatz von Multitouch-Wand und Tisch zu evaluieren. Da die Experimente und Evaluationen von Tisch und Wand, wie in der Analyse formuliert, unterschiedliche Ausrichtungen beinhalten, entstehen unterschiedliche Anforderungen an das Design der Applikationen.

Es sollte eine möglichst einfache wiederverwendbare Basis-Architektur geschaffen werden, mit deren Hilfe schnell unterschiedliche Arten von Applikationen entwickelt werden können, um eventuell auftretende Misstände in der Bedienung aufzulösen, ohne lange Entwicklungszeiten zu benötigen. Diese Basis-Architektur würde auch eine iterative Entwicklung der Applikationen unterstützen, welches dem Testen von intuitiven Bedienoberflächen zugute kommt.

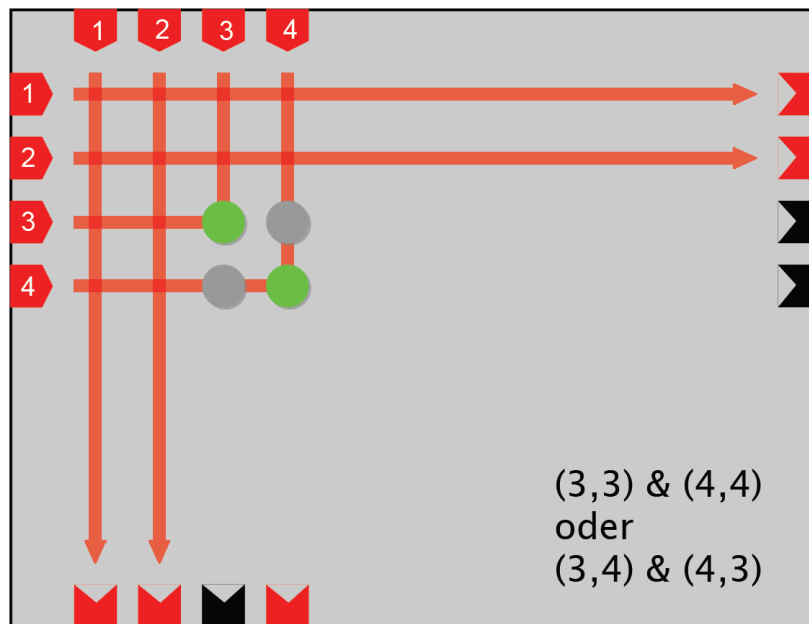


Abbildung 5.3: Grafische Darstellung von Ghost Points aus Gehn (2007a)

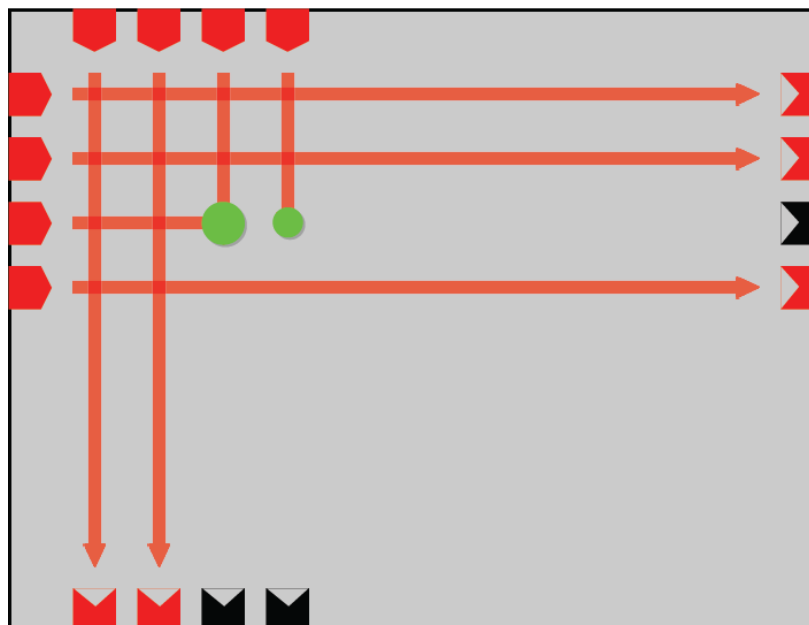


Abbildung 5.4: Grafische Darstellung von Überschattungen aus Gehn (2007a)

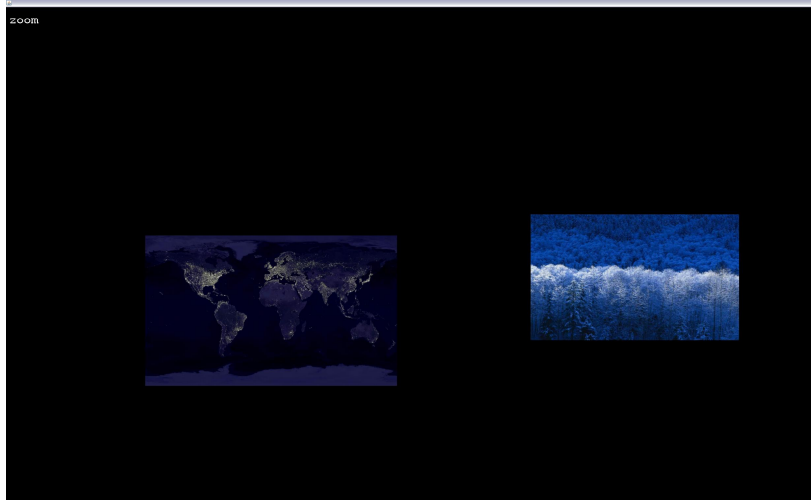


Abbildung 5.5: Prototyp der Bedienoberfläche der Multitouch Wand

Es werden nun kurz die Bedienoberflächen und Funktionalitäten der Wand (5.2.1,5.2.2) und des Tisches (5.2.3,5.2.4) vorgestellt. Darauf folgt die grundsätzliche Architektur der beiden Applikation und eine Beschreibung der einzelnen Subsysteme (5.3).

### 5.2.1 Bedienoberfläche der Multitouch Wand

Als Resultat aus den gewonnenen Erfahrungen mit den Testprogrammen (5.1.1) wird eine Anwendung erstellt, welche zur Darstellung multimedialer Inhalte dient. Neben Bildern und Videos mit Ton werden auch 3D Objekte mit festem Fluchtpunkt im Raum und 3D CAD Zeichnungen zur Anzeige gebracht. Die Abb. 5.5 zeigt eine erste Version der Multitouch Wand Bedienoberfläche, auf der zwei rechteckige Bilder, welche beliebige Datenobjekte darstellen, zu sehen sind. Der Hintergrund ist eine schwarze Wand, auf welcher die Datenobjekte, nach Pinnwand-Metapher, angepinnt und verschoben werden können. Die Applikation läuft auf einem vertikal zum Benutzer ausgerichteten Monitor, um möglichst nahe am Mental-Model zu sein. Die Ausrichtung der Datenobjekte ist in diesem Falle unerheblich, da durch die Ausrichtung des Bildschirms klar ist, wo der Benutzer stehen muss, um mit den Objekten zu interagieren.

Zur Darstellung der Objekte werden unterschiedliche Ansätze verfolgt. Gesucht wird eine Grafik-Engine, welche sich sehr gut für eine iterative Entwicklung eignet. Des Weiteren ist es nötig, dass Änderungen an der Engine machbar sind, da bei Multitouch sehr anpassungsfähige Umgebungen gebraucht werden, um die speziellen Eigenschaften umzusetzen

zu können.

Betrachtet werden hierbei Processing<sup>2</sup>, eine Grafik-Engine zur Manipulation von Bildern und animieren von Inhalten. Sie ermöglicht es, schnell zu einem lauffähigen Prototypen zu gelangen, welcher dann je nach Anforderung ausgebaut werden kann. Des Weiteren ist es ein Open Source Framework, somit ist es möglich Änderungen am Quellcode vorzunehmen und ist in Java implementiert, somit also plattformunabhängig.

Eine weitere in Betracht gezogene Engine ist die JMonkeyEngine<sup>3</sup>, welche sich durch performantes 3D Rendering in Java auszeichnet. Auch in Java implementiert ist Xith3D<sup>4</sup>, welches aber aufgrund mangelhafter Dokumentation nicht weiter zur Auswahl steht.

Da sie bessere Performance bieten werden auch in C++ entwickelte Engines begutachtet. Unter anderem wird hierbei die Open Source Irrlicht<sup>5</sup> und Ogre<sup>6</sup> Engine in Betracht gezogen. Neben diesen Engines käme auch Torque in Frage, aber hier würde das Erlernen der dafür eigens entwickelten Skriptsprache zu viel Zeit in Anspruch nehmen.

Als Programmiersprache kommen hier also hauptsächlich Java und C++ zum Einsatz. Engines beider Sprachen lassen sich auf unterschiedlichen Betriebssystemen lauffähig, wobei bei Java Engines die Plattformunabhängigkeit inhärent ist, während bei C++ Engines nach passenden Compilern gesucht werden muss.<sup>7</sup>

## 5.2.2 Funktionalitäten der Multitouch Wand

Mittels einfacher Handgesten können die verschiedenen Objekte nun mithilfe der Engine manipuliert werden. Aktionen wie freies Anordnen der Objekte auf dem Bildschirm, Vergrößern (s. Abb. 5.6), Verkleinern, Drehen um den Mittelpunkt, Flip um die horizontale und vertikale Achse können direkt durch Gesten erreicht werden. Durch Anfassen eines Objektes mit der kompletten Hand werden weitere Aktionen angezeigt, wie z.B. das Schließen eines Objektes oder das Herstellen der ursprünglichen Ausrichtung, welche dann ausgewählt werden können. Durch die Möglichkeit Objekte vom Bildschirm zu löschen, sollte auch das Öffnen von Inhalten ermöglicht werden. Es wird eine Interaktion mit dem Hintergrund eingeführt, welche es ermöglicht, einen Objektgenerator aufzurufen, über den dann Multimedia-Inhalte auf dem Bildschirm neu geöffnet werden können.

---

<sup>2</sup><http://www.processing.org>

<sup>3</sup><http://www.jmonkeyengine.com/>

<sup>4</sup><http://xith.org/>

<sup>5</sup><http://irrlicht.sourceforge.net/>

<sup>6</sup><http://www.ogre3d.org/>

<sup>7</sup>Wenn eine virtuelle Maschine verfügbar ist, ist in Java geschriebener Code einfacher portierbar.

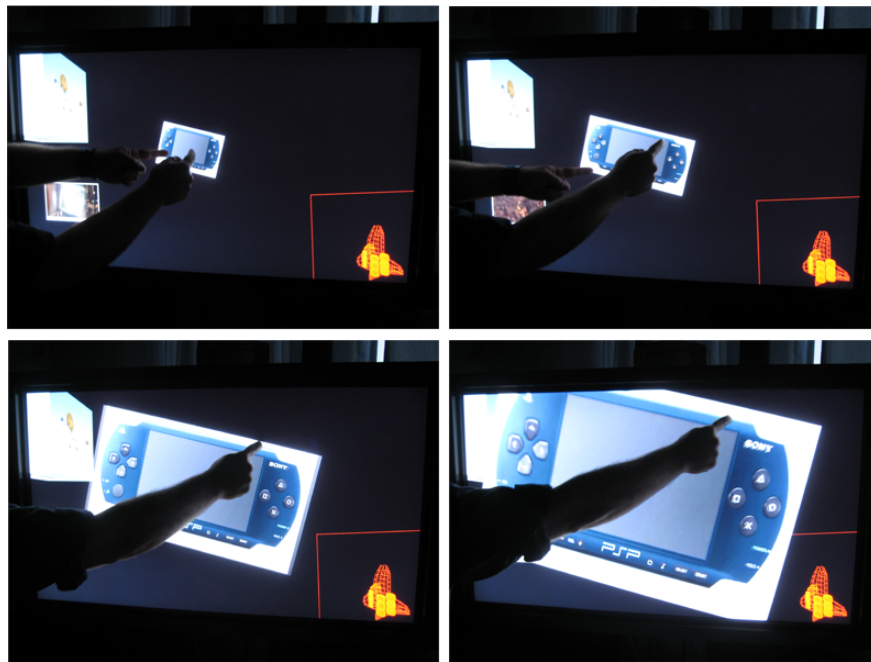


Abbildung 5.6: Vergrößern eines Objektes

Hier wird neben den Handgesten auch ein Konzept für aktive Flächen getestet. Nach der Auswahl des Bildes muss eine gewisse Zeit die Aktivierungsfläche gedrückt werden, um eines neues Objekt zur Anzeige zu bringen.

Die Anzeige von CAD Zeichnungen und 3D Objekten setzt eine 3D Umgebung voraus. Es wird eine Ebene erstellt, auf der sich sämtliche 2D Objekte befinden und auch in der Rotation diese Ebene als Ausgangspunkt betrachten, ähnlich wie bei einer Pinnwand (s. Abb. 5.7). Bei 3D Objekten wird ebenfalls diese Ebene als Mittelpunkt ausgewählt. Bei den CAD Zeichnungen wird auf eine Fixierung auf eine Ebene verzichtet, da die Betrachtung aus allen Richtungen und die Möglichkeit in das Objekt zu zoomen, Grundlagen für die Interaktion mit CAD Zeichnungen sind.

Es werden je nach Inhalt der Objekte verschiedene Funktionen extra implementiert. So können Videos gestartet oder gestoppt werden, Bilder können Vorder- und Rückseiten besitzen und 3D Objekte wie ein Würfel werden nicht wie bei Bildern und Videos um  $180^\circ$  gedreht, sondern lediglich um  $90^\circ$ , um eine Seitenansicht zu ermöglichen. Experimentell wird auch das Bewegen von zwei Bildern gleichzeitig implementiert. Aufgrund der von der Hardware ausgehenden Schwierigkeiten, muss auf eine komplette Realisierung verzichtet werden.

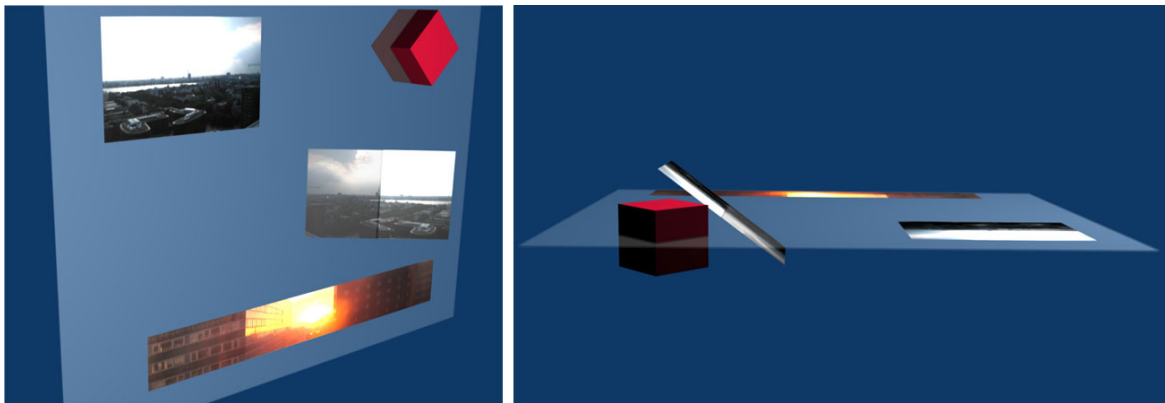


Abbildung 5.7: Pinnwandmetapher aus verschiedenen Ansichten

### 5.2.3 Bedienoberfläche des Multitouch-Tisches

Für den Multitouch-Tisch werden die oben genannten Grafik-Engines auch in Betracht gezogen, vor allem interessant sind hierfür die JMonkeyEngine, Irrlicht und Ogre. Alle drei bieten die Möglichkeit an, eine Physik-Engine zu benutzen oder, wenn nicht vorhanden, einzubinden.

Als eine fertige Applikation wird auch Dynambient als Fundament für die tischbasierte Applikation evaluiert.

Dynambient wurde im Rahmen der Masterarbeit von [Roßberger \(2007\)](#) erstellt. Als Grafik-Engine dient hier Irrlicht<sup>8</sup>, welche zur Erstellung von 3D Spielen genutzt wird. Die physikalische Simulation wird von der PhysX-Engine von NVIDIA<sup>9</sup> übernommen.

Als alternative Physik-Engines stehen auch ODE (Open Dynamics Engine<sup>10</sup>) und die Bullet Physics Engine<sup>11</sup> zur Verfügung, wobei die Letztere aufgrund mangelnder Zeit für die Einarbeitung nicht weiter betrachtet wird. ODE wurde in den ersten Prototypen verwendet, wobei sich herausstellte, dass die Konfiguration zu zeitintensiv ist, um die Verwendung weiter zu verfolgen.

Die erste Experimente zur Organisation der Bedienoberfläche orientieren sich stark an dem Mental Model für einen realen Tisch, auf dem Karten mit aufgedruckten Bildern liegen. In [Abb. 5.8](#) sind einige dieser Karten zu sehen, die auf einer planen und statischen Oberfläche liegen, die mit einer Stofftextur überzogen ist. Sie werfen einen Schatten der durch eine Lichtquelle, die oberhalb von der Kamera angebracht ist, erzeugt wird.

Die Karten liegen, wie bei einem echten Schreibtisch Papiere, auf der Tischfläche und überdecken sich zum Teil. Es kann eine gewisse Dicke dieser Objekte ausgemacht werden.

<sup>8</sup><http://irrlicht.sourceforge.net/>

<sup>9</sup>[http://www.nvidia.com/object/nvidia\\_physx.html](http://www.nvidia.com/object/nvidia_physx.html)

<sup>10</sup><http://www.ode.org/>

<sup>11</sup><http://www.bulletphysics.com/>





Abbildung 5.8: Prototyp der Bedienoberfläche des Multitouch Tisch

Ihre Ausrichtung ist unterschiedlich, da sie rotiert werden können und auch die Skalierung kann unterschiedlich sein.

Jede der Karten hat eine bestimmte Masse und Oberflächenbeschaffenheit. Die führt bei der verwendeten Physik-Simulation dazu, dass die Objekte sich beim Bewegen so verhalten, als wären es reale Karten auf einem Tisch. Die Entscheidung, eine Physik-Engine für den Tisch zu verwenden, wurde gefällt, da sie die Möglichkeit bietet, intuitive Funktionalitäten für eine virtuelle Tischoberfläche zu entwickeln.

#### 5.2.4 Funktionalität des Multitouch-Tisches

Der Benutzer kann die virtuellen Datenobjekte auf dem Tisch mit einem Finger bewegen, indem ein Objekt anfasst und dann mit gehaltenem Finger über die virtuelle Tischfläche zieht (s. Abb. 5.9). Dies ermöglicht das Sortieren und Organisieren von vielen Informationsträgern, welche auf dem Tisch platziert werden können. Bei Kollisionen von unterschiedlichen Objekten während des Vorschiebens, wird das Verhalten dynamisch von der Physik-Engine berechnet. Die Datenobjekte verhalten sich somit so, wie sich auch reelle Karten auf einem echtem Tisch verhalten würden.

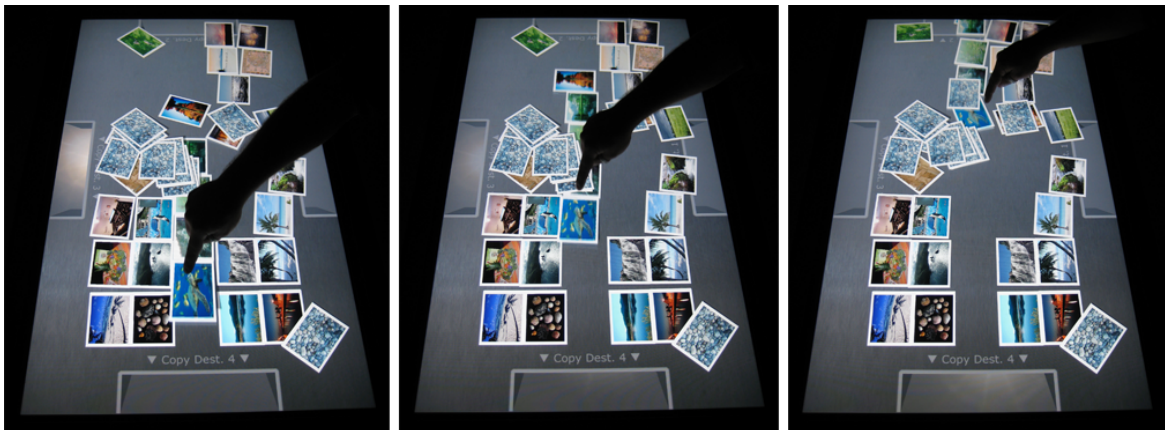


Abbildung 5.9: Bewegen eines Objektes

Durch das Anfassen an einer Ecke des Datenobjektes kann eine Rotation durchgeführt werden, indem eine kreisförmige Bewegung ausgeführt wird. Dies dient dazu, die Ausrichtung des Informationsträgers für einen gegenüberstehenden Benutzer zu korrigieren, damit auch dieser die Informationen sehen oder auch lesen kann. Ein Datenobjekt kann rotiert und gleichzeitig verschoben werden, um es einem anderen Benutzer zuzuwerfen. Die aufgenommene Kraft, und somit die Geschwindigkeit, in der das Objekt fliegt, ist abhängig von der Geschwindigkeit in der diese Geste ausgeführt wird.

Ein doppeltes kurzes Anfassen eines Datenobjektes führt zu einer Vergrößerung, sodass Details betrachtet werden können. Diese Vergrößerung wird durch ein einfaches, kurzes Anfassen wieder aufgehoben.

### 5.3 Architektur

Es handelt sich hier um ein visionäres Design, an dem sich die entwickelten Applikationen orientieren und es auch weitestgehend realisieren. Durch die iterative Entwicklung, speziell in Bezug auf die Schwierigkeiten mit der Hardware, wurde die Software sowie die Architektur immer wieder angepasst, um eine Verbesserung der Ergebnisse zu erreichen. Der in der technischen Umsetzung gezeigte Stand stellt somit einen aktuellen Entwicklungsstand dar. Erfahrungen aus Gesprächen mit Benutzern und Testern führen zu stetig neuen Aspekten, die in die Entwicklung mit einfließen. Die konkrete Umsetzung der Architektur wird in [5.4](#) beschrieben.

Aus der Analyse und den Anforderungen konnten drei Aufgaben identifiziert werden, welche ein System für die Interaktion mit Computersystemen auf Basis von Multitouch Technologie erfüllen muss.

So wird eine virtuelle Repräsentation der Objekte benötigt, eine Komponente welche diese Objekte anzeigt und eine Verwaltung der Objekte, welche Interaktionen mit diesen ermöglicht.

Ein Entwurfsmuster, welches in der Informatik häufige Anwendung findet und der benötigten Aufgabenteilung entspricht, ist das Model View Controller Pattern ([Gamma u. a. \(1995\)](#)).

### 5.3.1 MVC

Das MVC Pattern wurde als allgemeine Lösung der Problematik eines Benutzers, viele komplexe Daten zu verwalten, entwickelt. Die Intention von MVC ist es, die Lücke zwischen dem Mental Model des Benutzers und der Software-Modelle im Computer möglichst gut zu überbrücken.

Applikationen werden hierzu im Wesentlichen in drei Komponenten aufgeteilt ([Reenskaug \(2003\)](#)).

**Das Model** repräsentiert Wissen. Es kann ein einzelnes Objekt sein oder auch eine größere Struktur von mehreren Objekten. Es ist dabei zu beachten, dass das Model und seine Bestandteile eine Eins-zu-Eins-Verbindung zu den zu beschreibenden Objekten in der Realität haben.

**Die View** ist eine visuelle Repräsentation des Models. Es hat die Aufgabe verschiedene Teilaspekte des Models besonders hervorzuheben und andere komplett auszublenden. Somit stellt die View eine Art visuellen Filter dar.

Eine View ist immer verbunden mit einem dazugehörigen Model und erhält von diesem die Daten, die notwendig sind, um eine Darstellung zu generieren, indem es dem Model passende Nachrichten sendet. Somit muss die View die Semantik der Attribute des Models kennen<sup>12</sup>.

**Der Controller** bildet die Schnittstelle zwischen Benutzer und System. Er versorgt den Benutzer mit Eingabemöglichkeiten, bei denen der Benutzer über die View Kommandos abgeben kann. Dies wird realisiert, indem die Kommandos vom Controller entgegengenommen, in Nachrichten übersetzt und an die entsprechenden Models weitergeleitet werden. Der Controller kann keinen Einfluss auf die View ausüben, also keine Veränderungen vornehmen. Andersherum sollte die View keine Kenntnis über die Eingaben des Benutzers haben, sodass es jederzeit möglich ist im Controller Eingaben zu generieren, ohne die View verändern zu müssen.

---

<sup>12</sup>Stärkere Modularisierung kann dazu führen, dass die View die Semantik nicht kennen muss.

Dieses Paradigma wurde auf die Architektur der Multitouch Applikationen übertragen. Wie genau diese nun aussehen wird im Folgenden erläutert.

### 5.3.2 MVC in der Multitouch Architektur

#### Übersicht

Eine schematische Übersicht über die Architektur der Multitouch Anwendungen zeigt Bild 5.10. Nachfolgend werden die Einzelnen Komponenten genauer vorgestellt. Hierbei werden die Zusammenhänge der Komponenten verdeutlicht.

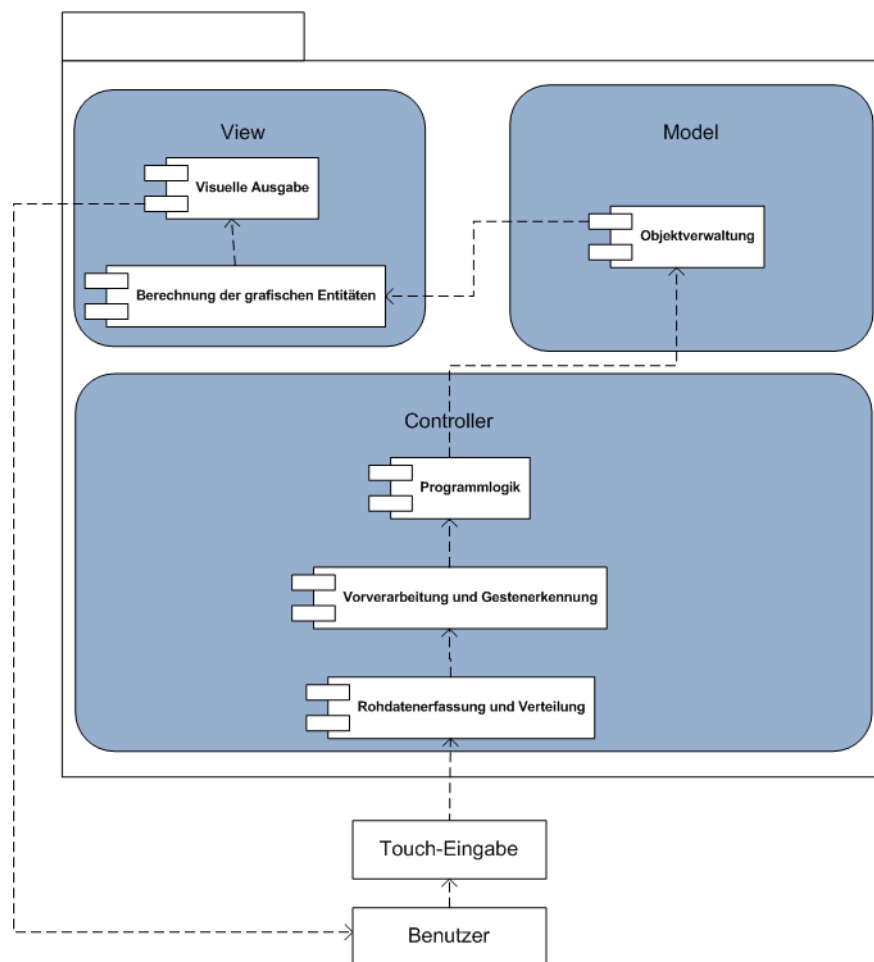


Abbildung 5.10: Schematische Komponentenübersicht

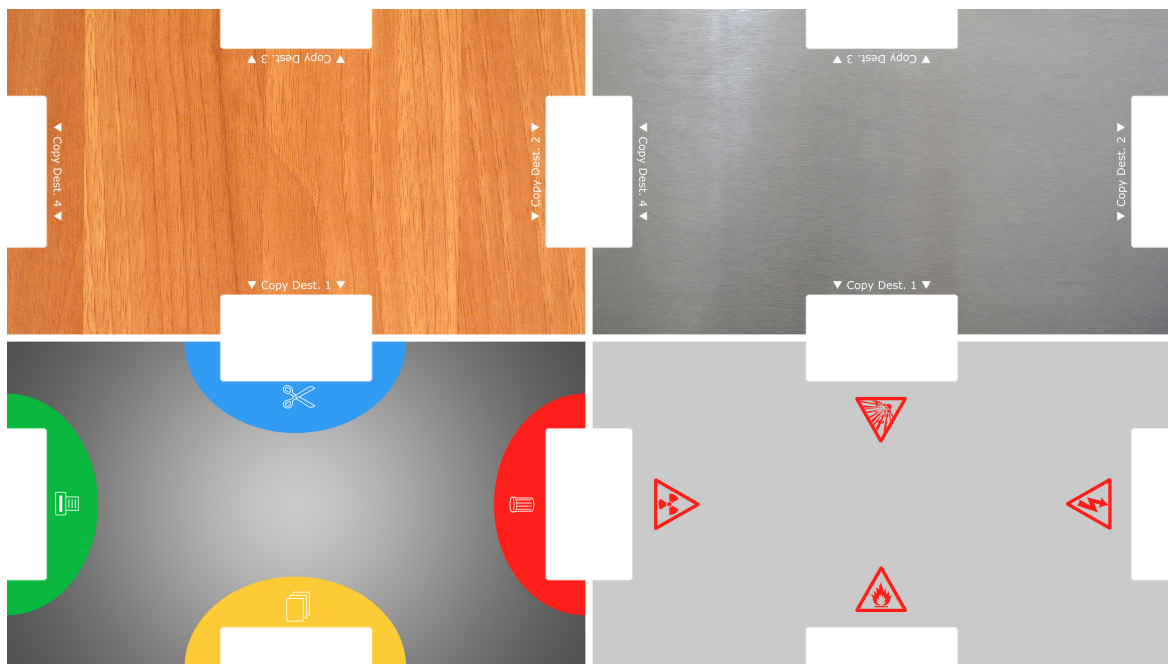


Abbildung 5.11: Verschiedene Textur Varianten für Dynambient aus [Roßberger \(2007\)](#)

## View

Betrachtet man die Szenarien, so wird ein View benötigt, das möglichst universell einsetzbar ist und unabhängig von der Touch Hardware genutzt werden kann.

Die Informationen über Form, Ausrichtung, Texturen und anzuzeigenden Informationen erhält das View über das Model. Dabei kann zur Unterstützung und zum schnelleren Einordnen von Daten z.B. eine Medien typische Umrandung den Benutzer assistieren. Bilder können den Look eines Polaroid Fotos bekommen und Videos die Umrandung einer Filmrolle. Bei gestoppten Videos hilft es so bei der schnellen Zuordnung des Medientypes und dient der intuitiven und einfachen Bedienung.

Die gestalterischen Möglichkeiten sind gerade im Bereich von Multitouch Applikationen vielseitig und unterliegen noch keinen gängigen oder einheitlichen Richtlinien.

Es besteht eine Abhängigkeit des Views von Ausrichtung und Anwendungskontext. Um die intuitive Bedienung zu erreichen, wird sich stets auf ein mentales Modell gestützt [2.2.1](#). Dies hat dann auch direkte Auswirkungen auf die gestalterische Umsetzung und die Sicht auf die Objekte. Es würden sich je nach Ausrichtungen der Kamera oder durch gestalterische Aspekte auch die Erwartungshaltungen an die Steuerungen oder Funktionen ändern. Es können bei DynAmbient je nach Hintergrundtextur(s. [5.11](#)) unterschiedliche Erwartungen

der Funktion auftreten. Änderungen des Kamerawinkels können, gerade in physikbasierten Umgebungen, zu einer veränderten Erwartungshaltung und zu einer weniger intuitiven Eingabe führen. Durch Änderungen der Kameraposition oder des Winkels auf die virtuelle Repräsentation werden z.B. Objektflächen perspektivisch verzerrt oder sind nur noch von der Seite sichtbar. Dies führt zu kleineren *anfassbaren* Objekten, was bei ungünstiger Position der Sicht auf die Arbeitsfläche eine Interaktion unmöglich machen könnte. Auch die Übersichtlichkeit könnte, bei zu weitem herausfahren der Kamera leiden. Eine genaue Betrachtung der Sicht ist dementsprechend wichtig, um die gleichbleibende Bedienbarkeit zu garantieren. Eine Möglichkeit, die Problematiken einzuschränken ist die feste Positionierung der Kamerasicht. Ein Konzept, in der die Ansicht auf die Arbeitsfläche geändert werden konnte, stellte zuletzt Microsoft mit seiner Touch-Wall auf der Microsoft CEO Summit 2008<sup>13</sup> vor<sup>14</sup>.

## Model

Das Model stellt ein virtuelles Objekt dar, mit dem interagiert werden kann. Neben der Form und den relevanten Informationen enthält das Model auch die Logik über die Aktionen, welche ausgeführt werden können. Die Informationen werden von dem View zur Anzeige gebracht.

Je nach Information, welche die Objekte enthalten, müssen andere Aktionen möglich sein. So müssen verschiedene Typen von Objekten unterschiedlich auf Aktionen reagieren können. Bei Videos wäre ein Stoppen und ein Vor- und Rücklauf nötig, was bei einem Bild keinen Sinn machen würde. Allerdings möchte man gewisse Grundfunktionalitäten, die bei allen Objekten gleich sind. So soll das Bewegen eines Objektes immer gleich bei allen Objekttypen funktionieren.

Um die Applikation schnell und einfach um weitere Medientypen zu erweitern, wurde das Entwurfsmuster der Schablonenmethode (engl. template method) eingesetzt (s. Abb. 5.12).

Bei dieser Methode werden grundlegende Funktionen durch ein Interface beschrieben und von einer abstrakten Klasse implementiert. Ein neuer Objekttyp erbt von der abstrakten Klasse die grundlegenden Funktionen und implementiert eigene Funktionen selber, welche auf den Typen speziell zugeschnitten sind. Sollen bestimmte Funktionen, welche durch die abstrakte Klasse implementiert wurden nicht möglich sein, können sie überschrieben werden.

<sup>13</sup><http://www.microsoft.com/presspass/events/ceosummit/default.aspx>

<sup>14</sup>[http://msstudios.vo.llnwd.net/o21/presspass/mpeg2/Touch\\_Wall.mpg](http://msstudios.vo.llnwd.net/o21/presspass/mpeg2/Touch_Wall.mpg)

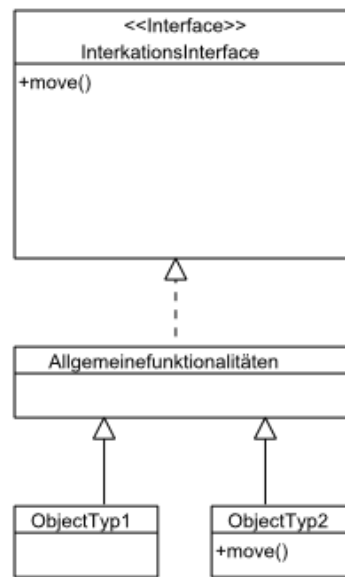


Abbildung 5.12: Beispielhafte Modellierung der selbst verwaltenden Objekte

Betrachtet man die Aufgaben des Models in der Architektur, wird eine genauere Betrachtung des Models und der in ihr enthaltenen Informationen notwendig.

**Position** Das Model muss alle Informationen über die Positionierung im virtuellen Raum enthalten. Da unterschiedliche Räume als Arbeitsebenen zugrunde liegen können, muss hier je nach Anwendung eine Repräsentation gefunden werden. Die wohl gängigste ist die des 3D Raums, welche sich aus drei Koordinaten (x,y,z) zusammensetzt, um eine Position im diesem Raum zu bestimmen.

**Form** Neben der Position müssen auch Informationen über die Form des Objektes gespeichert werden. Je nach zugrunde liegendem virtuellem Raum müssen auch hier, wie bei der Position, entsprechende Modelle angepasst werden. Ebenfalls kann der Inhalt Einfluss auf den Sinn der Form haben. Videos auf kugelförmigen Objekten anzuzeigen, kann zwar einen netten visuellen Effekt haben, aber der Sinn ist zu hinterfragen. Ein weiterer wichtiger Punkt ist die Größe der Form. Zu kleine Objekte können eventuell mit dem Finger nicht mehr angefasst werden. Es kann also sinnvoll sein, eine minimale Größe zu definieren, so dass eine Bedienbarkeit gewährleistet werden kann.

**Ausrichtung** Gerade bei tischbasierten Applikationen ist die Ausrichtung des Objektes von besonderer Wichtigkeit. So wird ein, von einem Bezugspunkt abhängiger Winkel zur Ausrichtung benötigt. Der Bezugspunkt spielt hierbei eine entscheidende Rolle in der

Bedienung. Liegt der Bezugspunkt in der Mitte einer Form, so wird um den Mittelpunkt des Objektes gedreht. Liegt er an einer Ecke oder am Rand der Form, so resultiert daraus ein völlig anderes Rotationsverhalten. In physikbasierten Anwendungen liegt der Bezugspunkt an der Stelle, an der das Objekt angefasst wird, und ist somit leicht zu erklären. In nicht Physik basierten Anwendungen müssen diese Punkte festgelegt werden und legen somit auch die möglichen Arten der Rotation fest.

**Inhalt** Die Inhalte der Objekte können vielfältig sein. Neben einfach medialen Inhalten, wie Texten, Videos, Bildern oder Musik könnten auch komplexe Inhalte wie Applikationen oder spezielle Dateiformate zur Anzeige gebracht werden. Das Model hält lediglich die Information über die Art des Inhaltes und eine Referenz auf den Inhalt selber.

**Physikalische Eigenschaften** In physikbasierten Umgebungen, wie DynAmbient, werden die physikalischen Eigenschaften der Objekte benötigt. Masse, Reibung, Elastizität, Substanz und auch Form müssen beachtet werden. So kann in physikalischen Umgebungen die Größe auch Auswirkungen auf das Gewicht haben. Die Abhängigkeiten zwischen den Eigenschaften sind hier genau zu beachten und erfordern eine genaue Abstimmung, um bedienbare Modelle und Umgebungen zu schaffen.

**Verwaltung** Um eine Verwaltung von mehreren Objekten zu ermöglichen, müssen gewisse zusätzliche Informationen eingeführt werden. Ein Beispiel ist ein Wert für eine Ebene, um feststellen zu können, welches Objekt vor dem anderen liegt. Dies hat nicht nur Auswirkungen auf die Darstellung, sondern auch auf die Interaktion von Objekten. Überdecken sich zwei Objekte, soll die Interaktion beim Anfassen mit dem darüber liegenden passieren. Solche Verwaltungsaufgaben können auch von verwendeten Komponenten angeboten werden. Grafik- oder Physik-Engines bringen viele Informationen und Funktionen zur Verwaltung mit und können dementsprechend genutzt werden.

Die Persistenz dieser Daten wird hier nicht weiter betrachtet. Auf die Interaktion hat es keine Auswirkungen, wo die Daten liegen, solange der Zugriff schnell genug ist. Eine Abstraktion der Persistenz innerhalb der Applikation ist empfehlenswert, um verschiedenste Datenquellen wie Datenbanken oder das Dateisystem einfach nutzbar zu machen.

## Controller

Der Controller soll die die Multitouch Interaktion mit den Objekten ermöglichen und stellt somit einen äußerst wichtigen Teil der Realisierung dar. Heutige Betriebssysteme besitzen keine nativen Möglichkeiten, mehrere Punkte zur Eingabe auszuwerten (Multipoint Fähigkeit). Bestehende Systeme nutzen nur einen Punkt wie den Mauszeiger, um Eingaben zu ermöglichen. Neuste Bestrebungen zeigen zwar, dass daran gearbeitet wird ([Windows 7](#), [The Multi-Pointer X Server \(2008\)](#), und sämtliche [Apple Inc.](#) Bestrebungen), allerdings hat



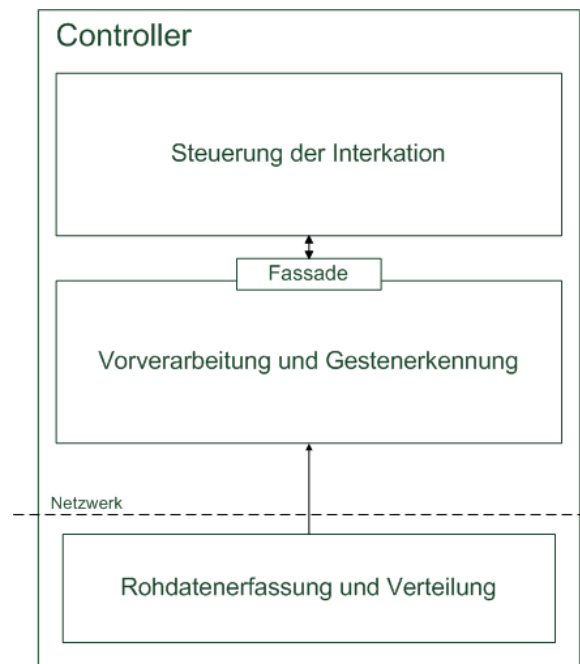


Abbildung 5.13: Aufbau der 3 Subsysteme

keines dieser Systeme ausreichende Stabilität erreicht, sind nicht über den Ankündigungs-Status hinaus oder haben weiterhin ein indirektes Verständnis<sup>15</sup> (MacBook Air Multitouch über das Trackpad ([MacBook Air](#))). Aufgrund dieser fehlenden Standards wird ein individuelles System entwickelt, welches das Lesen von Rohdaten sowie das Auswerten und das Interpretieren von Daten ermöglicht. Da hierbei viele einzelne Teilaspekte betrachtet werden müssen, wird der Controller hier ausführlicher und in seinen Einzelteilen betrachtet. Hierzu werden drei Subsysteme entwickelt, um Aufgabenbereiche voneinander zu trennen. (siehe Abb. 5.13).

- Rohdatenerfassung und Verteilung
- Vorverarbeitung und Gestenerkennung
- Programmlogik

**Aufbau der Subsysteme** Um die einzelnen Subsysteme voneinander unabhängig zu halten, wird zum einen das Entwurfsmuster der Fassade genutzt, um eine einheitliche Benutzung zu gewährleisten. Das Prinzip des Entwurfsmusters wird hier kurz vorgestellt und

<sup>15</sup>Mit indirektem Verständnis ist die nicht direkte Eingabe gemeint, der Benutzer manipuliert über ein Trackpad und nicht direkt auf der Bildschirmfläche.

anschließend die konkrete Verwendung erläutert. Im Fall der Rohdatenerfassung wird ein Datenserver realisiert, um die fehlende Unterstützung von Betriebssystemen bezüglich der Multipoint Fähigkeiten auszugleichen und das Verteilen der Rohdaten an mehrere Anwendungen zu ermöglichen.

**Subsystem der Rohdatenerfassung und Verteilung** Die fehlende Betriebssystemunterstützung zum Verstehen und Weiterleiten von mehreren Punkten macht ein Subsystem nötig, welches von der Hardware die Multitouch Daten liest und an verschiedene Anwendungen verteilt.

Hierbei sollte es sich um ein möglichst einfaches System handeln, welches weder Veränderungen an den Daten, abgesehen von Prüfungen auf Übertragungsebene, vornimmt oder interpretierende Aufgaben übernimmt. So lassen sich für dieses Subsystem zwei Aufgabenbereiche identifizieren. Zum einen die Implementation des Protokolls für die Hardware, dazu gehört eine mögliche Initialisierung der Hardware, sowie das Versenden der Daten an Applikationen, welche diese Daten benötigen.

**Lesen** Abhängig von der Hardware muss ein eigenes Protokoll implementiert werden. Nach Möglichkeit soll die Hardware im späteren Subsystemen keine bzw. eine sehr untergeordnete Rolle spielen, um eine Portierbarkeit auf andere Hardware oder spätere Hardware-Revisionen zu vereinfachen und Anpassungen so gering und einfach wie möglich zu gestalten. Vorstellbar ist auch, eine Auswahl von Hardware zu geben, welche genutzt werden kann.

Nach Möglichkeit werden Überprüfungen der Datenintegrität wie CRC Prüfungen in diesem Subsystem durchgeführt, um fehlerhafte Daten frühzeitig zu erkennen und ggf. zu verwerfen oder als fehlerhaft zu markieren. Auch fehlerhafte Datenpakete können Informationen enthalten, die von Nutzen sind. Werden dauerhaft fehlerhafte Daten erkannt, kann das System entsprechend reagieren und den Nutzer drauf hinweisen oder eventuell die Hardware neu initialisieren, sofern die Möglichkeit besteht und das zur Fehlerbehebung beitragen kann.

**Erfassung** Je nach Hardware müssen Daten von der Verteilung entsprechend aufbereitet werden. Unterschiedliche Informationen stehen zur Verfügung und erweitern das Protokoll um die Inhalte. So kann die Breite einer Unterbrechungen, Zeitstempel und andere optionale Informationen zur Verbesserung der Interaktion beitragen oder stellen Grundlagen für Gesten dar und müssen unbedingt an nachfolgende Systeme weitergegeben werden. Der mögliche Overhead, der entsteht, weil die Informationen nicht gebraucht oder verworfen werden, wirkt sich bei den heutigen, vorhandenen Rechen-

leistungen nicht negativ aus. Ein erster Entwurf, wie ein solches Protokoll aussehen könnte, entstand im Rahmen der in der Evaluation genutzten Hardware.

**Verteilung** Nach Lesen und Konvertierung in ein allgemeingültiges Protokoll wird ein Mechanismus benötigt, um beliebig vielen Anwendungen den Zugriff auf diese Daten zu gewähren. Um dies zu ermöglichen, sind zwei Arten der Kommunikation vorgesehen. Eine der Kommunikationsleitungen läuft über das localhost Interface per UDP Stream. Dieses muss für jede Anwendung eine Verbindung aufbauen und erfordert eine Anmeldung vom Client. Sowie eine Multicast Implementation, welche neben der Anwendung auf dem lokalen Gerät auch eine Grundlage für die Einbindung im CCW sein kann.

Durch dieses Subsystem soll, bis entsprechende Unterstützungen von Betriebssystemen gegeben werden, eine Basis zur Verfügung gestellt werden, um die Multitouch-Daten der Hardware auf verschiedene Anwendungen zu verteilen und somit die Grundlage für die direkte Interaktion zu schaffen.

**Subsystem der Vorverarbeitung und Gestenerkennung** Ziel dieses Subsystems in der Architektur ist es, eine allgemein gültige Logik zur Verfügung zu stellen, um Multitouch Daten aufzubereiten und anzubieten. Dieses Subsystem muss verschiedene Aufgaben übernehmen:

- Filter bereitstellen, die zur Beseitigung von Störungen dienen können
- zugriff auf einen unveränderten oder auch gefilterten Datenstrom an nachfolgende Systeme bieten
- die Möglichkeit eine Erkennung von Gesten vorzunehmen
- aufbau einer Historie und der Erstellung eines erweiterbaren Repertoires an Gesten

Wie aus der Analyse und dem Kapitel über Gesten hervorgeht, gibt es in dieser Hinsicht einige Problemstellungen, die eine Konzeption eines solchem Systems erschweren. Zum Einen die Ungewissheit über den Anfang und das Ende von Gesten bzw. die Übergänge untereinander. Zum Anderen könnte eine dauerhafte Auswertung von Gesten einen erheblichen Rechenaufwand erfordern. Um das Erkennen von Gesten über dieses Subsystem zu ermöglichen, soll es möglich sein, Gesten über einen gewissen Zeitraum aus der Historie und anhand von übergebener Daten zu erkennen. Auf die einzelnen Teilaspekte wird nachfolgend gesondert eingegangen. Als Zugriff auf dieses Subsystem wird, um die Arbeit mit dem System zu erleichtern, das Entwurfsmuster der Fassade verwendet, welches im folgenden Exkurs genauer erläutert wird.

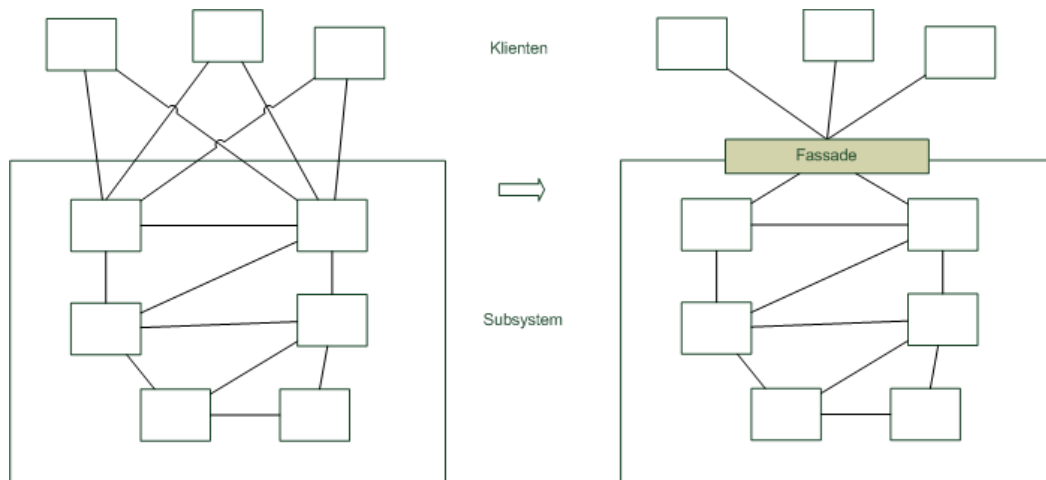


Abbildung 5.14: Aufbau einer Fassade

**Fassade Entwurfsmuster** Um Komplexitäten von Systemen zu minimieren, hilft eine Aufteilung in Subsysteme. Dabei ist es das Ziel, die Abhängigkeiten der Subsysteme untereinander möglichst gering zu halten. Durch das Erstellen einer Fassade wird genau dieses erreicht. Das Strukturmuster stellt eine vereinfachte Schnittstelle zu einem Subsystem zur Verfügung, über die die Klienten mit dem Subsystem kommunizieren können (s. Abb. 5.14).

Die Fassade besteht aus einem Objekt, welches weiß, welche angebotenen Funktionalitäten der Schnittstelle an welchen Teil des Subsystems gerichtet werden. Es delegiert die Anfragen an die entsprechenden Teile des Subsystems. Die Klassen, welche die eigentliche Funktionalität ausführen, wissen nichts von der Fassade, sondern führen nur die Aufgaben aus.

Die Vorteile, die der Einsatz dieses Entwurfsmusters bringt, sind

- die Abschirmung der Klienten von den einzelnen Subsystem-Komponenten
- die Förderung der losen Kopplung der einzelnen Subsysteme, wodurch ein hoher Grad an Modularität erreicht und das Austauschen von Subsystemen erheblich erleichtert wird, ohne dabei das Gesamtsystem zu destabilisieren. Ebenfalls unterstützt es die Aufteilung in verschiedene Schichten.
- die weiterhin bestehende Möglichkeit, direkt auf Teile des Subsystems zuzugreifen. Es wird zwar eine allgemeine Schnittstelle zur Verfügung gestellt, das Entwurfsmuster verhindert aber nicht den Zugriff auf Teile des Subsystems. Somit können andere Subsysteme, sofern es nötig ist, direkt auf Funktionalitäten zugreifen.

Durch diese abstrakten Schnittstellen wird die Benutzung der Subsysteme vereinfacht (Gamma u. a. (1995)).

**Aufbau der Fassade** Die Fassade soll folgende Funktionalitäten anbieten:

- Hinzufügen und Entfernen von Filtern
- Zugriff auf einen kontinuierlichen Datenstrom bieten
- Zugriff auf Teile oder die komplette Historie bieten
- Gestenerkennung über die Daten eines bestimmten Zeitraumes der Historie
- Gestenerkennung über eine mitgelieferte Datenmenge

**Filter** Eine Vorverarbeitung von Daten kann in vielen Fällen Sinn machen. Zum Beispiel können zitternde Punkte über verschiedene Algorithmen geglättet werden und starke Ausreißer können so abgefangen werden. Zu beachten ist, dass in dieser Phase möglichst nur minimale und eindeutige Fehler korrigiert werden sollten. Größere Veränderung von Daten kann die Semantik der Bewegung beeinträchtigen und somit die Interaktion erschweren.

Ein Filter um mögliche Ausreißer zu entdecken und zu eliminieren, ist zum Beispiel die Median-Glättung. Jeder Wert wird durch den Median seiner Umgebung bestimmt. So werden X- und Y-Werte jeweils per Median über n Werte ermittelt und als neues Wertepaar eingesetzt. Bei einem Median über drei Werte, werden diese Werte nach Größe sortiert und der middle Wert, wird als neuer Wert eingesetzt. Bei den Werten 5,4,2 würde nach Sortierung 2,4,5 als Auswahl dienen, wobei der Median in diesem Fall 4 ist. Median Glättung wird hier beispielhaft genannt, es könnten auch andere Methoden verwendet werden.

Bei einem fiktiven Beispiel wird gezeigt, wie durch den Median eine Glättung und speziell starke Ausreißer gefiltert werden können.

In Abbildung 5.15 kann man drei verschiedene Verläufe erkennen. In diesem Beispiel wird davon ausgegangen, dass ein Fehler der Y-Werte aufgetreten ist. So zeigt der blaue Verlauf den entstandenen Fehler, der gelbe die reale Eingabe und der orangefarbige Verlauf die Ergebnisse nach der der Median-Glättung. Vergleicht man die Verläufe, so kann man feststellen, dass nach der Median Glättung zwar nach wie vor ein gewisser Fehler bleibt, dieser aber erheblich verringert wurde.

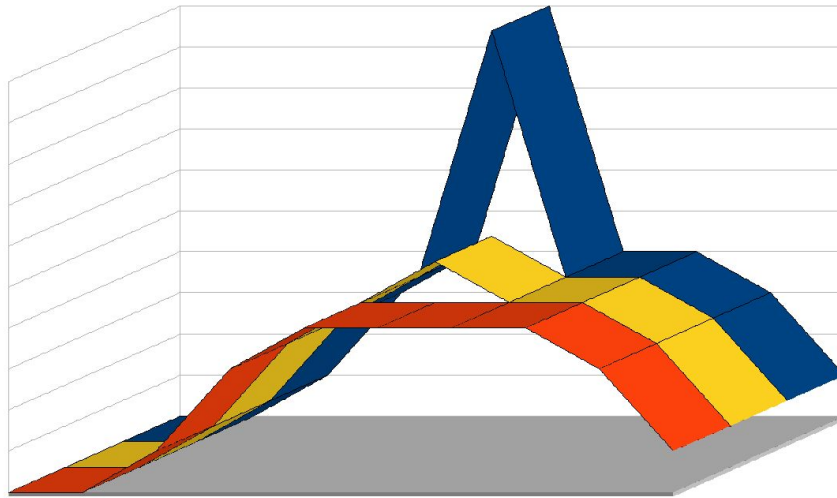


Abbildung 5.15: Beispielhafte Median Glättung einer Eingabe

Ein ähnlicher Ansatz wäre die Mittlung<sup>16</sup> über verschiedene Werte. Die Gefahr von Datenverlust in Bezug auf die Genauigkeit der Daten ist bei kleiner Auflösung<sup>17</sup> hoch. Wird der Grad der Glättung zu groß, können auch gewollte Bewegungen als Fehler betrachtet und somit unterdrückt werden.

Die Möglichkeiten Filter einzusetzen, sind vielfältig und abhängig von der eingesetzten Hardware und deren Eigenschaften.

**Kontinuierlicher Datenstrom** Aufgrund der Kombination von Gesten und kontinuierlichen Änderungen muss ein permanenter Zugriff auf aktuelle Touch-Daten ermöglicht werden. Nach einer möglichen Vorverarbeitung durch Filter sollten diese Daten möglichst in Echtzeit zur Verfügung gestellt werden, da jede Verzögerung einen massiven Einfluss auf die Bedienung von Multitouch Systemen hat. Entsteht eine zu große Verzögerung, nimmt die Intuitivität rapide ab und das System wird kaum bedienbar.

Hier ist ebenfalls ein Protokoll von Nöten, welches neben den X-Y Koordinaten, Timestamps und anderen erforderlichen Merkmalen auch eventuelle Änderungen durch einen Filter beinhalten könnte. Dieses Protokoll muss vollkommen unabhängig von der Hardware sein.

**Historie** Neben Weitergabe der kontinuierlichen Änderungen muss eine Historie der gesammelten Daten erstellt werden. Diese wird einerseits benötigt, um Filtern die Möglichkeit

<sup>16</sup>Bildung des Mittelwertes über n Werte

<sup>17</sup>Der Wertebereich, welchen die Hardware für ihre Touch-aktive Fläche benutzt.

zu geben eventuelle Störungen zu erkennen, aber auch für die nachfolgende Gestenerkennung ist eine Speicherung unabdingbar. Im einfachsten Fall besteht die Historie aus einer einfachen Listenstruktur, welche auf eine bestimmte Größe beschränkt ist. Diese Struktur kann ein Array sein oder auch eine komplexere Struktur, wie z.B. eine Queue. Je nach Anwendung ließe sich eine Historie beliebig erweitern. Sollte man für Usability Tests, z.B. eine dauerhafte Speicherung wünschen, wäre eine Speicherung in eine Datenbank oder auf einem Dateisystem denkbar. Da die Möglichkeiten vielseitig sind, sollen die weiteren Überlegungen über die Persistenz dieser Daten nicht Gegenstand der Arbeit sein. Die Gesten betreffend, ist allerdings eine Bibliothek zu schaffen, welche Persistent vorhanden ist, damit der Benutzer und das System auf einen Repertoire von Gesten zugreifen können.

**Gesten Bibliothek** Die Anzahl der möglichen Gesten ist, wie in 4 geschildert, aufgrund der vielen unterschiedlichen Bewegungsmöglichkeiten von Fingern und Händen sehr groß. Dem Benutzer einer solchen Gesten Bibliothek muss also ein Werkzeug an die Hand gegeben werden, mit dem diese Fülle an Gesten organisiert werden können. Hierzu zählt auch das Entfernen von nicht mehr benötigten und das Hinzufügen von neuen Gesten.

Ab einer zu großen Anzahl von Gesten wird es für den Benutzer auch schwieriger, sich die Gesten zu den Aktionen zu merken, daher wäre ein visuelles Tool zur Betrachtung der gespeicherten Gesten angebracht. In einem eingereichten Patent zeigt Apple, wie so ein Tool aussehen könnte (siehe Abb. 5.16)

Ein Mensch, der versucht, sich an Worte und deren Bedeutung einer fremden Sprache zu erinnern, nimmt ein Wörterbuch zur Hand. Passend hierzu könnte eine Gestenbibliothek dazu dienen, den Benutzer beim Memorieren der Gesten zu unterstützen. Sie hält hierzu ein Repertoire an Gesten bereit, welche mit vordefinierten Gesten versehen ist ([Apple Inc. Multi-touch Gesture Dictionary Patent Application \(2007\)](#)).

Auch das System benötigt Anhaltspunkte, welche Gesten zu welchen Aktionen gehören. Diese Informationen sind in der Gestenbibliothek gespeichert und können jederzeit abgerufen werden. Im Design sind modellartige Beschreibungen zu Gesten vorgesehen, welche im Betrieb nachgeladen werden können. Das Format dieser Gestenbeschreibungen wird im Rahmen dieser Arbeit nicht mehr festgelegt, allerdings wurden einige erste Gestenmodelle zu Testzwecken im XML-Format entwickelt.

Diese Bibliothek ist als dauerhafter Hintergrunddienst konzipiert, bei dem es vorgesehen ist, dass Eingabedaten zwecks Gestenerkennung an diesen bei Bedarf weitergeleitet werden können. So ist es möglich, systemweit einen konsistenten Status über die Gesten zu halten.

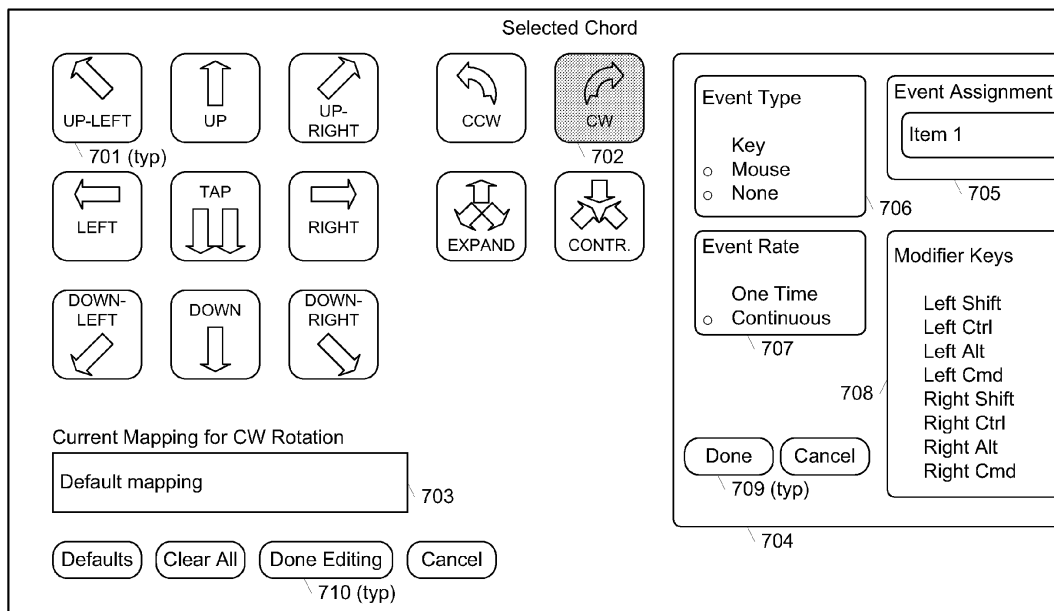


Abbildung 5.16: Aussehen eines visuellen Tools für eine Gesten Bibliothek aus [Apple Inc. Multi-touch Gesture Dictionary Patent Application \(2007\)](#)

**Gestenerkennung** Die Gesten können, wie oben beschrieben, je nach Bedarf abgerufen werden. Hierzu ist es nötig, eine gewisse Form von Gestenerkennung zu implementieren, da klar erkannt werden muss, um welche Geste es sich handelt. Wenn eine Geste unbekannt ist, muss es die Möglichkeit gegeben, neue Gesten hinzuzufügen.

Die Erkennung der Gesten anhand der Daten aus dem kontinuierlichen Datenstrom erfolgt über einen als Hintergrunddienst implementierten Gestenerkennung. Der Grund hierfür sind die zeitintensiven Berechnungen, welche bei der Erkennung von Gesten durchgeführt werden müssen. Daher wird von einer fortlaufenden Gestenerkennung abgesehen und eine „On Demand“ Gestenerkennung eingeführt<sup>18</sup>. Als Auslöser hierfür dient die Übergabe von Eingabedaten, aus dem kontinuierlichen Datenstrom, von der Applikation zum Hintergrunddienst. Dieser kann dann, unter Zuhilfenahme der Gestenbibliothek ermitteln, welche Geste zu einer bestimmten Zeit ausgeführt wurde. Die Information über die ausgeführte Geste wird dann wieder an die betreffende Applikation gesandt, welche dann weitere, für sie zutreffende Maßnahmen ergreifen kann.

Als Datenformat eignet sich hierzu in der Prototyp-Phase eine XML-Struktur, da sie leicht

<sup>18</sup>Die Entwicklung einer Gestenerkennung auf C++ Basis ist aufgrund der besseren Performance zu empfehlen, wird aber in dieser Arbeit aus Zeitmangel nicht mehr implementiert.



vom Menschen lesbar ist<sup>19</sup>. Dieses Übergabeformat sollte einheitlich und unabhängig von der benutzten Hardware sein, um schnell die Integration anderer Hardware zu ermöglichen. So müssen also je nach verwendeter Hardware bestimmte Interfaces implementiert werden.

Die eigentliche Gestenerkennung kann mit Hilfe von Hidden Markov Modellen, Supportet Vektor Machines und Space-Time Linearisierung umgesetzt werden.

Die Gestenerkennung darf keine semantische Bewertung der erkannten Gesten vornehmen, da die Bedeutung der Gesten je nach Anwendung stark variieren kann. Somit bleibt die Interpretation der Geste allein der Programmlogik der entsprechenden Multitouch Applikation überlassen.

**Programmlogik** Da die Gestenerkennung keine semantische Interpretation einer erkannten Geste vornehmen soll, muss diesem Teil des Designs geschehen.

Dieser überwacht ob ein Objekt angefasst wurde und überprüft bei längerem Benutzerkontakt mit dem Objekt, ob eine bestimmte Geste ausgeführt wurde. Dazu werden die bereitgestellten Funktionen der Fassade genutzt. Bei einer erkannten Geste werden die nötigen Veränderungen im Model vorgenommen (s. Abb. 5.17).

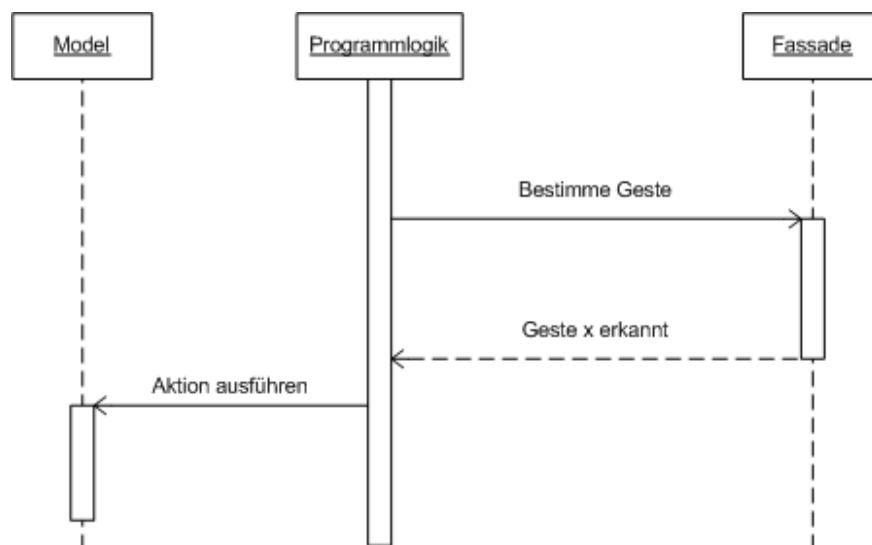


Abbildung 5.17: Ablauf der Gestenerkennung

Je nach Anwendung muss die Programmlogik angepasst werden, um das Zusammenspiel der einzelnen Komponenten und Subsysteme zu ermöglichen. Dem Anwendungsprogram-

<sup>19</sup>Die angesprochenen Performanceprobleme werden durch den Einsatz von XML nicht verbessert, allerdings ist es beim Debugging in der Prototyp-Phase ein angemessenes Mittel.

mierer wird überlassen wie sich Aktionen auf das System auswirken oder z.B. Multiuser-Fähigkeiten definiert werden. Dazu gehört auch die beschriebenen (s. 3.3.1) Herausforderungen zum Bewältigen der unterschiedlichen Ausrichtung bei Arbeitsräumen.

## 5.4 Technische Umsetzung

In diesem Abschnitt wird beschrieben, wie das Design unter Berücksichtigung der gesammelten Anforderungen aus 3.3 konkret umgesetzt wurde. Hierzu wird dargestellt, wie die aktuellen Versionen aussehen, welche Komponenten zum Einsatz gekommen sind und wie diese implementiert wurden.

Ebenso wird diskutiert, welche Grafik- und Physik-Engines zur Erstellung der Applikationen letztendlich verwendet wurden, und wieso die Entscheidung auf diese fiel.

### 5.4.1 Oberflächengestaltung

Die Oberflächengestaltung der Applikationen erzeugen, wie in 5.3.2 beschrieben, eine Erwartungshaltung beim Benutzer. Um eine möglichst große Überschneidung vom Mental Model des Benutzers und der Funktionalität der Applikationen zu schaffen, wurde viel Entwicklungsaufwand bei dem Entwurf der Oberflächen betrieben. Wie die Oberflächen der einzelnen Applikationen aussehen, soll hier geschildert werden.

#### Oberflächengestaltung der Multitouch-Wall

Die Verwendung von Processing als Grafik-Engine für die Applikation der Wand stellte sich relativ schnell als positiv heraus, da sich diese Engine gut zum schnellen Entwickeln von Prototypen eignet und eine iterative Programmierung unterstützt.

Schon nach ersten Prototypen wurden Evaluationen mit Benutzern durchgeführt, welche zu Veränderungen im Design der Oberfläche und zusätzlichen Funktionalitäten führten.

Auf diese Weise konnten alle Anforderungen aus der Analyse erreicht werden.

Die Oberfläche der Wand-Applikation ist ein virtuelles schwarzes Brett auf dem Bilder, Videos oder 3D-Modelle aufgehängt werden können. Diese können nun mittels der beschriebenen Gesten aus 4 manipuliert werden. Unter anderem können die Objekte verschoben und vergrößert werden, wobei sie sich, wie bei einem schwarzen Brett auch, gegenseitig überdecken können. Auch eine Rotation der Objekte ist möglich, sowie ein Flip (s. Abb. 5.18)



Abbildung 5.18: Durchführung einer Flip-Geste

von einem oder mehreren Objekten zugleich.

Es wurde des Weiteren noch ein Menü implementiert welches, wie bei von Betriebssystemen bekannten Menüs, weitere Optionen anzeigt, die je nach Medientyp variieren.

Wird das Menü auf einer leeren Fläche aufgerufen, so kann über dieses Menü ein Objekt-Lader eingeblendet werden, über den weitere Medien auf die Oberfläche geladen werden können. Zusätzlich kann über dieses Menü die Bildwiederholrate angezeigt werden, um die Performance zu überprüfen.

### **Oberflächengestaltung des Touch-Table**

Nach langer Evaluation der möglichen Grafik- und Physik-Engines wurden mehrere Prototypen der Tisch-Applikation entwickelt. Wie in Abb. 5.8 zu sehen, waren die ersten Versionen noch sehr rudimentär. Das Design der Oberfläche wurde mittels einfacher Boxen realisiert, welche mit verschiedenen Texturen versehen wurden. Alle Boxen sind aus einer grafischen und einer physikalische Entität zusammengesetzt, wobei bei den ersten Prototypen keine dedizierten Objekte vorhanden waren, sondern eine Entität beide Eigenschaften vereinte. Dies erschwerte die Wartbarkeit und Erweiterbarkeit massiv, da einzelne Veränderungen der grafischen oder physikalischen Eigenschaften eine Neuentwicklung vieler Teile der Applikation mit sich brachten, da einfache Justierungen an einzelnen Eigenschaften nicht möglich waren.

Die Tischfläche ist eine statische Box, also eine Box die ihre Position im Raum beibehält, dennoch reagieren andere Objekte mit physikalischen Eigenschaften auf diese Box. Die anderen kleineren Boxen sind dynamische Objekte, welche ihre Position im Raum ändern können. Sie reagieren hierbei auf statische und dynamische Elemente ihrer Umwelt. Aus diesem Grund ist es möglich, die Objekte zu verschieben und auf der Tischfläche umherzuschmeißen, wie es von einer realen Tischplatte mit aufliegenden Karten zu erwarten wäre. Fliegen die Objekte aus dem sichtbaren Bereich des Bildschirms, fallen sie von der Tischkante und werden zu Testzwecken wieder in die Mitte des Tisches versetzt.

Dieser Ansatz wurde fallen gelassen, da die Applikation insgesamt nicht flüssig genug dargestellt wurde. Die verwendete Grafik-Engine Stattdessen wurde versucht, eine ähnliche auf einer performanteren C++ Basis aufzubauen, was allerdings aufgrund von mangelnder Dokumentation des verwendeten Frameworks fehlschlug.

Nach Abschluss von Philipps Roßbergers Masterarbeit, wurde seine Applikation als Basis für den Tisch übernommen (s. Abb. 5.19).

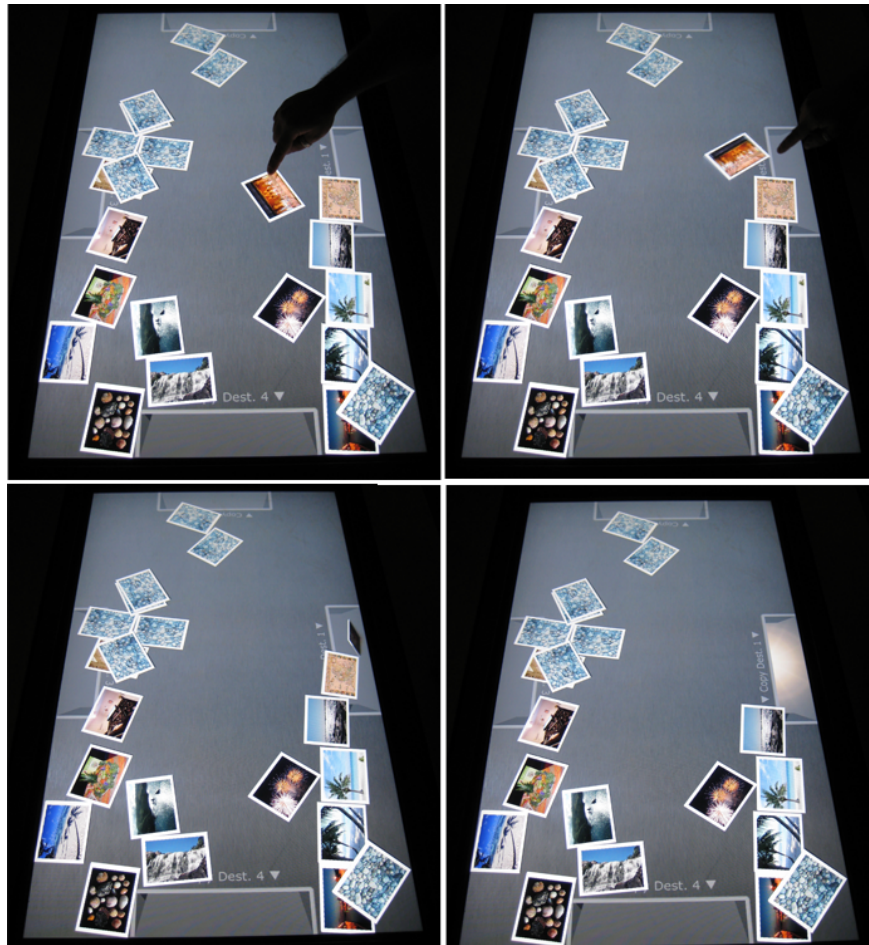


Abbildung 5.19: Interaktion mit DynAmbient

## 5.4.2 Verwendete und entwickelte Komponenten

Die Komponenten zur Realisierung der Applikationen sollen hier kurz dargelegt werden. Zum Einen gilt es hier, die hinzugezogenen fertigen Komponenten aufzulisten und zum Anderen,

die zusätzlich entwickelten Komponenten zu beschreiben.

Es hat sich während der Entwicklung eine Basiskomponente herausgestellt, die später genauer beschrieben wird.

### **Komponenten der Multitouch-Wall**

Die Darstellung, also der View, wird bei der Multitouch-Wall, wie oben geschildert, von Processing übernommen.

Das Model und der Controller sind, wie in 5.3.2 beschrieben, umgesetzt worden. Es gibt also ein Interface und eine implementierende abstrakte Klasse für die Objekte, welche von der View dargestellt werden. Sie stellen das Model dar und beinhalten zusätzlich zu der in Abb. 5.12 gezeigten move Methode, zum Bewegen von Objekten, auch eine Zoom-Methode, welche das Vergrößern realisiert.

Das Model enthält also Zustandsmerkmale über Position, Ausrichtung, Form und Inhalt der Objekte. Aufgabe der Model Komponente ist es auch, diese Objekte zu verwalten. Hierzu werden alle aktuell auf dem Schirm sichtbaren Objekte im Speicher vorgehalten und neu erstellte mit aufgenommen. Als Datenstruktur dient hierfür eine Liste, welche dynamisch wachsen und schrumpfen kann.

Die Controller-Komponente wurde zum Teil in der Applikation realisiert. Hier wird die Verbindung zwischen Gesten und Aktion geschaffen, indem die Daten vom SerialServer (5.4.2) entgegengenommen und passende Aktionen auf dem Model ausgeführt werden.

### **Komponenten des Touch-Table**

Da sich die Prototypen der Tisch Applikation, welche auf Basis von JmonkeyEngine als Grafik-Engine und ODE als Physik-Engine verwendeten, nicht durchsetzen konnten, wurde DynAmbient als Basis verwendet.

Durch Modellierung physikalischer Eigenschaften, wie Masse und Oberflächenbeschaffenheit, verhalten sich digitale Objekte in DynAmbient wie es Anwender in der Realität erwarten würden. Dieses neuartige Bedienkonzept Physik basierter Interaktion erlaubt die Nutzung vertrauter Arbeitstechniken und verbessert somit die intuitive Bedienbarkeit von Programmen in kollaborativen computergestützten Umgebungen.

So wurde eine Multitouch Funktionalität von Roßberger angedacht, aber nicht umgesetzt. Mit den Bestrebungen, die in dieser Arbeit durchgeführt wurden bietet sich auch eine Fortsetzung an.

Da aufgrund der Physik-Simulation eine viel nähere Anlehnung an Mentale Modelle von Nöten ist, wurde die Auswahl von Gesten angepasst. Das Bewegen von mehreren Objekten ist zwar möglich, wurde aber aufgrund von den beschriebenen Problemen mit der Hardware und der fehlenden Multipoint Fähigkeiten in Betriebssystemen noch nicht umgesetzt. Stattdessen wurden spezielle Gesten an Maus Events gebunden was so die Interaktion ermöglicht.

Durch die physikalische Basis, welche bei DynAmbient gegeben ist, ergeben sich viele der Gesten durch Anfassen und Bewegen des Datenobjektes durch einen Finger. So wird die Rotation über das Anfassen an eine der Ecken und entsprechendes Bewegen des Objektes realisiert. Bei einfachem Anfassen der Objekte mit einem Finger, werden diese über eine Gummiband-Metapher bewegt. Es ist aber auch möglich, ein Objekt direkt, ohne den Gummiband, anzufassen diese Funktionalität wurde auf ein Anfassen mit 2 Fingern abgebildet.

DynAmbient verwendet als Grafik-Engine Irrlicht und als Physik-Engine NVIDIA PhysX. Durch die Aufteilung der grafischen und physikalischen Entitäten werden Erweiterungen stark vereinfacht.

Die View Komponente besteht hier also aus Irrlicht, während die Model Komponente von PhysX bereitgestellt wird. Die Controller Komponente von Roßberger wurde um Multitouch Funktionalität erweitert, indem diese als weiteres Eingabegerät hinzugefügt wurde. Als Hardware-Abstraktionsschicht dient auch hier der SerialServer.

### **Basiskomponente SerialServer**

Im SerialServer sind die Funktionen der Rohdatenerfassung und Vorverarbeitung implementiert. Eine Vorbereitung für Gestenerkennung wurde eingefügt, so dass diese per Modul nachgerüstet werden kann.

Umgesetzt ist der SerialServer als Hintergrunddienst, welcher die Daten der Hardware entgegennimmt und je nach Wunsch des Benutzer unterschiedlich weitergibt. Die Daten werden vom Rahmen Byte-weise verschickt, vom SerialServer empfangen, in eine Listenstruktur übertragen und zum Versenden über die jeweilige Schnittstelle vorbereitet.

Erste Versionen wurden mit Java realisiert, welche sich allerdings als sehr instabil herausstellten, da die Anbindung von Hardware in Java von Sun nur sehr träge und fehleranfällig implementiert wurde. Daher wurde die finale Version mit Hilfe von C# und der passenden .NET Bibliotheken entwickelt. Das Resultat ist ein sehr stabiles und performantes Programm, welches sich häufig als erweiterbar und wartungsarm erwiesen hat.

Auch hier wurde eine iterative Programmierung verfolgt, sodass verschiedene Funktionalitätä-

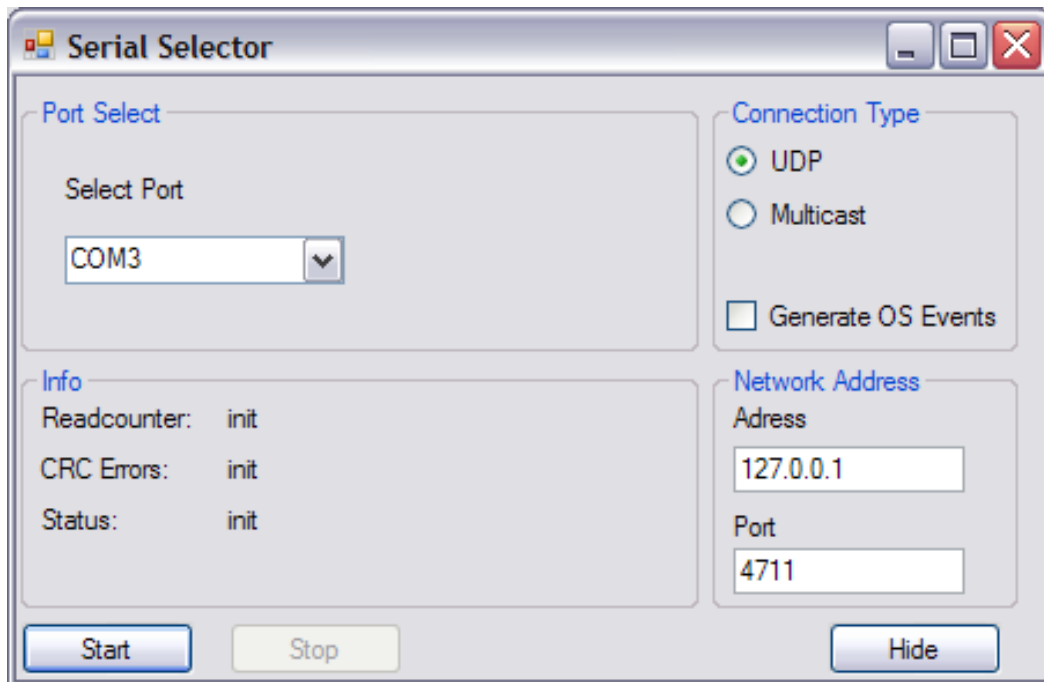


Abbildung 5.20: Die Oberfläche des SerialServers

ten mit der Zeit hinzugefügt werden konnten.

In Abb. 5.20 ist die grafische Oberfläche des SerialServer zu sehen. Diese wurde als Serial Selector bezeichnet, da sich hier unter dem Punkt „Port Select“ auswählen lässt, auf welcher seriellen Schnittstelle der SerialServer auf Hardwareeingaben hören soll.

Nach der Auswahl des Ports ist unter „Connection Type“ noch auszuwählen, welcher Verbindungstyp gewünscht wird. Es stehen UDP- und Multicast-Verbindungen zur Verfügung, über die dann die Daten an die Netzwerkschnittstelle weitergeleitet werden. Die Netzwerkadresse, an die weitergeleitet werden soll, kann unter „Network Address“ angegeben werden. Per Default werden Standardadressen verwendet. Zudem kann noch gewählt werden, ob lokale Betriebssystem Events erzeugt werden sollen, die sich wie die Eingaben von Maus und Tastatur, von beliebigen Programmen abrufen lassen. Unter anderem ist so eine Steuerung des Mauszeigers über Touch Eingaben implementiert worden.

Unter „Info“ können Informationen zum Status des Programmes angesehen werden, also z.B. ob es gestartet ist oder eventuelle Fehlermeldungen vorhanden sind. Auch CRC Errors werden hier angezeigt, sowie die Anzahl der Pakete pro Sekunde unter „Readcounter“.

### 5.4.3 Kompensation von fehlenden Informationen der Hardware

Aufgrund der schon genannten Schwierigkeiten mit der Hardware (s. 5.1.1) mussten einige Funktionen angepasst werden, um Multitouch Funktionalitäten zu ermöglichen und vor allem Intuitivität zu gewährleisten.

Als Konsequenz aus der Tatsache, dass bei zunehmender Menge an Punkten eine Aussage über die Bedeutung der Daten nahe zu unmöglich wurde, wurden teilweise sehr einfache und abstrakte Annahmen getroffen.

Funktionalitäten wurden an die Anzahl der gefunden Unterbrechungen gebunden. Durch diese Maßnahme musste auf die Mehrbenutzer Fähigkeit mit dieser Hardware verzichtet werden. Eine in Aussicht stehende verbesserte Hardware soll dieses Problem allerdings lösen. Ebenfalls mussten bei einigen Sonderfällen gewisse Annahmen getroffen werden, welche die Vielfalt an Möglichkeiten weiter einschränkte.

**1 Finger** Eine Unterbrechung wurde mit der Bewegung von Objekten verknüpft. Da es nur einen Punkt gibt, kann dieser genau verfolgt und die Bewegung eins zu eins umgesetzt werden. Ebenfalls wurden Eingaben über das Menü mit einem Finger realisiert.

**2 Finger** Es wurden zum einen das Vergrößern und Verkleinern, wie auch die Rotation realisiert. Aufgrund der schon beschriebenen Probleme mussten hier einige Einschränkungen vorgenommen werden, um eine Umsetzung zu ermöglichen. Es wurde klar abgegrenzt zwischen Rotation und Zoom. Eine Kombination aus beiden war anfangs möglich, führte aber dazu, dass keine der Funktion akzeptabel bedienbar waren. Bei Überdeckung von 2 Fingern, wenn sie auf einer geraden Achse liegen, reagiert das System nicht. Eine Ersetzung der Punkte mit vorheriger Verfolgung wäre umsetzbar, aber durch den Verlust von Werten, wie in den Hardware Eigenschaften beschrieben, kann es zu falschen Ersetzungen und somit zu vollkommen falschen und sprunghaften Ergebnissen kommen.

**3 Finger** Bei drei Fingern stellt das Fehlen an Zuordnungsinformationen von X- und Y-Werten erhebliche Probleme dar. Bei drei X- und drei Y-Werten entstehen neun mögliche Wertepaare, welche nur schwer auf die realen drei reduziert werden können. Eine Interpretation von 3 Werten wurde auf ein Minimum beschränkt. Die Geste muss dahingehend zwei Bedingungen erfüllen: Zum einen müssen 3 Finger auf einer Achse und zum anderen nur ein Finger auf der anderen erkannt werden.

**4 und 5 Finger** Bei vier und fünf Fingern wird das Menü geöffnet. Da es bei dieser Menge oftmals zu Überschattung kommt und somit oftmals 5 Finger nicht eindeutig erkannt werden konnten, wurde das Menü ebenfalls bei 4 Fingern geöffnet was die Bedienung erheblich verbesserte. Das Objekt, welches ausgewählt werden soll, wird über den Mittelpunkt der Bounding Box ermittelt.



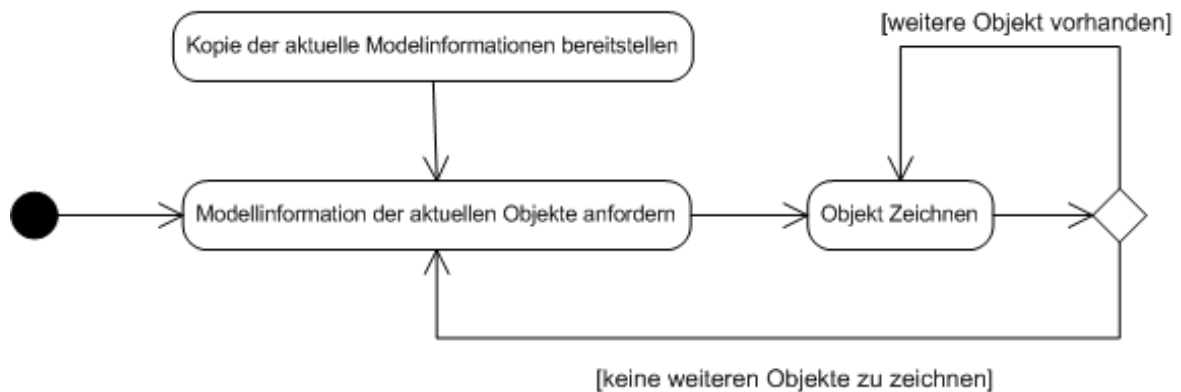


Abbildung 5.21: Ablauf der Objektdarstellung

#### 5.4.4 Kommunikation innerhalb der Anwendung

Um die verschiedenen einzelnen Komponenten zu verbinden und einen Informationsfluss zu schaffen, wurde eine interne Kommunikation benötigt. Ebenfalls musste dabei beachtet werden, dass Informationen trotz des Threadings aktuell und korrekt bleiben. Es wurden für die einzelnen Systeme spezialisierte Kommunikations- und Datenflusswege aufgestellt, um einen reibungslosen Ablauf zu ermöglichen. Die Unabhängigkeit zwischen der grafischen Darstellung, dem Lesen der Rohdaten und dem Verarbeiten von diesen ist zu gewährleisten. Hierzu wird zum Beispiel im View mit einer Kopie der aktuellen Model Daten gearbeitet (Abbildung 5.21). Somit musste keine Synchronisation der beiden Systeme vorgenommen werden, um Darstellungsfehler zu vermeiden. Eine Synchronisation wäre problematisch gewesen, da Daten unregelmäßig und mit einer nicht ausreichenden Datenrate ankommen, um eine flüssige Darstellung zu gewährleisten. Die Erstellung der Kopie ist hier der einzige Vorgang, welcher vor mehrfachem Zugriff unterschiedlicher Threads geschützt werden musste. Hier konnte auf die in Java 5 hinzugekommenen Funktionalitäten des Concurrent Packages zurückgegriffen werden.

Ein weiterer wichtiger Punkt in der Kommunikation im System ist der kontinuierliche Datenstrom. Da die Datenauswertung und die Kontrolle der Interaktion jeweils in ihrem eigenen Thread laufen, aber direkt voneinander abhängig sind, musste eine Synchronisation der Threads erfolgen. Die Wahl fiel hier auf BlockingQueue. Einer der Hauptgründe hierbei ist, dass keine Daten verloren gehen. Ist die Datenrate der Hardware hoch genug, dass mehr Datenpakete eingeht als die Kontrolle der Interaktion verarbeiten kann, gehen die Daten nicht verloren und die Kontrolle kann selbst Daten anhand des Zeitstempels verwerfen. Werden nicht genug Daten geliefert, kann der Thread schlafen um Ressourcen zu schonen, um dann wieder angesprochen zu werden, wenn neue Daten vorhanden sind. Eine vereinfachte Darstellung des Systems ist in der Abbildung 5.22 dargestellt.

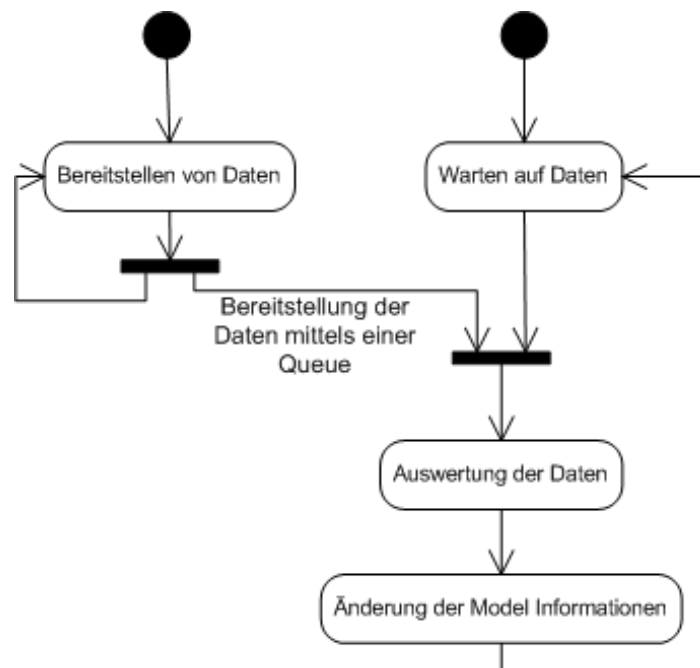


Abbildung 5.22: Synchronisation über den Datenstrom

Da die Hardware keine Daten mehr sendet, sobald keine Unterbrechung mehr festgestellt wird, erkennt das System diesen Zustand und nicht gebrauchte Systeme werden schlafen gelegt, so dass Ressourcen für andere Programme und Anwendungen wieder freigegeben werden.

## 5.5 Evaluation

Um die aufgestellten Interaktionsmöglichkeiten auf ihre Funktionalität und Intuitivität zu überprüfen, konnten in verschiedenen Phasen der Entwicklung immer wieder Testläufe mit verschiedenen Personengruppen durchgeführt werden.

Folgende Gesichtspunkte wurden immer wieder hinterfragt und überprüft:

**Grundlegende Funktionen:** Wie schnell lassen sich grundlegende Funktionen herausfinden?

**Erlernbarkeit der Funktionen:** Wie lange dauert es die Funktionen im angemessenen Maß zu beherrschen?

**Mitgebrachtes Wissen über Gesten und Interaktion:** Welches sind die ersten Versuche und Gesten, die der Nutzer ausprobiert?

**Ergonomie der Gesten:** Welche Gesten lassen sich gut ausführen? Treten Probleme beim Ausführen bestimmter Gesten auf?

### 5.5.1 Durchführung der Evaluation

Es wurden Vorfürhungen für Besuchergruppen im Ambient Labor der HAW Hamburg durchgeführt und in diesem Rahmen konnte beobachtet werden, wie verschiedene Personen mit dem System interagiert haben.

Ein weiterer intensiver Test konnte auf der Kiosk Europe Expo 2008<sup>20</sup> durchgeführt werden. Im Rahmen einer Kooperation mit [Elektrosil \(2008\)](#) wurde die Applikation dort vorgestellt. So wurde der Kreis der Personengruppe noch einmal erheblich vergrößert. Ebenfalls wurden Hard- und Software unter erschwerten Bedingungen, wie Messebeleuchtung und Dauereinsatz, getestet.

Durch die Vielseitigkeit der Rückmeldungen und der aufgetretenen Irritationen konnte somit in den verschiedensten Stadien der Entwicklung frühzeitig eingegriffen und reagiert werden.

### 5.5.2 Erkenntnisse aus der Evaluation

Die Erkenntnisse, die aus der Evaluation gewonnen wurden, entstanden aus der Beobachtung verschiedener Personen bei Benutzung des Systems und anschließenden Gesprächen. Dabei wurden manchen Personengruppen die Funktionen im Vorfeld erklärt und gezeigt, andere hingegen aufgefordert, die Systeme selbstständig und ohne Erklärung zu benutzen.

#### Grundlegende Funktionen

Es wurde festgestellt, dass Personen, welche eine Einweisung in Funktionen bekommen hatten, weitestgehend alle Gesten und Aktionen im Bereich der Bewegungsverfolgung und der kontinuierlichen Gesten (siehe Kapitel 4.2) ohne größere Probleme bedienen konnten. Die Fähigkeit die Funktionen zu handhaben, steigerte sich nach kurzer Zeit, meist sogar nach einmaligem Ausprobieren. Gelegentlich traten Irritationen, welche durch die Hardware verschuldet wurden, beim Phänomen der Ghost Points (siehe 5.1.1) auf. Meist wurde in diesen Fällen, nach mehrmaligem Ausprobieren, nachgefragt. Nach kurzer Erklärung konnte man dann fortlaufend eine Vermeidung der Situationen feststellen, in denen die Probleme

---

<sup>20</sup><http://www.kioskeurope-expo.com/>

auftraten.

Bei den Gruppen, welche keine Unterweisung erhielten, wurde festgestellt, dass oftmals die Gesten, deren mentalen Modelle sehr nah an der Realität lagen, sofort beherrscht wurden. Das einfache Verschieben der Objekte oder das Rotieren wurde sofort angenommen. Das Menü wurde oftmals zufällig entdeckt. Nach der Entdeckung wurde der Mechanismus, das Menü aufzurufen, schnell erkannt und in das Repertoire der Interaktion aufgenommen. Die Interaktion des Flip wurde nur in den seltensten Fällen entdeckt. Wenn die Aktion entdeckt wurde, konnte sie meist erst mit Hilfestellung reproduziert werden.

### **Erlernbarkeit der Funktionen**

Auffällig war, dass Gesten, sobald sie erkannt waren, sehr schnell angenommen und eingesetzt wurden. Gerade das Bewegen von Bildern, das Rotieren oder das Vergrößern wurde fast komplett ohne Nachfrage eingesetzt.

Beeindruckend war die spielerische Herangehensweise aller Personen. Mit zunehmendem Alter der Personen war eine schneller einsetzende Frustration festzustellen, wenn erste Versuche nicht gelangen. Bei jüngeren Teilnehmern wirkte es wie eine Herausforderung, die Geste zu meistern.

Vermehrte Schwierigkeiten hatten viele Personen mit der Flip Geste, wenn das Bild nicht parallel zum Rahmen ausgerichtet war, sondern leicht gedreht (siehe Bild 5.23). Da der Flip die Rotation des Objektes nicht berücksichtigt, führte es oftmals zu Fehlversuchen, da sich Tester an der Ausrichtung des Bildes orientierten und versuchen die Geste parallel zur Objektkante durchzuführen (Bild 5.23 links). Nach dem Hinweis, dass die Orientierung zur Bildschirmkante zum Erfolg führt (Bild 5.23 rechts), konnte man optische und motorische Schwierigkeiten feststellen, sich nicht auf die Ausrichtung des Objektes zu orientieren. Trotz des Wissens über die richtige Anwendung verleitete die Schräglage des Bildes, die Geste in der gleichen Schräglage durchzuführen. Zu untersuchen ist hier, ob bei einer Anpassung der Geste an die Ausrichtung der Objekte auch eine Verbesserung der Benutzbarkeit der Geste eintritt.

### **Mitgebrachtes Wissen über Gesten und Interaktion**

Die häufigste Geste, die als Erstes ausprobierte wurde, war die von Apple in ihrer Werbung für das iPhone gezeigte Zoom-Geste. Ein sehr großer Teil begann damit die Multitouch Fä-

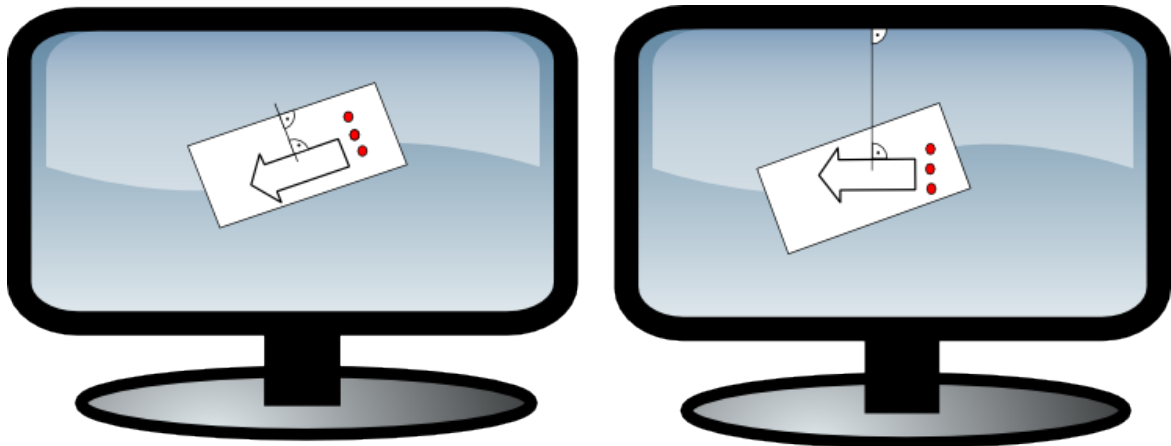


Abbildung 5.23: Orientierung der Flip Geste an verschiedenen Bezugspunkten

higkeit anhand dieser Geste zu testen. Danach wurden Rotation und Bewegung der Bilder ausprobiert. Da für die Zoom-Geste keine reale Entsprechung existiert, sondern nur anhand von Gummiband-Metaphern erklärt wird, ist umso erstaunlicher, dass anscheinend der Begriff Multitouch und Gesteninteraktion durch Werbemaßnahmen so stark geprägt wurden, dass bestimmte Gesten sofort damit in Verbindung gebracht werden. Zu beobachten ist, ob bei weiteren Werbemaßnahmen mit anderen Gesten der gleiche Effekt eintritt und somit eine durch Werbung generierte Erwartungshaltung an diese Systeme geschaffen wird oder es ausschließlich mit dem damaligen Hype mit der Einführung des iPhones verbunden war.

### **Ergonomie der Gesten**

Bei vielen Gesten konnte man eine natürliche Bedienung erkennen und auch die bequeme Bedienbarkeit feststellen. Auffällig war allerdings abermals die Flip Geste. Viele verkrampften beim Versuch, ihre Hand in eine entsprechende Position zu bringen und die Bewegung auszuführen, und schafften es dann nicht, die Aktion auszuführen. Oftmals wurde ein zusätzlicher Finger der Hand erkannt und verhinderte die Erkennung. An diesem Punkt muss zukünftig drauf geachtet werden, dass das System entweder mehr Freiheit gibt, in der Art und Weise wie Gesten ausgeführt werden können oder die Gesten müssen in Hinsicht auf Hand und Fingerstellung überdacht werden.

## 5.6 Bewertung der Umsetzung

Durch die fortwährenden Evaluationsprozesse, welche die Prototypen der Anwendungen durchliefen, konnten viele der Anforderungen umgesetzt werden.

Der immense Erfahrungsgewinn durch diese Vorgehensweise, ermöglichte es Applikationen zu schaffen, welche in vielen Hinsichten zufriedenstellende Ergebnisse liefern konnten.

Die entstandene Architektur ermöglicht es weitere Anwendungen zu entwickeln und sie auf ihre Bedienbarkeit zu untersuchen. Durch die Abstraktion der Hardware ist es zukünftig möglich neue Hardware schnell einzusetzen. Das entwickelte Design und die Anwendungen können zukünftig als Basis zur weiteren Experimenten mit dieser Technologie führen. Durch die Neuartigkeit des Themas und den bestehenden Problemen mit der Hardware, ist der Entwicklungsprozess noch nicht abgeschlossen. Weiterführende arbeiten sind notwendig, um eine Verbesserung des Systems zu erreichen und das Verständnis für die Technologie und die Interaktion mit dieser zu vertiefen.

# 6 Schluss

## 6.1 Zusammenfassung

Im Rahmen dieser Arbeit wurden neue Bedienkonzepte auf Basis von Multitouch Technologie vorgestellt. Anhand von zwei Szenarien (s. 3.1) konnte eine Wand- und eine Tisch Applikation konzipiert und entwickelt werden (s. 3.1.3), welche mit Hilfe der Technologie umgesetzt wurden. Zum Einen wurde die Multitouch-Wall entwickelt, eine gestenbasierte Applikation zum Darstellen von, und Interagieren mit Medieninhalten auf Basis einer Pinnwandmetapher. Zum Anderen das TouchTable, eine weitere gestenbasierte Applikation zum Darstellen von, und Interagieren mit Medieninhalten auf Basis einer Tischoberfläche, also einer Desktopmetapher.

In der Einführung (s. 1) dient ein Zitat und die dahinterliegende Idee von Mark Weiser als Motivation für die Entwicklung der neuen Bedienkonzepte und der zugehörigen Systeme. Das Überwinden des von ihm erwähnten „Tunnels“, oder auch als Barriere zu sehende Hürde, der heutigen Mensch-Maschine-Interaktionen war bei der Entwicklung erklärtes Hauptziel der Arbeit. Hierzu dienten die Grundlagen als Einstieg in die Informationsvielfalt, die zu diesem Thema vorhanden ist.

Auf den theoretischen Grundlagen (s. 2), welche von Weiser, Ishii, Greif und Pierce gebildet wurden, baut diese Arbeit eine mögliche Realisierung der Konzepte auf. Somit waren Teilziele die Entwicklung natürlicher und intuitiver Eingabemethoden (s. 2.2). Berücksichtigt wurde dabei die Einbettung der entwickelten Systeme in das Ambient Labor der HAW Hamburg, welches als Raum für den Themenbereich der kollaborativen Arbeit dient. In diesem Zusammenhang wurde als weitere theoretische Grundlage das Group Information Environment eingeführt, welches das verteilte Wissen in Teamstrukturen darstellt.

Ein erster Schritt in Richtung der Realisierung von natürlichen Eingabemethoden, sind Touch-Interfaces, bei denen durch direktes Anfassen der dargestellten Informationen die Diskrepanz zwischen der Erwartungshaltung eines Menschen und tatsächlicher Bedienung eines Systems minimiert wird. Aus den verwendbaren in den Grundlagen aufgeführten Touch-Technologien, wurde ein Infrarot-Touch Rahmen gewählt, der für die Realisierung am Besten schien.

Die Anforderungen an die Systeme wurden in der Anforderungsanalyse formuliert (s. 3). Um diese aus den Szenarien zu erhalten, wurde ein Anwendungsfall aufgestellt, aus denen eine Reihe von funktionalen und nichtfunktionalen Anforderungen hervorgehen. Dem Benutzer müssen einige grundlegende Funktionen bereitgestellt werden, sodass ein Informationsgewinn aus den dargestellten Informationen gezogen werden kann. Hierzu wurden Gesten eingeführt, welche im Gesten Kapitel (s. 4) erarbeitet wurden.

Die bisherigen Gestentaxonomien im Bereich der Mensch-Computer-Interaktion beriefen sich auf eine Kommunikation zwischen Mensch und Computer (s. 4). Die in dieser Arbeit verwendeten Interaktionen sind aber eher manipulativer Natur. So wurde eine Erweiterung der Gestentaxonomie von Pavlovic et. al. vorgenommen, bei der der Teilbaum der manipulativen Gesten um drei Gestenarten ergänzt wurde.

Anhand der geforderten Funktionalitäten und Eigenschaften an die Systeme wurde, wie in Design und Realisierung beschrieben (s. 5), eine Architektur konzipiert und umgesetzt, die es ermöglicht intuitive Bedienkonzepte auf Basis von Multitouch-Technologie zu realisieren. Parallel zur Entwicklung wurden Evaluation der intermediären Ergebnisse durchgeführt, um Missstände in der Konzeption und Umsetzung sofort korrigieren zu können. Hierbei wurden die Systeme unter anderem auf einer Messe ausgestellt, bei der sie sich bewähren konnten.

Aus den Erfahrungen der Evaluation (s. 5.5) ergaben sich viele Möglichkeiten für die Weiterentwicklung der Systeme, die im folgenden Ausblick aufgezeigt werden.

## 6.2 Ausblick

### Erweiterung des Gestenrepertoires

Die mit den Benutzern durchgeführten Evaluationen haben gezeigt, dass durchaus mehr Gesten als bisher vorgesehen möglich sind. Durch die intuitive Bedienung konnten die meisten Benutzer die vorgegebenen Gesten in relativ kurzer Zeit umsetzen und sogar Vorschläge für neue Gesten machen. Es sind weitere Studien durchzuführen, die aufzeigen wie intuitiv diese neuen Gesten sind. Die Einschätzung von Apple, dass bis zu 300 Gesten pro Hand möglich sind, erscheint allerdings unwahrscheinlich.

Hierzu ist vor allen Dingen eine Implementierung einer, in dieser Arbeit vorgesehenen aber nicht umgesetzten, Gestenbibliothek von Nöten. Durch diese könnten Gesten genauer spezifiziert und weitere durch Anlernen des Systems hinzugefügt werden. In diesem Zusammenhang sollten auch elegantere Verfahren zur Gestenerkennung eingeführt werden.



Um diese Ziele zu ermöglichen, müssen Mängel der Hardware beseitigt werden, welche eine effiziente Umsetzung bisher unmöglich machen.

## **Verbesserte Hardware**

In Gesprächen mit einer Herstellerfirma von Infrarot Touchrahmen, wurden die Schwierigkeiten mit der verwendeten Hardware geschildert. Dazu gehört auch die Vereinfachung der Zuordnung von Punkte- Wertepaaren, also die Auflösung von Ghost Points, und die Erkennung von mehr als nur Punkten auf dem Schirm, wie z.B. die Erkennung von Konturen. Diese Schwierigkeiten können mit unterschiedlichen Mitteln korrigiert werden.

Wenn diese Mängel aus dem Weg geräumt sind, ergeben sich neben den folgenden Verbesserungsmöglichkeiten an den Applikationen noch viele weitere Funktionen und Möglichkeiten.

## **Verbesserte Applikationen**

Die Multitouch-Wall könnte mit einer neuen Grafik Engine wesentlich verbessert werden. Dies ist sinnvoll, wenn viele Videos und Bilder mit hoher Auflösung dargestellt werden sollen. Auch interessant wäre die höhere Performance für die ruckelfreie Darstellung von 3D Modellen mit einer hoher Anzahl an Polygonen.

In Hinblick auf eine weitere Messe, auf der die Multitouch-Wall ausgestellt werden soll, werden neue Funktionalitäten hinzugefügt. Unter anderem wird hierfür auf der virtuellen Pinnwand eine dedizierte Fläche hinzugefügt auf der Produktbilder abgelegt werden können, um auf einem angeschlossenen USB-Stick die spezifischen Produktinformationen zu hinterlegen. So erhält der Kunde einen individualisierten Produktkatalog, den er selber zusammenstellen und mitnehmen kann.

Für das TouchTable wäre denkbar, durch bestehende Technik, wie z.B. RFID Tags, Interessen eines wichtigen VIP Kunden zu hinterlegen. Die Status-Informationen können dann als Profil abgerufen werden, so dass das System personalisiert agieren kann, indem es nur Informationen oder Produkte anzeigt, die für den jeweiligen Kunden interessant sind.

Bei einer Interessengemeinschaft von mehreren Herstellern könnte man sich vorstellen, dass der Kunde für ihn interessante Produkte unterschiedlicher Hersteller auf seine Profil-Karte speichert, die dann bei Ankunft am Multitouch-System vom einem zentralen Server auf dem Bildschirm präsentiert werden, um dort dann genauer betrachtet werden zu können. Um festzustellen, ob sich ein Messebesucher in der Nähe des Systems befindet, können

Abstandssensoren eingesetzt werden.

## Multimodale Interaktion

Das Hinzufügen von vielen Funktionen könnte unter Umständen die Intuitivität der Systeme stark beeinträchtigen. Der Einsatz von multimodaler Interaktion, also die Verwendung von unterschiedlichen Eingabemethoden, ermöglicht es, zu komplexe Gesten zu vermeiden. Als weitere Eingabemethode eignet sich die Spracheingabe, über die Kommandos abgesetzt werden können. Die Kombination von Zeigen und Sprechen ist in der menschlichen Kommunikation üblich und würde daher die natürliche Interaktion begünstigen.

Da visuelles Feedback alleine oft nicht ausreicht, um dem Benutzer eine Aktion zu bestätigen, wäre der Einsatz von Vibrationsarmbändern denkbar, die ein haptisches Feedback erzeugen.

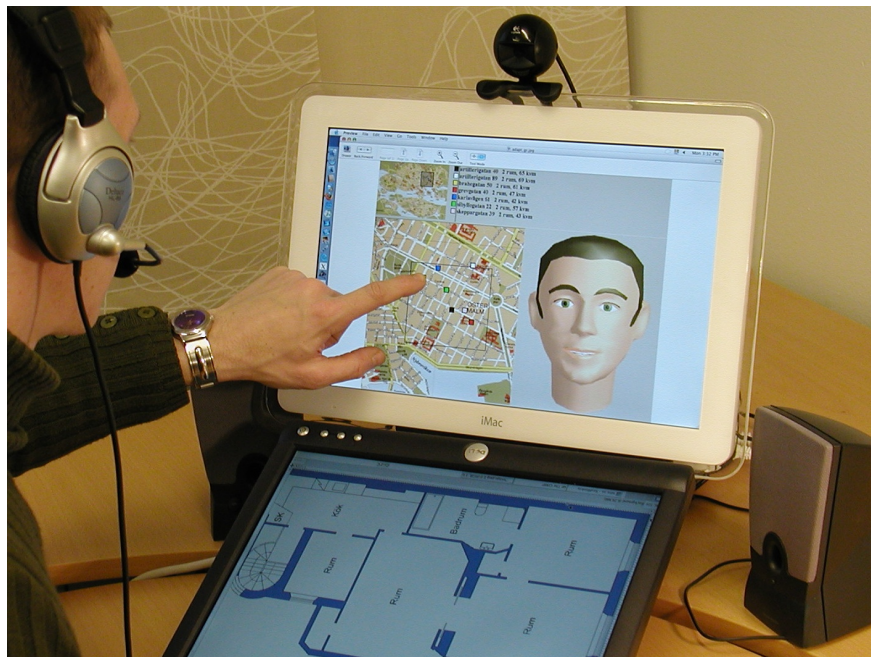


Abbildung 6.1: Multimodale Interaktion (von [KTH Speech, Music and Hearing \(2008\)](#))

## Fazit

Die oben beschriebenen vielfältigen Erweiterungsmöglichkeiten zeigen das Potential der Multitouch-Wall und des TouchTable, welches noch in diesen Systemen steckt.

Sie sind jedoch nicht die einzigen Systeme, die zu dem Thema entwickelt wurden und werden. Die zunehmende Anzahl an Veröffentlichungen von Forschungsarbeiten und Produkten auf dem Markt, beweisen den steigenden Bedarf an den entwickelten Systemen. Unter anderem gehören dazu das von Microsoft entwickelte Surface Computing und Sphere System und das Apple iPhone. Ein weiterer Kandidat ist die Nintendo Wii, welche sich gegen besser Ausgestattete Rivalen durchsetzen konnte, mit der Hilfe eines intuitiveren Bedienkonzeptes.

Mit dieser Arbeit ist ein kleiner Schritt in Richtung intuitiv bedienbarer Systeme getan. Weitere Untersuchungen zu diesem Thema sind notwendig, um einfach mit Computern in der Umgebung zu interagieren. Somit ist zu erwarten, dass in naher Zukunft viele weitere spannende Geräte und Forschungsprojekte im Bereich der Seamless Interaction zu erwarten sind.

# Literaturverzeichnis

- [ACM 2008] ACM: *ACM SIGCHI Curricula for Human-Computer Interaction*. Webseite. 2008. – URL <http://sigchi.org/cdg/cdg2.html>. – Letzter Aufruf am 14. August 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 Press, 2006, S. 1283–1292. – ISBN 1-59593-372-7
- [Apple Inc. 2007] APPLE INC.: *iPhone*. Webseite. 2007. – URL <http://www.apple.com/iphone/>. – Letzter Aufruf am 14. Mai 2008
- [Apple Inc. 2008a] APPLE INC.: *iPhone*. Webseite. 2008. – URL <http://www.apple.com/>. – Letzter Aufruf am 14. August 2008
- [Apple Inc. 2008b] APPLE INC.: *MacBook Air*. Webseite. 2008. – URL <http://www.apple.com/macbookair/features.html>. – Letzter Aufruf am 14. August 2008
- [Apple Inc. 2008c] APPLE INC.: *MobileMe*. Webseite. 2008. – URL <http://www.apple.com/mobileme>. – Letzter Aufruf am 14. August 2008
- [Apple Inc. Multi-touch Gesture Dictionary Patent Application 2007] APPLE INC. MULTI-TOUCH GESTURE DICTIONARY PATENT APPLICATION: *Multi-touch Gesture Dictionary Patent Application*. Webseite. 2007. – URL <http://appft1.uspto.gov/netacgi/nph-Parser?Sect1=PTO1&Sect2=HITOFF&d=PG01&p=1&u=%2Fnetacgi%2FPTO%2Fsrchnum.html&r=1&f=G&l=50&s1=%2220070177803%22.PG NR.&OS=DN/20070177803&RS=DN/20070177803>. – Letzter Aufruf am 14. August 2008
- [AT&T Inc. 2008] AT&T INC.: *AT&T Inc.* Webseite. 2008. – URL <http://www.att.com/>. – Letzter Aufruf am 14. August 2008
- [Boetzer 2008] BOETZER, Joachim: *Bewegungs und gestenbasierte Applikationssteuerung auf Basis eines Motion Trackers*, HAW Hamburg, Bachelorarbeit, 2008

- [Brown u. a. 1985] BROWN, John S. ; CASHMAN, Paul M. ; MALONE, Thomas: Interfaces in organizations (panel session): supporting group work. In: *SIGCHI Bull.* 16 (1985), Nr. 4, S. 65. – ISSN 0736-6906
- [Carnegie Mellon 2008] CARNEGIE MELLON: *Carnegie Mellon HCI*. Webseite. 2008. – URL <http://www.hcii.cmu.edu/>. – Letzter Aufruf am 14. August 2008
- [Craik 1943] CRAIK, K.: *The Nature of Explanation*. 1943
- [D: All Things Digital 2007] D: ALL THINGS DIGITAL: *Windows7*. Webseite. 2007. – URL [http://d6.allthingsd.com/20080527/gates\\_ballmer/](http://d6.allthingsd.com/20080527/gates_ballmer/). – Letzter Aufruf am 14. August 2008
- [Deutsches Zentrum für Luft- und Raumfahrt 2008] DEUTSCHES ZENTRUM FÜR LUFT- UND RAUMFAHRT: *Deutsches Zentrum für Luft- und Raumfahrt*. Webseite. 2008. – URL [http://www.dlr.de/DesktopDefault.aspx/tabid-4530/3681\\_read-10978/gallery-1/gallery\\_read-Image.1.4481/](http://www.dlr.de/DesktopDefault.aspx/tabid-4530/3681_read-10978/gallery-1/gallery_read-Image.1.4481/). – Letzter Aufruf am 14. August 2008
- [Elektrosil 2008] ELEKTROSIL: *Elektrosil*. Webseite. 2008. – URL <http://www.elektrosil.com/>. – Letzter Aufruf am 14. August 2008
- [Fischer 2007] FISCHER, Christian: *Entwicklung eines multimodalen Interaktionssystems für computergestützte Umgebungen*, Hochschule für Angewandte Wissenschaften (HAW) Hamburg, Diplomarbeit, 2007. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/master/fischer.pdf>
- [Gamma u. a. 1995] GAMMA, Erich ; HELM, Richard ; JOHNSON, Ralph ; VLISSIDES, John: *Design patterns: elements of reusable object-oriented software*. Addison-Wesley Professional, 1995
- [Gehn 2007a] GEHN, Stefan: *Evaluation einer infrarotbasierten Multitouchhardware*. Projektbericht an der Hochschule für Angewandte Wissenschaften Hamburg. 2007. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master07-08-proj/gehn/report.pdf>
- [Gehn 2007b] GEHN, Stefan: *Techniken für Interaktion mittels Bewegungen und Gesten*. Seminararbeit an der Hochschule für Angewandte Wissenschaften Hamburg. 2007. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master07-08-aw/gehn/folien.pdf>
- [Gehn 2008] GEHN, Stefan: *Forthcoming*. 2008. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/papers.html>

- [Han 2005] HAN, Jefferson Y.: Low-cost multi-touch sensing through frustrated total internal reflection. In: *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology*. New York, NY, USA : ACM Press, 2005, S. 115–118. – ISBN 1-59593-271-2
- [Harrah's Entertainment 2008] HARRAH'S ENTERTAINMENT: *Harrah's Entertainment*. Webseite. 2008. – URL <http://www.harrahs.com/>. – Letzter Aufruf am 14. August 2008
- [Heitsch 2008] HEITSCH, Johann: Ein Framework zur Erkennung von dreidimensionalen Gesten / Hochschule für Angewandte Wissenschaften Hamburg. 2008. – Bachelorarbeit
- [Hiroshi Ishii 2008] HIROSHI ISHII: *Hiroshi Ishii Publications*. Webseite. 2008. – URL <http://web.media.mit.edu/~ishii/publications.html>. – Letzter Aufruf am 14. August 2008
- [Hollatz 2007] HOLLATZ, Dennis: *Einsetzbarkeit von AJAX-basierten Applikationen für kooperatives Arbeiten*. Bachelorarbeit an der Hochschule für Angewandte Wissenschaften Hamburg. 2007. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/bachelor/hollatz.pdf>
- [HowStuffWorks 2008] HOWSTUFFWORKS: *HowStuffWorks? iPhone*. Webseite. 2008. – URL <http://electronics.howstuffworks.com/iphone.htm>. – Letzter Aufruf am 14. August 2008
- [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
- [Ishii und Ullmer 1997] ISHII, Hiroshi ; ULLMER, Brygg: *Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms*. 1997. – URL [citeseer.ist.psu.edu/ishii97tangible.html](http://citeseer.ist.psu.edu/ishii97tangible.html)
- [JazzMutant Lemur 2008] JAZZMUTANT LEMUR: *JazzMutant Lemur*. Webseite. 2008. – URL [http://www.jazzmutant.com/lemur\\_overview.php](http://www.jazzmutant.com/lemur_overview.php). – Letzter Aufruf am 14. August 2008
- [Jeff Pierce 2008] JEFF PIERCE: *Personal Information Environments*. Webseite. 2008. – URL <http://www.almaden.ibm.com/cs/projects/pie/>. – Letzter Aufruf am 14. August 2008
- [Köckritz 2007] KÖCKRITZ, Oliver: *Virtueller Collaborative Workspace mit Remote Windows*. Seminararbeit. Februar 2007. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master06-07-aw/koeckritz/report.pdf>

- [KTH Speech, Music and Hearing 2008] KTH SPEECH, MUSIC AND HEARING: *KTH*. Webseite. 2008. – URL <http://www.speech.kth.se/courses/DT2140/>. – Letzter Aufruf am 14. August 2008
- [Marceau Foundation 2008] MARCEAU FOUNDATION: *Marceau Foundation*. Webseite. 2008. – URL <http://www.marceaufoundation.org/>. – Letzter Aufruf am 14. August 2008
- [McNeill 2008] MCNEILL: *Center for Gesture and Speech Research*. Webseite. 2008. – URL <http://mcneilllab.uchicago.edu/>. – Letzter Aufruf am 14. August 2008
- [Microsoft Corporation 2007] MICROSOFT CORPORATION: *Microsoft Surface*. Webseite. 2007. – URL <http://www.microsoft.com/surface/>. – Letzter Aufruf am 14. August 2008
- [Microsoft Corporation 2008] MICROSOFT CORPORATION: *LiveMesh*. Webseite. 2008. – URL <http://www.mesh.com/>. – Letzter Aufruf am 14. August 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>
- [NextWindow Limited 2008] NEXTWINDOW LIMITED: *NextWindow Limited*. Webseite. 2008. – URL <http://www.nextwindow.com>. – Letzter Aufruf am 14. August 2008
- [NextWindow Limited, Multitouch Whitepaper 2008] NEXTWINDOW LIMITED, MULTITOUCH WHITEPAPER: *NextWindow Limited, Multitouch Whitepaper*. Webseite. 2008. – URL [http://www.nextwindow.com/pdf/nextwindow\\_multitouch.pdf](http://www.nextwindow.com/pdf/nextwindow_multitouch.pdf). – Letzter Aufruf am 14. August 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
- [Perceptive Pixel 2007] PERCEPTIVE PIXEL: *Perceptive Pixel*. Webseite. 2007. – URL <http://www.perceptivepixel.com/>. – Letzter Aufruf am 14. August 2008
- [Piening 2007] PIENING, Andreas: *Katastrophen-Leitstand: Current work and projects*. Seminararbeit an der Hochschule für Angewandte Wissenschaften Hamburg. 2007. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master06-07-aw/piening/report.pdf>
- [Reenskaug 2003] REENSKAUG, Trygve: *The Model-View-Controller (MVC) Its Past and Present*. 2003. – URL <http://heim.ifi.uio.no/~trygver/2003/javazone-jaoo/HM1A93.html>

- [Roßberger 2007] ROSSBERGER, Philipp: *Physikbasierte Interaktion in kollaborativen computergestützten Umgebungen*, Hochschule für Angewandte Wissenschaften (HAW) Hamburg, Diplomarbeit, 2007. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/master/rossberger.pdf>
- [Schilit u. a. 1994] SCHILIT, B ; ADAMS, N ; WANT, R: *Context-aware computing applications*. 1994
- [Scott 2005] SCOTT, Stacey D.: *Territoriality in Collaborative Tabletop Workspaces*, University of Calgary, Dissertation, March 2005
- [Stanford 2008a] STANFORD: *Interactive Workspaces*. Webseite. 2008. – URL <http://iwork.stanford.edu/>. – Letzter Aufruf am 14. August 2008
- [Stanford 2008b] STANFORD: *Stanford HCI*. Webseite. 2008. – URL <http://hci.stanford.edu/>. – Letzter Aufruf am 14. August 2008
- [Stanford HCI Group's iTable 2008] STANFORD HCI GROUP'S ITABLE: *Stanford HCI Group's iTable*. Webseite. 2008. – URL <http://hci.stanford.edu/research/itable.html>. – Letzter Aufruf am 14. August 2008
- [Tandler 2004] TANDLER, Peter: *Synchronous Collaboration in Ubiquitous Computing Environments*, TU Darmstadt, Fachbereich Informatik, Dissertation, 2004. – URL <http://elib.tu-darmstadt.de/diss/000506>
- [Telepolis 1999] TELEPOLIS: *Interview mit Mark Weiser*. Webseite. 1999. – URL <http://www.heise.de/tp/r4/artikel/5/5033/1.html>. – Letzter Aufruf am 14. August 2008
- [The Multi-Pointer X Server 2008] THE MULTI-POINTER X SERVER: *The Multi-Pointer X Server*. Webseite. 2008. – URL <http://wearables.unisa.edu.au/mpx/?q=mpx>. – Letzter Aufruf am 8. August 2008
- [Tim Roth 2008] TIM ROTH: *Multitouch Dev Blog*. Webseite. 2008. – URL <http://iad.projects.zhdk.ch/multitouch/>. – Letzter Aufruf am 14. August 2008
- [Touchscreenguide 2008] TOUCHSCREENGUIDE: *Touchscreen Guide*. Webseite. 2008. – URL <http://www.touchscreenguide.com>. – Letzter Aufruf am 14. August 2008
- [Ubiquitous ID Center 2008] UBIQUITOUS ID CENTER: *Ubiquitous Computing Shema*. Webseite. 2008. – URL [http://www.uidcenter.org/english/ubi\\_te.html](http://www.uidcenter.org/english/ubi_te.html). – Letzter Aufruf am 14. August 2008
- [Weiser 1991] WEISER, Mark: The Computer for the Twenty-First Century. In: *Scientific American* 265 (1991), S. 94–104



- 
- [Wikipedia 2008] WIKIPEDIA: *Wikipedia*. Webseite. 2008. – URL [http://de.wikipedia.org/wiki/Chinesische\\_Zahlen](http://de.wikipedia.org/wiki/Chinesische_Zahlen). – Letzter Aufruf am 14. August 2008
- [Winograd 2003] WINOGRAD, Terry: *Stanford Interactive Workspaces Project Overview*. Webseite. 2003. – URL <http://iwork.stanford.edu/>. – Letzter Aufruf am 14. August 2008

# Glossar

**3D Rendering** ist die Erzeugung einer auf dem Bildschirm darstellbaren Grafik aus räumlichen Beschreibungen von 3D-Modellen, Lichtquellen, Oberflächenbeschaffenheit und anderen Eigenschaften.

**Ambient-Assisted-Living** ist ein europäisches Technologie- und Innovations Förderprogramm. Weiteres ist unter <http://www.aal169.org/> nachzulesen.

**API** Applikation Programming Interface ist eine Programmierschnittstelle, die anderen Programmen zur Verfügung gestellt wird.

**BlockingQueue** Eine Liste, mit blockierendem Zugriff, mit Hilfe derer mehrere Zugriffe auf eine Ressource geregelt werden können.

**Bounding Box** ist eine Box, die in einer virtuellen 3D-Welt die Umrandung eines Modells darstellt.

**Compiler** oder auch Übersetzer, wandelt Quellcode vom Programmierer in einen von der Maschine verarbeitbaren Code um.

**Concurrent Package** ist eine von Sun seit Java 5 zur Verfügung gestellte Library, mit der Threading spezifische Aufgaben gelöst werden können.

**CRC** Cyclic Redundancy Check ist ein Verfahren zur Erkennung von Fehlern bei der Datenübertragung.

**Framework** stellt einen Rahmen oder ein Grundgerüst für die Entwicklung von Programmen dar.

**GPS** Global Positioning System ist ein Satelliten gestütztes weltweites Ortungssystem, welches häufig für Navigationssysteme eingesetzt wird.

**Grafik-Engines** sind Bibliotheken oder eigenständige Computerprogramme, welche sich um die Anzeige und Darstellung von grafischen Objekten auf dem Bildschirm kümmern.

**Haptik** Vom griechischen Wort haptikos, greifbar. Darunter versteht man die Wahrnehmung durch den Tastsinn und Griff-Eigenschaften des Materials.

**Head-Up-Displays** sind Anzeigen, die einem Benutzer, welcher dieses System auf dem Kopf trägt oder vor sich stehen hat, Informationen direkt vor das Sichtfeld projizieren.

**Hype** engl. für Publicity, Rummel oder gar Reklameschwindel

**Mainframe** Mainframes sind Großrechner, die sich durch hohe Zuverlässigkeit und Ein-/Ausgabe-Leistung auszeichnen.

**Passant** ist ein evtl. interessierter vorbeigehender Besucher.

**Physik-Engines** sind Computerprogramme oder Teile von Computerprogrammen. Sie dienen dazu physikalische Prozesse zu simulieren, um so Eigenschaften von Materialien und Objekten aus der Realität nachzubilden. Anwendung finden Physik-Engines hauptsächlich in der Spieleindustrie und sorgen dort bei modernen Spielen für realistische Effekte von Modellen.

**RFID Tags** sind Plastikkarten, welche mit einem Radio Frequency Identification Chip (RFID Chip) ausgestattet werden, um die Karte mittels elektromagnetischer Wellen identifizierbar zu machen.

**SDK** abk. für Software Development Kit. Es handelt sich dabei um eine Sammlung von Dokumentationen und Teilprogrammen, welche es dem Entwickler ermöglichen, Applikationen in der vom SDK bereitgestellten Umgebung zu entwickeln.

**Skriptsprachen** sind Programmiersprachen, die sich für die schnelle Entwicklung von Applikationen eignen. Es wird dabei bewusst auf einige Sprachelemente anderer Sprachen verzichtet, um eine prototypische Anwendungsentwicklung zu ermöglichen.

**Stylus** Stylus, oder auch Touchpen genannt, ist ein Stift zur Eingabe auf Touchscreens. Der Ursprung des Begriffes ist das lateinische Wort Stilus und bedeutet Stift.

**Textur** Eine Textur ist im Bereich der Computergrafik, ein Bild welches über ein 3D-Modell, als Haut gezogen wird.

**Threading** beschreibt das simultane Arbeiten von mehreren Prozessen an einer Entität. Es greifen also mehrere Prozesse gleichzeitig auf eine geteilte Ressource zu.

**UDP und Multicast** sind Netzwerkübertragungsarten, welche sich durch wenig Netzwerkauslastung auszeichnen.

**XML** ist eine Abkürzung für Extensible Markup Language. Es ist eine Auszeichnungssprache bei der Daten in hierarchischen Strukturen verpackt werden. Verwendet wird XML zur Datenübertragung über das Internet und andere Medien.

# Versicherung über Selbstständigkeit

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

Hamburg, 18. August 2008

Ort, Datum

Unterschrift

Hamburg, 18. August 2008

Ort, Datum

Unterschrift