

Fachhochschule Köln  
Cologne University of Applied Sciences

# **Gestenbasierte Interfaces**

## **Konzeption und Entwicklung unter Verwendung der Microsoft Kinect**

### **Masterarbeit**

im Fachgebiet Medieninformatik  
zur Erlangung des akademischen Grades

Master of Science (M.Sc.)

an der Fachhochschule Köln, Campus Gummersbach

vorgelegt von: Florian Gebauer

Studiengang: Medieninformatik Master

Matrikelnummer: 11048082

Erster Prüfer: Prof. Dr. Gerhard Hartmann

Zweiter Prüfer: Prof. Dr. Horst Stenzel

Gummersbach, im Februar 2012



---

## Zusammenfassung

Ein Ziel der Mensch Computer Interaktion besteht in der Verbesserung der Interaktion zwischen Mensch und Computer. Ein Ansatz besteht darin, die Interaktion zwischen Mensch und Computer so zu gestalten, dass der Mensch auf natürliche, bereits erlernte Fähigkeiten zurückgreifen kann. Der Paradigmenwechsel von Graphical User Interfaces zu Natural User Interfaces bringt diesen Wandel mit sich. Natural User Interfaces verwenden natürliche Eingaben des Menschen in Form von Gestik und Sprache und versetzen ihn in die Lage, direkt mit Inhalten zu interagieren.

Diese Arbeit beschäftigt sich mit der Konzeption und prototypischen Realisierung eines Natural User Interfaces. Die Arbeit gliedert sich dabei in drei Teile. Im ersten Teil werden Natural User Interfaces untersucht. Es werden Eigenschaften beschrieben und Konzepte zur Gestaltung vorgestellt. Der zweite Teil beschäftigt sich mit der Verwendung von Gesten als Eingabemodalität. Dabei werden Ansätze zur Einordnung von Gesten sowie die Erfassung von Gesten untersucht. Der dritte Teil der Arbeit besteht in der Konzeption sowie prototypischen Umsetzung eines Natural User Interface. Als Eingabemodalität wird die Microsoft Kinect verwendet. Die grafische Benutzeroberfläche wird in Unity3D gestaltet. Die Konzeption und Realisierung erfolgt anhand der beschriebenen Konzepte in dieser Arbeit.

## Abstract

In the area of Human Computer Interaction (HCI) exists a long-established demand for advanced methods to improve the interaction between humans and computers. One approach consists in arranging the HCI in such a way, that the humans can make use of their natural, acquired skills. The paradigm shift from Graphical User Interfaces to Natural User Interfaces is bringing about this change. Natural User Interfaces use natural input modalities of humans and enable them to interact directly with the content. This work deals with the conception and the prototypical implementation of Natural User Interfaces and is divided into three parts. The first part investigates the Natural User Interfaces by describing its qualities and introducing different design concepts. The second part takes a closer look at the use of gestures as an input modality. For this purpose, several approaches of the classification of gestures and the entry of gestures are examined. The third part of the work deals with the conception and the prototypical conversion of a Natural User Interface. In this case, the Microsoft Kinect is used as an input modality. The user interface is



---

designed in Unity3D. The conception and realization take place on the basis of the described concepts of this work.





# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	<b>IV</b>
------------------------------	-----------

<b>Verzeichnis der Listings</b>	<b>VII</b>
---------------------------------	------------

<b>1. Einleitung</b>	<b>1</b>
----------------------	----------

1.1. Problembeschreibung . . . . .	1
1.2. Ziel der Arbeit . . . . .	2
1.3. Abgrenzung der Arbeit . . . . .	3
1.4. Aufbau der Arbeit . . . . .	4

<b>2. Verwandte Arbeiten</b>	<b>6</b>
------------------------------	----------

2.1. Gestenbasierte Interaktion . . . . .	6
2.2. Gestenbasierte Interfaces . . . . .	8
2.3. Gestenbasierte Anwendungen . . . . .	9

<b>3. Grundlagen</b>	<b>11</b>
----------------------	-----------

3.1. Mensch Computer Interaktion . . . . .	11
3.2. Gesten und Menschen . . . . .	12
3.3. Natürliches und intuitives Handeln . . . . .	14
3.4. Microsoft Kinect . . . . .	15
3.4.1. Hardware der Kinect . . . . .	16
3.4.2. Funktionsweise der Kinect . . . . .	18
3.4.3. Beurteilung der Kinect . . . . .	21
3.5. Verwendete Software . . . . .	22
3.5.1. OpenNI . . . . .	23
3.5.2. NITE . . . . .	24
3.5.3. Unity3D . . . . .	26

<b>4. Gestenbasierte Eingabe</b>	<b>27</b>
----------------------------------	-----------

4.1. Definition von Gesten . . . . .	27
4.2. Gestentypen . . . . .	29
4.3. Erfassung von Bewegungen . . . . .	32
4.3.1. Tragbare Geräte . . . . .	33



4.3.2. Tragbare Anzüge . . . . .	34
4.3.3. Computer Vision . . . . .	35
4.4. Erfassung von Gesten . . . . .	36
4.4.1. Gestenmodellierung . . . . .	36
4.4.2. Segmentierung . . . . .	38
4.4.3. Merkmalsbestimmung . . . . .	39
4.4.4. Klassifizierung . . . . .	39
4.5. Interkulturalität . . . . .	40
4.6. Gestenbasierte Systeme . . . . .	42
4.7. Zwischenfazit . . . . .	44
<b>5. Gestenbasierte Interfaces</b>	<b>45</b>
5.1. Definition von NUI . . . . .	45
5.2. Vom CLI zu NUI . . . . .	47
5.3. Eigenschaften von NUI . . . . .	48
5.4. Anwendungsbeispiele . . . . .	51
5.4.1. Apple iPhone . . . . .	51
5.4.2. Microsoft Surface . . . . .	52
5.4.3. Microsoft Xbox360 . . . . .	53
5.5. Ergonomie gestenbasierter Interfaces . . . . .	54
5.6. OCGM - Interface Metaphern . . . . .	56
5.7. Zwischenfazit . . . . .	58
<b>6. Konzeption</b>	<b>60</b>
6.1. Vorgehensweise . . . . .	60
6.1.1. Größe der Interface-Elemente . . . . .	61
6.1.2. Anordnung der Interface-Elemente . . . . .	61
6.1.3. Verhalten der Interface-Elemente . . . . .	62
6.1.4. Farbgebung . . . . .	62
6.2. Gestenbasierte Eingaben . . . . .	63
6.3. Anwendungsbeispiel . . . . .	63
6.3.1. Startoberfläche . . . . .	64
6.3.2. Aktuelles . . . . .	64
6.3.3. Mensa . . . . .	67
<b>7. Realisierung</b>	<b>70</b>
7.1. Architektur . . . . .	70
7.2. Anmeldung . . . . .	72
7.3. Handverfolgung . . . . .	73



7.4. Selektion . . . . .	75
7.5. Push-Geste . . . . .	76
<b>8. Zusammenfassung und Fazit</b>	<b>78</b>
<b>Literaturverzeichnis</b>	<b>82</b>
<b>Eidesstattliche Erklärung</b>	<b>93</b>
<b>A. Anhang</b>	<b>i</b>
A.1. Inhalt der DVD . . . . .	i
A.2. Interface Abbildungen . . . . .	i



## Abbildungsverzeichnis

3.1. Das interdisziplinäre Forschungsfeld der Mensch Computer Interaktion nach Bongers . . . . .	12
3.2. Modell der Interaktion nach Norman . . . . .	13
3.3. OSIT Handlungsmodell nach Henseler . . . . .	15
3.4. Kernkomponenten der Microsoft Kinect . . . . .	17
3.5. Primesense Referenzdesign . . . . .	18
3.6. Infrarot Punktmuster der Kinect . . . . .	19
3.7. Versatz der Lichtpunkte im Punktmuster . . . . .	21
3.8. Links: Mit Kinect aufgenommenes RGB-Bild. Rechts: Das entsprechende Tiefenbild . . . . .	22
3.9. Architektur des OpenNI Frameworks . . . . .	23
3.10. Links: Kalibrierungspose unter OpenNI. Rechts: Aktivierte Handverfolgung . . . . .	24
3.11. Architektur des OpenNI Frameworks mit NITE . . . . .	25
4.1. Phasen von Gesten nach Kendon . . . . .	29
4.2. Taxonomie von Gesten in der MCI nach Karam . . . . .	31
4.3. Taxonomie der Handgesten für die MCI nach Pavlovic . . . . .	32
4.4. Tragbare Geräte zur Bewegungserfassung. Links: 3D-Maus. Rechts: Nintendo WiiMote . . . . .	33
4.5. Tragbare Anzüge zur Bewegungserfassung. Links: Elektro-mechanisches Exoskelett. Rechts: P5 Datenhandschuh . . . . .	34
4.6. Spezielle Kamerasysteme zur Bewegungserfassung . . . . .	35
4.7. Skizzierter Aufbau eines typischen Gestenerkennungssystems . . . . .	36
4.8. Verfahren zur Modellierung von Gesten nach Pavlovic . . . . .	38
4.9. Links: Symbolische Geste mit kulturell unterschiedlicher Bedeutung. Recht: Einsatz symbolischer Gesten bei Ländern . . . . .	41
4.10. iPointPresenter des Fraunhofer Instituts für Nachrichtentechnik . . . . .	42
4.11. Links: Funktionsprinzip des GestPoint Maestro3D Rechts: GestPoint Maestro3D in der Anwendung . . . . .	43
5.1. Links: Command Line Interface, Rechts: Graphical User Interface . . . . .	47



5.2. Apple iPhone 4S . . . . .	52
5.3. Multitouch Tisch Microsoft Surface . . . . .	53
5.4. Benutzeroberfläche der Xbox360 unter Verwendung der Kinect . . . . .	54
5.5. OCGM Metaphern nach George und Blake . . . . .	57
6.1. Vergleich von Größe und Anzahl der Interface-Elemente bei GUI (links) und NUI (rechts) . . . . .	61
6.2. Für die Konzeption verwendetes Farbklima . . . . .	62
6.3. Startoberfläche des Informationssystems für Studenten . . . . .	65
6.4. Selektion von Container „Aktuelles“ . . . . .	65
6.5. Überblick über die Container innerhalb des Containers „Aktuelles“ . . . . .	66
6.6. Selektion des Containers . . . . .	66
6.7. Zugriff auf das Objekt . . . . .	67
6.8. Selektion des Mensa-Containers . . . . .	68
6.9. Container innerhalb des Mensa-Containers . . . . .	68
6.10. Selektion des Objektes . . . . .	69
6.11. Interaktion mit dem Objekt . . . . .	69
7.1. Verwendete Systemarchitektur . . . . .	71
7.2. Anmeldebildschirm des Prototyps . . . . .	73
A.1. Selektion des Containers Personen. . . . .	ii
A.2. Bestätigung anhand einer Push-Geste. . . . .	ii
A.3. Container Personen: Die Darstellung der Objekte umfasst den Namen und ein Bild. . . . .	iii
A.4. Selektion eines Objektes: Weitergehende Inhalte des Objektes werden dargestellt. . . . .	iii
A.5. Selektion des Containers Räume. . . . .	iv
A.6. Bestätigung anhand einer Push-Geste. . . . .	iv
A.7. Container Räume: Anhand der Etagen wird eine Unterscheidung vor- genommen. . . . .	v
A.8. Container Räume: Die Darstellung der Objekte orientiert sich am Anwendungszweck. Eine angemessene Darstellung für Räume wäre z. B. ein Video, das den Weg zum Ziel beschreibt. . . . .	v
A.9. Selektion des Containers Schwarzes Brett. Eine Voransicht auf die hinterlegten Objekte wird angeboten. . . . .	vi
A.10. Selektion eines Objekts im Container Schwarzes Brett. . . . .	vi
A.11. Selektion des Containers Vorlesungen. . . . .	vii
A.12. Bestätigung anhand einer Push-Geste. . . . .	vii



A.13. Selektion der Container Bachelor, Master. . . . .	viii
A.14. Selektion des Container Studiengang. . . . .	viii
A.15. Bestätigung anhand einer Push-Geste. . . . .	ix
A.16. Selektion des Containers Semester. Daraufhin gelangt man zum Objekt, dem persönlichen Vorlesungsplan. . . . .	ix



## Verzeichnis der Listings

7.1. OpenNI XML Konfiguration . . . . .	72
7.2. Anmeldung am System . . . . .	73
7.3. Methode zur Selektion von Interface-Elementen . . . . .	75
7.4. Verwendung von Push-Gesten in Unity . . . . .	76



# 1. Einleitung

## 1.1. Problembeschreibung

Der Einsatz der Multitouch Technologie hat in den vergangenen Jahren immer weiter zugenommen. Endgeräte, wie das Apple iPad oder das Samsung Galaxy zeigen, wie sich innovative Benutzerschnittstellen an die Erfordernisse der Nutzer anpassen und immer mehr auf realitätsnahe Handlungsmuster setzen. Der Einsatz dieser, auf Handgesten basierenden, berührungsempfindlichen Systeme, ist jedoch nicht für jeden Anwendungsfall praktikabel. Es existieren Szenarien, in denen der Nutzer Berührungen vermeiden möchte. Dies kann zum Beispiel im häuslichen Umfeld oder aber in bestimmten Arbeitsumfeldern der Fall sein. In öffentlichen Einrichtungen besteht zudem die Gefahr, dass die Multitouch-Oberfläche beschmutzt oder gar beschädigt werden kann.

Um eine solche berührungsfreie Interaktion zwischen Nutzer und System auf der Basis von Gesten zu ermöglichen, existieren bereits eine Reihe von Möglichkeiten: Datenhandschuhe oder ähnliche nicht-optische Systeme übermitteln über Detektoren die Position der Hand und machen eine gestenbasierte Interaktion möglich. Der Nachteil dieser Systeme ist die zu tragende Hardware (z. B. Datenhandschuh), so dass sie für den Alltagsgebrauch ausgeschlossen ist. Optische Verfahren hingegen kommen ohne zusätzliche Hardware aus. Die Erfassung der Bewegung erfolgt hier durch die Verwendung spezieller Kamerasysteme, wobei unterschiedliche Ansätze wie 3D Stereo Rekonstruktion, Time of Flight oder Light Coding unterschieden werden müssen.

Der Einsatz von gestenbasierten Interfaces, unabhängig davon, ob sie auf berührungsempfindliche oder berührungsfreie Interaktion setzen, wird vor allem von der aktuellen Generation von Smartphones, Tablets und Videospielkonsolen (Nintendo Wii, Playstation Move, Microsoft Kinect) genutzt. Im Bereich der Desktop Systeme wurde diesem Thema bisher wenig Beachtung geschenkt, obwohl der eindeutige Nutzen bereits in mehreren Anwendungsfällen demonstriert wurde (Kollaborative Umgebungen, Informationsszenarien, Bildungsszenarien). Die Hauptprobleme liegen dabei zum einen in der Erstellung und Verarbeitung eines geeigneten Gesten-





### 1. Einleitung

---

Vokabulars, welches die zu erbringende Aufgabe des Nutzers möglichst eindeutig widerspiegelt. Zum anderen stellt die Konzeption des grafischen Interfaces eine Vielzahl an Herausforderungen an den Designer, da bekannte Interface-Metaphern wie z. B. WIMP (Windows, Icons, Menus, Pointer) nur bedingt für die gestenbasierte Eingabe geeignet sind.

Die Microsoft Kinect stellt in diesem Kontext eine kostengünstige Möglichkeit dar, berührungsfreie Interaktion zu ermöglichen. Durch die Öffnung der Treiber für den PC ergeben sich neue Möglichkeiten, die Kinect auch außerhalb der Spieledomäne nutzen zu können. Aktueller Gegenstand universitärer Forschung und kommerzieller Entwicklungen ist die Verwendung der Kinect als Eingabegerät in Anwendungsgebieten, wie Robotik oder Medizin. Mit ihren technischen Charakteristika eignet sich die Kinect jedoch vorrangig als Eingabegerät für die Steuerung gestenbasierter Interfaces.

## 1.2. Ziel der Arbeit

Ziel dieser Arbeit ist es, auf Grundlage von bekannten, gestenbasierten Interaktionskonzepten und Studien, die den Einsatz von berührungsfreien Interaktionstechniken evaluieren, den Mehrwert der gestenbasierten Interaktion auf Desktop-Systemen aufzuzeigen. Als Eingabegerät soll dabei die Microsoft Kinect verwendet werden. Mit dem zu erzeugenden Gesten-Vokabular sollen anfallende Aufgaben möglichst intuitiv erledigt werden können. Darüber hinaus besteht ein weiteres Ziel in der Gestaltung der Benutzeroberfläche, die es dem Benutzer ermöglicht direkt mit Inhalten zu interagieren. Die Ergebnisse sollen später anhand einer prototypischen Anwendung für ein interaktives Informationssystem demonstriert werden. Zusammengefasst soll die vorliegende Arbeit Antworten auf folgende Forschungsfragen liefern:

- Welche Möglichkeiten der Systeminteraktion bietet die Microsoft Kinect ?
- Welche Handgesten ermöglichen eine intuitive Interaktion mit dem System ?
- Welche Anforderungen bestehen bei der Gestaltung der Benutzeroberfläche für ein gestenbasiertes Interface ?



### 1.3. Abgrenzung der Arbeit

Ein Großteil aktueller Forschungen im Bereich gestenbasierter Interfaces beschäftigt sich mit den technischen Möglichkeiten der Eingabe. Der Fokus liegt meist in der Erfassung und Verarbeitung von Gesten, unterschiedlicher Eingabegeräte. Die Eingabemodalität ist jedoch nur eine Facette von Natural User Interfaces. Diese Arbeit zielt darauf ab, Natural User Interfaces bei Desktop Systemen unter Verwendung der Microsoft Kinect zu untersuchen. An dieser Stelle sollen nun die Grenzen der Arbeit festgelegt werden. Zusammengefasst behandelt diese Arbeit keine Verwendung von Zeichensprache und keine Multimodalität. Des Weiteren wird ausschließlich die berührungsfreie Interaktion untersucht. Die Alternative der berührungssensitiven Eingabe, wird in dieser Arbeit nur kurz angesprochen.

Große Teile aktueller Forschungen konzentrieren sich auf die Verwendung von Zeichensprache. Dabei wird unter anderem die Verwendung von Zeichensprache bei Lernprogrammen für Kinder behandelt [30]. Weitere Arbeiten in diesem Gebiet beschäftigen sich mit der Erfassung und Segmentierung menschlicher Bewegungen und Gestenerkennung [63]. Des Weiteren wurden auch erste Versuche unternommen die Microsoft Kinect als System für die Erkennung von Zeichensprache einzusetzen, wie von LANG beschrieben. Die Verwendung von Zeichensprache ist eine sehr spezielle Form der gestenbasierten Eingabe. Zeichensprache kann als eine eigene Interaktionsform einer speziellen Zielgruppe angesehen werden und wird für diese Arbeit nicht weiter verfolgt.

Das von Richard A. Bolt erste entwickelte, gestenbasierte Interface „Put That There“ [6], zeigte bereits damals das große Potential im Bereich multimodaler Anwendungen. Durch eine Kombination von Zeigegesten und Sprachkommandos ist es dem Nutzer bei dieser Anwendung möglich, geometrische Formen auf einer Leinwand zu erzeugen und zu manipulieren. Die Kombination mehrerer natürlicher Eingabemodalitäten bietet sich für ein Natural User Interface an. Aufgrund des großen Themengebiets beschränkt sich die vorliegende Arbeit auf die Eingabe natürlicher Handgesten. Die natürliche Eingabe über Sprache wird jedoch kurz angesprochen.

Die in dieser Arbeit verwendete Hardware und Software bietet grundsätzlich die Möglichkeit, Gesten unter Berücksichtigung der meisten Körperteile (Kopf, Hände, Füße) zu definieren und anschließend zu erfassen. Die in dieser Arbeit entwickelte, prototypische Anwendung soll jedoch ausschließlich den Einsatz von Handgesten unterstützen. Dem interessierten Leser sei die Arbeit von MAES U. A. empfohlen. Sie beschäftigen sich mit der Konzeption und Entwicklung eines Systems, das die gestenbasierte Interaktion mit verschiedenen Software Agenten ermöglicht. Des Weiteren



werden berührungssensitive Eingabesysteme wie z. B. Tablet-PCs oder Smartphones nur kurz angesprochen.

## 1.4. Aufbau der Arbeit

Zu Beginn der Arbeit wird in Abschnitt 1.1 die Problembeschreibung vorgestellt. In Abschnitt 1.2 werden die Forschungsfragen für diese Arbeit definiert. Eine Abgrenzung der Arbeit wird in Abschnitt 1.3 vorgenommen.

Kapitel 2 gibt einen Überblick über ausgewählte Arbeiten, die Ansätze zur Beantwortung der in dieser Arbeit definierten Forschungsfragen liefern. Das Kapitel ist in drei unterschiedliche Abschnitte unterteilt. Abschnitt 2.1 gibt einen Überblick über Arbeiten, die sich mit der gestenbasierten Eingabe beschäftigen. Im Fokus stehen sowohl technische als auch allgemeine Fragestellungen der gestenbasierten Interaktion mit einem System. In Abschnitt 2.2 werden Arbeiten vorgestellt, die sich mit der Gestaltung des Interfaces bei gestenbasierten Systemen beschäftigen. Darüber hinaus werden Untersuchungen zur Ergonomie und Usability von gestenbasierten Systemen beschrieben. Abschließend werden in Abschnitt 2.3 bedeutende Arbeiten vorgestellt, die die Entwicklung von gestenbasierten Systemen geprägt haben.

Im Kapitel 3 werden die Grundlagen für diese Arbeit beschrieben. Dazu wird in Abschnitt 3.1 ein kurzer Überblick über das interdisziplinäre Forschungsgebiet der Mensch Computer Interaktion gegeben. Im darauffolgenden Abschnitt 3.2 wird das Ziel beschrieben, die Interaktion zwischen Mensch und Computer auf bereits erlernte Fähigkeiten des Menschen auszulegen. Aufbauend auf diesem Ziel werden im Abschnitt 3.3 die für diese Arbeit wichtigen Begriffe *Natürlichkeit* und *Intuitivität* definiert. Im Hinblick auf die Konzeption und Realisierung eines gestenbasierten Interface, werden in den Abschnitten 3.4 und 3.5 die in dieser Arbeit verwendete Hardware und Software beschrieben.

Kapitel 4 befasst sich mit der gestenbasierten Eingabe. Zu Beginn wird in Abschnitt 4.1 der Begriff *Gestik* aus unterschiedlichen Perspektiven definiert. Danach werden in Abschnitt 4.2 Gesten anhand ihrer Charakteristika in Gestentypen unterteilt und in Taxonomien eingeordnet. Im darauf folgenden Abschnitt 4.3 werden Möglichkeiten zur Erfassung von menschlichen Bewegungsabläufen vorgestellt. Die aufgezeigten Möglichkeiten stellen die Grundlage für eine gestenbasierte Interaktion dar. Ausgehend von einem allgemeinen Verständnis über die Erfassung von Bewegungen, befasst sich der nächste Abschnitt 4.4 mit dem Erfassungsvorgang von Gesten. Dazu werden die typischen Prozessstufen eines Gestenerkennungssystems vorgestellt. In Abschnitt 4.4.1 wird die Modellierung von Gesten beschrieben. Danach befasst sich



## 1. Einleitung

---

Abschnitt 4.4.2 mit der Segmentierung. Die Merkmalsbestimmung wird in Abschnitt 4.4.3 behandelt. Die Klassifizierung von Gesten wird in Abschnitt 4.4.4 beschrieben. Danach wird in Kapitel 4 das Thema Interkulturalität von Gesten in Abschnitt 4.5 untersucht. Der folgende Abschnitt 4.6 stellt zwei kommerzielle, gestenbasierte Systeme vor. Abschließend wird in Abschnitt 4.7 ein Zwischenfazit gezogen.

In Kapitel 5 werden gestenbasierte Interfaces behandelt. Aufgrund der Forderung nach einer natürlichen Interaktion wird das Konzept der Natural User Interfaces (NUI) vorgestellt. Zu Beginn werden in Abschnitt 5.1 Ansätze zur Definition gegenübergestellt und diskutiert. Danach wird in Abschnitt 5.2 die zeitliche Entwicklung der Benutzerschnittstelle beschrieben. Abschnitt 5.3 stellt die Eigenschaften eines NUI vor. Darauf aufbauend werden in Abschnitt 5.4 Produkte vorgestellt, die das Konzept der NUI umsetzen. Abschnitt 5.5 befasst sich anschließend mit der Ergonomie von berührungsfreien Eingabesystemen. Aufgrund des Paradigmenwechsels von GUI zu NUI müssen neue Metaphern gefunden werden über die ein Benutzer interagieren kann. Ein solcher Ansatz wird in Abschnitt 5.6 vorgestellt. Abgeschlossen wird das Kapitel in Abschnitt 5.7 mit einem Zwischenfazit.

Kapitel 6 befasst sich mit der Konzeption eines NUI. Zu Beginn wird in diesem Kapitel die Vorgehensweise in Abschnitt 6.1 vorgestellt. Darauf aufbauend wird in den folgenden Abschnitten 6.1.1 und 6.1.2 die Gestaltung der Interface-Elemente beschrieben. Danach wird in Abschnitt 6.1.3 das Verhalten der Interface-Elemente erklärt. Die Farbgebung wird in Abschnitt 6.1.4 behandelt. Nachdem das Vorgehen bei der Gestaltung der Benutzeroberfläche beschrieben wurde, wird in Abschnitt 6.2 das verwendete Gestenvokabular festgelegt. Abgeschlossen wird das Kapitel in Abschnitt 6.3. Dort wird auf Grundlage der festgelegten Vorgehensweise ein NUI für ein Studenteninformationssystem vorgestellt.

Kapitel 7 behandelt abschließend die prototypische Realisierung eines NUI. Zu Beginn des Kapitels wird in Abschnitt 7.1 die verwendete Systemarchitektur vorgestellt und das Zusammenspiel der einzelnen Komponenten erklärt. Die darauffolgenden Kapitel befassen sich mit der Umsetzung der Grundfunktionalitäten. Dazu wird in Abschnitt 7.2 der Anmeldevorgang am System beschrieben. Darauf aufbauend befasst sich der nächste Abschnitt 7.3 mit der Erstellung des Handcursors und der Verwendung von Zeigegesten. Danach wird in Abschnitt 7.4 die Selektion von Interface-Elementen in Unity3D erklärt. Im abschließenden Abschnitt 7.5 wird die Möglichkeit zur Bestätigung anhand von Push-Gesten beschrieben.



## 2. Verwandte Arbeiten

Dieses Kapitel umfasst eine Auswahl an verwandten Arbeiten, die sich mit den für diese Arbeit relevanten Themenschwerpunkten Verwendung von Gesten, gestenbasierte Interfaces und gestenbasierte Anwendungen befassen. Im Abschnitt 2.1 werden Arbeiten vorgestellt, die sich vorwiegend mit dem Erfassungsvorgang, der Klassifikation und der Evaluation von gestenbasierten Systemen auseinandersetzen. Danach werden im Abschnitt 2.2 Arbeiten aufgezeigt, die sich mit den Problemen und Möglichkeiten bei der Konzeption gestenbasierter Interfaces, unterschiedlicher Anwendungsgebiete, befassen haben. Der letzte Abschnitt, 2.3, stellt wichtige Arbeiten vor, die die zeitliche Entwicklung von gestenbasierten Anwendungen aufzeigen. Ziel dieses Kapitels ist es, dem Leser einen Überblick über verwandte Arbeiten zu geben, die für die Umsetzung einer gestenbasierten Steuerung im Rahmen dieser Arbeit eine hohe Relevanz aufweisen.

### 2.1. Gestenbasierte Interaktion

Die Arbeit von KARAM [38] befasst sich mit Entwicklung und Einsatz eines Frameworks, das als Werkzeug für die Erforschung von gestenbasierter Interaktion dienen soll. Zu diesem Zweck werden vier Kategorien definiert und deren Einfluss auf das Benutzerverhalten untersucht. Die Kategorien umfassen dabei physische Gesten, Eingabegeräte, Ausgabegeräte und *user goals*<sup>1</sup>. Die Versuchsergebnisse vermitteln ein theoretisches Verständnis über gestenbasierte Interaktion im Kontext der *Mensch Computer Interaktion* (MCI).

Eine weitere Arbeit, die den Einsatz von Handgesten untersucht, stammt von EPPS U. A. [17]. Der Versuch besteht dabei in der Beobachtung von Probanden, die durch den Einsatz von nicht vorgegebenen Handgesten verschiedene Aufgaben an einem gestenbasierten Interface lösen müssen. Die Arbeit zielt darauf ab, Vorlieben für Handformen zu identifizieren. Die Erkenntnisse über die Vorlieben für einzelne Handformen sind für die Aufstellung eines geeigneten Gestenvokabulars wichtig.

---

<sup>1</sup>Vgl. [14], S. 125



## 2. Verwandte Arbeiten

---

Die Arbeit „*Comparison of techniques for mixed-space collaborative navigation*“ von [STAFFORD U. A.](#) [79] befasst sich mit einer Gegenüberstellung unterschiedlicher Interaktionsformen zur Navigation in virtuellen Welten. Im Mittelpunkt der Arbeit steht die Untersuchung von Effizienz und Effektivität zur Aufgabenbewältigung unter Verwendung von Handgesten und Maus. [STAFFORD U. A.](#) kommen dabei zu dem Ergebnis, dass die Navigation unter Benutzung von Handgesten minimal effizienter ist, als mit der Maus. Des Weiteren halten [STAFFORD U. A.](#) fest, dass Handgesten ausdrucksstärker sind und in Situationen in denen keine verbale Kommunikation möglich ist, zu einer schnelleren Aufgabenbewältigung führen.

Viele Forschungen beschäftigen sich zudem mit dem Erfassungs- und Verarbeitungsvorgang von Gesten. [PAVLOVIC U. A.](#) geben in ihrer Arbeit einen umfassenden Überblick über die visuelle Interpretation von Handgesten im Kontext MCI. Zu diesem Zweck unterteilen sie den Erfassungsvorgang von Gesten in vier aufeinander folgende Phasen: Modellierung von Gesten, Analyse von Gesten, Erfassung von Gesten und gestenbasierte Systeme und Anwendungen [64]. [PAVLOVIC U. A.](#) beschreiben den Ablauf der unterschiedlichen Phasen und gehen auf Möglichkeiten und Schwierigkeiten für die MCI ein .

Weitere Arbeiten, die sich explizit mit dem Erfassungsvorgang von Gesten beschäftigen stammen von [CAPO U. A.](#) und [TANG](#). Sie stellen in ihrer Arbeit einen Ansatz zur Erkennung von Gesten durch Einsatz eines optischen *Motion Capturing*<sup>2</sup> Systems vor. Die Erkennungsgenauigkeit des vorgestellten Ansatzes liefert dabei 87% korrekte Ergebnisse [10]. Eine ähnliche Untersuchung stammt von [TANG](#). Als Eingabegerät wird bei seinen Versuchen die Microsoft Kinect verwendet. Der Ansatz zur Erfassung von Gesten beruht auf der Analyse der vorliegenden RGB- und Tiefenbilder. Anhand des Einsatzes eines *SURF Descriptors* werden die Tiefenbilder analysiert. Laut [TANG](#) beträgt die Erkennungsgenauigkeit des Systems zwischen 92% und 96% [86].

Die Wahl einer geeigneten Klassifikation von Gesten wird in den Arbeiten von [TRIGO UND PELLEGRINO](#) und [STERN U. A.](#) behandelt. [TRIGO UND PELLEGRINO](#) stellen in ihrer Arbeit verschiedene Ansätze zur Merkmalsextraktion von Handgesten vor. Das von ihnen vorgeschlagene Verfahren zur Merkmalsextraktion beruht, ähnlich wie in [86], auf der Analyse der Handformen. Die Klassifikation anhand der extrahierten Merkmale erfolgt auf Grundlage der analysierten Formen [88].

Die Arbeit von [STERN U. A.](#) beschreibt eine Methode zur Selektion von relevanten Gesten für ein Gestenvokabular. [STERN U. A.](#) beziehen in ihrem analytischen Ansatz mehrere Faktoren ein. Der Ansatz berücksichtigt sowohl psycho-physiologische

---

<sup>2</sup>Vgl. Unter Motion Capture (engl. Bewegungserfassung) versteht man den Prozess, Bewegungen mit Hilfe von Sensoren aufzunehmen und diese in eine für den Computer lesbare, verarbeitbare Form zu übersetzen



als auch algorithmische Faktoren der Gestenerkennung [82]. In einem ersten Schritt wird die Bildung einer Teilmenge von Gesten vorgenommen, die eine minimale Erfassungsgenauigkeit voraussetzen. Im zweiten Schritt werden Gesten zu einer Aufgabe zugeordnet. Die Entscheidung, welche der möglichen Gesten für das Gestenvokabular in Frage kommt, wird letztlich anhand einer *Pareto-Verteilung*<sup>3</sup> entschieden [82].

## 2.2. Gestenbasierte Interfaces

Die in diesem Abschnitt vorgestellten Arbeiten befassen sich mit den Möglichkeiten und Problemen im Kontext der gestenbasierten Interaktion mit grafischen Benutzeroberflächen. [MOESLUND U. A.](#) untersuchen im Rahmen ihrer Arbeit die gestenbasierte Navigation in virtuellen Welten [55]. Im Fokus steht die Erprobung der Genauigkeit von Zeigegesten. [MOESLUND U. A.](#) stellen abschließend fest, dass der Einsatz von visuellem Feedback großen Einfluss auf die Genauigkeit hat. Das visuelle Feedback für den Benutzer wird dabei über eine helle Linie dargestellt, die die Zeigerichtung andeutet.

Die Arbeit von [GRANDHI U. A.](#) behandelt die berührungsfreie, gestenbasierte Interaktion mit Systemen. Anhand der Analyse von Videos, die den Einsatz gestenbasierter Eingaben in unterschiedlichen Anwendungsgebieten zeigen, gehen [GRANDHI U. A.](#) der Frage nach, welche Art von Gesten am intuitivsten und natürlichsten ist. [HOFMEESTER UND WIXON](#) beschreiben in ihrer Arbeit die Konzeption eines gestenbasierten Interfaces unter besonderer Berücksichtigung von Metaphern. Ziel der Untersuchung ist die Beantwortung dreier zentraler Forschungsfragen: Funktionieren die spezifischen Mechaniken des Interfaces für den Benutzer? Kommunizieren die Mechaniken des Interfaces die Metapher? Wie wird der Benutzer von der Metapher angesprochen? [29].

Die Konzeption geeigneter Menüstrukturen für gestenbasierte Benutzereingaben werden von [LENMAN U. A.](#) untersucht. Ausgehend von der Wahl der zu Verfügung stehenden Kommandos, werden kognitive Aspekte, artikulatorische Aspekte und technologische Aspekte untersucht [45]. Abschließend wird ein Prototyp des Kommandomenüs zur Steuerung von Fernsehgeräten vorgestellt.

Die Arbeit „*On the usability of gesture interfaces in virtual reality environments*“ [9] von [CABRAL U. A.](#) befasst sich mit Problemen im Bereich der Usability bei gestenbasierten Systemen. Zur Untersuchung werden drei verschiedene Szenarios betrachtet: Gestenbasierte Eingaben anstelle von Maus-Eingaben, gestenbasierte Eingabe als

---

<sup>3</sup>Vgl. [SCHLITTEGEN](#), S. 216-218



Navigationswerkzeug und gestenbasierte Eingaben als Visualisierungswerkzeug [9]. CABRAL U. A. stellen fest, dass die Nachteile des Systems in technischen Einschränkungen sowie in Ermüdungserscheinungen der Benutzer liegen. Laut CABRAL U. A. stellen gestenbasierte Interfaces eine einfach zu lernende und zu benutzende Möglichkeit dar, die sich vorwiegend für den kurzzeitigen Einsatz anbietet.

Die Arbeit von JOHNSON U. A. befasst sich mit der konkreten Anwendung eines gestenbasierten Interfaces in der interventionellen Radiologie. Das Interface wird dabei zur Navigation und Analyse von digitalen Aufnahmen verwendet. Der Vorteil der gestenbasierten Interaktion besteht in Möglichkeit unter aseptischen Bedingungen arbeiten zu können [37]. Probleme bestehen dagegen in der räumlichen Flexibilität des Systems und dem kollaborativem Zugriff auf das System [37].

## 2.3. Gestenbasierte Anwendungen

Der Gedanke, Gesten als natürliche, intuitive Methode für die MCI einzusetzen, ist nicht neu. Die ersten Gesten, die der Interaktion mit dem Computer dienten, wurden von SUTHERLAND in seiner Arbeit „*Sketchpad: A man-machine graphical communication system*“ vorgestellt. Durch einen Lichtgriffel war es dem Benutzer möglich, gestenbasierte Eingaben auf einem CRT Monitor zu tätigen. Die Eingaben wurden zur Erstellung und Manipulation von grafischen Objekten genutzt [84]. Diese Form der gestenbasierten Eingabe wird auch heute noch vereinzelt für z. B. PDA's verwendet.

1980 stellte BOLT mit „*Put-That-There*“ ein System vor, das unter Verwendung von Handgesten und Sprache die Manipulation von virtuellen Objekten an einem großen Display zuließ. Virtuelle Objekte konnten über Zeigegesten selektiert und anschließend über Spracheingaben gesteuert bzw. manipuliert werden.

Ferner entwickelte BUXTON U. A. ein gestenbasiertes Interfaces für einen Tablet-PC. Das Gestenvokabular orientierte sich an den Eingaben, die Benutzer mit Bleistift auf einem Skizzenblock machten. Ähnlich zur Arbeit von SUTHERLAND wurden die Eingaben zur Erstellung und Manipulation von grafischen Objekten genutzt [8].

Die 1989 fortgeschrittenen Möglichkeiten zur Erfassung von Bewegungsabläufen und anschließender Übersetzung in computerlesbare Form, verwendete STURMAN U. A. für seine Arbeit „*Hands-on interaction with virtual environments*“ . Über einen Datenhandschuh wurden Handbewegungen des Benutzers erfasst und verarbeitet. Untersucht wurde die gestenbasierte Interaktion für die Navigation in virtuellen Welten [83].





### *2. Verwandte Arbeiten*

---

[FREEMAN UND WEISSMAN](#) stellten [1995](#) in ihrer Arbeit ein gestenbasiertes System zur Steuerung von Fernsehgeräten vor. Zur Erfassung der Handbewegungen verwendeten [FREEMAN UND WEISSMAN](#) ein spezielles Kamerasystem. Die Untersuchung umfasste zwei Schwerpunkte: die möglichst intuitive Kommunikation zwischen Mensch und Maschine und die verwendeten Erfassungstechniken für den konkreten Anwendungsfall [\[20\]](#).



## 3. Grundlagen

Im Rahmen der Grundlagen sollen wesentliche Aspekte für den anschließenden Hauptteil vorgestellt werden. Der erste Abschnitt 3.1 gibt einen Überblick über das Forschungsfeld, in das diese Arbeit fällt. Im Zuge dessen wird ein Modell der Interaktion vorgestellt. Im anschließenden Abschnitt 3.2 werden grundlegende Erkenntnisse zusammengefasst, die die gestenbasierte Interaktion im Kontext der MCI evaluieren. Danach werden in Abschnitt 3.3 die für diese Arbeit wichtigen Begriffe Natürlichkeit und Intuitiv definiert. Die letzten Abschnitte 3.4 und 3.5 beschäftigen sich mit Hardware und Software, die zur Realisierung eines gestenbasierten Interfaces in dieser Arbeit eingesetzt werden.

### 3.1. Mensch Computer Interaktion

Die MCI beschäftigt sich im Allgemeinen mit der Beziehung zwischen Menschen und ihrer technologischen Umgebung [7]. Untersuchungen auf diesem Gebiet befassen sich mit Interaktionsmöglichkeiten, die den Austausch von Informationen zwischen Mensch und Computer ermöglichen. Die Untersuchung von Interaktionsmöglichkeiten erfordert dabei ein interdisziplinäres Wissen. Soziale Aspekte spielen ebenso eine Rolle wie Kenntnisse aus den Bereichen Design und Technik (Vgl. Abbildung 3.1).

Das grundlegende Verständnis von dem Zusammenspiel dieser drei Forschungsfelder hat dazu geführt, dass die Art und Weise, wie Menschen am Computer Aufgaben lösen, besser verstanden werden kann. NORMAN modelliert die Interaktion zwischen Mensch und Computer in seinem Interaktionsmodell *seven stages of action* (Vgl. Abbildung 3.2). Das Modell umfasst Prozessstufen, die ein Anwender bei der Interaktion mit einem interaktiven System durchläuft. Die Aussage seines Modells ist, dass menschliche Ziele (*goals*) durch kognitive Prozesse gebildet werden und dann in der mentalen Welt vorliegen. Da sich Ziele jedoch immer auf die angestrebten Zustände eines Systems beziehen, orientiert sich die Zielerreichung an den evaluierbaren Zielzuständen fest, die in der realen Welt vorliegen. NORMAN beschreibt



### 3. Grundlagen

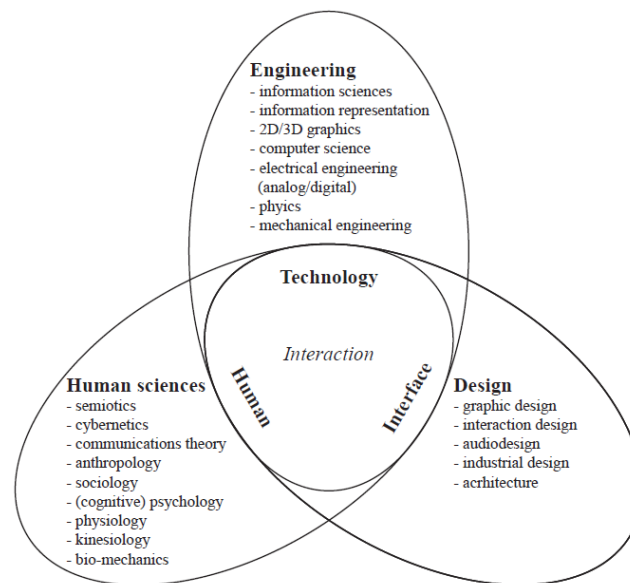


Abbildung 3.1.: Das interdisziplinäre Forschungsfeld der Mensch Computer Interaktion nach BONGERS [7]

in seinem Modell die Diskrepanz zwischen den mental vorliegenden Zielen und den verfügbaren Aktionen und Informationen auf technologischer Ebene als *gulf of execution*. Für eine erfolgreiche Interaktion muss demnach eine Brücke über diese Kluft „erbaut“ werden, um die mentale Welt mit der physikalischen Welt zu verbinden. Ähnlich verhält es sich mit dem *gulf of evaluation*. Dieser beschreibt die Diskrepanz zwischen Systemstatus und den Erwartungen des Anwenders [60]. Für eine erfolgreiche Interaktion gilt es auch hier, eine Brücke zwischen der physikalischen und der mentalen Welt zu „errichten“.

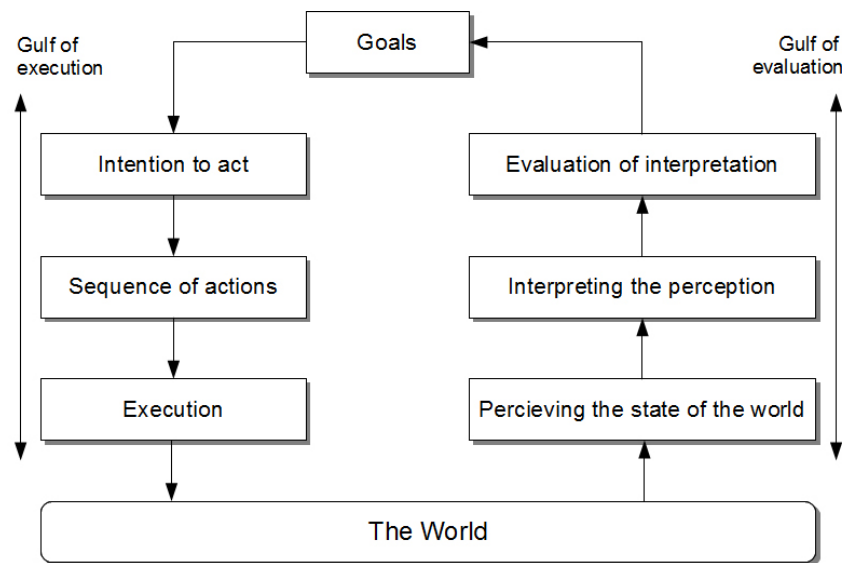
Gegenstand dieser Arbeit ist der intuitive Gebrauch von Handgesten zur Interaktion. Bezogen auf das Modell von NORMAN bedeutet dies, dass durch den Gebrauch von intuitiven Gesten versucht wird, die Diskrepanz des *gulf of execution* zu minimieren.

## 3.2. Gesten und Menschen

Schon in frühem Alter beginnen Kinder mittels Gestik und Vokalisationen zu kommunizieren. Die Kommunikation über Wörter bzw. den Sprachgebrauch wird erst später von ihnen erlernt [48]. Neben Vokalisationen stellt Gestik demnach die erste vom Menschen erworbene Kommunikationsmöglichkeit in der Beziehung Mensch-Mensch dar. Tänze und Zeremonien unterschiedlichster Kulturen sind ein weiteres



## 3. Grundlagen

Abbildung 3.2.: Modell der Interaktion nach [NORMAN](#) [59]

Beispiel dafür, wie tief Gestik in der Kommunikation zwischen Individuen verwurzelt ist [33]. Gestik wird häufig unbewusst als sprachunterstützendes Kommunikationsmittel verwendet. Sogar beim Telefonieren werden Gesten verwendet, da sie dem Sprecher dabei helfen das Gesprochene zu strukturieren [51]. Nach [IVERSON UND GOLDIN-MEADOW](#) wird Gestik auch von geburtsblinden Menschen eingesetzt. Die verwendeten Gesten unterscheiden sich dabei nicht von denen sehender Menschen [34]. Des Weiteren wird Gestik häufig angewendet, wenn es um die Beschreibung bzw. die Manipulation von Objekten geht.

Die MCI beschäftigt sich mit Möglichkeiten, die Interaktion zwischen Mensch und Computer zu verbessern [59]. Studien belegen, dass sich Benutzer dann wohl fühlen, wenn die Kommunikation mit dem Computer auf möglichst intuitive und natürliche Weise geschieht [76]. Diese Forderung kann jedoch nur erfüllt werden, wenn Entwickler die physischen und mentalen Anforderungen des Benutzers verstehen, wie [DIX U. A.](#) feststellen. Solche Systeme, die bei der Darstellung und der Interaktion auf bereits erlernte Fähigkeiten des Benutzers zurückgreifen, bezeichnet man als *Natural User Interfaces*. Bevor in Kapitel 5 konkret auf NUI Bezug genommen wird, gilt es die Begriffe *Natürlichkeit* und *Intuitiv* für diese Arbeit zu definieren.



### 3.3. Natürliches und intuitives Handeln

Der Begriff Natürlichkeit bedeutet in seinem Ursprung, dass etwas in unveränderter Form in der Natur vorkommt, ohne dass es vom Menschen verändert wurde. Des Weiteren wird der Begriff Natürlichkeit für etwas verwendet, was von Geburt an vorhanden ist. Darunter fallen Eigenschaften wie sehen, fühlen, schmecken oder riechen. Im Laufe eines Lebens eignen sich Menschen eine Reihe von Fähigkeiten an. Diese müssen zuerst erlernt werden. Ein Kind muss Fähigkeiten wie Laufen, Springen, Reden erst erlernen. Einmal erlernte Fähigkeiten werden vom Menschen dann zunehmend als natürliche Handlungen empfunden. William Buxton definiert natürliche Handlungen als:

„...skills that we have aquired through a lifetime of living in the world.“ <sup>4</sup>

Um den Begriff intuitives Handeln zu definieren, bedarf es der Vorstellung eines weiteren Konzeptes. Das Konzept der *mentalen Modelle*. Mentale Modelle werden vorrangig in der Kognitionspsychologie sowie in der MCI verwendet. In der Kognitionspsychologie bezeichnen mentale Modelle Wissensrepräsentationen zu Ideen, Konzepten, Handlungen und Problemlösungen. In der MCI bezeichnen mentale Modelle die Vorstellung eines Benutzers über ein technisches System. Die Definition beider Disziplinen fokussiert sich auf unterschiedliche Merkmale mentaler Wissensrepräsentationen. Im Rahmen der Definition des Begriffs Intuitiv wird auf den kognitionspsychologischen Ansatz Bezug genommen. Nach HENSELER ist intuitives Handeln die wichtigste Facette von natürlichem Handeln:

„Eine Sache intuitiv bedienen oder benutzen zu können bedeutet, seine mentalen Modelle für etwas einsetzen zu können.“ [28]

Für die Nutzung eines Systems bedeutet dies, dass intuitives Handeln immer dann gegeben ist, wenn die mentalen Modelle des Benutzers mit den funktionalen Modellen des Systems deckungsgleich sind. Je deckungsgleicher diese beiden Modelle sind, desto intuitiver ist das System. Um diesen Sachverhalt zu verdeutlichen, soll folgendes Beispielszenario dienen: Ein Benutzer möchte ein System verwenden, kann dieses jedoch erst durch Studium des Handbuchs bedienen. In diesem Fall ist das mentale Modell des Benutzers nicht ausreichend um das System zu bedienen. Der Benutzer ist nun aufgefordert sein mentales Modell anhand des Handbuchs, welches

---

<sup>4</sup>Website: <http://www.billbuxton.com/>



### 3. Grundlagen

das mentale Modell des Entwicklers widerspiegelt, zu „überarbeiten“. Erst nach der Überarbeitung seines mentalen Modells ist er in der Lage, das System zu bedienen. Dies entspricht jedoch einem wenig intuitiven Verhalten.

Für die Gestaltung *intuitiver Systeme* ist es daher wichtig zu verstehen, wie Benutzer in bestimmten Anwendungskontexten agieren. Es besteht demnach die Notwendigkeit nach einem Modell, welches die natürlichen und intuitiven Handlungen von Menschen beschreibt. Ein solches allgemeingültiges Handlungsmodell wurde von HENSELER auf Grundlage kognitionsergonomischer Kriterien entwickelt [27]. Das OSIT Handlungsmodell ist ein Akronym, dass sich aus den Worten Orientieren, Selektieren, Informieren und Transagieren zusammensetzt und wird in Abbildung 3.3 dargestellt.

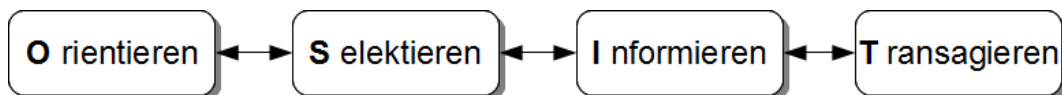


Abbildung 3.3.: OSIT Handlungsmodell nach HENSELER [27]

Der erste Schritt im OSIT-Handlungsmodell ist die Orientierung. Orientierung bedeutet einen Überblick über die Dinge zu erlangen. Nach der Orientierung, welche die Position bestimmt, erfolgt die Selektion in der bestimmte „Objekt“ aus dem näheren Umfeld ausgewählt werden. Nach erfolgter Selektion wird sich im Detail informiert. Dies bedeutet eine Fokussierung auf das selektierte „Objekt“. Die Detailansicht ermöglicht, dass mehr über das „Objekt“ in Erfahrung gebracht werden kann. Nachdem sich informiert wurde, wird das „Objekt“ verwendet oder in einen anderen Zustand überführt. Es findet eine Transaktion statt [27].

Das OSIT-Handlungsmodell gilt sowohl für menschliche Handlungen in der realen, als auch in der virtuellen Welt. Das Modell ist unabhängig von der kulturellen Herkunft und Alter gültig.

### 3.4. Microsoft Kinect

2009 stellte Microsoft auf der Spielemesse E3<sup>5</sup> erstmals seine neue Bewegungssteuerung Kinect für die Konsole Xbox360 vor. Das unter dem Projektnamen *Natal* in Kooperation mit dem Hersteller Primesense<sup>6</sup> entwickelte Eingabegerät sollte das

<sup>5</sup>Electronic Entertainment Expo, Website: <http://www.e3expo.com/>

<sup>6</sup>Website: <http://www.primesense.com/>



### 3. Grundlagen

---

Gegenstück zu den Konkurrenzprodukten aus dem Hause Sony und Nintendo darstellen. Microsoft reagierte dabei in erster Linie auf den großen Erfolg, den Nintendo mit seiner Spielekonsole Wii erzielt hatte. Das Alleinstellungsmerkmal der Wii war dabei ein neuartiger Controller, der über einen integrierten Bewegungssensor die Position und Bewegung des Controllers in Bewegungen des virtuellen Avatars umsetzte. Im Gegensatz zur Konkurrenz verfolgte Microsoft mit der Kinect den Ansatz, auf einen Controller, wie er bei Wii und Playstation zum Einsatz kam, zu verzichten. Die Bewegungssteuerung erfolgt bei der Kinect über eine Kombination aus Infrarot-Kamera und Infrarot-Projektor, welche die Bewegungen des Benutzers auf die Bewegungen des virtuellen Avatars umsetzt.

Nach Veröffentlichung der Kinect im November 2010 dauerte es nicht lange, bis sich unabhängige Entwickler damit befassten, Gerätetreiber für den PC zu entwickeln. Die Kinect war von Microsoft ursprünglich exklusiv für die Spielekonsole Xbox360 vorgesehen. Der Fokus lag dabei auf der Steuerung von Spielen. Die Verwendung am PC machte es Entwicklern möglich, die Kinect auch außerhalb der Spieledomäne zu nutzen. Die Community von Entwicklern wuchs nach Veröffentlichung der ersten Gerätetreiber schnell an, auch weil Primesense mit OpenNI ein unabhängiges Framework für die Entwicklung vorstellte. Microsoft reagierte auf diese Entwicklung und brachte im Juni 2011 ein eigenes Software Development Kit (SDK) heraus.

Dieses Kapitel befasst sich mit der Microsoft Kinect, die für die vorliegende Arbeit als Eingabegerät verwendet wird. Beginnend mit einer Beschreibung der technischen Charakteristika der Kinect in Abschnitt 3.4.1 wird im nächsten Abschnitt 3.4.2 die Funktionsweise der Kinect erklärt. Abgeschlossen wird das Kapitel in Abschnitt 3.4.3 mit einer Beschreibung der Stärken und Schwächen der Kinect sowie einer Begründung, warum die Kinect als Eingabegerät für die gestenbasierte Interaktion genutzt wird.

#### 3.4.1. Hardware der Kinect

Dieser Abschnitt befasst sich mit den technischen Charakteristika der Kinect. Dazu werden die einzelnen Komponenten der Kinect beschrieben sowie deren technische Eckdaten genannt. Die Beschreibung beschränkt sich dabei auf die wesentlichen Komponenten der Kinect. Dem interessierten Leser sei die Untersuchung „Kinect Teardown“ von Prof. Rahul Mangharam<sup>7</sup> empfohlen.

---

<sup>7</sup>Website: <http://www.seas.upenn.edu/~rahulm/Shared/Slides/Teardown-01-Kinect.pdf>



### 3. Grundlagen

Die Kinect besteht im Wesentlichen aus folgenden fünf Kernkomponenten, welche in Abbildung 3.4 dargestellt werden:

- RGB-Kamera
- Infrarot-Projektor
- Infrarot-Kamera
- Multi-Array-Mikrofon
- Neigungsmotor

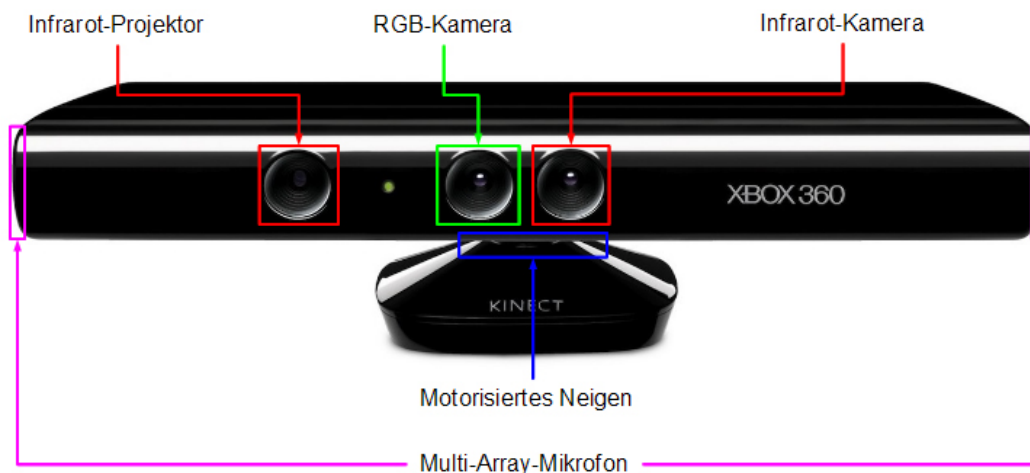


Abbildung 3.4.: Kernkomponenten der Microsoft Kinect<sup>8</sup>

Die **RGB-Kamera** ist in der Mitte der Kinect-Sensorleiste angebracht. Der als System on Chip (SoC) ausgeführte CMOS Sensor liefert bei einer Bildwechselfrequenz von 30 fps eine VGA Auflösung mit 640\*512 Pixel. Die Bildinformationen werden dabei mit einer Farbtiefe von 16 bit übertragen. Der Nachteil der RGB-Kamera ist die geringe Bildwechselfrequenz. Die Verfolgung schneller Bewegungen ist dadurch nicht sehr präzise. Das Sichtfeld der RGB-Kamera umfasst 57° in der Horizontalen und 43° in der Vertikalen. Die Aufgabe des **Infrarot-Projektors** besteht in der Aussendung eines speziellen Lichtmusters. Das Infrarotlicht wird mit einer Wellenlänge von 780 nm ausgestrahlt. Damit ist die Kinect gegen Umgebungslicht weitestgehend unempfindlich. Die Verwendung der Kinect im Freien, bei direktem Sonnenlicht, ist jedoch nicht zu empfehlen, da die Sonneneinstrahlung zu Interferenzen führen kann. Die Aussendung des infraroten Lichts geschieht bei der Kinect über die Zeit konstant. Um das projizierte Lichtmuster zu erfassen, ist in der Kinect ein monochromatischer,

<sup>8</sup>Website: <http://www.microsoft.com/>





### 3. Grundlagen

lichtempfindlicher CMOS Sensor verbaut, die **Infrarot-Kamera**. Der Sensor der Infrarot-Kamera verwendet einen Bandbreitenfilter und verfügt über eine Auflösung von 1280\*1024 Pixeln. Durch die Verwendung eines 2\*2 Pixel-Binnings<sup>9</sup> bei einer Bildwechselfrequenz von 30 fps beträgt die reduzierte Auflösung 640\*512 Pixel. Die Bildinformationen werden dabei mit einer 11 bit Intensitätstiefe vom Sensor übertragen. Das Audiosystem der Kinect besteht aus einem **Multi-Array-Mikrofon**, welches aus vier einzelnen Mikrofonen besteht. Jeder der vier Mikrofonkanäle wird vom Mikroprozessor als 16 bit Audio mit einer Abtastrate von 16 kHz verarbeitet. Der **Neigungsmotor** befindet sich im Standfuß der Kinect. Er ermöglicht das motorisierte Neigen der Sensorleiste um  $\pm 28^\circ$ . Der aktuell vorliegende Neigungswinkel wird dabei über einen Beschleunigungssensor bestimmt.

#### 3.4.2. Funktionsweise der Kinect

Nachdem im vorangehenden Abschnitt die technischen Spezifikationen der Kinect beschrieben wurden, werden in diesem Abschnitt die Funktionsweise und das Zusammenspiel der Komponenten erläutert. Zu Beginn wird die Kommunikation zwischen Sensoren und Mikroprozessor erklärt. Im Anschluss wird das Verfahren zur Erzeugung von Tiefenbildern, *Light Coding*, beschrieben. Abschließend wird kurz auf die erzeugten Kamerabilder (RGB, IR) eingegangen.

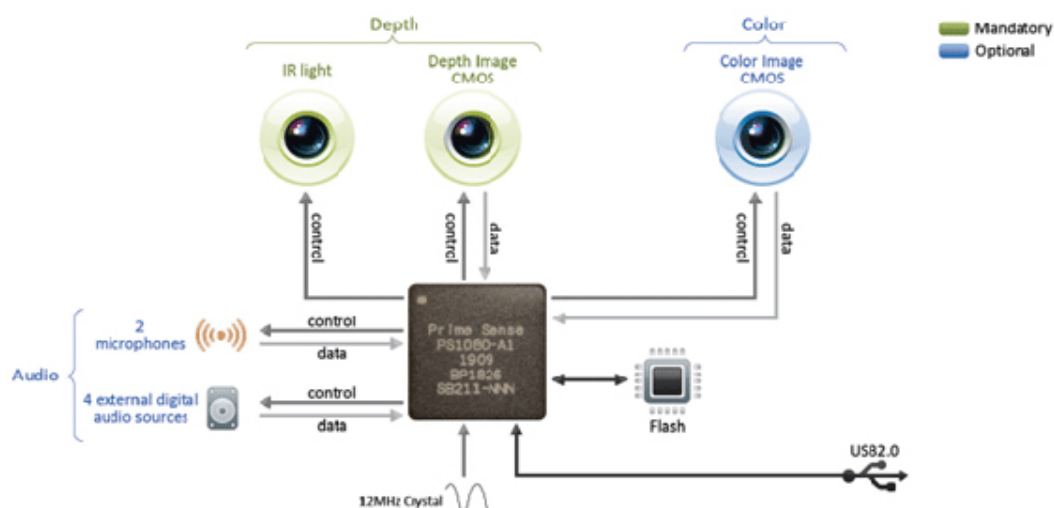


Abbildung 3.5.: Primesense Referenzdesign<sup>10</sup>

<sup>9</sup>Binning: Das Zusammenfassen benachbarter Bildelemente auf dem Bildsensor selbst. Durch die Bildung von Pixelblöcken wird eine höhere Lichtempfindlichkeit pro Pixel zu Lasten der Auflösung erreicht.

<sup>10</sup>Website: <http://www.primesense.com/>



### 3. Grundlagen

---

Die Microsoft Kinect verwendet das von der Firma Primesense patentierte Referenzdesign „Primesensor“ . In der Kinect kommt dieses in abgeänderter Form zum Einsatz. Das Referenzdesign umfasst die drei Komponenten: Infrarotprojektor, Infrarot-Kamera und Primesense PS1080 System on Chip (PS1080 SoC). Zusätzlich zum Referenzdesign ist in der Kinect eine RGB-Kamera verbaut. Eine Darstellung des Referenzdesigns mit den zusätzlich in der Kinect verbauten Komponenten wird in Abbildung 3.5 gegeben. Der PS1080 SoC ist der Kern der Kinect: Er steuert den Infrarot-Projektor, welcher ein großflächiges, fest definiertes Infrarot-Punktmuster in den Raum projiziert. Die Infrarot-Kamera empfängt die reflektierten Infrarotstrahlen und liefert die Daten an den Mikroprozessor. Dieser errechnet auf Basis des (verzerrten) empfangenen Punktmusters und des Referenz-Punktmusters über ein Triangulationsverfahren ein Tiefenbild der Szene. Die Erzeugung der Tiefenbilder geschieht also direkt auf dem Chip und erfordert keine zusätzlichen Software-Berechnungen. Über einen separaten Datenstrom wird das Farbbild der RGB-Kamera an den PS1080 SoC übertragen. Per USB 2.0 werden die Daten anschließend an den Host übertragen. Für multimodale Anwendungen übernimmt der PS1080 SoC zusätzliche audiospezifische Berechnungen, wie Rauschunterdrückung oder die Lokalisation von Audioquellen [68].

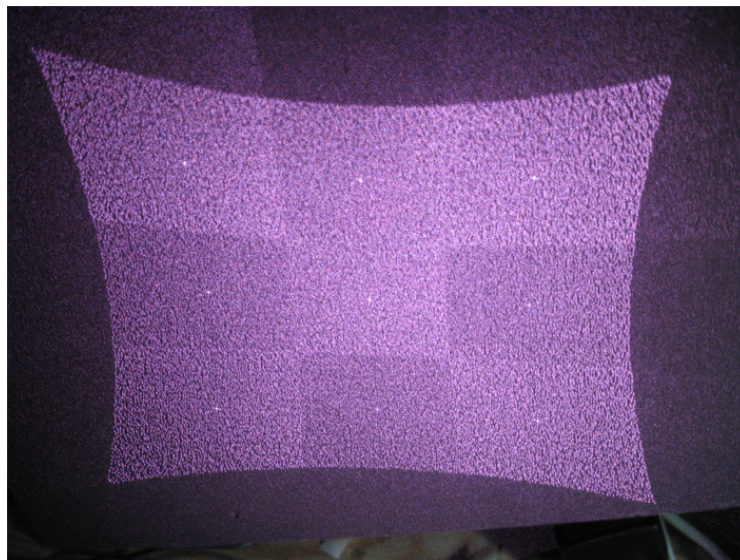


Abbildung 3.6.: Infrarot Punktmuster der Kinect<sup>11</sup>

Die interne Erstellung von Tiefenbildern durch den PS1080 SoC stellt eine der wichtigsten Aufgaben der Kinect dar. Die Kinect verwendet zu diesem Zweck ein von Primesense entwickeltes und patentiertes Verfahren, *Light Coding*. Der genaue Al-

---

<sup>11</sup>Website: <http://www.futurepicture.org/?p=129/>



### 3. Grundlagen

---

gorithmus dieses Verfahrens ist nicht offengelegt. Es lassen sich jedoch anhand von Patenten und Untersuchungen Dritter Rückschlüsse auf das Verfahren ziehen<sup>12</sup>.

Entgegen vieler Annahmen nutzt die Kinect keine teure *ToF-Technik*<sup>13</sup>, sondern nutzt das Prinzip von strukturiertem Licht. Bei diesem wird Infrarotlicht in definierten, wechselnden Mustern ausgesendet und von einem Infrarot-CMOS Sensor<sup>14</sup> aufgenommen. Aufgrund des nicht offen gelegten Algorithmus' lassen sich nicht alle Facetten des Verfahrens erklären. Dadurch, dass die Berechnung der Tiefeninformationen bereits auf dem PS1080 SoC durchgeführt werden, liegt die Vermutung nahe, dass es sich bei dem projizierten Punktmuster um eine Art Code handelt, der die finale Berechnung vereinfacht. Zur Erzeugung von Tiefeninformationen projiziert der Infrarot-Projektor ein fest definiertes Punktmuster unterschiedlicher Intensitätsstufen. Das auf eine Leinwand projizierte Punktmuster des Infrarot-Projektors wird in Abbildung 3.6 dargestellt. Bei näherer Betrachtung von Abbildung 3.6 fällt auf, dass sich das Punktmuster nicht durch eine zufällige Anordnung definiert. Vielmehr lassen sich mehrere „Blöcke“ identifizieren, in dessen Zentrum ein besonders heller Lichtpunkt steht. Das Muster wiederholt sich insgesamt dreimal, jeweils nach 211 Pixeln in horizontaler und 165 Pixeln in vertikaler Richtung. Damit ergibt sich insgesamt eine „3x3 Blockmatrix“. Multipliziert man die Pixel mit der Anzahl der Blöcke, kommt man auf 633\*495 Pixel, was ungefähr einer VGA Auflösung gleich kommt. Eine ausführliche Untersuchung der Zusammensetzung des Musters wird von REICHINGER in [71] beschrieben.

Die Besonderheit der Kinect ist, dass die Tiefeninformationen aus der Kombination von Infrarot-Projektor und Infrarot-Kamera berechnet werden. Dies stellt einen großen Vorteil bei der Menge der anfallenden Berechnungen im Vergleich zu Stereo-Kamerasystemen dar. Anhand des Musters, das in einem zur Lichtquelle bekannten Abstand entsteht, kann über die Abweichung zwischen reflektiertem und bekanntem Muster die Tiefeninformation berechnet werden. Zur Berechnung wird ein Triangulationsverfahren eingesetzt. Abbildung 3.7 verdeutlicht den räumlichen „Versatz“ der Lichtpunkte, der entsteht, wenn sich Objekte im Sichtfeld des Infrarot-Projektors befinden.

Der optimale Abstand zur Kinect beträgt laut Herstellerangaben zwischen 1,2m und 3,5m [52]. Objekte, die sich außerhalb des empfohlenen Abstandes befinden, liefern gar keine oder nur sehr ungenaue Tiefeninformationen [40]. Unter Einbeziehung von

---

<sup>12</sup>Vgl. [19], [22], [77], [78]

<sup>13</sup>ToF: Time of Flight, Laufzeitverfahren um Distanzen zu messen. Dabei wird die Szene mittels eines Lichtpulses ausgeleuchtet. Die Kamera misst für jedes Pixel die Zeit, die das Licht zum Objekt und zurück braucht. Die Zeit ist dabei proportional zur Distanz

<sup>14</sup>Bei der Kinect: Infrarot-Kamera

<sup>15</sup>Website: <http://www.futurepicture.org/?p=116>



### 3. Grundlagen

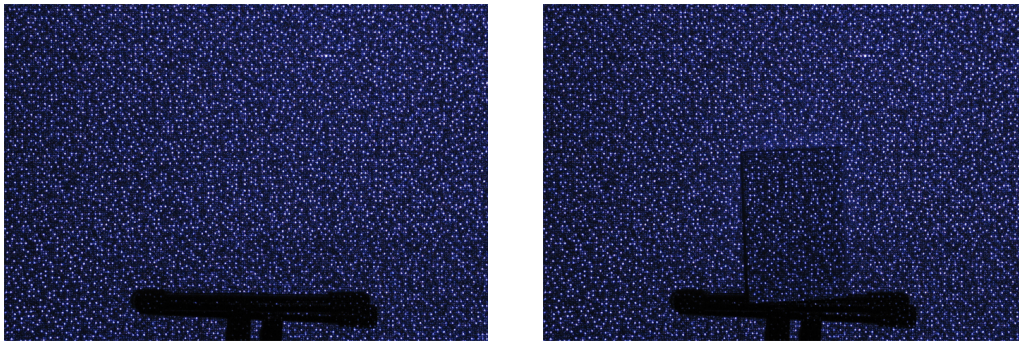


Abbildung 3.7.: Versatz der Lichtpunkte im Punktmuster<sup>15</sup>

Sichtfeld und Kameraabstand ergibt sich ein ungefähres Aufnahmefeld<sup>16</sup> von  $4\text{m}^2$ . Die Genauigkeit der Infrarot-Kamera beträgt bei einem Abstand von 2m zur Kinect für  $x, y = 3\text{mm}$  und für  $z = 100\text{mm}$  [68].

Abbildung 3.8 zeigt ein mit der Kinect aufgenommenes RGB Bild und das zugehörige Tiefenbild. Die RGB Kamera besitzt gegenüber der Infrarot-Kamera ein größeres Sichtfeld. Auffallend bei der Kinect ist, dass die Tiefenbilder mit einer Auflösung von  $640 \times 480$  Pixeln anstatt der angegebenen  $640 \times 512$  Pixeln ausgegeben werden. Die Tiefeninformationen werden mit einer Genauigkeit von 11 bit zurückgegeben. Die Intensitätstiefe der Tiefenbilder entspricht also einem 11 bit Wert, womit sich für jedes Pixel ein Wertebereich von 0 – 2047 ergibt. Bei der Betrachtung der Tiefenbilder fallen vereinzelte schwarzen Bereiche auf (Vgl. Abbildung 3.8). Diese liefern keine Tiefeninformationen und entstehen, wenn Objekte mit spiegelnden, reflektierenden Oberflächen aufgenommen werden oder dann, wenn an besagter Stelle kein Punktmuster abgebildet wurde. Des Weiteren ist bei Tiefenbildern an der rechten Seite ein dünner schwarzer Rand zu bemerken, der ebenfalls keine Tiefeninformation liefert.

#### 3.4.3. Beurteilung der Kinect

Anhand der vorgestellten technischen Daten in Abschnitt 3.4.1 und der Funktionsweise der Kinect in Abschnitt 3.4.2 lassen sich einige Vor- und Nachteile der Kinect feststellen.

Zum einen bietet die Kinect mit einem durchschnittlichen Verkaufspreis von 120 \$ eine preisgünstige Möglichkeit zur Nutzung einer 3D Kamera. Kamerasysteme die auf anderen optischen Verfahren der Tiefenmessung wie z.B. Time of Flight beru-

<sup>16</sup>Mit Aufnahmefeld ist die nutzbare Fläche gemeint, in der der Akteur von der Kamera optimal erfasst werden kann.





### 3. Grundlagen

---



Abbildung 3.8.: Links: Mit Kinect aufgenommenes RGB-Bild. Rechts: Das entsprechende Tiefenbild

hen, sind um ein Vielfaches teurer<sup>17</sup>. Das in der Kinect eingesetzte *Light Coding* Verfahren bietet einige Vorteile: Aufgrund der Verwendung von Infrarotlicht ist die Kinect gegen andere Lichtquellen weitestgehend unempfindlich. Interferenzen treten nur bei direkter Sonneneinstrahlung auf. Damit ist die Grundvoraussetzung für einen Einsatz in geschlossenen Räumen, wie z. B. in öffentlichen Einrichtungen, gegeben. Des Weiteren wird eine berührungsfreie Interaktion möglich. Dies ist besonders für die Anwendung in öffentlichen Einrichtungen von Vorteil, um das System vor Verschmutzung und Vandalismus zu schützen.

Eine Einschränkung der Kinect ist die mit 1,2m – 3,5m relativ geringe Reichweite des Sensors. Die Nutzungsfläche für die Erfassung von Bewegungen ist dadurch auf ca. 4 m<sup>2</sup> eingeschränkt. Bei einem möglichen Einsatz in öffentlichen Einrichtungen spielt dies jedoch eine untergeordnete Rolle, da sich der Nutzer ähnlich wie bei einem Terminal nur wenig bewegt. Die Messungenauigkeit der Tiefenwerte bei zu großem Abstand zur Kinect ist demnach ebenso zu vernachlässigen.

Zusammenfassend lässt sich aussagen, dass die technischen Eigenschaften der Kinect alle Grundvoraussetzungen erfüllen, um als Eingabegerät in Interaktionsszenarien eingesetzt zu werden.

## 3.5. Verwendete Software

In diesem Kapitel wird die für diese Arbeit genutzte Software beschrieben, die zur Realisierung eines gestenbasierten Interfaces eingesetzt wird. Dazu wird in Abschnitt 3.5.1 OpenNI vorgestellt. OpenNI ist ein Framework mit dem die Verwendung der Kinect an einem PC ermöglicht wird. Danach wird in Abschnitt 3.5.2 die Middlewar-

---

<sup>17</sup>Als Vergleichsobjekt dient die SR4000 des Herstellers Mesa Imaging: <http://www.mesa-imaging.ch/>. Der Preis liegt bei ca. 9000 \$



### 3. Grundlagen

komponente NITE behandelt. NITE stellt Module zur Verfügung, die eine Erfassung und Verfolgung von Handgesten ermöglichen. Abschließend wird in Abschnitt 3.5.3 die Spieleengine Unity3D beschrieben, in der die Benutzeroberfläche für diese Arbeit erstellt wird.

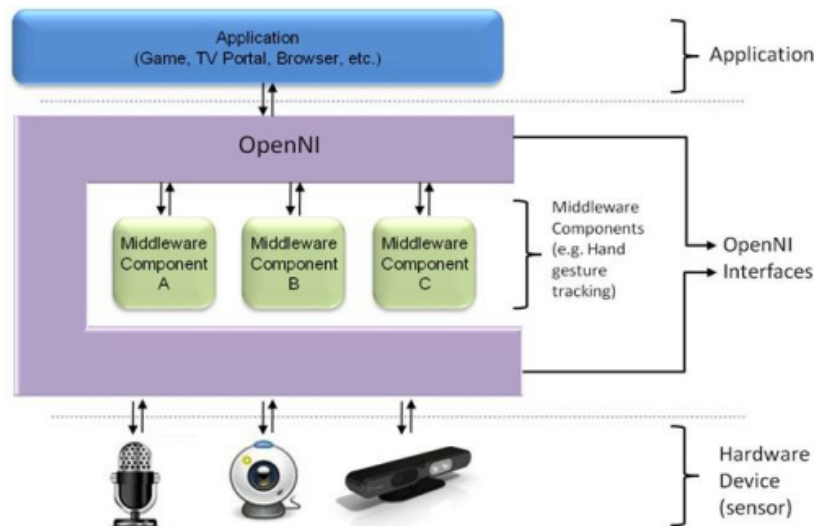


Abbildung 3.9.: Architektur des OpenNI Frameworks<sup>18</sup>

#### 3.5.1. OpenNI

Hinter dem Begriff OpenNI (*Open Natural Interaction*) verbirgt sich eine nicht profitorientierte Organisation, bestehend aus Primesense, Willow Garage<sup>19</sup>, SideKick<sup>20</sup> und Asus<sup>21</sup>. Die Organisation hat sich zum Ziel gesetzt, die Nutzung und Förderung von Kompatibilität und Interoperabilität natürlicher Interaktions-Geräte, -Anwendungen und -Middleware voranzutreiben [66].

OpenNI ist ein Framework, das mehrere *Application Programming Interfaces* (APIs) für die Entwicklung von Anwendungen mit natürlicher Interaktion zur Verfügung stellt. Das Framework stellt sowohl Schnittstellen für die visuellen und auditiven Sensoren (z. B. Microsoft Kinect) als auch für Middlewarekomponenten zur Verfügung. OpenNI unterstützt die Entwicklung in den Sprachen *C* und *C#* und kann unter den Betriebssystemen *Windows*, *LINUX* und *Mac OSX* verwendet werden.

<sup>18</sup>OpenNI: <http://75.98.78.94/default.aspx/>

<sup>19</sup>Website: <http://www.willowgarage.com/>

<sup>20</sup>Website: <http://www.sidekick.co.il/>

<sup>21</sup>Website: <http://www.asus.de/>



### 3. Grundlagen

Ein Merkmal von OpenNI sind die sogenannten *Production Nodes*. Production Nodes beinhalten die eigentlichen Funktionalitäten, beispielsweise werden durch einen *Depth Generator* die unverarbeiteten Tiefeninformationen des Sensors als Tiefenbild ausgegeben. Abhängig vom Anwendungskontext können mehrere Production Nodes definiert werden. Die Regulierung des Datenflusses und der Kommunikation wird vom OpenNI Framework gewährleistet [66]. Die Architektur von OpenNI lässt sich in drei unterschiedliche Schichten unterteilen. Dabei wird eine strikte Trennung zwischen erzeugenden, verarbeitenden und nutzenden Programmteilen angestrebt. Eine grafische Darstellung der Architektur wird in Abbildung 3.9 gegeben. Die unterste Schicht beinhaltet die Hardware, welche die visuellen und/oder auditiven Elemente der Szene über Sensoren erfasst. Darüber befindet sich in der mittleren Schicht OpenNI. In dieser Schicht werden Schnittstellen zur Kommunikation bereitgestellt, die sowohl mit den Sensoren als auch mit den Middleware-Komponenten interagieren. Die Middleware-Komponenten analysieren dabei die Daten, die vom Sensor erfasst werden. Die oberste Schicht repräsentiert Software, die die Anwendungen mit natürlicher Interaktion implementiert.



Abbildung 3.10.: Links: Kalibrierungspose unter OpenNI. Rechts: Aktivierte Handverfolgung

#### 3.5.2. NITE

Ebenfalls von Primesense wird die Middleware-Komponente *NITE* angeboten [67]. NITE stellt mehrere Production Nodes zur Verfügung, die speziell auf die Verarbeitung von Tiefeninformationen abzielen. Die NITE-Middleware beinhaltet sowohl Module für die Identifizierung und Verfolgung von Personen und deren Bewegungen als auch Module zur Erfassung und Verarbeitung von Handgesten. Die Erkennung von Handgesten wird unter NITE durch die *PointControl-Klasse* realisiert. PointControl liefert einen kontinuierlichen Datenstrom aus 3D Raumkoordinaten



### 3. Grundlagen

der Hand in der Form  $P_t = (x, y, z)$  über die Zeit. NITE verfügt über bereits vordefinierte Handgesten: *wave*, *push*, *swipe*, *circle* und *steady*. Um Gesten auf Grundlage der PointControl-Klasse zu erkennen, verfügt jede Handgeste über eine *Detector-Klasse*. In dieser Klasse sind die Merkmale definiert, auf deren Basis ein Vergleich mit der vom Benutzer ausgeführten Geste vollzogen werden kann [67].

Die Architektur von NITE, die in Abbildung 3.11 dargestellt ist, umfasst zwei Schichten: *NITE Algorithms* und *NITE Controls* [67]. In der NITE Algorithms Schicht wird auf die von OpenNI erfassten Daten der Sensoren zugegriffen. Die Sensordaten werden in dieser Schicht verarbeitet und über eine Schnittstelle den Middleware-Komponenten zur Verfügung gestellt. Die NITE Controls Schicht ist eine Applikationsschicht. In dieser Schicht werden Module zur Gestenerkennung und zur Interaktion mit grafischen Oberflächen bereitgestellt. OpenNI ermöglicht dabei die Kommunikation zwischen der Algorithms und Controls Schicht. Für die Erzeugung des Skelettmodells durch den *User Production Node* benötigt OpenNI eine Kalibrierungspose (Vgl. Abbildung 3.10). Diese „Psi Pose“, die wenige Sekunden gehalten werden muss, nutzt OpenNI, um die Koordinaten des Skelettmodells auf Grundlage der Tiefeninformationen zu berechnen. Erst nach erfolgreicher Kalibrierung ist das Tracking für die jeweilige Person aktiviert.

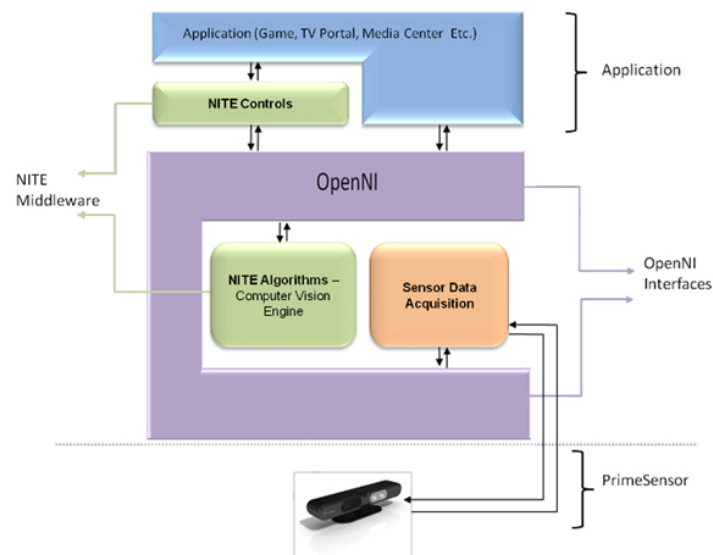


Abbildung 3.11.: Architektur des OpenNI Frameworks mit NITE [67]





### 3.5.3. Unity3D

Unity3D<sup>22</sup> ist eine Software zur Erstellung von 3D Anwendungen. Der Fokus von Unity3D liegt dabei auf der Erstellung von Spielen und grafischen Benutzeroberflächen. Für die vorliegende Arbeit wird Unity3D zur Realisierung der grafischen Benutzeroberfläche verwendet.

Unity3D ist vergleichbar mit anderen renommierten Spieleengines wie z. B. der *UnrealEngine* von Epic Games<sup>23</sup> oder der *CryEngine3* von Crytek<sup>24</sup>. Die Entwicklung ist für viele Plattformen möglich. Die aktuelle Version 3.4.2 unterstützt die Entwicklung unter Windows, MacOSX, Firefox, Safari, Nintendo Wii, iPhone, iPad, Android, Microsoft Xbox360 und Sony Playstation 3. Unity3D wird in zwei unterschiedlichen Versionen angeboten: Unity3D und Unity3D Pro. Der Unterschied liegt dabei in dem angebotenen Funktionsumfang. Der Funktionsumfang von Unity3D umfasst die Module Modellierung, Animation, Audio, Beleuchtung, Rendering, Physik, Netzwerk und Programmierung. Für diese Arbeit sind vorrangig die Funktionalitäten aus den Modulen Modellierung und Programmierung von Interesse. Die Erstellung der grafischen Benutzeroberfläche wird durch Modellierungsmethoden durchgeführt. Unity3D stellt zudem Schnittstellen zur Verfügung, die es ermöglichen, die vom Kinectsensor erfassten und in OpenNI und NITE verarbeiteten Daten zu nutzen. Die Programmierung kann in Unity3D in den Sprachen C#, JavaScript und Python in einem eigens mitgelieferten Editor erfolgen. Eine Synchronisation zu anderen Entwicklungsumgebungen, wie z. B. Visual Studio, wird unterstützt. Unity3D lässt sich grob in zwei Umgebungen unterteilen: *Scene* und *Game*. In der Scene Umgebung wird die Anwendung modelliert, animiert und programmiert. Die Game Umgebung ermöglicht das direkte Testen der Anwendung ohne Kompilierungsvorgang.

---

<sup>22</sup>Website: <http://unity3d.com/>

<sup>23</sup>Website: <http://www.unrealengine.com/>

<sup>24</sup>Website: <http://mycryengine.com/>



## 4. Gestenbasierte Eingabe

Im vorigen Abschnitt wurden die Begriffe *Natürlichkeit* und *Intuitivität* definiert. Es wurde festgehalten, dass die Eingabe auf bereits erlernten, natürlichen Handlungen des Menschen basieren muss. Eine natürliche Form von Eingaben sind Gesten. Das Verständnis über den Gebrauch von Gesten wird bereits in einem frühem Alter erlernt. Zu Beginn wird in Abschnitt 4.1 der Begriff Gestik aus unterschiedlichen Perspektiven definiert. Darauf aufbauend, wird im nächsten Abschnitt 4.2 ein Überblick über Gestentypen gegeben. Dazu wird die allgemeine Taxonomie von Gesten für die Mensch Computer Aktion nach KARAM und die Taxonomie von Handgesten nach PAVLOVIC U. A. vorgestellt. Danach wird in Abschnitt 4.3 ein State-of-the-Art Überblick über bestehende Technologien zur Bewegungserfassung gegeben. Die vorgestellten Technologien stellen die Grundlage für eine spätere Erfassung von Gesten dar. Ausgehend von einem Verständnis für die Erfassung von Bewegungen befasst sich der folgende Abschnitt 4.4 mit der Erfassung von Gesten. Aufgrund der im vorigen Kapitel 3.5.1 beschriebenen fehlenden Möglichkeit, die Algorithmen von OpenNI direkt zu untersuchen, wird der Erfassungsprozess allgemein betrachtet. Dazu werden die typischen Prozessstufen eines Gestenerkennungssystems betrachtet. Beginnend mit der Modellierung von Gesten in Abschnitt 4.4.1 werden Segmentierung in Abschnitt 4.4.2, Merkmalsbestimmung in Abschnitt 4.4.3 sowie der abschließende Klassifizierungsprozess in Abschnitt 4.4.4 beschrieben. Nachdem der Erfassungsvorgang beschrieben wurde, befasst sich Abschnitt 4.5 mit dem Thema Interkulturalität von Gesten. Danach wird im darauf folgenden Abschnitt 4.6 ein Überblick über bereits verfügbare gestenbasierte Systeme gegeben. Abschließend wird in Abschnitt 4.7 ein Zwischenfazit gezogen.

### 4.1. Definition von Gesten

Zum Begriff Gestik lassen sich abhängig vom Anwendungskontext unterschiedliche Definitionen in der Literatur wiederfinden. Die Definitionen stammen dabei hauptsächlich aus linguistischen, anthropologischen sowie psychologischen Untersuchungen und beschränken sich dabei meist auf menschliche Ausdrucksweise und soziale



#### 4. Gestenbasierte Eingabe

---

Interaktion. An dieser Stelle sollen nun einige Definitionen vorgestellt werden. Der deutsche Linguist und Semiotiker [NÖTH](#) definiert den Begriff Gestik wie folgt:

*„Gestik im engeren Sinne umfasst das semiotische Ausdruckspotential des menschlichen Körpers mittels der Arme, der Hände und des Kopfes[61]“*

Eine allgemeinere Definition wird von [KENDON](#) gegeben. Er definiert Gestik als:

*„... Körperhandlungen nichtsprachlicher Art, mit der Absicht etwas zum Ausdruck zu bringen. [39]“*

Die Kernaussage beider Definitionen fokussiert sich dabei auf vom Menschen beabsichtigte Bewegungen mit dem Ziel, etwas zum Ausdruck zu bringen. Für die MCI ist diese Definitionen jedoch nicht ausreichend. Die Verwendung von Handgesten zur Kommunikation mit dem Computer erfordert neben der Steuerung von Funktionen durch Handgesten auch die Manipulation virtueller Objekte durch natürliche Handgesten. Dies wurde auch von [MITRA UND ACHARYA](#) erkannt, die zu folgender Definition von Gestik kommen. Sie definieren Gestik als:

*„... expressive, meaningful body motions involving physical movements of the fingers, hands, head, face or body with the intent of: 1. conveying meaningful information or 2. interacting with the environment. [53]“*

[MITRA UND ACHARYA](#) erweitern die bisherigen Definitionen um eine weitere Facette, die Interaktion mit der Umgebung. Abschließend soll die Definition von [PAVLOVIC U. A.](#) vorgestellt werden. Er definiert Gesten ebenfalls aus einer informationstechnischen Perspektive:

*„A gesture is a stochastic process in the gesture model parameter space  $M_t$  over a suitably defined time interval  $I$ . [64]“*



#### 4. Gestenbasierte Eingabe

Wichtig bei der Definition von [PAVLOVIC U. A.](#) ist die Feststellung, dass Gesten nur über einen bestimmten Zeitraum, dem Zeitintervall  $I$ , mit festem Anfang und Ende definiert sind. Er bezieht sich bei seiner Definition auf die Untersuchungen von [KENDON](#), der Gesten in drei unterschiedliche Phasen einteilt: *preparation*, *stroke*, *recovery*. In der Vorbereitungsphase (*preparation*) wird der Anwender in eine für den Empfänger sichtbare Position gebracht. Danach erfolgt die eigentliche Geste (*stroke*). [KENDON](#) bezeichnet diese beiden Phasen als *gesture phrase*. Abschließend erfolgt die Rückzugsphase (*recovery*), in der der Körper in eine Ruheposition gebracht wird. Den sequentiellen Ablauf der drei Phasen bezeichnet [KENDON](#) als *gesture unit*. Eine grafische Darstellung der Phasen von Gesten wird in Abbildung 4.1 gegeben.

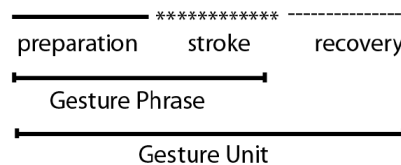


Abbildung 4.1.: Phasen von Gesten nach [KENDON](#) [39]

## 4.2. Gestentypen

Im Hinblick auf die Entwicklung eines gestenbasierten Systems ist die Frage nach einer geeigneten Unterscheidung von Gesten von besonderem Interesse. Entsprechend der unterschiedlichen Definitionen in Kapitel 4.1 ist aber auch die Unterscheidung von Gestentypen in der Literatur nicht eindeutig bestimmt. Oftmals werden unterschiedliche Begriffe synonym verwendet. [KARAM](#) stellt fest, dass beispielsweise symbolische Gesten in der Literatur auch als ikonische Gesten [42] oder Strichgesten bezeichnet [41] werden. Für die Aufstellung einer Taxonomie soll zu Beginn eine geeignete Einteilung von Gesten in Klassen<sup>25</sup> vollzogen werden. Darauf aufbauend werden im Anschluss zwei unterschiedliche Taxonomien betrachtet: die allgemeine Taxonomie für die MCI nach [KARAM](#) und die ausschließlich auf die Verwendung von Handgesten ausgelegte Taxonomie nach [PAVLOVIC U. A.](#).

Eine aus technischer Sicht grundlegende Frage ist, ob Gesten über die Zeit größeren positionellen Änderungen unterworfen sind. Anhand dessen lässt sich eine erste Unterscheidung von Gesten vornehmen. Man differenziert dabei zwischen zwei Arten

<sup>25</sup>Gestenklassen werden in der Literatur auch häufig als Styles oder Gesture Styles bezeichnet.



#### 4. Gestenbasierte Eingabe

---

von Gesten:

- *Statische Gesten*: werden auch als Posen bezeichnet. Statische Gesten lassen sich anhand ihrer Position, Rotation und Form bestimmen. Ein Beispiel für statische Gesten ist Zeichensprache.
- *Dynamische Gesten*: sowohl rein auf Bewegung basierende Gesten, bei denen die Form keine Rolle spielt (z. B. Zeigegesten), als auch Gesten die sich auf Grundlage ihrer Bewegung und Form definieren, werden als dynamische Gesten bezeichnet.

Nachdem eine erste grundlegende Unterscheidung vorgenommen wurde, wird im Folgenden der Ansatz von **KARAM** vorgestellt. **KARAM** unterteilt dabei Gesten anhand ihres Anwendungskontextes:

- **Deiktische Gesten**

Deiktische Gesten umfassen Zeigehandlungen, die die Selektion oder Identifikation von Objekten möglich machen. Diese Gesten werden in der Literatur als eine Basisform zur Kommunikation beschrieben [51]. Bereits in der „Put That There“ Arbeit von **BOLT** wurde der intuitive Gebrauch von deiktischen Gesten aufgezeigt.

- **Manipulierende Gesten**

Nach **QUEK U. A.** sind manipulierende Gesten dadurch definiert, dass die Bewegungen der Hand / des Arms, den Bewegungen des virtuellen Objektes zugeordnet sind. Sie ermöglichen grundlegende Objekttransformationen durch die direkte Beziehung zwischen Hand und virtuellem Objekt. **KARAM** unterscheidet dabei zwischen berührungsfreier, gestenbasierter Eingabe und berührungsempfindlicher, gestenbasierter Eingabe.

- **Semaphorische Gesten**

Semaphorische Gesten werden nach **KARAM** als ein Signalsystem mit festem Vokabular, das sowohl statische als auch dynamische Gesten beinhalten kann, verstanden. Statische Gesten werden dabei durch eine definierte Körperhaltung bzw. Pose repräsentiert. Im Gegensatz zu deiktischen Gesten sind semaphorische Gesten weniger intuitiv [91].



#### 4. Gestenbasierte Eingabe

- **Gestikulation**

Gestikulation oder „Coverbale Gesten“ sind laut [KARAM](#) eine der natürlichsten Formen von Gesten. Dies liegt darin begründet, dass Gestikulation ein wesentlicher, sprachunterstützender Bestandteil menschlicher Dialoge darstellt. [WEXELBLAT](#) beschreibt Gestikulation als eine spezifische, spontane Hand- oder Armbewegung während des Sprechens, die vom Benutzer keinen Lernaufwand und keine bestimmte Körperhaltung erfordert [90].

- **Zeichensprache**

Zeichensprache ist zu den bisher vorgestellten Gesten unabhängig zu behandeln. Zeichensprache basiert ursprünglich auf linguistischen Untersuchungen. Die eingesetzten Zeichen können über Handform, Orientierung, Position und Bewegung beschrieben werden. Gesten einer Zeichensprache sind jedoch nicht zwangsläufig symbolisch. Neben einigen deiktischen Gesten zählen auch manche nachahmende Gesten zur Zeichensprache[51]. In der MCI spielt Zeichensprache vor allem bei der Kommunikationsunterstützung eine große Rolle, z. B. beim Unterrichten von Zeichensprache für Kinder [44].

[KARAM](#) identifiziert demnach die fünf Gestentypen: deiktische Gesten, manipulierende Gesten, semaphorische Gesten, Gestikulation und Zeichensprache. Der Gedanke dieser Einteilung besteht darin, angemessene Gestenklassen für jegliche Interaktionsszenarien der MCI zu definieren. Eine grafische Darstellung der anhand der identifizierten Gestenklassen aufgestellten Taxonomie wird in Abbildung 4.2 gegeben.

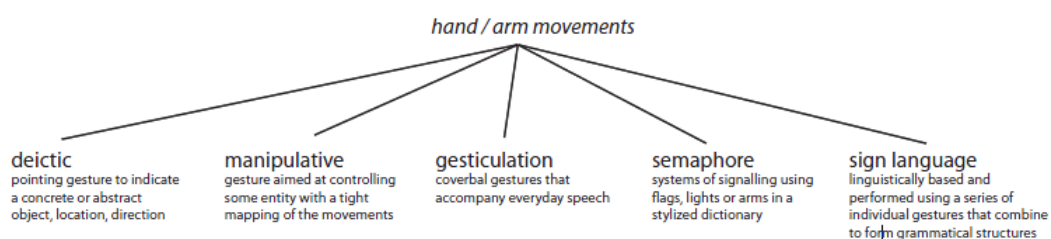


Abbildung 4.2.: Taxonomie von Gesten in der MCI nach [KARAM](#) [38]

Ein weiterer Ansatz zur Unterscheidung von Gesten wird von [PAVLOVIC U. A.](#) beschrieben und basiert auf Untersuchungen von [QUEK U. A.](#). Die Motivation besteht für Pavlovic in der Einteilung von Gesten für die Interaktion zwischen Mensch und Computer. Die Interaktion geschieht dabei ausschließlich durch die Verwendung von Hand- bzw. Armbewegungen. Eine grafische Darstellung der Taxonomie von Pavlovic wird in Abbildung 4.3 gegeben.



#### 4. Gestenbasierte Eingabe

Eine erste Unterteilung von Gesten besteht nach [PAVLOVIC U. A.](#) in der Bildung zweier Hauptklassen: *Gesten* und *unbeabsichtigte Bewegungen*. *Unbeabsichtigte Bewegungen* sind Bewegungen der Hände/Arme, die keine aussagekräftigen Informationen übermitteln. *Gesten* werden wiederum in zwei Modalitäten unterteilt: *kommunikative* und *manipulative* Gesten. *Manipulative Gesten* sind unmittelbar mit einem Objekt verbunden und verändern die Rotation, Position oder Orientierung des virtuellen Objektes. *Kommunikative Gesten* unterteilen [PAVLOVIC U. A.](#) in *Handlungen* und *Symbole*. *Symbole* umfassen dabei alle Gesten, die eine linguistische Rolle spielen. Sie sind entweder *referentieller* Natur, z. B. eine kreisförmige Bewegung des Zeigefingers bezieht sich auf ein Rad oder aber *modalistischer* Natur. *Modalistische Gesten* untermauern im Allgemeinen sprachliche Anweisungen. Im Kontext der MCI nehmen *symbolische Gesten* eine wichtige Rolle ein. Sie lassen sich durch statische Posen beschreiben [64], sind jedoch häufig mit einem kulturellen Kontext verknüpft. *Handlungen* umfassen sowohl *mimetische* als auch *deiktische* Gesten. *Deiktische Gesten* umfassen wie bei [KARAM](#) Zeigehandlungen. *Mimetische Gesten* umfassen dagegen Gesten, die bekannte Handlungsabläufe imitieren.

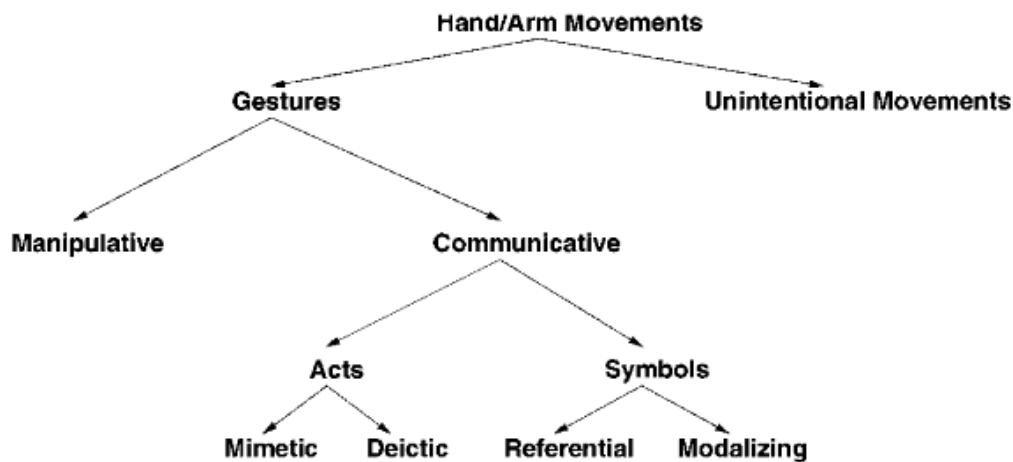


Abbildung 4.3.: Taxonomie der Handgesten für die MCI nach [PAVLOVIC U. A.](#) [64]

### 4.3. Erfassung von Bewegungen

Die technologische Weiterentwicklung bietet neue Möglichkeiten für die gestenbasierte Interaktion. In diesem Abschnitt wird ein *State-Of-The-Art* Überblick über bestehende Möglichkeiten gegeben, menschliche Bewegungen zu erfassen. Dies stellt



#### 4. Gestenbasierte Eingabe

---

die Grundlage für eine spätere Erfassung von Gesten dar. Für eine bessere Übersicht wird eine Einteilung in drei Gruppen von Eingabegeräten vorgenommen: tragbare Geräte werden in Kapitel 4.3.1 behandelt, Computer Vision in Kapitel 4.3.3 und tragbare Anzüge in Kapitel 4.3.2.



Abbildung 4.4.: Tragbare Geräte zur Bewegungserfassung.

Links: 3D-Maus<sup>26</sup>. Rechts: Nintendo WiiMote<sup>27</sup>

##### 4.3.1. Tragbare Geräte

Ein Ansatz für die Erfassung von Bewegungen bieten tragbare Geräte (*Handhelds*), die über integrierte Sensoren die Bewegungen direkt an das Interface weitergeben. Viele dieser Geräte bieten zudem über einen oder mehrere Tasten zusätzliche Interaktionsmöglichkeiten an. Ein Vorteil tragbarer Geräte besteht darin, dass der Benutzer über das kontinuierlich gesendete Signal des Gerätes eindeutig zu identifizieren ist [18]. Damit eignet sich diese Technik laut MORRIS U. A. besonders für den Einsatz in einer kooperativen Umgebung [57]. In Untersuchungen von BERARD U. A. wurde zudem festgehalten, dass tragbare Geräte dem Benutzer eine höhere mentale Anstrengung abverlangen als bei der Eingabe mit einer Maus [2]. Tragbaren Geräte unterscheiden sich anhand ihrer Erfassungsmerkmale. Die Erfassung der Bewegungen durch die Geräte kann entweder extern oder intern geschehen. Externe Erfassungsverfahren verwenden entweder *Computer Vision* oder externe Sensoren. Auf diese Verfahren soll jedoch gesondert in Kapitel 4.3.3 und Kapitel 4.3.2 eingegangen werden. Die Erfassung von Bewegungen geschieht bei den meisten praxistauglichen, tragbaren Geräten intern. Die interne Erfassung wird häufig durch den kombinierten Einsatz von intern verbauten Beschleunigungssensoren und Gyroskopen ermöglicht. Ein Beispiel für ein solches tragbares Gerät sind 3D-Mäuse (Vgl.

---

<sup>26</sup>Website: <http://www.nintendo.de/>

<sup>27</sup>Website: <http://www.3dconnexion.com/>





#### 4. Gestenbasierte Eingabe

Abbildung 4.4, links). Ein weiteres populäres Beispiel ist die Nintendo WiiMote, die in Abbildung 4.4 rechts dargestellt ist. Die interne Erfassung von Bewegungen wird bei der Wiimote anhand der Detektion von ausgesendeten Infrarotstrahlen durch eine verbaute Kamera ermöglicht.

#### 4.3.2. Tragbare Anzüge

Ein pragmatischer Ansatz zur Erfassung von Bewegungen besteht darin, die Sensoren zur Bewegungserfassung direkt am Körper anzubringen. Spezielle „Erfassungszüge“ ermöglichen eine komplette Erfassung menschlicher Bewegungen (Vgl. Abbildung 4.5 links). Für die exklusive Erfassung von Handbewegungen existieren spezielle Datenhandschuhe (Vgl. Abbildung 4.5 rechts). Anzüge und Handschuhe unterscheiden sich typischerweise in der Anzahl der Sensoren, Auflösung der Sensoren und Samplingrate der Sensoren. Des Weiteren unterscheidet man die technischen Charakteristika der Sensoren.



Abbildung 4.5.: Tragbare Anzüge zur Bewegungserfassung. Links: Elektromechanisches Exoskelett<sup>28</sup>. Rechts: P5 Datenhandschuh<sup>29</sup>

Elektromechanische Anzüge (Exoskelette) oder Handschuhe verwenden an den Gelenken angebrachte elektromechanische Sensoren, um Bewegungen zu erfassen. Bewegungen des Akteurs resultieren in einer analogen Spannungsänderung, die im Anschluss in ein digitales Signal umgewandelt werden kann. Die Vorteile dieser Anzüge liegen im Wegfall des Verdeckungsproblems sowie geringe Kosten. Der Nachteil besteht vorwiegend in den eingeschränkten Bewegungsmöglichkeiten des Akteurs. Anstelle der elektromechanischen Sensoren lassen sich auch magnetische Sensoren verwenden. Die Erfassung der Bewegungen geschieht hier über die Abtastung eines

<sup>28</sup>Website: <http://www.animazoo.com/>

<sup>29</sup>Website: <http://www.initiation.co.uk/>



#### 4. Gestenbasierte Eingabe

---

niederfrequenten magnetischen Feldes, das von einem Transmitter ausgesendet wird. Die abgetasteten Werte werden anschließend an eine Kontrolleinheit gesendet, die Position und Ausrichtung im Raum errechnet. Der Vorteil dieser Anzüge besteht in einer hohen Reichweite und keinem Verdeckungsproblem. Die Nachteile sind eine schwierige Kalibrierung und die Anfälligkeit für fremde Magnetfelder.

### 4.3.3. Computer Vision

Computer Vision oder „maschinelles Sehen“ beschreibt im Allgemeinen die computergestützte Lösung von Aufgaben, die sich mit der Erfassung und Analyse von Bildern beschäftigen. Durch die Verwendung spezieller Kamerasysteme (Vgl. Abbildung 4.6) ist es möglich, die Bewegungen von Menschen zu erfassen. Im Gegensatz zu den bisher vorgestellten Möglichkeiten in Kapitel 4.3.1 und 4.3.2 liefern Computer Vision Systeme die Bewegungsdaten nicht direkt in final-verarbeiteter Form an den Computer. Die an den Computer übertragenen Bilder müssen erst anhand spezieller Algorithmen verarbeitet werden, bevor Bewegungsdaten vorliegen. Detaillierte Untersuchungen wurden bereits von [MOESLUND U. A.\[54\]](#) und [POPPE\[65\]](#) durchgeführt. Die Vorteile von Computer Vision Systemen liegen in der geringen Latenzzeit, wodurch sie auch für die Erfassung in Echtzeit geeignet sind, dem Wegfall von zu tragender Hardware und der Robustheit gegen äußeren Einflüssen wie z. B. Magnetfelder. Die Probleme von Computer Vision Systemen liegen, wie [MOESLUND U. A.\[55\]](#) anführen, vor allem in der Verdeckung. Es bestehen jedoch Ansätze des maschinellen Lernens, die versuchen, dass Verdeckungsproblem durch Vorhersagen zu lösen. Dazu seien die Untersuchungen von [J. U. A.\[35\]](#) und [WELCH UND BISHOP\[89\]](#) aufgeführt. Computer Vision Systeme lassen sich in zwei Bereiche unterteilen: markerbasierte Erfassung und markerlose Erfassung.



Abbildung 4.6.: Spezielle Kamerasysteme zur Bewegungserfassung<sup>30</sup>



## 4.4. Erfassung von Gesten

Nachdem im vorangehenden Kapitel 4.3 Technologien zur Erfassung von Bewegungen beschrieben wurden, befasst sich dieses Kapitel mit der Verarbeitung von Gesten. Wie bereits ausgeführt, werden abhängig von der eingesetzten Erfassungstechnologie unterschiedliche Daten zur Weiterverarbeitung erfasst. Datenhandschuhe oder tragbare Geräte (Vgl. Kapitel 4.3.1 und 4.3.2) liefern im Normalfall eine Sequenz der dreidimensionalen Raumposition in der Form  $P_t = (x, y, z)$ . Computer Vision basierte Technologien liefern jedoch ausschließlich Bilder, weshalb eine geeignete Vorverarbeitung vorgenommen werden muss. Die Verarbeitungsschritte lassen sich in Gestenmodellierung (Kapitel 4.4.1), Segmentierung (Kapitel 4.4.2), Merkmalsbestimmung (Kapitel 4.4.3) und Klassifizierung (Kapitel 4.4.4) unterteilen. Eine grafische Darstellung der einzelnen Verarbeitungsschritte wird in Abbildung 4.7 aufgezeigt. Im Folgenden werden die aufgeführten Verarbeitungsschritte einzeln betrachtet und der aktuelle Stand der Technik auf dem betreffenden Gebiet wird wiedergegeben.

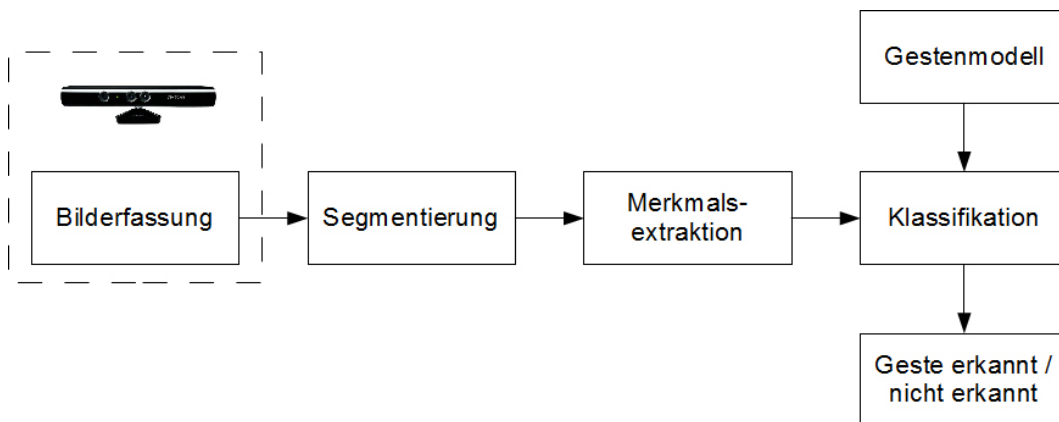


Abbildung 4.7.: Skizzierter Aufbau eines typischen Gestenerkennungssystems

### 4.4.1. Gestenmodellierung

Die Voraussetzung für die Erfassung von Gesten ist ein Gestenmodell. Das Gestenmodell beschreibt, anhand welcher Merkmale und Parameter eine Geste dargestellt bzw. unterschieden wird. Laut PAVLOVIC U. A. stehen dabei Gestenmodell und Anwendungskontext in einem direkten Zusammenhang. Aus technischer Sicht lassen

<sup>30</sup>Website: <http://www.vicon.com/>



#### 4. Gestenbasierte Eingabe

---

sich Gesten in Form von Trajektorien<sup>31</sup> (dynamische Gesten) oder Vektoren (statische Gesten) im Phasenraum abbilden. Bei dynamischen Gesten beschreibt die Trajektorie das Verhalten der Geste über die Zeit. Statische Gesten werden durch einen einzelnen Vektor im Phasenraum repräsentiert. Abhängig vom Anwendungskontext werden laut PAVLOVIC U. A. zwei Verfahren der Modellierung unterschieden: Die *ansichtsbasierten* (*appearance based*) und *modellbasierten Verfahren* (*3D model based*) [64]. Die Wahl des Verfahrens ist abhängig vom Anwendungskontext und hat einen direkten Einfluss auf die eingesetzten Methoden, die bei der späteren Segmentierung (Kapitel 4.4.2), Merkmalsbestimmung (Kapitel 4.4.3) und Klassifikation (Kapitel 4.4.4) eingesetzt werden. Abbildung 4.8 zeigt die von PAVLOVIC U. A. vorgenommene Einteilung der Verfahren zur Gestenmodellierung.

*Ansichtsbasierte Verfahren* werden in der Regel für Anwendungen genutzt, die ein simples Gestenmodell erfordern (z. B. die Steuerung des Fernsehers durch Zeigegesten). Ziel ist es, anhand von Bildsequenzen die charakteristischen Bildmerkmale zu erfassen und zu verfolgen. Je nach Anwendungskontext kann es demnach ausreichend sein, z. B. durch einen einfachen Mustervergleich die Hände zu erfassen. Die restlichen Informationen im Bild sind nicht relevant. *Ansichtsbasierte Verfahren* haben den Vorteil, dass sie etablierte Verfahren der Bildverarbeitung nutzen können. Die Fokussierung auf konkrete Bildmerkmale erfordert zudem eine geringere Rechenleistung<sup>32</sup>, was eine Erfassung in Echtzeit möglich macht.

*Modellbasierte Verfahren* verwenden Vorwissen über den menschlichen Körper, um diesen in einer Bildsequenz zu erkennen und zu verfolgen. Diese Verfahren werden für Anwendungen genutzt die ein komplexes Gestenmodell erfordern (z. B. Zeichensprache). Abhängig von der Komplexität werden einfache Strichmodelle oder detaillierte 3D-Modelle des menschlichen Körpers (oder einzelner Körperteile) verwendet. Die Anzahl der verwendeten Parameter (Vgl. Abbildung 4.8) bestimmen dabei den Detailreichtum des Modells. Ein weiterer wichtiger Faktor für die Wahl des Modells ist der Anwendungskontext. Verfolgt man das Ziel, komplexe Handstellungen (z. B. für Zeichensprache) zu erkennen, müssen im Modell die einzelnen Fingerglieder modelliert werden. Für eine einfache Erkennung und Verfolgung der ganzen Hand sind weitaus weniger detaillierte Modelle notwendig.

---

<sup>31</sup>Trajektorie: Eine Trajektorie ist der Pfad, den ein Objekt im n-dimensionalen Raum, in dem es sich bewegt, zurücklegt. [1]

<sup>32</sup>Im direkten Vergleich zu modellbasierten Verfahren

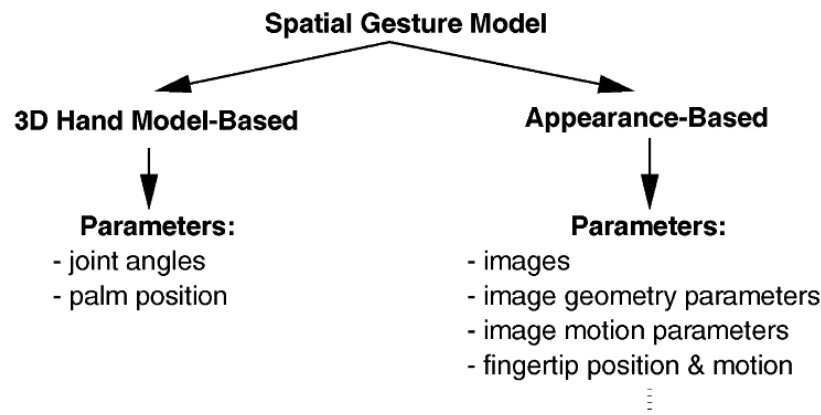


Abbildung 4.8.: Verfahren zur Modellierung von Gesten nach PAVLOVIC U. A. [64]

#### 4.4.2. Segmentierung

Die Vordergrund- und Hintergrundsegmentierung ist ein wichtiges Verfahren in der Computer Vision. Ziel der Segmentierung ist es, ein Bild in einzelne, voneinander getrennte und für eine gewisse Eigenschaft homogene Bereiche zu unterteilen. Die Segmentierung ist ein anwendungsspezifisches Verfahren, das vom Aufnahmeprozess abhängt<sup>33</sup>. Es existieren mehrere Ansätze der bildbasierten Segmentierung. An dieser Stelle sollen exemplarisch die kantenbasierte Segmentierung und die farbbasierte Segmentierung vorgestellt werden.

Bei der **kantenbasierten Segmentierung** wird in einem ersten Schritt versucht, durch lokale Operationen Kanten im Bild zu erkennen. Im Anschluss werden über einen Konturverfolgungsalgorithmus die Konturen aus dem Bild heraus extrahiert. Als Kantendetektionsoperatoren werden häufig *LaPlace*, *Sobel* oder der *Canny-Operator* eingesetzt. Die Arbeiten von LI UND GREENSPAN [46] und SHI U. A. [75] beschreiben den Einsatz von kantenbasierter Segmentierung im Rahmen der Gestenerkennung. Bei der **farbbasierten Segmentierung** geschieht die Segmentierung durch eine Farbe, die sich eindeutig vom Hintergrund absetzt. Viele Arbeiten auf diesem Gebiet setzen die farbbasierte Segmentierung zur Erkennung der Hautfarbe ein [96]. Es existieren mehrere Ansätze, die auf unterschiedliche Methoden zurückgreifen, Hautfarbe in komplexen Bildern zu erkennen. Abhängig vom Ansatz, wird die Segmentierung in unterschiedlichen Farbräumen durchgeführt (HSV/HSI, YcbCr, RGB). Zur Unterscheidung der Pixel werden Schwellenwertverfahren, Histogramme oder Lookup-Tabellen verwendet [81]. Die Arbeiten von ZARIT U. A. [96] und WU U. A. [94] seien als Beispiele für den Einsatz der farbbasierten Segmentierung bei der Ges-

<sup>33</sup>Vgl. [81], S. 203



tenerkennung aufgeführt.

#### 4.4.3. Merkmalsbestimmung

Nachdem im vorangehenden Abschnitt die Segmentierung beschrieben wurde, besteht die nächste Prozessstufe eines Gestenerkennungssystems in der Extraktion bzw. Bestimmung der Merkmale. Die Merkmalsbestimmung wird in der Literatur auch als *feature extraction* bezeichnet. Ziel ist es, auf Grundlage der extrahierten Merkmale eine Unterscheidung oder Abschätzung der Geste vornehmen zu können. Man unterscheidet dabei zwei unterschiedliche Ansätze der Merkmalsbestimmung: die Bestimmung anhand von zweidimensionalen Merkmalen und die Bestimmung anhand von dreidimensionalen Merkmalen.

Die Bestimmung von zweidimensionalen Merkmalen geschieht anhand formbeschreibender Parameter wie der Silhouette des Segments, segmentumschließender Rechtecke oder dem Segmentsschwerpunkt. Anhand von Fourierdeskriptoren [25], Hauptkomponentenanalyse (PCA<sup>34</sup>) [13] oder Hu-Momenten [32] wird der Merkmalsraum in seiner Dimension reduziert. Der Vorteil dieser Verfahren besteht in der Invarianz der Bildsegmente gegenüber Translation, Rotation und Skalierung. Die Kombination mehrerer Verfahren zur Merkmalsbestimmung ermöglicht es, Abhängigkeiten der Umgebung weiter zu reduzieren [62].

Der zweite Ansatz beschreibt die Ermittlung von dreidimensionalen Merkmalen anhand eines dreidimensionalen Modells. Durch den Einsatz von Stereokamerasystemen kann beispielsweise die konvexe Hülle der Hand rekonstruiert werden. Weitere Analysen der konvexen Hülle lassen z. B. Rückschlüsse auf die Positionen der Fingerspitzen zu [73]. Im Gegensatz zur zweidimensionalen Merkmalsbestimmung benötigen Ansätze der dreidimensionalen Merkmalsbestimmung deutlich mehr Rechenaufwand. Eine Möglichkeit dieses Problem zu lösen, besteht in der Berechnung durch den Grafikprozessor (GPU<sup>35</sup>) [47].

#### 4.4.4. Klassifizierung

Bei der Klassifikation werden die zuvor ermittelten Merkmale (Vgl. Kapitel 4.4.3) zur Unterscheidung von Gesten verwendet. Es wird eine Entscheidung getroffen, ob es sich um eine bekannte oder unbekannte Geste handelt. Die Basis der Klassifikation

---

<sup>34</sup>PCA: Principle Component Analysis

<sup>35</sup>GPU: Graphics Processing Unit



#### 4. Gestenbasierte Eingabe

---

bilden Merkmalsvektoren (statische Gesten) oder Merkmalstrajektorien (dynamische Gesten), die die Gesten beschreiben. Die Merkmalsvektoren bzw. -trajektorien werden aus den zuvor extrahierten Merkmalen erzeugt und bilden die Gesten in einem Merkmalsraum ab. Die Dimensionalität des Merkmalsraums entspricht der Anzahl der extrahierten Merkmale. Allgemein wird für die Erkennung der Abstand des Merkmalsvektors bzw. -trajektorie zu der bekannten Gestenklasse<sup>36</sup> ermittelt. Liegt der Merkmalsvektor bzw. -trajektorie in dem Gebiet dieser Gestenklasse, wird die Geste erkannt.

Die Klassifikation umfasst bei aktuellen Ansätzen zwei Teilaufgaben. Die erste Teilaufgabe umfasst eine Trainings- oder Lernphase, in der extrahierte Merkmale zu einer Gestenklasse zugeordnet werden. Das Anlegen der Trainingsdaten geschieht abhängig vom Ansatz, manuell oder automatisch. J. U. A. verwenden in ihrer Arbeit „*Real Time Human Pose Recognition in Parts from Single Depth Images*“ ein Trainingsdatensatz von 500.000 Bildern, die dem Klassifikator eine Bewertung ermöglichen [35]. Die zweite Teilaufgabe besteht in dem eigentlichen Erkennungsvorgang. In der Literatur sind dazu viele unterschiedliche Verfahren beschrieben. Eine häufig verwendete Methode der Klassifikation ist das *Hidden Markov Modell* (HMM), das unter anderem von YANG UND XU [95] und WILSON UND BOBICK [92] zur Klassifikation von Gesten verwendet wird. Eine weitere, häufig verwendete Methode sind künstliche neuronale Netze [16][11]. CARBINI U. A. stellen in ihrer Arbeit ein System vor, das dem Anwender die Interaktion mittels Zeigegesten an einem großen Display ermöglicht. Das neuronale Netz wird verwendet, um nach einer farbbasierten Segmentierung zu entscheiden, ob die Bildsegmente den Kopf oder die Hände des Anwenders darstellen. Für die Klassifikation existieren noch weitere Methoden, wie die PCA [26], Dynamic Time Warping (DTW) [46] oder die Condensation Trajectory Recognition (CTR), eine Generalisierung der HMM [69].

### 4.5. Interkulturalität

Dieses Kapitel behandelt das Thema Interkulturalität im Kontext Gesten. Ausgehend von der Tatsache, dass symbolische sowie manipulierende Gesten einen kulturellen Kontext besitzen, wird die Fragestellung betrachtet, inwieweit sich dies auf die Verwendung von Gesten in der MCI auswirkt.

In der interkulturellen Kommunikation spielt nicht nur die Sprache eine Rolle, sondern auch der sprachbegleitende Einsatz von Mimik und Gestik. Die Verwendung von

---

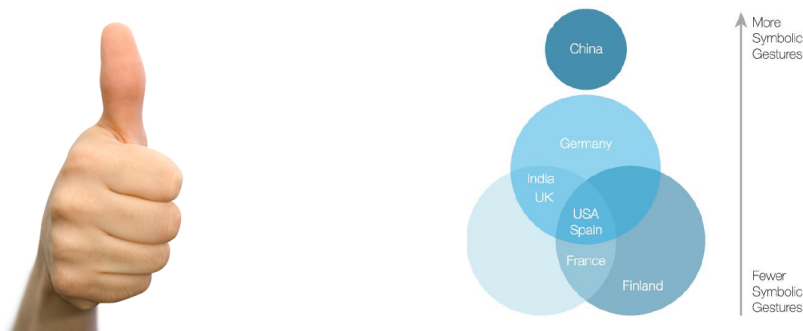
<sup>36</sup>Die Gestenklasse liegt dabei in der Form eines Vektors oder Trajektorie vor.





#### 4. Gestenbasierte Eingabe

Gesten geschieht oft unbewusst und kann in verschiedenen Kulturen unterschiedliche Bedeutungen haben. Was in der einen Kultur als eine positiv zu wertende Geste aufgefasst wird, kann in einer anderen Kultur zu Verärgerung führen. Beispielweise symbolisiert eine zur Faust geballte Hand mit erhobenem Daumen (Vgl. Abbildung 4.9) in den meisten Ländern soviel wie „Ok“ oder „gut so“. In Japan wird diese Geste allerdings als „fünf“ aufgefasst, sie kann aber auch als ein Zeichen fürs „Trampen“ verstanden werden. Dieses Beispiel macht deutlich, dass Gesten häufig eine kulturell unterschiedliche Bedeutung haben. Für die Konzeption gestenbasierter Interfaces müssen diese kulturellen Aspekte bei der Wahl der Gesten berücksichtigt werden.



Abbildungung 4.9.: Links: Symbolische Geste mit kulturell unterschiedlicher Bedeutung. Recht: Einsatz symbolischer Gesten bei Ländern [50]

Es existieren jedoch Gesten, die unabhängig von der Kultur verstanden werden. CASSELL stellte fest, dass im Wesentlichen symbolische sowie manipulierende Gesten unterschiedlich aufgefasst werden [12]. Die Untersuchungen von MAUNEY U. A. befassten sich mit den kulturell unterschiedlichem Gebrauch von Gesten an einem Touchscreen. Dazu wurden in 9 Ländern<sup>37</sup> 340 Probanden getestet. Es wurde festgestellt, dass z. B. der Gebrauch einer „Scroll Down“ Aktion unterschiedlich aufgefasst wird. 45 % der Probanden führten eine Bewegung von unten nach oben aus, 37 % eine Bewegung von oben nach unten. Des Weiteren wurde festgestellt, dass Chinesen im Vergleich zu anderen Probanden signifikant mehr symbolische Gesten verwenden (Vgl. Abbildung 4.9) [50]. Die Ergebnisse dieser Studie liefern trotz des Schwerpunkts auf berührungsempfindliche Geräte Anhaltspunkte für die Wahl der Gesten. Darüber hinaus kann angezweifelt werden, ob die Ergebnisse des „Scroll Down“ Versuchs kultureller Herkunft sind oder ob es sich bei den Ergebnissen eher um ein Resultat der unterschiedlichen Erfahrungen mit berührungsempfindlichen Geräten handelt.

<sup>37</sup>China, Indien, Spanien, UK, USA, Frankreich, Deutschland, Italien, Finnland





## 4.6. Gestenbasierte Systeme

In diesem Abschnitt werden zwei kommerzielle, gestenbasierte Systeme vorgestellt, die eine solche Interaktion umsetzen. Namentlich sind dies der *IPointPresenter*, der am *Fraunhofer Heinrich-Hertz-Institut* entwickelt wurde und das *GestPoint-System* der amerikanischen Firma *GestureTrek*. Beide Systeme zeichnen sich dadurch aus, dass die Interaktion allein durch die Verwendung von Zeigegesten geschieht. Die berührungsfreie Interaktion wird jeweils durch ein Zwei-Kamerasystem erreicht, das keine weiteren Hilfsmittel<sup>38</sup> erforderlich macht.



Abbildung 4.10.: iPointPresenter des Fraunhofer Instituts für Nachrichtentechnik<sup>39</sup>

### • IPoint Presenter

Der am Fraunhofer Institut für Nachrichtentechnik in Berlin entwickelte *IPoint Presenter* verwendet zwei Kameras, die die Position der Zeigefinger des Nutzers ermitteln und anschließend deren Bewegung verfolgen. Das System wird in unterschiedlichen Ausführungen angeboten: als Barebone PC oder als einzelner Videoprojektor. Die Barebone-Variante eignet sich in erster Linie für den mobilen Einsatz ohne die Notwendigkeit einer separaten Kamerainstallation. Das System ermöglicht gestenbasierte Interaktion in Echtzeit. Die Kameras arbeiten dabei mit einer Bildwechselfrequenz von 50 fps<sup>40</sup> und sind gegen Umgebungslicht weitestgehend unempfindlich. Der *IPoint Presenter* ist in der Lage bis zu acht Finger des Benutzers gleichzeitig zu erfassen und zu verfolgen, was eine beidhändige Interaktion ermöglicht. Die Interaktionsmöglichkeiten die das System beim Umgang mit virtuellen Objekten bietet, umfassen rotieren, skalieren, greifen, zeigen oder loslassen. Verschiedene Interpretationen von Fingerbewegungen können als Eingabegesten verwendet werden und zusätzlich

<sup>38</sup>z. B. Datenhandschuhe Vgl. Kapitel 4.3

<sup>39</sup>Website: <http://www.hhi.fraunhofer.de/>

<sup>40</sup>fps: frames per second, Maß für die Bildwechselfrequenz



### 4. Gestenbasierte Eingabe

um weitere Gesten erweitert werden. Des Weiteren bietet der *IPoint Presenter* die Möglichkeit des Mehrbenutzerbetriebs an. Als Nutzungssituationen kommen z. B. Informationssysteme für öffentliche Bereiche (Point of Sale, Point of Information), Spielekonzepte oder die Steuerung medizinischer Anwendungen, in denen eine aseptische Umgebung erforderlich ist, in Frage.

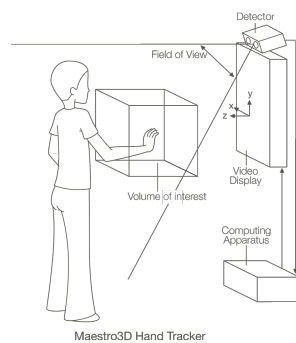


Abbildung 4.11.: Links: Funktionsprinzip des GestPoint Maestro3D  
Rechts: GestPoint Maestro3D in der Anwendung<sup>41</sup>

#### • GestPoint

Die Firma GestureTek aus den Vereinigten Staaten ist Marktführer auf dem Gebiet der kamerabasierten Gestenerkennungssysteme. Das Produktportfolio umfasst interaktive Multitouch-Tische sowie verschiedene Systeme zur berührungsfreien Gestenerkennung. Das unter dem Namen Maestro 3D veröffentlichte System, verwendet eine 3D Kamera und ermöglicht berührungsfreie gestenbasierte Interaktion. Die mitgelieferte Software nutzt patentierte Algorithmen, um eine unmittelbare Erfassung des Benutzers ohne Kalibrierungsphase zu gewährleisten. Abhängig vom Anwendungskontext werden drei unterschiedliche Erfassungsmöglichkeiten genutzt: Zwei-Hand-Erfassung, Vier-Punkt-Erfassung (Kopf, Hände, Oberkörper) und die Erfassung des Skeletts anhand Tiefenbild-Segmentierung. Durch den „Volume of interest“ (VOI) wird entschieden, welche Erfassungsmöglichkeit genutzt wird (Vgl. Abbildung 4.11). Befinden sich z. B. die Hände des Benutzers im VOI so wird die Zwei-Hand-Erfassung genutzt. Damit lassen sich fünf bereits konfigurierte Handgesten verwenden: Circle, Swipe, Poke, Steering, Scale/Rotate. Das mitgelieferte Software Development Kit (SDK) gestattet die Erstellung eigener Handgesten. Die Anwendungsgebiete des Maestro3D umfassen laut GestureTek Videokonferenzen, Präsentationen, Spiele, aseptische Umgebungen und Heimanwendungen.

<sup>41</sup>Website: <http://www.gesturetek.com/>



## 4.7. Zwischenfazit

In diesem Abschnitt wurden Gesten als eine natürliche Interaktionsform vorgestellt. Aufgrund des breiten Forschungsfeldes beschränkte sich die Untersuchung auf die berührungsfreie Interaktion durch Gesten. Berührungssensitive und sprachbasierte Eingabesysteme wurden nicht weiter verfolgt. Die vorgestellten Definitionen und Taxonomien für Gesten sind jedoch auf berührungssensitive Eingabesysteme übertragbar. Die Algorithmen zur Erfassung von Gesten sind unter OpenNI nicht zugänglich. Aus diesem Grund wurde der Fokus auf ein allgemeines Verständnis und einer Beschreibung verschiedener Möglichkeiten bei der Erfassung von Gesten gelegt.

Die Interaktion sollte sich an bereits erlernten Fähigkeiten des Menschen orientieren. Gesten stellen eine früh erlernte Möglichkeit zur Kommunikation bzw. Interaktion dar. Die berührungsfreie Erfassung von Gesten bringt einige Anforderungen mit sich. Eine Anforderung besteht in der Verwendung eines Eingabemediums, das auf bereits erlernten, natürlichen Fähigkeiten des Menschen beruht. Die gestenbasierte Interaktion mit den Händen stellt eine solche Eingabemöglichkeit dar. Computer Vision basierte Ansätze zur Gestenerkennung ermöglichen dabei die gestenbasierte Interaktion ohne zusätzliche Hilfsmittel. Tragbare Geräte wie z. B. die WiiMote erfordern dagegen einen gewissen Lernaufwand über die Bedienung. Weitere Anforderungen bestehen in der Erfassung in Echtzeit und einer hohen Erfassungsgenauigkeit. Um eine direkte, berührungsfreie Interaktion zu gewährleisten müssen zudem geringe Latenzzeiten erreicht werden. Die Latenzzeiten müssen so gering sein, dass sie vom Menschen nicht mehr wahrgenommen werden können.



## 5. Gestenbasierte Interfaces

Im vorigen Kapitel wurde die natürliche Interaktion anhand von Gesten beschrieben. Das Ziel einer natürlichen Interaktion zwischen Mensch und Computer kann jedoch nur erreicht werden, wenn sowohl Eingabe als auch die Interaktion mit Elementen der Benutzeroberfläche auf bereits erlernten Fähigkeiten basiert. Daher wird in diesem Kapitel das Konzept der Natural User Interfaces vorgestellt werden. NUI stellen nach Graphical User Interfaces (GUI) die nächste Generation der Benutzerschnittstelle in der MCI dar. NUI zeichnen sich nicht ausschließlich durch die natürliche Eingabe aus. Sie ermöglichen die direkte Interaktion mit Inhalten in menschenangemessener Form. Aufgrund des noch jungen Forschungsgebiets besteht die Notwendigkeit nach einem allgemeingültigen Verständnis über NUI. Dieses Kapitel greift offene Forschungsfragen auf und gibt den aktuellen Stand der Technik wieder. Dazu werden in Abschnitt 5.1 Ansätze zur Definition von NUI gegenübergestellt und diskutiert. Abschnitt 5.2 gibt einen Überblick über die zeitliche Entwicklung der Benutzerschnittstelle in der MCI. Ausgehend von den Definitionen werden in Abschnitt 5.3 die Eigenschaften eines NUI beschrieben. Darauf aufbauend werden im folgenden Abschnitt 5.4 Produkte vorgestellt, die eine direkte Interaktion anhand NUI ermöglichen. Abschnitt 5.5 befasst sich mit der Ergonomie von berührungsfreien Eingabesystemen. Durch den Paradigmenwechsel von GUI zu NUI müssen neue Metaphern gefunden werden über die ein Benutzer interagieren kann. In Abschnitt 5.6 wird ein solcher Ansatz vorgestellt. Abschließend wird in Abschnitt 5.7 ein Zwischenfazit gezogen.

### 5.1. Definition von NUI

Natural User Interfaces stellen nach den Command Line Interfaces (CLI) und den Graphical User Interfaces die dritte Generation der Benutzerschnittstelle dar. Die Gestaltung von Darstellung und Interaktion erfolgt bei NUI auf natürliche und intuitive Weise, indem auf Fähigkeiten zurückgegriffen wird, die ein Benutzer im Laufe seines Lebens erworben hat. Aufgrund der vielfältigen Interaktionsmechanismen



### 5. Gestenbasierte Interfaces

---

und -technologien, die von Touchscreens über Smartphones bis hin zu Videospielen reichen, existiert noch keine einheitliche Definition von NUI. Dieser Abschnitt soll einige Ansätze zur Definition von NUI vorstellen.

Eine Definition aus dem Jahr 2009 stammt von [BOLLHOEFER U. A.](#). Im Rahmen einer Untersuchung des Microsoft Surface stellten [BOLLHOEFER U. A.](#) folgende Definition auf:

„Ein Natural User Interface beschreibt ein Interface, welches unmittelbar durch einen oder mehrere Sinne des Benutzers bedient wird [5].“

[BOLLHOEFER U. A.](#) betonen in ihrer Definition zwei Aspekte: Die Verwendung eines oder mehrerer Sinne (Multimodalität) und die unmittelbare, direkte Interaktion mit dem Interface. Eine weitere Definition wird von [MONSON-HAEFEL](#) vorgenommen. Er definiert den Begriff folgendermaßen:

„A Natural User Interface is a human-computer interface that models interactions between people and the natural environment [56].“

Die Definition von [MONSON-HAEFEL](#) sagt aus, dass ein NUI eine besondere Ausprägung der MCI darstellt. Er betont, dass ein NUI die Interaktion des Benutzers mit seiner natürlichen Umgebung abbildet. Abschließend soll in diesem Abschnitt die Definition von [BLAKE](#) vorgestellt werden. Er definiert den Begriff NUI wie folgt:

„A natural user interface is a user interface designed to reuse existing skills for interacting directly with content [3].“

[BLAKE](#) beschreibt in seiner Definition explizit die Notwendigkeit eines „Design“. Ein NUI zeichnet sich demnach nicht nur durch die direkte Interaktion aus, sondern bedarf auch einer gezielten Gestaltung, die die Interaktion in geeigneter Form unterstützt. Des Weiteren stellt er den „reuse of existing skills“ heraus, also die Wiederverwendung bereits erlernter Fähigkeiten, die der Benutzer im Laufe seines Lebens erworben hat. Der letzte Punkt in seiner Definition „interacting directly with the content“ beschreibt die direkte Interaktion mit dem Inhalt. Anders als bei GUI interagiert der Benutzer nicht indirekt über Anwendungen, die ihm Zugang zu den



## 5. Gestenbasierte Interfaces

Inhalten ermöglichen, sondern unmittelbar und direkt.

Vergleicht man die vorgestellten Definition, so weisen alle eine Gemeinsamkeit auf: die direkte Interaktion mit Inhalten durch den Benutzer. Darüber hinaus lässt sich feststellen, dass noch kein allgemeiner Konsens bei der Definition von NUI besteht.

### 5.2. Vom CLI zu NUI

Im vorangehenden Abschnitt wurde der Begriff des Natural User Interfaces definiert. Um ein besseres Verständnis über die grundlegenden Eigenschaften von NUI zu bekommen, befasst sich dieser Abschnitt mit der zeitlichen Entwicklung von Benutzerschnittstellen. Beginnend mit einer kurzen Beschreibung der Command Line Interface wird im Anschluss der Wandel vom Graphical User Interface zum Natural User Interface erklärt.

Das CLI (Vgl. Abbildung 5.1 links) war das erste weitverbreitete Dialogsystem, das die Interaktion zwischen Mensch und Computer ermöglichte. CL-basierte Systeme wie MS-DOS oder Apples ProDOS verwenden textbasierte, kryptische Eingaben zur Interaktion mit dem Computer. Über eine meist monochrome Darstellung am Monitor kann der Benutzer Manipulationen am System und an Dateien vornehmen. Ein allgemeines Grundwissen sowie eine strikte Einhaltung der Syntax eines Befehls sind für die Interaktion an einem CLI Voraussetzung. Als primäres Eingabegerät dient die Tastatur.

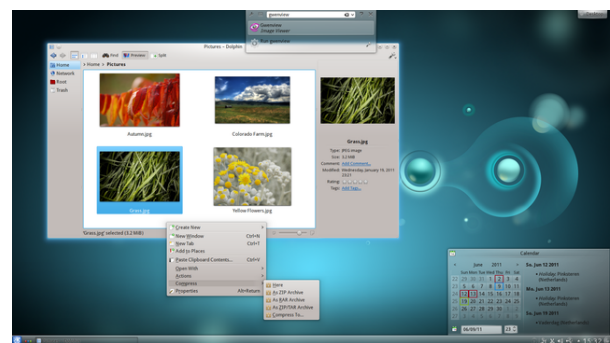
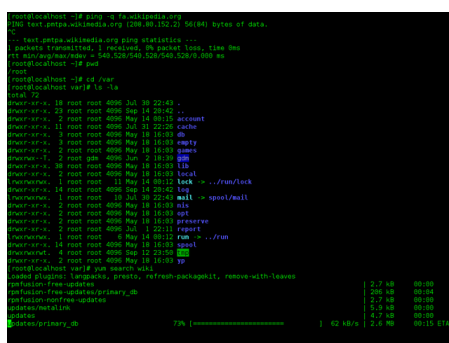


Abbildung 5.1.: Links: Command Line Interface<sup>42</sup>(CLI)  
Rechts: Graphical User Interface<sup>43</sup>(GUI)

<sup>42</sup>Website: [http://en.wikipedia.org/wiki/Command-line\\_interface](http://en.wikipedia.org/wiki/Command-line_interface)

<sup>43</sup>Website: [http://en.wikipedia.org/wiki/Graphical\\_user\\_interface](http://en.wikipedia.org/wiki/Graphical_user_interface)



### 5. Gestenbasierte Interfaces

Der Paradigmenwechsel vom CLI zu GUI besteht vorrangig in der grafischen Gestaltung des Interfaces. GUI (Vgl. Abbildung 5.1, rechts) lösen sich von einer textbasierten Darstellung und verwenden grafische Metaphern, um den Benutzer in seiner Aufgabenerfüllung zu unterstützen. Das Aussehen der Betriebssysteme rückt bei GUI stärker in den Fokus und steigert dadurch die Benutzerfreundlichkeit. Die Desktopmetapher, WIMP-Metapher (Window, Icon, Menu, Pointer) oder „Drag and Drop“ sind Beispiele für die Interaktion über Metaphern an einem GUI. Das primäre Eingabegeräte für GUI ist die Maus.

Die dritte Generation von Benutzerschnittstellen stellen Natural User Interfaces dar. Ähnlich zum Paradigmenwechsel von CLI zu GUI, begründet sich der Wechsel in dem Bedürfnis, eine möglichst natürliche Form der Interaktion zwischen Mensch und Computer anzubieten. Anhand der Definitionen lassen sich einige Eigenschaften von NUI festhalten. Das primäre Eingabemedium bei NUI ist die Hand. Die Interaktion geschieht durch Gesten in einer natürlichen Art und Weise. Anders als bei GUI geschieht die Interaktion mit dem Inhalt in direkter Form. Eine Gegenüberstellung der Eigenschaften von Interfaces wird in 5.1 gegeben. Die aufgeführten Eigenschaften von NUI sollen im folgenden Abschnitt näher betrachtet werden.

Tabelle 5.1.: Vergleich der Eigenschaften von CLI, GUI, NUI nach [HENSELER \[28\]](#)

	CLI	GUI	NUI
<b>Primäre Eingabemedien</b>	Tastatur	Tastatur und Maus	Hand und Griffel
<b>Interface</b>	Abstrakt	Indirekt	Unmittelbar
	Text	Grafik	Objekte
<b>Denken in</b>	Zahlen	Symbolen	Objekten
<b>Interaktion</b>	Unnatürlich	Semi-Natürlich	Natürlich
	Gelernt	Wiedererkennend	Intuitiv
<b>Mediale Ausprägung</b>	Monomedial	Multimedial	Multimodal
<b>Zielerreichung</b>	Getrieben	Explorativ	Kontext-Sensitiv
<b>User Experience</b>	Nüchtern	Anschaulich	Erlebnisorientiert
<b>Wirkungseffizienz</b>	Gering	Mittel	Hoch

### 5.3. Eigenschaften von NUI

Die vorangehenden Abschnitte haben den Begriff NUI definiert sowie einen Überblick über die zeitliche Entwicklung gegeben. In diesem Abschnitt wird anhand der



Definition von [BLAKE](#) beschrieben, wodurch sich ein NUI auszeichnet. Es wird dabei der Frage nachgegangen, welche Prinzipien eingehalten werden müssen, um eine natürliche Interaktion zwischen Mensch und Computer zu erreichen.

Die erste Aussage ist nach [BLAKE](#):

**„NUI are designed“**

Diese These sagt aus, dass NUI designt werden. Eine Reihe von Vorüberlegungen und eine spezifische Planung sind dabei nötig. Die Interaktion eines NUI muss für Benutzer, Inhalt und Kontext angemessen sein. [BLAKE](#) merkt an, dass der Designprozess eines NUI genauso viel Priorität besitzen sollte wie der Entwicklungsprozess. Der Unterschied zu GUI besteht in einer Verschiebung der Prioritäten beim Design. GUI und CLI definieren sich über die Eingabegeräte (Maus bzw. Tastatur). NUI definieren sich über die Interaktion mit Inhalten. Diese Verschiebung bringt eine neue Herausforderung für Designer mit sich.

**„NUI use natural human behaviors“**

Die zweite Aussage befasst sich mit dem natürlichen Verhalten von Menschen, was bei NUI die Grundlage der Interaktion darstellt. In Kapitel [3.3](#) wurde festgehalten, dass natürliche Handlungen beim Menschen auf bereits erlernten Fähigkeiten beruhen. Darunter fallen angeborene Fähigkeiten sowie weitere erlernte Fähigkeiten, die sich ein Mensch im Laufe seines Lebens angeeignet hat. [BLAKE](#) betont die Relevanz der Wiederverwendbarkeit dieser Fähigkeiten. Er differenziert zwischen zwei Arten von Fähigkeiten: einfache (*simple skills*) und zusammengesetzte (*composite skills*) Fähigkeiten. Composite Skills sind Fähigkeiten, die aus einfachen oder anderen composite skills bestehen [\[3\]](#). Sie ermöglichen dem Benutzer die Lösung von komplexen Aufgaben. Composite Skills sind schwierig zu erlernen, in ihrer Anwendung auf einen bestimmten Kontext beschränkt und erfordern vom Benutzer einen hohen *cognitive load*<sup>44</sup>. Die Ausübung eines Klicks mit der Maus ist ein composite skill. Der Benutzer muss über die Fähigkeiten verfügen, die Maus zu halten und zu bewegen sowie die Fähigkeit besitzen, mit der Maus ein Ziel auf der Benutzeroberfläche zu treffen. Um diese Fähigkeiten nutzen zu können, muss der Benutzer ein

---

<sup>44</sup>Cognitive load ist das Maß für die Auslastung des working memorys während der Ausübung einer Aufgabe. Das Konzept beschreibt unter Berücksichtigung der limitierten Kapazität des working memorys, wieviele Aufgaben ein Benutzer zur gleichen Zeit ausüben kann. [\[85\]](#)





### 5. Gestenbasierte Interfaces

---

konzeptuelles Verständnis von der Eingabemetapher besitzen. Einfache Fähigkeiten sind demgegenüber Fähigkeiten über die ein Benutzer bereits verfügt. Simple Skills zeichnen sich durch eine einfache Erlernbarkeit, einen geringen cognitive load sowie der Möglichkeit, sie in unterschiedlichen Anwendungskontexten anwenden zu können, aus. Die Eingabe mit der Hand auf einem Touchscreen ist ein simple skill. Die Anwendung erfordert vom Benutzer lediglich angeborene feinmotorische Fähigkeiten sowie eine Auge-Hand-Koordination. NUI sollten darauf ausgelegt sein, dass die Aufgaben die ein Benutzer stellt durch simple skills gelöst werden können.

#### **„NUI have direct interaction with content“**

Die dritte Aussage befasst sich mit der direkten Interaktion mit Inhalten. GUI nutzen Fenster, Menüs und Icons als primäre Interfaceelemente. Dies steht in direktem Kontrast zur obigen Aussage. Der Fokus der Interaktion liegt bei NUI auf der direkten Interaktionen mit dem Inhalt. Das bedeutet, dass Bedienelemente, wie Menüs, eine untergeordnete Rolle einnehmen. Im Mittelpunkt steht die direkte Manipulation von Inhalten. Um eine direkte Interaktion zu ermöglichen, können drei verschiedene Ansätze verfolgt werden. Abhängig von der Eingabemodalität kann eine direkte Interaktion durch räumliche Nähe, zeitliche Nähe oder parallele Aktionen ermöglicht werden. Zeitliche Nähe bedeutet, dass das Interface möglichst zeitnah auf die Eingaben des Benutzers reagiert. Die räumliche Nähe beschreibt die physische Nähe zwischen Eingabegerät und Ausgabe am Interface (z. B. bei Touchscreens). Parallele Aktionen ist die Abbildung einer oder mehrerer Freiheitsgrade der Benutzeraktionen mit einer oder mehrerer Freiheitsgrade der Interfacereaktion [3]. Am Beispiel eines Multitouch-Tisches wird die direkte Interaktion deutlich: Der Benutzer berührt den Touchscreen (räumliche Nähe), das Interface reagiert unmittelbar auf dessen Eingabe (zeitliche Nähe). Die horizontalen und vertikalen Bewegungen der Finger werden direkt auf die horizontalen und vertikalen Bewegungen des Objekts abgebildet (*parallel action*).

Die direkte Interaktion mit Inhalten bietet dem Benutzer einige Vorteile. Die Reduzierung auf Inhalte bei NUI erfordert vom Benutzer einen geringen cognitive load. Dies steht im Gegensatz zu GUI, wo dem Benutzer eine Vielzahl von Funktionen und Bedienelementen angeboten werden, die ihm aber einen hohen cognitive load abverlangen. Eine weitere Reduzierung der grafischen Elemente wird beim NUI durch die kontext-sensitive Verwendung von Funktionen erreicht. Funktionen werden bei NUI abhängig vom Kontext dargestellt und sind nicht wie bei GUI allgegenwärtig. Aufgrund des geringen cognitive loads ist der Benutzer bei NUI in der Lage, viele In-



teraktion in einer kurzen Zeit auszuüben. **BLAKE** bezeichnet dies als *high frequency interaction*.

## 5.4. Anwendungsbeispiele

Der vorangehende Abschnitt hat die Eigenschaften eines Natural User Interfaces beschrieben. In diesem Abschnitt werden bereits verfügbare Produkte vorgestellt, die dem Benutzer eine direkte Interaktion durch NUI ermöglichen. Dazu werden Produkte aus drei unterschiedlichen Anwendungsgebieten betrachtet. Abschnitt 5.4.1 stellt, exemplarisch für die meisten Smartphones, das *iPhone* von Apple vor. Danach wird in Abschnitt 5.4.2 der Multitouch-Tisch *Surface* von Microsoft beschrieben. Abschließend wird in Abschnitt 5.4.3 das NUI der *Xbox360* unter Verwendung der Kinect behandelt. Ziel dieses Kapitels ist es, einen Überblick über bereits bestehende Produkte zu geben, deren Interaktion auf NUI basiert. Des Weiteren werden exemplarische Aufgaben und deren Zielerreichung am NUI beschrieben.

### 5.4.1. Apple iPhone

Das *iPhone* vom US-amerikanischen Hersteller Apple ist ein mobiles Telefon, das über ein Multitouch-Display verfügt (Vgl. Abbildung 5.2). Der Begriff Multitouch-Interaktion bedeutet, dass Interaktionen durch mehrere zeitgleiche Berührungen (z. B. mehrere Finger) auf einer sensitiven Oberfläche registriert werden. Das *iPhone* gilt dabei als eines der bekanntesten Produkte mit Multitouch Fähigkeit. Die Bedienung erfolgt beim iPhone primär durch Interaktion mit dem Display anhand Single- oder Multitouch-Gesten. Die unmittelbare, direkte Interaktion zwischen Mensch und „Computer“ wird demnach unterstützt. Die Anmeldung geschieht beim iPhone durch die Eingabe eines persönlichen, vierstelligen Passwortes. Alternativ kann das Passwort auch durch eine Geste eingegeben werden, indem eine durchgehende Linie über den vierstelligen Zahlencode gezogen wird. Das „Hauptmenü“ des iPhones bietet einen Überblick über die wichtigsten Funktionalitäten und dient der Orientierung. Standardmäßig werden Funktionen zum Telefonieren, Email, Internet, Musik und andere Apps angeboten. Apps (Kurzform für *engl.: application*) sind Anwendungsprogramme aus den unterschiedlichsten Bereichen (z. B. Business, Sport, Lifestyle). Bei NUI besteht grundsätzlich der Gedanke, sich von Anwendungsprogrammen, wie wir sie heute kennen, zu lösen. Zur Lösung eines konkreten Problems nehmen



### 5. Gestenbasierte Interfaces

kontext-sensitive Dienste die Rolle der Anwendungsprogramme ein. Viele der angebotenen Apps entsprechen bereits der Definition eines Dienstes. So ist es anhand einer entsprechenden „Wetter-App“ möglich, aktuelle Wetterinformationen abzurufen.



Abbildung 5.2.: Apple iPhone 4S<sup>45</sup>

Bei der Eingabe eines Textes wird der Benutzer in seiner Interaktion kontext-sensitiv unterstützt. Anhand der markierten oder nicht markierten Objekte wird entschieden, welche Möglichkeiten dem Benutzer zur Interaktion angeboten werden. Ist nichts markiert, so stehen dem Benutzer die Funktionen *select*, *select all*, *paste* zur Verfügung. Für markierte Objekte bestehen dagegen die kontext-sensitiven Funktionen *cut*, *copy*, *paste*.

#### 5.4.2. Microsoft Surface

Vom US-amerikanischen Hersteller Microsoft wird der Multitouch-Tisch *Surface* angeboten (Vgl. Abbildung 5.3). Die Interaktion mit digitalen Objekten erfolgt ähnlich wie beim iPhone über eine berührungssensitive Multitouch Oberfläche. Der Ansatz zur Verwendung von Multitouch Eingaben geht dabei auf Untersuchungen von REKIMOTO [72] zurück. Das Anwendungsfeld der Surface Tische besteht vorwiegend in Informations- sowie Unterhaltungsangeboten. Eine wichtige Funktionalität des Surface besteht im berührungsfreien, kabellosen Datenaustausch mit anderen Endgeräten. Für die Interaktion mit physischen Gegenständen aus der realen Welt müssen diese mit einem „Tag“<sup>46</sup> versehen werden. Die Möglichkeit mit physischen Objekten aus der realen Welt zu interagieren, ist ein weiterer Schritt, die reale mit der virtuellen Welt zu verschmelzen: Legt man eine compatible Kamera auf den Surface, so werden die gespeicherten Bilder und Videos der Kamera unmittelbar auf dem Surface dargestellt. Anhand der Interaktion über Multi- oder Singletouch können im

<sup>45</sup>Website: <http://www.apple.com/de/iphone/>

<sup>46</sup>Ähnlich einem Barcode; Informationen zu Tags:

[http://msdn.microsoft.com/en-us/library/ee804823\(v=surface.10\).aspx](http://msdn.microsoft.com/en-us/library/ee804823(v=surface.10).aspx)



### 5. Gestenbasierte Interfaces

---

Anschluss kontext-sensitive Funktionen mit den Bildern und Videos durchgeführt werden, wie z. B. Verschieben, Skalieren oder Rotieren. Legt man eine zweite compatible Kamera auf den Surface, besteht die Möglichkeit, Objekte unmittelbar über Zeigegesten von einem Endgerät zum anderen zu übertragen. Ähnlich wie das iPhone verwendet das Microsoft Surface keine Anwendungsprogramme, sondern Dienste, um den Benutzer in seinen konkreten Problemstellungen zu unterstützen. Großes Potential besteht beim Microsoft Surface im Bereich der Kollaboration. Hierzu existieren verschiedene Szenarien: mehrere Benutzer können gleichberechtigt in einem Dienst interagieren oder interagieren in verschiedenen Diensten. In diesem Fall werden die Benutzer vom System unterschieden, nutzen aber das gleiche Basissystem.



Abbildung 5.3.: Multitouch Tisch Microsoft Surface<sup>47</sup>

#### 5.4.3. Microsoft Xbox360

Die Spielekonsole *Xbox360* stammt wie das Surface vom US-amerikanischen Hersteller Microsoft. Seit Anfang 2010 wird zusätzlich die Kinect als Eingabegerät angeboten. Die Kinect ermöglicht die Steuerung von Anwendungen anhand von natürlichen Gesten und Sprachkommandos. Die Kinect ist demnach ein multimodales System, das die Interaktion über mehrere Sinneskanäle unterstützt. Im Rahmen der Veröffentlichung der Kinect wurde auch die Benutzeroberfläche der Xbox360 überarbeitet, die in Abbildung 5.4 dargestellt ist. Die Benutzeroberfläche wurde dabei auf das *Metro-Design*, das auch vom Windows-Phone und dem noch nicht veröffentlichten Windows 8 verwendet wird, umgestellt. Das Hauptmenü bietet Anbindungen

---

<sup>47</sup>Website: <http://www.microsoft.com/surface/>



### 5. Gestenbasierte Interfaces

zu sozialen Netzwerken wie Facebook oder Twitter, Videos, Spiele und Apps. Die gestenbasierte Interaktion wird vorrangig über Zeigegesten und Wischgesten ermöglicht. Um den Benutzer bei der gestenbasierten Eingabe einen Orientierungspunkt auf der Benutzeroberfläche zu bieten, repräsentiert eine virtuelle Hand die aktuelle Position der realen Hand. Um Objekte zu aktivieren, muss der Benutzer seine Hand über das Objekt bewegen und dort für 2 Sekunden verweilen. Ein teiltransparenter Ladekreis zeigt dem Benutzer dabei an, dass die Eingabe bearbeitet wird. Neben der gestenbasierten Interaktion unterstützt die Kinect auch die sprachbasierte Interaktion. Über das definierte Sprachkommando „Xbox“ wird die Spracherkennung der Kinect aktiviert. Danach wird dem Benutzer eine Liste der kontext-sensitiven Sprachkommandos angezeigt. Bei Filmen können z. B. die Sprachkommandos *play*, *pause*, *fast-forward*, *rewind* verwendet werden.

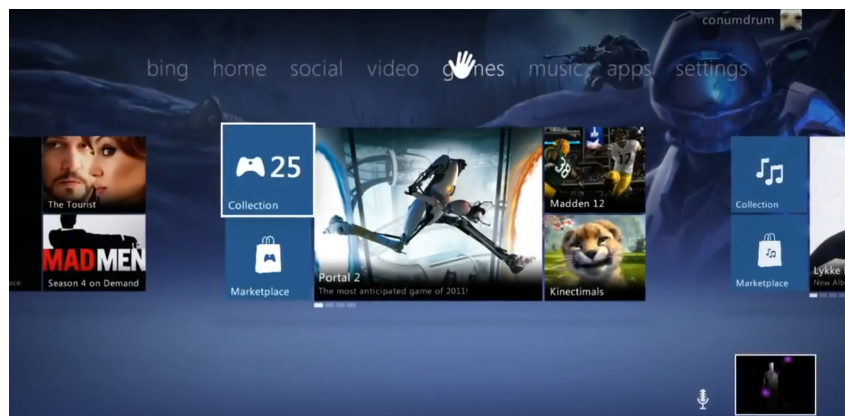


Abbildung 5.4.: Benutzeroberfläche der Xbox360 unter Verwendung der Kinect<sup>48</sup>

## 5.5. Ergonomie gestenbasierter Interfaces

Beim NUI lösen die Hände die Maus als primäres Eingabemedium ab. Die bisherigen Normen<sup>49</sup> zur Ergonomie von Computer Arbeitsplätzen befassen sich ausschließlich mit der Verwendung von Maus und Tastatur an einem Bildschirm. Die ISO 9241-9 [80] enthält Anforderungen und Empfehlungen für Eingabegeräte abgesehen von Tastaturen. Darunter fallen z. B. Maus, Joystick, Trackball oder Touchscreen. Die berührungssensitive Eingabe wird in dieser Norm beschrieben, nicht aber die berührungsfreie Eingabe im dreidimensionalen Raum.

<sup>48</sup>Website: <http://www.xbox.com/de-DE>

<sup>49</sup>EG-Richtlinie 90/270/EWG, [http://www.gaa.baden-wuerttemberg.de/servlet/is/16050/2\\_1\\_05.pdf](http://www.gaa.baden-wuerttemberg.de/servlet/is/16050/2_1_05.pdf)



### 5. Gestenbasierte Interfaces

---

Dieser Abschnitt beschäftigt sich mit der Ergonomie gestenbasierter Systeme. Dazu wird zu Beginn eine Definition des Begriffs Ergonomie vorgestellt. Diese wird im Anschluss um die Definition Software-Ergonomie im Kontext MCI erweitert. Des Weiteren werden Untersuchungen vorgestellt, die sich mit der Ergonomie von gestenbasierten Systemen beschäftigt haben.

Unter dem Begriff *Ergonomie* versteht man die Wissenschaft der Gesetzmäßigkeiten menschlicher Arbeit. Ziel der Ergonomie ist die Anpassung der Arbeitsbedingungen an die menschlichen Fähigkeiten und Fertigkeiten. [FRÖHLICH UND J.](#) definieren den Begriff Ergonomie wie folgt:

„Der Begriff Ergonomie ist die zusammenfassende Bezeichnung für ein interdisziplinäres Forschungsgebiet, das sich sowohl mit den funktionellen Grenzen und Möglichkeiten des Menschen im Bereich spezifischer Arbeitsplätze und Arbeitsabläufe als auch mit den Bedingungen und Methoden der Arbeitsplatzgestaltung und Arbeitsorganisation befasst, die als Voraussetzung für die Aufrechterhaltung der Arbeitsfähigkeit, Gesundheit und Sicherheit gelten.“[\[21\]](#)

Der Begriff Ergonomie beschreibt in seinem Ursprung die Arbeitsplatzgestaltung an Maschinen, den körperlichen Einsatz sowie Einflüsse der Umgebung. Mit der Entwicklung der Personal Computer wurde der Begriff Ergonomie auf Software-Ergonomie (*engl.: usability engineering*) ausgeweitet. Nach [EBERLEH U. A.](#) umfasst Software-Ergonomie die benutzergerechte Gestaltung der MCI, d.h. die Gestaltung der wesentlichen Aspekte eines interaktiven Computersystems, die von Software gesteuert wird und an einer Benutzeroberfläche wirksam wird. Die Benutzeroberflächen mit denen der Benutzer unmittelbar agiert, sind ein wichtiger Gegenstand der Gestaltung [\[15\]](#).

Die Verwendung von berührungsfreien Gesten ist mit einer körperlichen Anstrengung verbunden. Anders als bei der Eingabe mit Maus und Tastatur erfordert die berührungsfreie Interaktion ein Ausstrecken der Arme. Für die Gestaltung von Gesten müssen diese ergonomischen Aspekte berücksichtigt werden. [NIELSEN U. A.](#) befassen sich in ihrer Arbeit mit ergonomischen Grundsätzen von Gesten. Auf Grundlage von biologischen und biomechanischen Untersuchungen stellen sie folgende Grundsätze auf [\[58\]](#):

- Vermeidung von äußeren Positionen. Keine Überstreckung



- Ständige Wiederholungen vermeiden
- Muskeln Gelegenheit zur Entspannung geben
- Entspannte Haltung ermöglichen
- Statische Haltungen vermeiden
- Externe sowie interne Krafteinwirkungen auf Gelenke vermeiden

Die vorgestellten Grundsätze haben einen großen Einfluss auf die technischen Charakteristika eines berührungsfreien, gestenbasierten Systems. Bei der Erfassung von Gesten muss das System eine gewisse Toleranz gegenüber der gestenbasierten Eingabe des Benutzers besitzen. Zu geringe Toleranzen führen zu Fehleingaben, wodurch der Benutzer die Geste wiederholen muss. Zu hohe Toleranzen führen dagegen zur Erfassung nicht beabsichtigter Gesten, wodurch der Benutzer eine Korrektur durchführen muss. Des Weiteren muss der Erfassungs-Algorithmus tolerant gegenüber Auflockerungsbewegungen sein: Auflockerungsbewegungen dürfen nicht als Gesten erkannt werden. Gesten sollten daher möglichst durch einfache Handbewegungen dargestellt werden.

## 5.6. OCGM - Interface Metaphern

Die Interaktion mit GUI basiert auf den vier Komponenten Window, Icon, Menu und Pointer. WIMP stellt die grundlegenden Metaphern eines GUI dar, über die der Benutzer mit dem Interface interagiert. Für NUI existieren dagegen noch keine allgemeingültigen Metaphern. Aufgrund des noch jungen Forschungsgebietes der NUI existieren bisher nur wenige Ansätze, die ähnlich zu WIMP ein Grundkonzept für NUI beschreiben. In diesem Abschnitt soll die OCGM-Metaphern vorgestellt werden. OCGM ist ein Akronym und setzt sich aus *Object*, *Container*, *Gestures*, *Manipulations* zusammen. OCGM beschreibt die grundlegenden Metaphern eines NUI und wurde 2010 von [BLAKE UND GEORGE](#) vorgestellt.

GUI verwenden die Desktop-Metapher um dem Benutzer ein Verständnis über die Interaktion mit dem Computer zu geben. Metaphern sind bildhafte Repräsentationen eines abstrakten Konzepts. Darüber hinaus sind Metaphern die Trigger, die darüber entscheiden, welches mentale Modell vom Benutzer verwendet wird.

*Objects* sind die Metapher für den Inhalt. *Container* sind die Metaphern für die Beziehung zwischen mehreren Inhalten. *Gestures* sind nach [BLAKE UND GEORGE](#) die





Metaphern für diskrete, indirekte, intelligente Interaktionen. *Manipulations* stellen die Metaphern für direkte, unmittelbare Interaktionen mit der Umgebung dar [4]. Eine Darstellung der OCGM-Metaphern wird in Abbildung 5.5 gegeben.

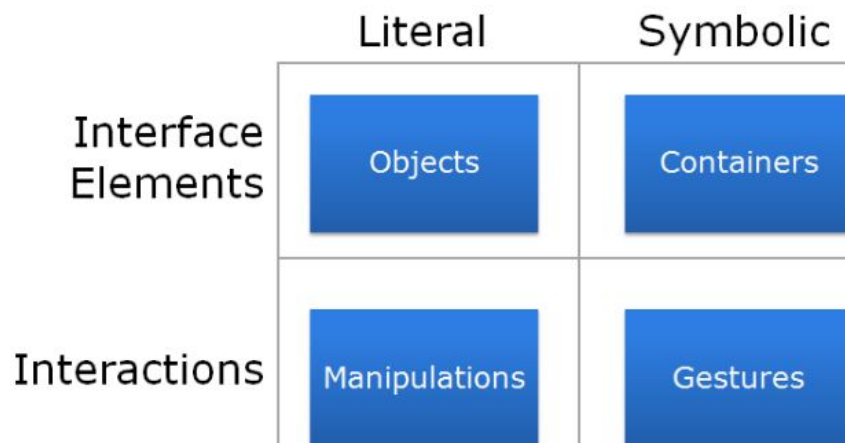


Abbildung 5.5.: OCGM Metaphern nach BLAKE UND GEORGE [4]

Ausgehend von Abbildung 5.5 sollen die grundlegenden Metaphern näher beschrieben werden.

- **Objects**

Objekte sind die Basiselemente eines NUI, die den eigentlichen Inhalt bzw. die Daten darstellen. Jedes einzigartige Stück Inhalt wird am Interface als ein Objekt dargestellt. Objekte besitzen eine 1:1 Beziehung mit dem Inhalt.

- **Container**

Container sind die Metaphern für die Beziehung zwischen verschiedenen Objekten und können als „Sammelbehälter“ für Objekte verstanden werden. Container sind nicht zwangsläufig hierarchisch organisiert. Sie sollen dem Benutzer dabei helfen, die Beziehungen zwischen den Objekten in möglichst natürlicher Form zu verstehen. Abhängig von den Objekten, die im Container vorliegen, kann die Beziehung durch Räumlichkeit, alphabetische Reihenfolge oder Gruppierungen hervorgehoben werden.

- **Gestures**

Gesten repräsentieren eine indirekte, diskrete, symbolische Aktion des Benutzers. Indirekt und diskret, weil eine Geste eine Bewegung über die Zeit darstellt, die erst abgeschlossen werden muss, um die Bedeutung zu transportieren. Die Reaktion des Systems erfolgt erst nach kompletter Ausführung der





Geste. Gesten sind nach [BLAKE UND GEORGE](#) symbolischer Natur und erfordern vom Benutzer einen gewissen Lernaufwand. Gesten eignen sich daher für komplexere Aufgaben.

- **Manipulations**

Manipulationen sind direkte, kontinuierliche Aktionen, die eine unmittelbare Reaktion des Systems zur Folge haben. Manipulationen basieren auf den natürlichen Handlungen des Menschen. Im Gegensatz zu Gesten haben Manipulationen ein direktes Feedback des Systems zur Folge. Benutzer sind dadurch in der Lage, das Interface ohne unverhersehbare Konsequenzen zu erforschen.

[BLAKE UND GEORGE](#) begründen den OCGM-Ansatz anhand kognitionspsychologischer Untersuchungen bei Kleinkindern. Ein neun Monate altes Kind besitzt laut [BLAKE UND GEORGE](#) bereits alle kognitiven Fähigkeiten, um die OCGM-Metaphern zu verstehen. Sie verweisen dabei auf Arbeiten, die sich mit dem Verhalten von Kleinkindern bezüglich der Kategorisierung von Objekten [23], der Manipulation von Objekten [36] und der Erfassung und Ausübung von Zeigegesten [93] beschäftigen. Im Gegensatz dazu stehen die kognitiven Fähigkeiten, die für ein Verständnis der WIMP Metapher notwendig sind. Die notwendigen Fähigkeiten werden erst in einem Vorschulalter erlernt. Das kognitive Verständnis über Icons wird erst in einem Alter von dreieinhalb Jahren entwickelt [87]. Des Weiteren besteht zwischen Kindern und Erwachsenen ein großer Unterschied in der effektiven Nutzung der Maus als Eingabegerät [31]. Ein typisches GUI Menü zu bedienen, erfordert die Fähigkeit lesen zu können. Die Fähigkeit des Lesens wird in der Regel erst in einem Alter von fünf bis sechs Jahren erlangt.

OCGM stellt einen Ansatz zur Definition von Basis-Metaphern für NUI dar. Sie sind so gewählt, dass sie bereits in einem frühen Alter verstanden werden können. Der cognitive load bei Interfaces die OCGM-Metaphern verwenden ist dementsprechend gering. Für die Entwicklung von NUI Anwendungen/Diensten sind weitere Metaphern nötig, die anhand des Kontextes und Inhalts entwickelt werden müssen.

## 5.7. Zwischenfazit

In diesem Kapitel wurde das Konzept der NUI vorgestellt. Dazu wurde zu Beginn ein Überblick über bestehende Ansätze zur Definition gegeben. Die Definition von [BLAKE](#) wird dabei als angemessen erachtet, weil sowohl die direkte Interaktion mit Inhalten als auch die Verwendung bereits erlernter Fähigkeiten des Menschen berücksichtigt werden. Im Hinblick auf Multimodalität macht diese Definition Sinn,



### 5. *Gestenbasierte Interfaces*

---

da Sprache eine Fähigkeit ist, die der Mensch erst im Verlauf seines Lebens erlernt. Des Weiteren geht [BLAKE](#) in seiner Definition auf den Gestaltungsvorgang ein. Die Gestaltung NUI erfordert vom Designer ein Umdenken. Im Mittelpunkt des NUI-Designs steht die Gestaltung der Interaktion. Dies stellt einen Gegensatz zum bisherigen GUI-Design dar. Ziel beim NUI-Design ist eine angemessene Reduzierung der Interface-Elemente, um die direkte Interaktion mit Inhalten für den Benutzer zu gewährleisten. Die für GUI eingesetzte WIMP-Metapher ist damit nicht effektiv einsetzbar, obwohl bestimmte Interaktionselemente wie Scrollbars auch bei NUI verwendet werden können. Die Metapher wird bei NUI nur in anderer Form genutzt. Bei GUI ermöglicht die Scrollbar die Navigation in bestimmten Inhalten. NUI ermöglichen anhand ihrer Eingabemodalitäten die Navigation anhand von Gesten. Scrollbars dienen bei NUI demnach nur als Positions-Indikator für den Benutzer.

Aufgrund des Wegfalls der WIMP-Metapher müssen alternative Interaktionsmetaphern für NUI definiert werden. Dazu wurde das Konzept der OCGM-Metaphern vorgestellt. Die OCGM-Metaphern umfassen die grundlegenden Interfaceelemente und Interaktionsformen für ein NUI. Die abstrakten Metaphern sind so gewählt, dass sie vom Benutzer bereits früh erlernt bzw. verstanden werden. Sie stellen jedoch nur einen Startpunkt für die Gestaltung eines NUI dar. Für konkrete Szenarien müssen weitere Metaphern definiert werden. Dies gilt es in weitergehenden Forschungen zu untersuchen.

Die Erforschung von NUI befindet sich derzeit noch in einem frühem Stadium. Wie bereits in der Einleitung erwähnt, konzentrieren sich die meisten Untersuchungen auf die technische Realisierung der Eingabemodalitäten. Der Wegfall der Tastatur bringt jedoch einige Probleme in spezifischen Eingabeszenarien mit sich. Die Kombination mehrerer Modalitäten (Sprache und Gestik) stellt einen Ansatz zur Lösung dieses Problems dar.



## 6. Konzeption

In diesem Kapitel wird die Konzeption eines NUI beschrieben. Dazu wird zu Beginn in Abschnitt 6.1 die allgemeine Vorgehensweise bei der Konzeption erklärt. Darauf aufbauend widmen sich die folgenden Abschnitte 6.1.1 und 6.1.2 der Gestaltung der Interface-Elemente. Danach wird in Abschnitt 6.1.3 das Verhalten der Interface-Elemente erklärt. Die Farbgebung wird in Abschnitt 6.1.4 behandelt. Nachdem das Vorgehen bei der Gestaltung der Benutzeroberfläche beschrieben wurde, wird in Abschnitt 6.2 das verwendete Gestenvokabular festgelegt und begründet. Im abschließenden Abschnitt 6.3 wird auf Grundlage der festgelegten Vorgehensweise ein NUI für ein Studenteninformationssystem vorgestellt.

### 6.1. Vorgehensweise

Um Menschen bei ihrer Problemlösung in geeigneter Form zu unterstützen, müssen sowohl Eingabe als auch die Interaktion mit Elementen der Benutzeroberfläche auf natürliche Art und Weise geschehen. Der Ausgangspunkt für die Gestaltung einer natürlichen Interaktion ist das OSIT-Handlungsmodell<sup>50</sup>. Das OSIT-Handlungsmodell gilt sowohl für menschliche Handlungen in der realen, als auch in der virtuellen Welt. Darüber hinaus ist das Modell unabhängig von Alter und kultureller Herkunft gültig. Bei der Gestaltung der Benutzeroberfläche wird dieses Modell berücksichtigt. Der Benutzer soll so in die Lage versetzt werden, anfallende Aufgaben intuitiv zu lösen.

Anhand der in Abschnitt 5.6 auf S. 56 vorgestellten OCGM-Metaphern sollen die Interface-Elemente und die Interaktionen gestaltet werden. Die OCGM-Metaphern sind so gewählt, dass sie bereits in einem frühem Alter erlernt und verstanden werden. Die Interaktion mit dem System basiert auf der Verwendung von Handgesten. Die Benutzeroberfläche wird speziell für diese Eingabemodalität gestaltet. Die folgenden Abschnitte gehen nun genauer auf ausgewählte Aspekte bei der Gestaltung

---

<sup>50</sup>Vgl. Kapitel 3.3, S. 14



## 6. Konzeption

der Benutzeroberfläche ein.

### 6.1.1. Größe der Interface-Elemente

Die Größe der Interface-Elemente ist speziell für die berührungsfreie Interaktion von großer Bedeutung. Nach dem Gesetz von Fitts, beschreibt die Gleichung:  $T = a + \log_2 \left( \frac{S}{D} + \frac{1}{2} \right)$  die benötigte Zeit, um ein Zeigegerät aus dem Abstand  $S$  auf ein Ziel mit dem Durchmesser  $D$  zu positionieren. Daraus ergibt sich, dass Ziele mit größerem Durchmesser bei gleichem Abstand, schneller zu treffen sind. Die Interface-Elemente werden aus diesem Grund ausreichend groß dargestellt. Ferner besteht die Möglichkeit, die Interface-Elemente aufgrund ihrer Relevanz bzw. ihres Kontextes unterschiedlich groß darzustellen. Einen Vergleich zwischen GUI und NUI anhand Größe und Anzahl der Interface-Elemente wird in Abbildung 6.1 gegeben.

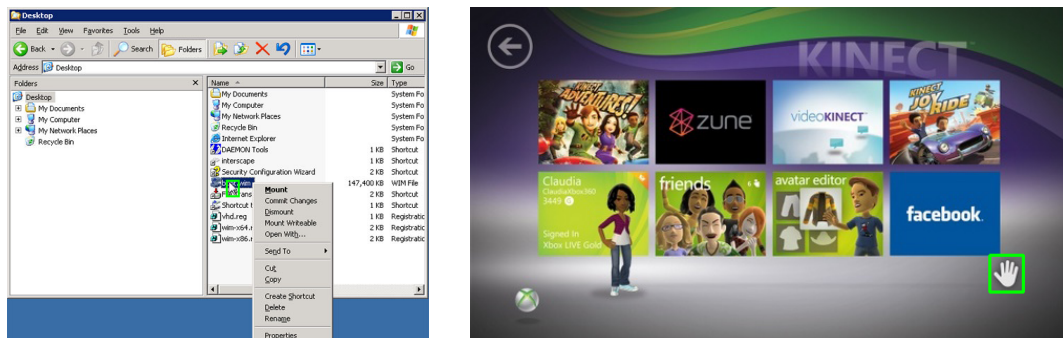


Abbildung 6.1.: Vergleich von Größe und Anzahl der Interface-Elemente bei GUI (links) und NUI (rechts)

### 6.1.2. Anordnung der Interface-Elemente

Die Anordnung der Interface-Elemente wird auf Grundlage der Ergebnisse in Abschnitt 5.5: Ergonomie von NUI getroffen. Bei der berührungsfreien Interaktion nimmt der Benutzer eine besondere Ausgangshaltung ein. In der Ausgangshaltung streckt er seinen Arm in einem 90° Winkel von sich weg. Auf dem Interface wird diese Ausgangshaltung durch einen im Zentrum dargestellten Handcursor repräsentiert. Um Anstrengungen weitestgehend zu vermeiden, werden die Interface-Elemente möglichst zentral angeordnet. Wichtigere Inhalte werden demnach zentraler angeordnet als weniger wichtigere Inhalte. Des Weiteren kann durch die Anordnung der Interface-Elemente ein struktureller Zusammenhang transportiert werden.



## 6. Konzeption

Nah beieinander liegende Elemente werden vom Benutzer als Einheit bzw. als zusammenhängend wahrgenommen.

### 6.1.3. Verhalten der Interface-Elemente

In der MCI wurde erkannt, dass das Aussehen (Look) eine genauso wichtige Rolle einnimmt wie das Verhalten (Feel) der Anwendung. HENSELER beschreibt im Rahmen des Paradigmenwechsels von GUI zu NUI eine Verlagerung dieses Schwerpunktes vom Look zum Feel [28]. Es besteht demnach die Notwendigkeit, das Verhalten der Anwendung möglichst natürlich zu gestalten. Für die konzipierte Anwendung bedeutet dies, dass der Benutzer zu Beginn in der Orientierungsphase unterstützt werden muss. Dies wird durch wenige dargestellte Interface-Elemente erreicht. In der darauf folgenden Selektionsphase wird der Benutzer „näher“ an die Inhalte herangeführt. Der selektierte Inhalt bewegt sich auf den Benutzer zu, alle anderen Inhalte rücken in den Hintergrund oder verschwinden. Des Weiteren sollten sich virtuelle Objekte in ihrem Verhalten an ihren realen Vorbildern orientieren.

### 6.1.4. Farbgebung

Bei der Wahl der Farben wurde auf das Farbklima<sup>51</sup> der Medieninformatik, Fachhochschule Köln, zurückgegriffen. Eine grafische Darstellung des verwendeten Farbklimas wird in Abbildung 6.2 gegeben. Die unterschiedliche Farbgebung der Interface-Elemente ermöglicht dem Benutzer, strukturelle Zusammenhänge zu erkennen. Die unterschiedlichen Farben werden in der Konzeption als Orientierungshilfe und zur Unterscheidung von Objekten eingesetzt.

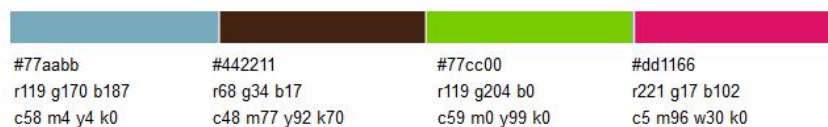


Abbildung 6.2.: Für die Konzeption verwendetes Farbklima

<sup>51</sup>Website: [www.medieninformatik.fh-koeln.de/website/general/general/designguide\\_3/de/de\\_designguide\\_articl\\_1.php](http://www.medieninformatik.fh-koeln.de/website/general/general/designguide_3/de/de_designguide_articl_1.php)



## 6.2. Gestenbasierte Eingaben

Die berührungsfreie Interaktion besitzt den Nachteil, dass der Benutzer kein haptisches Feedback bekommt. Der Benutzer muss dennoch jederzeit in der Lage sein, die Position seiner Hand auf dem Interface nachzuvollziehen. Nach [MOESLUND U. A.](#) hat das visuelle Feedback bei Zeigegesten großen Einfluss auf die Genauigkeit der ausgeführten Geste. Aus diesem Grund wird dem Benutzer ein visuelles Feedback in Form eines Hand-Cursors gegeben (Vgl. Abbildung 6.3 S. 65). Diese Darstellungsform wird in ähnlicher Art und Weise auch bei der Xbox360<sup>52</sup> oder der Nintendo Wii verwendet. Für die Konzeption wurde die Hand eines Rechtshänders verwendet. Die OCGM-Metaphern sehen Gesten und Manipulationen als Interaktionsmöglichkeiten bei NUI vor. Das konzipierte Interface verwendet drei Eingabemöglichkeiten: Zeigegesten zur Navigation, Push-Gesten zur Bestätigung von Selektionen sowie Wink-Gesten zur Anmeldung am System. Zeigegesten bzw. deiktische Gesten sind nach [BLAKE](#) Manipulationen. Sie ermöglichen eine direkte Interaktion. Die direkte Interaktion bei Zeigegesten besteht in der zeitlichen Nähe<sup>53</sup>. Das Interface reagiert unmittelbar auf die Eingaben des Benutzers. Die Bestätigung von Selektionsvorgängen wird durch Push-Gesten ermöglicht. Push-Gesten sind Bewegungen über die Zeit, die erst abgeschlossen werden müssen, um eine Bedeutung zu transportieren. Eine Systemreaktion erfolgt daher erst nach kompletter Ausführung der Geste. Die Verwendung von Push-Gesten muss dem Benutzer zu Beginn erklärt werden. Die Anmeldung am System erfolgt durch die Ausführung einer Wink-Geste. Die Wink-Geste ist ebenfalls eine indirekte Interaktionsmöglichkeit und muss dem Benutzer zu Beginn erklärt. Bei der Wahl der Gesten für das konzipierte System wurde darauf geachtet, dass sich die verwendeten Gesten ausreichend unterscheiden. Die Push-Geste ist eine alternierende Bewegung zur Z-Achse, während die Wink-Geste eine alternierende Bewegung zur X-Achse ist. Fehlinterpretationen der Gestenerkennung werden damit nahezu ausgeschlossen.

## 6.3. Anwendungsbeispiel

Nachdem die grundlegende Vorgehensweise beschrieben wurde, wird in diesem Abschnitt ein Anwendungsbeispiel gegeben. Als Anwendungsszenario dient ein Informationssystem für Studenten. Das System soll Studenten in informationellen und

---

<sup>52</sup>Vgl. Kapitel 5.4.3 S. 53

<sup>53</sup>Vgl. Kapitel 5.3 S.48



## 6. Konzeption

---

organisatorischen Problemstellungen unterstützen. Das konzipierte NUI könnte als berührungssensitive oder berührungsfreie Terminalanwendung in einer folgenden Arbeit umgesetzt werden.

Die prototypische Gestaltung der Benutzeroberfläche wurde in Adobe Photoshop<sup>54</sup> umgesetzt. Die folgenden Abschnitte gehen detailliert auf die Gestaltung des NUI ein und beschreiben anhand von Bildfolgen, wie die Interaktion in einem konkreten Szenario ablaufen kann. Dabei wird auf Eingaben des Benutzers, Systemreaktion sowie auf Möglichkeiten und Grenzen des Systems Bezug genommen. Aufgrund der ähnlichen Verhaltensweise beschränkt sich die Beschreibung auf die Container: *Aktuelles* und *Mensa*. Die restlichen konzipierten Container werden als kommentierte Bildfolgen im Anhang: A.2 angeboten.

### 6.3.1. Startoberfläche

Die *Startoberfläche* ermöglicht eine erste Orientierung über die angebotenen Inhalte, die für Studenten von Interesse sein könnten. Die Container, die den direkten Zugriff auf die Inhalte ermöglichen, sind unterschiedlich groß gestaltet. Große Container repräsentieren primäre Inhalte, kleine Container repräsentieren sekundäre Inhalte. Die primären Container bieten Zugriff auf Inhalte bezüglich Personen, Vorlesungen, aktuelle Nachrichten und Räume. Die Beschreibung des jeweiligen Inhalts wird über Text und Icon dargestellt. Die sekundären Container umfassen die Inhalte Schwarzes Brett, Campus-Kalender und Speisekarte der Mensa. Der jeweilige Inhalt wird allein durch Icons beschrieben. Darüber hinaus wird in der rechten, unteren Ecke die aktuelle Uhrzeit und das Datum als Objekt dargestellt. Die Startoberfläche ist in Abbildung 6.3 dargestellt.

### 6.3.2. Aktuelles

Der Container *Aktuelles* bietet Zugriff auf aktuelle Nachrichten, die als Objekte hinterlegt sind. Eine Positionierung der Hand<sup>55</sup> über dem Container hebt diesen hervor (Vgl. Abbildung 6.4). Der selektierte Container wird größer, rückt mehr in den Vordergrund. Alle anderen Interface-Elemente werden weniger sichtbar dargestellt. Die Gestaltung dieses grafischen Übergangs orientiert sich am menschlichen

---

<sup>54</sup>Website: <http://www.adobe.com/de/products/photoshop.html/>

<sup>55</sup>gemeint ist die Positionierung des *Handcursors* auf dem Interface. Der Einfachheit wird in den folgenden Abschnitten von einer Positionierung der Hand gesprochen.



6. Konzeption

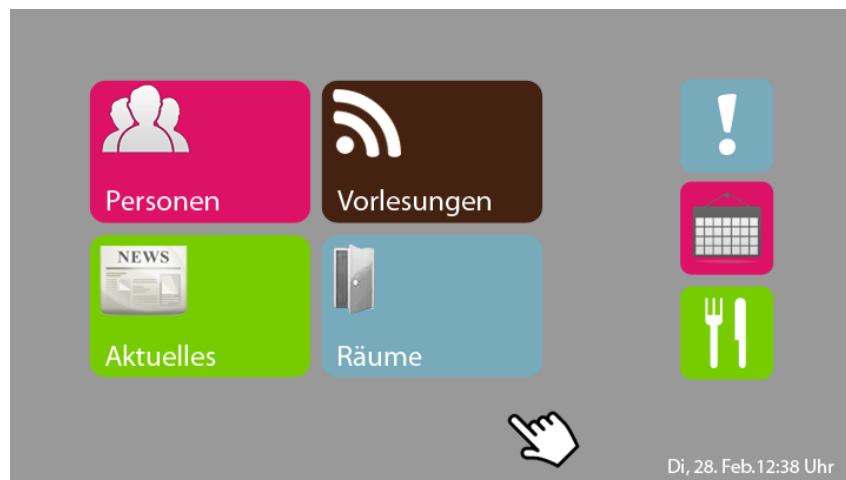


Abbildung 6.3.: Startoberfläche des Informationssystems für Studenten

Sehen. Fokussierte Objekte werden scharf wahrgenommen, die Umgebung wird unscharf wahrgenommen. Dem Benutzer wird so ein Feedback über die erfolgreiche Selektion gegeben.



Abbildung 6.4.: Selektion von Container „Aktuelles“

Durch eine *Push-Geste* bestätigt der Benutzer seine Eingabe und wird näher zum Inhalt gebracht. Dies wird in Abbildung 6.5 dargestellt. Der grafische Übergang besteht in einem Zoom in den Container. In diesem Fall befinden sich im Container *Aktuelles* weitere Container, die Zugriff auf die Objekte und damit den Inhalt geben. Die Container bieten dabei eine Voransicht auf die enthaltenen Inhalte. Abhängig vom Kontext kann diese Voransicht unterschiedlich dargestellt werden. Ein Contai-





## 6. Konzeption

ner mit Video-Inhalten könnte z. B. ein oder mehrere Videos zur Voransicht liefern.



Abbildung 6.5.: Überblick über die Container innerhalb des Containers „Aktuelles“

Interessiert sich ein Benutzer für ein Objekt, positioniert er seine Hand über dem entsprechenden Container (Vgl. Abbildung 6.6). Analog zu Abbildung 6.4 rückt das selektierte Element in den Vordergrund. Die restlichen Elemente rücken halbtransparent in den Hintergrund. Durch eine Push-Geste bestätigt der Benutzer seine Eingabe und gelangt schließlich zum eigentlichen Objekt, das ihm den Zugriff auf den kompletten Inhalt bietet (Vgl. Abbildung 6.7).



Abbildung 6.6.: Selektion des Containers

Möchte sich der Benutzer weitergehend informieren, kann er durch eine Push-Geste zum vorigen Container gelangen. Die Ausführung der Push-Geste an der linken unte-



## 6. Konzeption



Abbildung 6.7.: Zugriff auf das Objekt

ren Ecke führt den Benutzer zur Startoberfläche zurück. Der Handcursor wird dabei an den Grenzen des Bildschirms abgefangen<sup>56</sup>. Die Farbe des „Eselsohrs“ entspricht der Hintergrundfarbe der Startoberfläche. Dem Benutzer wird durch die Farbwahl eine Hilfestellung zur Orientierung gegeben. GUI verwenden in der Regel ein Haus-Icon als Metapher, um zur Startoberfläche zu gelangen. Das Eselsohr wird häufig als Metapher für ein Lesezeichen verwendet. Die Farbe des Eselsohrs liefert dabei zusätzlich die Informationen darüber, wohin der Benutzer gelangt. Die Defokussierung zum vorigen Container oder zur Startoberfläche wird durch einen grafischen Übergang dargestellt. Es wird aus der Szene herausgezoomt.

### 6.3.3. Mensa

Der Container Mensa bietet Zugriff auf die Speisekarte der Mensa. Eine Positionierung der Hand über dem Container vergrößert diesen und ermöglicht eine Voransicht auf die enthaltenen Inhalte. Alle anderen Interface-Elemente werden weniger sichtbar dargestellt. In diesem Fall sind dies die erhältlichen Tagesgerichte. Dieser Interaktionsschritt wird in Abbildung 6.8 dargestellt.

Durch eine Push-Geste bestätigt der Benutzer seine Eingabe und wird näher zum Inhalt gebracht. Der grafische Übergang besteht in einem Zoom in den Container hinein. Die Darstellung der Objekte orientiert sich an ihren realen Abbildern. Die in Abbildung 6.9 dargestellten Teller stellen Platzhalter für ein Foto des dazugehörigen

<sup>56</sup>Vgl. Kapitel 6.1.1: Gesetz von Fitts



6. Konzeption



Abbildung 6.8.: Selektion des Mensa-Containers

Gerichts dar. Diese Ansicht bietet grundlegende Informationen über die Tagesgerichte der Mensa. Des Weiteren besteht die Möglichkeit auf eine Wochenübersicht zu fokussieren. Möchte ein Benutzer weitergehende Informationen über die Inhalte, bewegt er seine Hand über das entsprechende Gericht. Das selektierte Gericht wird hervorgehoben und rückt in den Vordergrund. Die restlichen Interface-Elemente werden halbtransparent dargestellt. Dieser Schritt wird in Abbildung 6.10 dargestellt.



Abbildung 6.9.: Container innerhalb des Mensa-Containers

Durch eine Push-Geste gelangt der Benutzer zum Objekt. Der grafische Übergang besteht wieder in einem Zoom, der den Fokussierungsvorgang verdeutlichen soll. Das Objekt bietet Zugriff auf alle relevanten Inhalte (Vgl. Abbildung 6.11). In diesem Fall werden eine Ernährungstabelle sowie die verfügbaren Beilagen dargestellt. Die



6. Konzeption



Abbildung 6.10.: Selektion des Objektes

Schalen sind dabei Platzhalter für ein Foto ihrer realen Abbilder. Möchte sich der Benutzer weitergehend informieren, kann er sich durch eine Push-Geste vom aktuellen Objekt entfernen. Die Defokussierung des Inhalts wird durch einen negativen Zoom grafisch dargestellt. Ferner besteht die Möglichkeit über das Eselsohr zurück zur Startoberfläche zu gelangen. Dies wird ebenfalls durch einen negativen Zoom dargestellt.

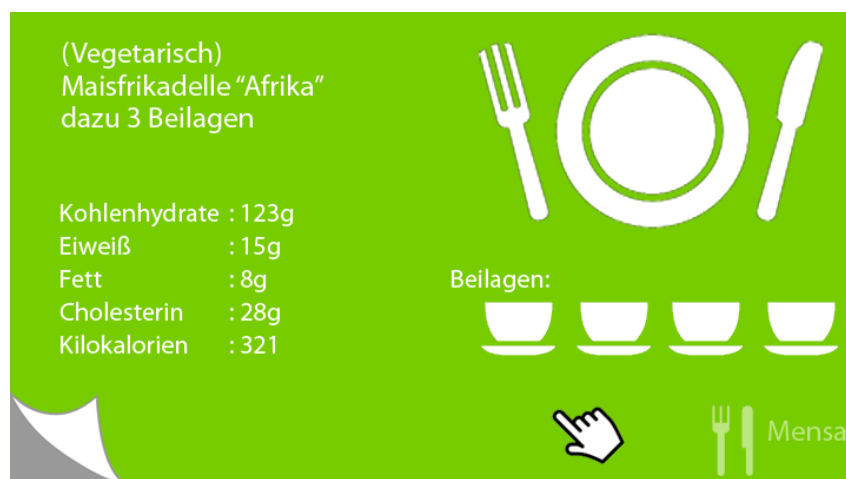


Abbildung 6.11.: Interaktion mit dem Objekt



## 7. Realisierung

Dieses Kapitel behandelt die Realisierung eines NUI unter Verwendung der bereits vorgestellten Software. Die prototypische Realisierung beschränkt sich dabei auf die Umsetzung der Grundfunktionalitäten. Zu Beginn des Kapitels wird in Abschnitt 7.1 die verwendete Systemarchitektur vorgestellt und das Zusammenspiel der einzelnen Komponenten erklärt. Die darauf folgenden Kapitel befassen sich mit der Umsetzung der Grundfunktionalitäten. Dazu wird in Abschnitt 7.2 der Anmeldevorgang am System beschrieben. Darauf aufbauend befasst sich der nächste Abschnitt 7.3 mit der Erstellung des Handcursors und der Verwendung von Zeigegesten. Danach wird in Abschnitt 7.4 die Selektion von Interface-Elementen in Unity3D auf Grundlage einer gestenbasierten Eingabe erklärt. Im abschließenden Abschnitt 7.5 wird die Möglichkeit zur Bestätigung anhand von Push-Gesten beschrieben.

### 7.1. Architektur

Die für diesen Abschnitt relevante Software wurde bereits in Abschnitt 3.5 vorgestellt. An dieser Stelle soll nun das Zusammenspiel der einzelnen Softwarekomponenten beschrieben werden.

Die prototypische Anwendung wird in Unity3D umgesetzt. Unity3D bietet die Möglichkeit, über Schnittstellen, die vom Kinectsensor erfasst und in OpenNI und NITE verarbeiteten Daten zu nutzen. Die Anbindung erfolgt über einen speziellen UnityWrapper<sup>57</sup>. Der Lieferumfang des UnityWrappers beinhaltet zusätzliche Unity3D-spezifische Skripte. Diese Skripte ermöglichen die komfortable Verwendung der NITE-Funktionalitäten in Unity3D. Die prototypische Realisierung in dieser Arbeit greift dabei ausschließlich auf die von NITE angebotenen Funktionalitäten zurück. Das OpenNI-Interface (Vgl. Abbildung 3.9, S.23) bietet jedoch die Möglichkeit, das Framework um eigene Funktionalitäten zu erweitern. So wäre es für weitergehende Arbeiten möglich, z. B. eine eigene Gestenerkennung oder ein Modul

---

<sup>57</sup>Website: <https://github.com/zigfu/UnityOpenNIBindings>



zur Fingererkennung in OpenNI zu implementieren<sup>58</sup>.

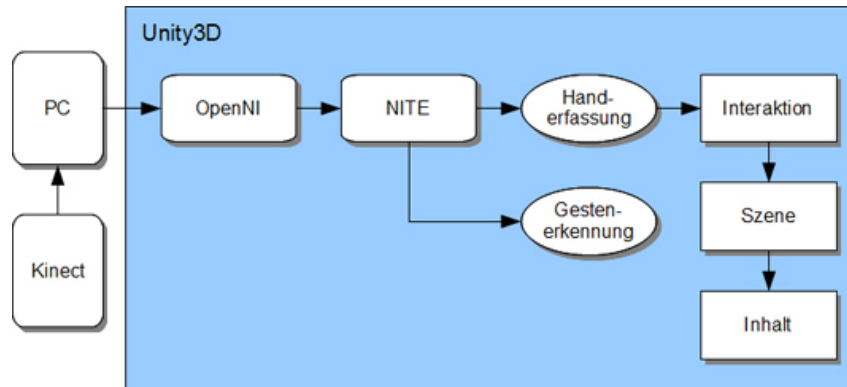


Abbildung 7.1.: Verwendete Systemarchitektur

Die verwendete Systemarchitektur wird in Abbildung ?? gegeben. Als Eingabegerät wird die Kinect verwendet, die über USB 2.0 an den PC angeschlossen ist. Die von den Kinectsensoren erfassten Informationen werden auf dem PS1080-SoC vorverarbeitet und an OpenNI übertragen. Über die NITE-Algorithm-Schicht wird auf die von OpenNI zur Verfügung gestellten Daten zugegriffen. Die Methoden zur Gestenerkennung und Handerfassung sind in der NITE-Controls-Schicht definiert. Der UnityWrapper ermöglicht die Verwendung dieser Methoden in Unity3D. Die in Unity3D verwendeten Tracking-Daten der Kinect werden normalisiert und als Eingabeparameter für die Handposition genutzt. Die Daten der Handposition, die einmal pro frame erneuert werden, sind die Grundlage für eine Interaktion in der Szene.

Anwendungen müssen zu Beginn ihrer Ausführung im OpenNI-Framework registriert werden. Dies geschieht über die Konfiguration einer XML-Datei. Die Konfiguration legt dabei den Kontext der Anwendung fest und definiert, welche Informationen verfügbar gemacht werden müssen. Am Beispiel von Listing 7.1 wird gezeigt, wie eine solche Konfiguration aussieht.

<sup>58</sup>Vgl. TANG [86]



Listing 7.1: OpenNI XML Konfiguration

```

1 <OpenNI>
2   <Licenses>
3     <License vendor="PrimeSense" key="OK0Ik2JeIBYClPWVnMoRKn5cdY4="/>
4   </Licenses>
5   <Log writeToConsole="true" writeToFile="false">
6     <LogLevel value="3"/>
7     <Masks>
8       <Mask name="ALL" on="true"/>
9     </Masks>
10    <Dumps>
11    </Dumps>
12  </Log>
13  <ProductionNodes>
14    <Node type="Image">
15      <Configuration>
16        <MapOutputMode xRes="640" yRes="480" FPS="30"/>
17        <Mirror on="true"/>
18      </Configuration>
19    </Node>
20    <Node type="Depth">
21      <Configuration>
22        <MapOutputMode xRes="640" yRes="480" FPS="30"/>
23        <Mirror on="true"/>
24      </Configuration>
25    </Node>
26    <Node type="User"/>
27    <Node type="Gesture"/>
28    <Node type="Hands"/>
29  </ProductionNodes>
30 </OpenNI>

```

Zu Beginn einer Konfiguration wird der Wurzelknoten OpenNI angegeben. Danach werden globale Konfigurationen wie z. B. *Loggen*, *License*<sup>59</sup> und die Production Nodes definiert. In diesem Beispiel werden die Production Nodes zur Erzeugung von Farbbild (image), Tiefenbild (depth), Skelettmodell (user), Gestenerkennung (gesture) und Handverfolgung (hands) angefordert. Während des Initialisierungsprozesses fordern die definierten Knoten beim Framework die benötigten Daten an, die durch die Sensoren erzeugt werden.

## 7.2. Anmeldung

Die Anmeldung am System soll vom Benutzer durch die Ausführung einer Wink-Geste geschehen. Der Benutzer muss über die Möglichkeit, wie er sich am System

<sup>59</sup>Wird die Kinect als Eingabegerät verwendet, muss der angegebene Lizenzschlüssel eingetragen werden. Dieser ist jedoch zur freien Nutzung bestimmt [66]



## 7. Realisierung

anmeldet in Kenntnis gesetzt werden. Aus diesem Grund wird beim Start der Anwendung eine Nachricht gegeben, wie der Benutzer sich am System anmelden kann. Der Quellcode zur Erzeugung eines Anmeldebildschirms ist in Listing 7.2 dargestellt.

Listing 7.2: Anmeldung am System

```
1 public class StartSessionMessage : MonoBehaviour {  
2     void OnGUI()  
3     {  
4         if (!OpenNISessionManager.InSession) {  
5             GUILayout.BeginArea(new Rect(Screen.width / 2 - 150, 50, 300, 300));  
6             GUILayout.Box("Zur Anmeldung freundlich winken !");  
7             GUILayout.EndArea();  
8         }  
9     }  
10 }
```

Der Anmeldevorgang wird nur prototypisch in Form einer einfachen Textnachricht umgesetzt. Anstelle einer Textnachricht wäre es in weiterführenden Arbeiten möglich, den Anmeldevorgang in Form eines kleinen Videos darzustellen. Die Darstellung der geforderten Anmeldegeste in Form eines Videos, wäre für den Benutzer eindeutiger als eine einfache Textnachricht. Eine grafische Darstellung des Anmeldevorgangs wird in Abbildung 7.2 gegeben.



Abbildung 7.2.: Anmeldebildschirm des Prototyps

## 7.3. Handverfolgung

Die grundlegende Interaktion der prototypischen Anwendung basiert auf der Verwendung von Zeigegesten. Zeigegesten sind nach der Definition von BLAKE Manipulationen, weil sie eine direkte Interaktion ermöglichen. Direkte Interaktion wird





## 7. Realisierung

---

bei Zeigegesten durch die zeitliche Nähe<sup>60</sup> ermöglicht.

Die Erstellung eines Handcursors in Unity3D gestaltet sich wie folgt. Zuerst muss ein GameObject, eine plane Fläche erstellt werden. Die plane Fläche wird im Anschluss mit der entsprechenden Handtextur verknüpft. In Unity3D wird dies durch eine Drag & Drop Operation ermöglicht. Es besteht die Möglichkeit, den Handcursor weitergehend zu animieren. Die Verknüpfung eines Partikelsystems mit dem Handcursor lässt diesen beispielsweise Funken sprühen. Solche oder ähnliche Verfahren sorgen für eine verbesserte Sichtbarkeit des Handcursors. Darüber hinaus kann das Partikelsystem eventbasiert eingesetzt werden, z. B. nur bei der Selektion eines Interface-Elements.

Der nächste Schritt besteht in der Verknüpfung des erstellten Handcursors mit der eigentlichen Funktionalität: der Handverfolgung. Die Scripte *FollowHandPoint* und *HandPointControl* werden bei der Installation des UnityWrappers mitgeliefert. *HandPointControl* fordert von OpenNI die benötigten Positionsdaten der erfassten Hände an. Die Positionsdaten werden von dem *FollowHandPoint*-Script abgerufen und für die Verwendung in einer Unity3D-Szene weiterverarbeitet. Im *FollowHandPoint*-Script werden zusätzlich die Begrenzungen der Szene definiert, an denen das verfolgte Objekte abgefangen wird. Damit der Handcursor den Bewegungen der Benutzerhand folgt, muss in Unity3D lediglich das *FollowHandPoint*-Script mit dem erstellten Handcursor verknüpft werden. Das *HandPointControl*-Script wird dabei automatisch zur Laufzeit erzeugt.

Beim Testen der Handverfolgung in Unity3D wurden einige Schwächen identifiziert. Diese sollen an dieser Stelle aufgeführt werden. Ein grundlegendes Problem der Handverfolgung besteht in dem von OpenNI verwendeten Algorithmus zur Erfassung der Hände. Es passiert häufig, dass die Erfassung verloren geht. Dies passiert, wenn die interne Auswertung der Tiefendaten keine Rückschlüsse auf die Position der Hand zulässt. Es wurde beobachtet, dass die Erfassung in den meisten Fällen versagt, wenn sich die Hand des Benutzers nahe am Körper befindet. Dies stellt ein großes Problem dar, weil der Benutzer, nachdem er eine Eingabe getätigt hat, eine entspannte Körperhaltung anstrebt. Des Weiteren wurde beobachtet, dass die Handverfolgung in Unity3D mit einer leichten Verzögerung dargestellt wird.

---

<sup>60</sup>Vgl. Kapitel 5.3 S.48



## 7.4. Selektion

Um Interface-Elemente selektieren zu können, muss die Handposition mit der Position der Interface-Elemente verglichen werden. Befindet sich die Hand über einem Interface-Element muss das System dem Benutzer ein Feedback darüber geben. Das Feedback kann beispielsweise anhand einer Translation, Skalierung oder einer Farbänderung des entsprechenden Interface-Elementes gegeben werden. Der Vergleich zwischen Handposition und Interface-Elementen wird in Unity3D durch die Raycast-Methoden umgesetzt. Bei der Raycast-Methode wird pro frame ein Strahl in die Szene geschossen und erkannt, ob dieser mit einem anderem GameObject kollidiert. Das Raycast-Script wird dabei mit dem Handcursors verknüpft. Um zu wissen, welches Interface-Element getroffen wurde, müssen die Interface-Elemente vom Typ *Transform* sein. Die Unity-Engine übernimmt dadurch die Berechnungen der Fläche der Interface-Elemente. Damit ist ein Vergleich zwischen dem Raycast-Objekt und dem Interface-Element möglich. Der Quellcode ist in Listing 7.3 dargestellt.

ei der Raycast-Methode schiesst jedes frame einen Strahl in die Szene und erkennt, ob dieser mit einem anderem GameObject kollidiert.

Listing 7.3: Methode zur Selektion von Interface-Elementen

```
1 void Update () {  
2  
3     RaycastHit hit;  
4     Ray ray = new Ray();  
5     ray.origin = transform.position;  
6     ray.direction = transform.TransformDirection (Vector3.forward);  
7  
8     if (Physics.Raycast (ray, out hit, 1000)) {  
9         foreach(Transform item in items) {  
10             if (hit.collider.gameObject.transform == item) {  
11                 item.renderer.material.color = Color.green;  
12             }  
13         }  
14     }  
15 }
```

Im Kapitel 6, Konzeption wurde festgehalten, dass die Interaktion des Benutzers anhand des OSIT-Handlungsmodells erfolgen soll. Nachdem der Benutzer sich orientiert hat, selektiert er die für ihn wichtigen Inhalte. Der Benutzer fokussiert den entsprechenden Inhalt. Der Fokussierungsvorgang wird in Unity3D durch eine Kamerafahrt auf das entsprechende Interface-Element realisiert. Für weitergehende Arbeiten könnten die Kamerafahrten um zusätzliche Effekte wie Tiefenunschärfe erweitert



werden.

## 7.5. Push-Geste

Um Selektionen zu bestätigen, wird eine Push-Geste verwendet. Die Charakteristik einer Push-Geste besteht in einer durchgängigen Handbewegung zur Kamera und wieder zurück. In Unity3D wird der Push-Detector zu Beginn einer Anwendung in der Start-Methode initialisiert. Danach können die Methoden des Push-Detectors verwendet werden. Der Push-Detector ist ein weiteres Script, das im Lieferumfang des UnityWrappers enthalten ist. In Listing 7.4 wird der Quellcode der allgemeinen Funktionsweise des Push-Detectors dargestellt. In diesem Beispiel bewegt sich die Kamera auf den Mittelpunkt des „gepushten“ Elementes.

Listing 7.4: Verwendung von Push-Gesten in Unity

```
1 void Start () {  
2     push = GetComponent<PushDetector>();  
3 }  
4  
5 void Update () {  
6  
7     if(push.IsClicked) {  
8         Camera.mainCamera.transform.localPosition = new Vector3(item);  
9     }  
10 }
```

Die Verwendung von Push-Gesten ist nur eine Möglichkeit Selektionen zu bestätigen. Das Problem von Push-Gesten besteht in einer häufigen Nicht-Erkennung der Geste und in einer auf Dauer großen körperlichen Anstrengung. Des Weiteren muss die Push-Geste vom Benutzer erlernt werden. In einer Anwendung muss dem Benutzer der Gebrauch dieser Geste erst erklärt werden. Dies kann ähnlich zur Anmeldung<sup>61</sup> in Form einer textbasierten Nachricht oder eines kurzen Videos geschehen. Eine alternative Möglichkeit zur Bestätigung besteht in der Verwendung eines Timers. Hierbei bewegt der Benutzer seine Hand über das Objekt. Solange sich die Hand über dem Objekt befindet, wird ein Timer gestartet. Nach einer definierten Zeit gilt die Selektion als bestätigt. Für den Benutzer wird der Fortschritt des Timers z. B. anhand eines Ladebalkens visuell dargestellt. Die Bestätigung durch einen laufenden Timer hat einige Vorteile. Bei der Timer-Methode wird dem Benutzer ein unmittelbares, visuelles Feedback gegeben. Ein weiterer Vorteil besteht darin, dass

---

<sup>61</sup>Vgl. Abschnitt 7.2



### *7. Realisierung*

---

die Timer-Methode keine Geste voraussetzt. Eine potentielle Nicht-Erkennung, wie bei der Push-Geste, ist somit ausgeschlossen.



## 8. Zusammenfassung und Fazit

Das Ziel dieser Masterarbeit bestand darin, auf Grundlage bekannter gestenbasierter Interaktionskonzepte den Mehrwert der gestenbasierten Interaktion auf Desktop-Systemen aufzuzeigen. Um eine abschließende Aussage darüber zu treffen, wurden drei zentrale Forschungsfragen formuliert, die in den einzelnen Kapiteln behandelt wurden.

Im Kapitel 4 wurden Gesten als natürliche Interaktionsform für NUI vorgestellt. Gesten stellen eine früh erlernte Möglichkeit zur Interaktion dar. Die in Abschnitt 5.6 vorgestellten OCGM-Metaphern unterteilen Interaktion in Manipulationen und Gesten. Manipulationen sind direkte, kontinuierliche Aktionen, die eine unmittelbare Reaktion des Systems zur Folge haben. Manipulationen basieren auf natürlichen Handlungen des Menschen und stellen somit die primäre Interaktionsform für ein NUI dar. Gesten repräsentieren hingegen eine indirekte, diskrete, symbolische Aktion des Benutzers. Die Systemreaktion erfolgt bei Gesten erst nach einer kompletten Ausführung. Gesten müssen zudem erst vom Benutzer erlernt werden. Sie eignen sich daher für komplexere Aufgaben, die nicht anhand natürlicher Manipulationen durchgeführt werden können. Darüber hinaus bestehen bei Gesten im Kontext der berührungsfreien Interaktion weitere Probleme. Diese Probleme lassen sich in benutzerspezifische und technische Probleme unterteilen. Aus technischer Sicht besteht für die Ausführung komplexer Gesten die Anforderung eines robusten Systems zur Gestenerkennung. Ferner besteht ein Folgeproblem, wenn viele komplexe Gesten vom System unterstützt werden. Die Ähnlichkeit in der Ausführung der Gesten kann zu häufigen Fehlinterpretationen führen. Von dem Benutzer erfordern komplexe Gesten die Aufbringung eines gewissen Lernaufwandes. Je mehr komplexe Gesten vom System unterstützt werden, desto höher ist der aufzubringende Lernaufwand. Des Weiteren entsteht durch die Verwendung vieler komplexer Gesten ein weiteres Problem in der Behaltensleistung des Benutzers. Daraus ergeben sich speziell für die berührungsfreie Interaktion mit einem NUI folgende Anforderungen: Die gestenbasierte Interaktion mit einem NUI geschieht vorwiegend durch die Verwendung von Manipulationen. Ferner werden Gesten nur für die Erfüllung komplexer Aufgaben



## 8. Zusammenfassung und Fazit

---

verwendet. Die Interaktion wird zudem so gestaltet, dass dem Benutzer in der Ausübung von Gesten und Manipulationen ausreichende Ruhephasen geboten werden. Abschließend wurde in Kapitel 4 das Thema Interkulturalität angesprochen. Bei der Wahl der zu verwendenden Gesten müssen kulturelle Aspekte berücksichtigt werden. Das Wissen darüber, dass z. B. Chinesen vorwiegend symbolische Gesten benutzen, kann für bestimmte Anwendungsgebiete von Vorteil sein.

Im Kapitel 5 wurde ausgehend von der Forderung einer natürlichen Interaktion anhand erlernter Fähigkeiten das Konzept der NUI vorgestellt. Die Definition von **BLAKE** wurde dabei als angemessen erachtet, weil sowohl die direkte Interaktion mit Inhalten als auch die Verwendung bereits erlernter Fähigkeiten des Menschen berücksichtigt werden. Ferner wird in der Definition die Wichtigkeit des Gestaltungsvorgangs von NUI hervorgehoben. Bei NUI steht die kontext-sensitive Interaktion mit Inhalten im Vordergrund. Daraus ergeben sich eine Reihe von Anforderungen, die sich grundlegend von bestehenden GUI unterscheiden. Sowohl die Eingabe als auch die Interaktion mit Interface-Elementen müssen bei NUI auf bereits erlernte Fähigkeiten des Menschen zurückgreifen. Daraus ergeben sich vor allem für den Designer. Zum einen muss die Charakteristik der berührungsfreien Eingabe berücksichtigt werden, zum anderen muss das Verhalten des Interfaces an natürliche Handlungen des Menschen angepasst werden. Um diese Probleme zu lösen wurden, zwei Konzepte vorgestellt: das OSIT-Handlungsmodell und die OCGM-Metaphern. Das OSIT-Handlungsmodell beschreibt das intuitive Handeln des Menschen anhand der Prozessstufen Orientieren, Selektieren, Informieren und Transagieren. Die Anwendung dieses Handlungsmodells bei der Gestaltung der Benutzeroberfläche besteht darin, dass das Interface die aufgeführten Prozessstufen widerspiegelt. Das zweite Konzept, das als Gestaltungsgrundlage für NUI angewendet werden kann, sind die OCGM-Metaphern. Die Metaphern *Container* und *Object* sind die grundlegenden Interface-Elemente eines NUI. Sie sind so gewählt, dass sie bereits in frühem Alter erlernt und verstanden werden können. Hier besteht wiederum ein großer Unterschied zu GUI. Die in GUI eingesetzten WIMP-Metaphern erfordern Fachwissen, das erst in fortgeschrittenem Alter erlernt wird.

Als Eingabegerät für eine berührungsfreie Interaktion wurde in dieser Arbeit die Microsoft Kinect verwendet. Die Kinect bietet aufgrund des eingesetzten Light-Coding-Verfahrens einige Vorteile. Die Unempfindlichkeit gegen Umgebungslicht macht sie für den Einsatz in geschlossenen Räumen nutzbar. Aufgrund der berührungsfreien



### 8. Zusammenfassung und Fazit

---

Interaktion ist sie speziell für Gefahrenzonen, in denen keine berührungssensitive Interaktion erfolgen darf und für aseptische Bereiche interessant. Ferner bietet die berührungsfreie Interaktion Schutz vor Vandalismus und Verschmutzung. Eine Einschränkung der Kinect liegt in der mit 1,2m – 3,5m relativ geringen Reichweite der Sensoren. Darüber hinaus verfügen die Sensoren über eine geringe Auflösung. Für die Echtzeiterfassung von Bewegungen fällt vor allem die geringe Bildwechselfrequenz von 30 fps ins Gewicht.

Ausgehend von der Konzeption eines NUI wurden einige Feststellungen gemacht. Diese sollen an dieser Stelle zusammengefasst werden. Aufgrund des noch jungen Forschungsfeldes der NUI bestehen, speziell für den Bereich der berührungsfreien Interaktion, wenige Anhaltspunkte zur Gestaltung der Interaktion. Aus diesem Grund wurde die Konzeption des NUI auf Grundlage des OSIT-Handlungsmodells und der OCGM-Metaphern durchgeführt. Das konzipierte NUI besitzt mehrere Vorteile. Die Fokussierung auf die Interaktion mit Inhalten bedeutet für den Benutzer, dass er sich auf nur wenige Elemente konzentrieren muss. Das Zusammenspiel der natürlichen Eingaben, der Fokussierung auf Inhalte sowie der Berücksichtigung seiner natürlichen Handlungen führt zu einem geringen cognitive load des Benutzers. Dadurch ist er in der Lage schnell aufeinander folgende Aktionen durchzuführen. Das konzipierte NUI kommt dabei ohne funktionale Buttons, wie z.B. „Zurück“ aus. Die einzigen Interface-Elemente sind Container und Objekte. Die kontext-sensitive Einblendung bzw. Ausblendung von Interface-Elementen sorgt für eine zusätzliche Entlastung des cognitive loads bei dem Benutzer. Die Orientierung wird beim konzipierten NUI durch die Farbgebung und dem Verhalten der Interface-Elemente gewährleistet. Das Heranzoomen bzw. Fokussieren auf die Container und Objekte führt den Benutzer kontinuierlich an die Inhalte heran. Anders als zu bekannten GUI-typischen Hierarchieebenen wird beim NUI kein harter Bruch aufgrund eines neuen Fensters erzeugt. Eine abschließende Bewertung des erstellten Konzeptes fällt aufgrund fehlender Evaluationsmöglichkeiten schwer. Die Evaluierung des konzipierten NUI könnte Gegenstand weiterführender Untersuchungen sein.

Bei der abschließenden Realisierung wurde sich auf die Grundfunktionalitäten des konzipierten NUI beschränkt. Die Gründe dafür sind vorwiegend technisch begründet. Die verwendete Systemarchitektur mit den Komponenten OpenNI, NITE und Unity3D ist zum derzeitigen Stand nicht stabil. Die Hauptproblematik bestand in häufigen Verbindungsabbrüchen der Kinect in Unity3D. Die Installation unterschiedlicher Versionen von OpenNI und NITE brachte keine spürbare Verbesserung. Dar-



### 8. Zusammenfassung und Fazit

---

über hinaus besteht eine wahrnehmbare Latenzzeit bei der Handverfolgung unter Unity3D. In speziellen Fällen trat zudem das Problem auf, dass die Kinect die Verfolgung der Hände kurzzeitig verlor oder ungenaue Daten lieferte.

Der Paradigmenwechsel von GUI zu NUI bringt grundlegende Änderungen in der Interaktion zwischen Mensch und Computer mit sich. Erfolgreiche Produkte wie das iPhone, iPad oder die Kinect zeigen, dass die Interaktion mit einem NUI vom Benutzer angenommen wird. Dies gilt vor allem für die Interaktion mit berührungssensitiven Eingabegeräten. Im Bereich der Desktop-Systeme besteht jedoch nach wie vor ein Mangel an praxis- bzw. massentauglichen Eingabegeräten, die eine NUI-basierte Interaktion ermöglichen. Die Anwendungsgebiete berührungsfreier Eingabegeräte sind stark beschränkt. Effektiver Nutzen besteht in den Anwendungsgebieten Home-Entertainment, Informationssystemen und aseptischen Bereichen. Das nächste Windows Betriebssystem, Windows 8, könnte die Entwicklung von NUI weiter vorantreiben. Die in Windows 8 verwendete *Metro-Oberfläche* weist mehrere Eigenschaften eines NUI auf. Die Verwendung der Kinect mit Windows 8 könnte demnach ein wichtiger Schritt in der Entwicklung der NUI sein.





## Literaturverzeichnis

- [1] ANDRIENKO, N. ; ANDRIENKO, G. ; PELEKIS, N. ; SPACCAPIETRA, S. ; GI-ANNOTTI, F. (Hrsg.) ; PEDRESCHI, D. (Hrsg.): *Basic Concepts of Movement Data*. Springer, 2007. – ISBN 978–3642094439 **31**
- [2] BERARD, F. ; IP, J. ; BENOVOY, M. ; EL-SHIMY, D. ; BLUM, J. R. ; COOPERSTOCK, J. R.: Did "Minority Report" Get It Wrong? Superiority of the Mouse over 3D Input Devices in a 3D Placement Task. In: *Proceedings of the 12th IFIP TC 13 International Conference on Human-Computer Interaction: Part II*, Springer-Verlag, 2009 (INTERACT '09). – ISBN 978–3–642–03657–6, 400–414 **4.3.1**
- [3] In: BLAKE, J.: *The natural user interface revolution*. Bd. 1. Manning, 2010. – ISBN 1935182811, 375 **5.1, 5.3, 5.7, 6.2, 7.3, 8**
- [4] BLAKE, J. ; GEORGE, R.: *Objects, Containers, Gestures, and Manipulations: Universal Foundational Metaphors of Natural User Interfaces*, 2010 **5.6, 5.5, 5.6**
- [5] BOLLHOEFER, K. W. ; MEYER, K. ; WITZSCHE, R.: *White Paper Microsoft Surface und das Natural User Interface (NUI)*. [http://www.pixelpark.com/de/pixelpark/\\_ressourcen/attachments/publikationen/090204\\_White\\_Paper\\_MS\\_Surface.pdf](http://www.pixelpark.com/de/pixelpark/_ressourcen/attachments/publikationen/090204_White_Paper_MS_Surface.pdf). Version: 2009 **5.1**
- [6] BOLT, R. A.: Put-that-there: Voice and gesture at the graphics interface. In: *SIGGRAPH Comput. Graph.* 14 (1980), July, S. 262–270. <http://dx.doi.org/http://doi.acm.org/10.1145/965105.807503>. – DOI <http://doi.acm.org/10.1145/965105.807503>. – ISSN 0097–8930 **1.3, 2.3, 4.2**
- [7] BONGERS, B.: Interaction with our electronic environment ? an e-cological approach to physical interface design. In: *Computer Vision in Human-Machine Interaction*, Bd. 34, Faculty of Journalism and Communication, Hogeschool van Utrecht, 2004 **3.1, 3.1**



- [8] BUXTON, W. ; FIUME, E. ; HILL, R. ; LEE, A. ; WOO, C.: Continuous Hand-Gesture Driven Input. In: *Proceedings of Graphics Interface 1983*, 1983 (CHC-CS '83), 191–195 2.3
- [9] CABRAL, M. C. ; MORIMOTO, C. H. ; ZUFFO, M. K.: On the usability of gesture interfaces in virtual reality environments. In: *Proceedings of the 2005 Latin American conference on Human-computer interaction*, ACM, 2005 (CLIHC '05). – ISBN 1–59593–224–0, S. 100–108 2.2
- [10] CAPO, A. J. ; VARONA, J. ; J., Perales Lopez F.: Representation of Human Postures for Vision-Based Gesture Recognition in Real-Time. In: *Gesture Workshop*, 2007, S. 102–107 2.1
- [11] In: CARBINI, S ; VIALLET, J E. ; BERNIER, O: *Pointing gesture visual recognition for large display*. 2004, 27–32 4.4.4
- [12] CASSELL, J.: A framework for gesture generation and interpretation. In: R.CIPOLLA (Hrsg.) ; PENTLAND, A. (Hrsg.): *Computer Vision in Human-Machine Interaction*, Cambridge University Press, 1998, S. 191–215 4.5
- [13] COOGAN, T. ; AWAD, G. ; HAN, J. ; SUTHERLAND, A.: Real Time Hand Gesture Recognition Including Hand Segmentation and Tracking. In: *Advances in Visual Computing* 1 (2006), 495–504. <http://www.springerlink.com/index/J762264281L62223.pdf> 4.4.3
- [14] DIX, A. ; FINLAY, J. E. ; ABOWD, G. D. ; BEALE, R.: *Human-Computer Interaction (3rd Edition)*. Upper Saddle River, NJ, USA : Prentice-Hall, Inc., 2003. – ISBN 0–130–46109–1 1, 3.2
- [15] EBERLEH, E. ; OBERQUELLE, H. ; OPPERMAN, R.: *Einführung in die Software-Ergonomie*. Berlin, New York : Walter de Gruyter, 1994. – ISBN 3–11–013814–X. 5.5
- [16] EGMONT PETERSEN, M. ; RIDDERB, D. de ; HANDELS, H.: Image processing with neural networks ?- a review. In: *Pattern Recognition* 35 (2002), Nr. 10, 2279–2301. <http://linkinghub.elsevier.com/retrieve/pii/S0031320301001789> 4.4.4
- [17] EPPS, J. ; LICHMAN, S. ; WU, M.: A study of hand shape use in tabletop gesture interaction. In: *CHI '06 extended abstracts on Human factors in computing systems*, 2006 (CHI EA '06). – ISBN 1–59593–298–4, S. 748–753 2.1
- [18] FIKKERT, F. W.: *Gesture Interaction at a Distance*. Enschede, Universiteit Twente, Diss., 2010. <http://doc.utwente.nl/69985/>, Abruf: 05.02.2012 4.3.1



- [19] FREEDMAN, B. ; SHPUNT, A. ; MACHLINE, M. ; ARIELI, Y.: *Depth Mapping Using Projected Patterns*. <http://www.freepatentsonline.com/20080240502.pdf/>. Version: 2010, Abruf: 01.02.2012 12
- [20] FREEMAN, W. T. ; WEISSMAN, C. D.: Television control by hand gestures. In: *International Workshop on Automatic Face and Gesture Recognition*, 1995, 179–183 2.3
- [21] FRÖHLICH, W. D. ; J., Drever: *Wörterbuch zur Psychologie*. 26. überarb. u. erw. Aufl. München : dtv, 1986. – ISBN 3423030313 5.5
- [22] GARCIA, J. ; ZALEVSKY, Z.: *Range Mapping Using Speckle Decorrelation*. <http://www.freepatentsonline.com/7433024.pdf/>. Version: 2010, Abruf: 01.02.2012 12
- [23] GOPNIK, A. ; MELTZOFF, A. N.: The development of categorization in the second year and its relation to other cognitive and linguistic developments. In: *Child Development* 58 (1987), 1523–1531. [http://ilabs.washington.edu/meltzoff/pdf/87Gopnik\\_Meltzoff\\_ChildDev.pdf](http://ilabs.washington.edu/meltzoff/pdf/87Gopnik_Meltzoff_ChildDev.pdf), Abruf: 08.02.2012 5.6
- [24] GRANDHI, S. A. ; JOUE, G. ; MITTELBERG, I.: Understanding naturalness and intuitiveness in gesture production. In: *Proceedings of the 2011 annual conference on Human factors in computing systems*, ACM, 2011 (CHI '11). – ISBN 978-1-4503-0228-9, S. 821–824 2.2
- [25] HARDING, P. R. G. ; ELLIS, T. J.: Recognizing Hand Gesture using Fourier Descriptors. In: *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 3 - Volume 03*, IEEE Computer Society, 2004 (ICPR '04). – ISBN 0-7695-2128-2, S. 286–289 4.4.3
- [26] HASANUZZAMAN, M. ; ZHANG, Tao ; AMPORNARAMVETH, Vuthichai ; BHUIYAN, M. A. ; SHIRAI, Yoshiaki ; UENO, Haruki: Face and Gesture Recognition Using Subspace Method for Human-Robot Interaction. In: *PCM (1)*, 2004, 369–376 4.4.4
- [27] HENSELER, W.: Absatzoptimierung im Internet. Zur konsumpsychologischen Gestaltung von Online-Shops. In: *i-com* 6 (2007), Nr. 2, S. 52–. <http://dx.doi.org/http://dx.doi.org/10.1524/icom.2007.6.2.52>. – DOI <http://dx.doi.org/10.1524/icom.2007.6.2.52> 3.3, 3.3
- [28] HENSELER, W.: Vom GUI zum NUI. <http://createordie.de/cod/artikel/Von-GUI-zu-NUI-2818.html/>. Version: 2010 3.3, 5.1, 6.1.3



- [29] HOFMEESTER, K. ; WIXON, D.: Using metaphors to create a natural user interface for microsoft surface. In: *Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems*, ACM, 2010 (CHI EA '10). – ISBN 978-1-60558-930-5, S. 4629–4644 2.2
- [30] HOLT, G. A. ; ARENSEN, J. ; RIDDER, H. de ; DOORN, A. J. ; REINDERS, M. J. T. ; HENDRIKS, E. A.: Sign language perception research for improving automatic sign language recognition. In: *Human Vision and Electronic Imaging* Bd. 7240, SPIE, 2009 (SPIE Proceedings). – ISBN 978-0-8194-7490-2, 72400 1.3
- [31] HOURCADE, J. P. ; BEDERSON, B. B. ; DRUIN, A. ; GUIMBRETIERE, title = Differences in pointing task performance between preschool children and adults using mice journal = ACM Trans. Comput.-Hum. Interact. volume = 11 issue = 4 year = 2004 issn = 1073-0516 pages = 357–386 numpages = 30 doi = <http://doi.acm.org/10.1145/1035575.1035577> acmid = 1035577 publisher = ACM F.: 5.6
- [32] HU, Ming-Kuei: Visual pattern recognition by moment invariants. In: *Information Theory IRE Transactions on* 8 (1962), Nr. 2, 179–187. [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=1057692](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1057692) 4.4.3
- [33] HUANG, T. S. ; PAVLOVIC, V. I.: Hand Gesture Modeling, Analysis, and Synthesis. In: *In Proc. of IEEE International Workshop on Automatic Face and Gesture Recognition*, 1995, S. 73–79 3.2
- [34] IVERSON, J. ; GOLDIN-MEADOW, S.: What's communication got to do with it? Gesture in children blind from birth. In: *Developmental Psychology* (1997), S. 453–467 3.2
- [35] J., Shotton ; FITZGIBBON, A. W. ; COOK, M. ; T., Sharp ; FINOCCHIO, M. ; MOORE, R. ; KIPMAN, A. ; BLAKE, A.: Real-time human pose recognition in parts from single depth images. In: *CVPR*, 2011, S. 1297–1304 4.3.3, 4.4.4
- [36] JOHNSON, C. P. ; BLASCO, P. A.: Infant growth and development. In: *Pediatrics in Review* 18 (1997), S. 224–242 5.6
- [37] JOHNSON, R. ; O'HARA, K. ; SELLEN, A. ; COUSINS, C. ; CRIMINISI, A.: Exploring the potential for touchless interaction in image-guided interventional radiology. In: *Proceedings of the 2011 annual conference on Human factors in computing systems*, ACM, 2011 (CHI '11). – ISBN 978-1-4503-0228-9, 3323–3332 2.2



- [38] KARAM, M.: *A framework for gesture-based human computer interactions*. Saarbrücken, Germany : VDM Verlag, 2009. – ISBN 9783639194395 2.1, 4, 4.2, 4.2
- [39] KENDON, A.: *Gesture – Visible Action as Utterance*. Cambridge : Cambridge University Press, 2004. – ISBN 0–521–54293–6 4.1, 4.1
- [40] KHOSHELHAM, K.: Accuracy Analysis of Kinect Depth Data. In: *Geo-Information Science* 38 (2010), Nr. 5/W12, 1–6. [http://www.isprs.org/proceedings/XXXVIII/5-W12/Papers/ls2011\\_submission\\_40.pdf](http://www.isprs.org/proceedings/XXXVIII/5-W12/Papers/ls2011_submission_40.pdf), Abruf: 01.02.2012 3.4.2
- [41] KOONS, D. B. ; SPARRELL, C. J.: Iconic: speech and depictive gestures at the human-machine interface. In: *Conference companion on Human factors in computing systems*, ACM, 1994 (CHI '94). – ISBN 0–89791–651–4, S. 453–454 4.2
- [42] KOPP, S. ; TEPPER, P. ; CASSELL, J.: Towards integrated microplanning of language and iconic gesture for multimodal output. In: *Proceedings of the 6th international conference on Multimodal interfaces*, ACM, 2004 (ICMI '04). – ISBN 1–58113–995–0, S. 97–104 4.2
- [43] LANG, S.: *Sign Language Recognition with Kinect*, Diplomarbeit, 2011. <http://page.mi.fu-berlin.de/block/abschlussarbeiten/Bachelor-Lang.pdf>, Abruf: 05.02.2012 1.3
- [44] LEE, S. ; HENDERSON, V. ; HAMILTON, H. ; STARNER, T. ; BRASHEAR, H. ; HAMILTON, S.: A gesture-based american sign language game for deaf children. In: *CHI '05 extended abstracts on Human factors in computing systems*, ACM, 2005 (CHI EA '05). – ISBN 1–59593–002–7, S. 1589–1592 4.2
- [45] LENMAN, S. ; BRETZNER, L. ; THURESSON, B.: Computer vision based hand gesture interfaces for human-computer interaction. In: *CID Stockholm Sweden* (2002). <http://cid.nada.kth.se/pdf/CID-172.pdf>, Abruf: 05.02.2012 2.2
- [46] LI, H. ; GREENSPAN, M.: Multi-Scale Gesture Recognition from Time-Varying Contours. In: *Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1 - Volume 01*, IEEE Computer Society, 2005. – ISBN 0–7695–2334–X–01, 236–243 4.4.2, 4.4.4
- [47] LI, M. ; MAGNOR, M. ; SEIDEL, H. P.: Hardware-Accelerated Visual Hull Reconstruction and Rendering. In: *In Graphics Interface 2003*, 2003, S. 65–71 4.4.3



- [48] LISZKOWSKI, U.: A differentiated perspective on infant gestures. In: *Gestures in language development* 8 (2008), S. 180–196. <http://dx.doi.org/http://dx.doi.org/10.1075/gest.8.2.04lis>. – DOI <http://dx.doi.org/10.1075/gest.8.2.04lis> 3.2
- [49] MAES, P. ; DARRELL, T. ; BLUMBERG, B. ; PENTLAND, A.: The ALIVE system: wireless, full-body interaction with autonomous agents. In: *Multimedia Syst.* 5 (1997), 105–112. <http://dl.acm.org/citation.cfm?id=959145.959153>. – ISSN 0942–4962 1.3
- [50] MAUNEY, Dan ; HOWARTH, Jonathan ; WIRTANEN, Andrew ; CAPRA, Miranda: Cultural similarities and differences in user-defined gestures for touchscreen user interfaces. In: *Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems*, ACM, 2010 (CHI EA '10). – ISBN 978–1–60558–930–5, 4015–4020 4.9, 4.5
- [51] MCNEILL, D.: *Language and Gesture*. Cambridge University Press, 2000. <http://dx.doi.org/10.2277/0521777615>. <http://dx.doi.org/10.2277/0521777615>. – ISBN 0–521–77761–5 3.2, 4.2
- [52] MICROSOFT: *Kinect Manual*. [download.microsoft.com/download/0/C/2/0C21CEBE-0031-489E-966E-513E6070995A/kinect\\_en\\_fr-FR\\_de-DE\\_it-IT\\_es-ES.pdf/](http://download.microsoft.com/download/0/C/2/0C21CEBE-0031-489E-966E-513E6070995A/kinect_en_fr-FR_de-DE_it-IT_es-ES.pdf/). Version: 2009, Abruf: 05.02.2012 3.4.2
- [53] MITRA, S. ; ACHARYA, T.: Gesture recognition: A survey. In: *IEEE Transactions On Systems, Man And Cybernetics - Part C* 37 (2007), Nr. 3, S. 311–324 4.1
- [54] MOESLUND, T. B. ; HILTON, A. ; KRÜGER, V.: A survey of advances in vision-based human motion capture and analysis. In: *Comput. Vis. Image Underst.* 104 (2006), November, S. 90–126. <http://dx.doi.org/http://dx.doi.org/10.1016/j.cviu.2006.08.002>. – DOI <http://dx.doi.org/10.1016/j.cviu.2006.08.002>. – ISSN 1077–3142 4.3.3
- [55] MOESLUND, T. B. ; STORRING, M. ; GRANUM, E.: A Natural Interface to a Virtual Environment through Computer Vision-Estimated Pointing Gestures. In: *Revised Papers from the International Gesture Workshop on Gesture and Sign Languages in Human-Computer Interaction*, Springer-Verlag, 2002 (GW '01). – ISBN 3–540–43678–2, 59–63 2.2, 4.3.3, 6.2
- [56] MONSON-HAEFEL, R.: *The iPad Entrepreneur: NUI Defined*. <http://theclevermonkey.blogspot.com/2010/01/nui-defined.html>. Version: 2010, Abruf: 03.02.2012 5.1



- [57] MORRIS, M. R. ; HUANG, . ; PAEPCKE, A. ; WINOGRAD, T.: Cooperative gestures: multi-user gestural interactions for co-located groupware. In: *Proceedings of the SIGCHI conference on Human Factors in computing systems*, ACM, 2006 (CHI '06). – ISBN 1–59593–372–7, S. 1201–1210 [4.3.1](#)
- [58] NIELSEN, M. ; STÖRRING, M. ; MOESLUND, T. B. ; GRANUM, E.: A procedure for developing intuitive and ergonomic gesture interfaces for HCI. In: *Gesture-Based Communication in HumanComputer Interaction* 2915 (2004), 105?106. <http://www.springerlink.com/index/WMP0F5NED2MTEF26.pdf> [5.5](#)
- [59] NORMAN, D. A.: *The Psychology of Everyday Things*. New York, U.S.A. : Basic Books, 1988. – ISBN 0–465–06709–3 [3.1](#), [3.2](#), [3.2](#)
- [60] NORMAN, Donald A. ; DRAPER, Stephen W.: *User Centered System Design; New Perspectives on Human-Computer Interaction*. Hillsdale, NJ, USA : L. Erlbaum Associates Inc., 1986. – 31–61 S. – ISBN 0898597811 [3.1](#)
- [61] NÖTH, W.: *Handbuch der Semiotik (Handbook of Semiotics)*. Stuttgart and Weimar: Verlag JB Metzler, 2000. – ISBN 3–476–01226–3 [4.1](#)
- [62] O'HAGAN, R. ; ZELINSKY, A.: Visual Gesture Interfaces for Virtual Environments. In: *Proceedings of the First Australasian User Interface Conference*. Washington, DC, USA : IEEE Computer Society, 2000. – ISBN 0–7695–0515–5, 73– [4.4.3](#)
- [63] ONG, S. C. ; R., Surendra: Automatic Sign Language Analysis: A Survey and the Future beyond Lexical Meaning. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 27 (2005), Nr. 6, S. 873–891. <http://dx.doi.org/http://doi.ieeecomputersociety.org/10.1109/TPAMI.2005.112>. – DOI <http://doi.ieeecomputersociety.org/10.1109/TPAMI.2005.112> [1.3](#)
- [64] PAVLOVIC, V. I. ; SHARMA, R. ; HUANG, T. S.: Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 19 (1997), S. 677–695. <http://dx.doi.org/http://dx.doi.org/10.1109/34.598226>. – DOI <http://dx.doi.org/10.1109/34.598226>. – ISSN 0162–8828 [2.1](#), [4](#), [4.1](#), [4.2](#), [4.2](#), [4.3](#), [4.4.1](#), [4.8](#)
- [65] POPPE, R.: A survey on vision-based human action recognition. In: *Image Vision Comput.* 28 (2010), June, S. 976–990. <http://dx.doi.org/http://dx.doi.org/10.1016/j.imavis.2009.11.014>. – DOI <http://dx.doi.org/10.1016/j.imavis.2009.11.014>. – ISSN 0262–8856 [4.3.3](#)





- [66] PRIMESENSE: *OpenNI User Guide*. [http://www.openni.org/images/stories/pdf/OpenNI\\_UserGuide\\_v4.pdf/](http://www.openni.org/images/stories/pdf/OpenNI_UserGuide_v4.pdf/). Version: 2010, Abruf: 03.02.2012 3.5.1, 59
- [67] PRIMESENSE: *Prime Sensor NITE 1.3 Controls Programmers Guide*. <http://andrealtazar.files.wordpress.com/2011/02/nite-controls-1-3-programmers-guide.pdf/>. Version: 2010, Abruf: 03.02.2012 3.5.2, 3.11
- [68] PRIMESENSE: *The Primesense Reference Design*. [http://primesense.360.co.il/files/FMF\\_2.PDF/](http://primesense.360.co.il/files/FMF_2.PDF/). Version: 2010, Abruf: 01.02.2012 3.4.2, 3.4.2
- [69] PSARROU, A. ; GONG, S. ; WALTER, M.: Recognition of human gestures and behaviour based on motion trajectories. In: *Image Vision Comput.* 20 (2002), Nr. 5-6, S. 349–358. [http://dx.doi.org/http://dx.doi.org/10.1016/S0262-8856\(02\)00007-0](http://dx.doi.org/http://dx.doi.org/10.1016/S0262-8856(02)00007-0). – DOI [http://dx.doi.org/10.1016/S0262-8856\(02\)00007-0](http://dx.doi.org/10.1016/S0262-8856(02)00007-0) 4.4.4
- [70] QUEK, F. ; MCNEILL, D. ; BRYLL, R. ; DUNCAN, S. ; MA, X. F. ; KIRBAS, C. ; MCCULLOUGH, K. E. ; ANSARI, R.: Multimodal human discourse: gesture and speech. In: *ACM Trans. Comput.-Hum. Interact.* 9 (2002), September, 171–193. <http://dx.doi.org/http://doi.acm.org/10.1145/568513.568514>. – DOI <http://doi.acm.org/10.1145/568513.568514>. – ISSN 1073–0516 4.2, 4.2
- [71] REICHINGER, A.: *Kinect Pattern Uncovered*. <http://azttm.wordpress.com/2011/04/03/kinect-pattern-uncovered/>. Version: 2010, Abruf: 01.02.2012 3.4.2
- [72] REKIMOTO, J.: SmartSkin: an infrastructure for freehand manipulation on interactive surfaces. In: *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves*, ACM, 2002 (CHI '02). – ISBN 1–58113–453–3, 113–120 5.4.2
- [73] SCHLATTMANN, M. ; KLEIN, R.: Simultaneous 4 gestures 6 DOF real-time two-hand tracking without any markers. In: *Proceedings of the 2007 ACM symposium on Virtual reality software and technology*, ACM, 2007 (VRST '07). – ISBN 978–1–59593–863–3, S. 39–42 4.4.3
- [74] SCHLITTGEN, R.: *Einführung in die Statistik: Analyse und Modellierung von Daten*. München : Oldenbourg, 2008. – ISBN 978–3486587746 3
- [75] SHI, L. ; WANG, Y. ; LI, J.: A real time vision-based hand gestures recognition system. In: *Proceedings of the 5th international conference on Advances*





- in computation and intelligence*. Berlin, Heidelberg : Springer-Verlag, 2010 (ISICA'10). – ISBN 3-642-16492-7, 978-3-642-16492-7, 349-358 4.4.2
- [76] SHNEIDERMAN, B.: *Designing the User Interface – Strategies for Effective Human-Computer-Interaction*. 3. Addison-Wesley Longman Inc., 1998. – ISBN 0-201-69497-2 3.2
- [77] SHPUNT, A.: *Optical Designs for Zero Order Reduction*. <http://www.freepatentsonline.com/20090185274.pdf>. Version: 2009, Abruf: 01.02.2012 12
- [78] SHPUNT, A.: *Depth Mapping Using Multi Beam Illumination*. <http://www.freepatentsonline.com/20100020078.pdf/>. Version: 2010, Abruf: 01.02.2012 12
- [79] STAFFORD, A. ; THOMAS, B. H. ; PIEKARSKI, W.: Comparison of techniques for mixed-space collaborative navigation. In: *Proceedings of the Tenth Australasian Conference on User Interfaces - Volume 93*, Australian Computer Society, Inc., 2009 (AUIC '09). – ISBN 978-1-920682-74-3, 61-70 2.1
- [80] STANDARDIZATION, International O.: *ISO 9241-9: Ergonomic requirements for office work with visual display terminals (VDTs) - Part 9: Requirements for non-keyboard input devices*. 2000 5.5
- [81] STEINBRECHER, Rainer: *Bildverarbeitung in der Praxis*. München : Oldenbourg Verlag, 1993. – ISBN 3-486-22372-0 4.4.2, 33
- [82] STERN, H. ; WACHS, J. ; EDAN, Y.: *Gesture-Based Human-Computer Interaction and Simulation*, Springer-Verlag, 2009. – ISBN 978-3-540-92864-5, S. 57-68 2.1
- [83] STURMAN, D. J. ; ZELTZER, D. ; PIEPER, S.: Hands-on interaction with virtual environments. In: *Proceedings of the 2nd annual ACM SIGGRAPH symposium on User interface software and technology*, ACM, 1989 (UIST '89). – ISBN 0-89791-335-3, S. 19-24 2.3
- [84] SUTHERLAND, I. E.: Sketch pad a man-machine graphical communication system. In: *Proceedings of the SHARE design automation workshop*, ACM, 1964 (DAC '64), S. 6.329-6.346 2.3
- [85] SWELLER, J.: Cognitive Load Theory and E-Learning. In: *AIED*, 2011, S. 5-6 44



- [86] TANG, M.: Recognizing Hand Gestures with Microsoft Kinect. In: *stanfordedu* (2011). [http://www.stanford.edu/class/ee368/Project\\_11/Reports/Tang\\_Hand\\_Gesture\\_Recognition.pdf](http://www.stanford.edu/class/ee368/Project_11/Reports/Tang_Hand_Gesture_Recognition.pdf), Abruf: 05.02.2012 2.1, 58
- [87] TOLAR, T. ; LEDERBERG, A. ; GOKHALE, S. ; TOMASELLO, M.: The development of the ability to recognize the meaning of iconic signs. In: *Journal of Deaf Studies and Deaf Education* 13 (2008), 71–86. <http://jdsde.oxfordjournals.org/content/13/2/225.full>, Abruf: 08.02.2012 5.6
- [88] TRIGO, T. R. ; PELLEGRINO, S. R. M.: An Analysis of Features for Hand-Gesture Classification. In: *17th International Conference on Systems, Signals and Image Processing*, 2010 (IWSSIP 2010) 2.1
- [89] WELCH, G. ; BISHOP, G.: An Introduction to the Kalman Filter. Chapel Hill, NC, USA : University of North Carolina at Chapel Hill, 1995. – Forschungsbericht 4.3.3
- [90] WEXELBLAT, A.: An approach to natural gesture in virtual environments. In: *ACM Trans. Comput.-Hum. Interact.* 2 (1995), S. 179–200. <http://dx.doi.org/http://doi.acm.org/10.1145/210079.210080>. – DOI <http://doi.acm.org/10.1145/210079.210080>. – ISSN 1073–0516 4.2
- [91] WEXELBLAT, A.: Research Challenges in Gesture: Open Issues and Unsolved Problems. In: *Proceedings of the International Gesture Workshop on Gesture and Sign Language in Human-Computer Interaction*, Springer-Verlag, 1998. – ISBN 3–540–64424–5, 1–11 4.2
- [92] WILSON, A. D. ; BOBICK, A. F.: Parametric Hidden Markov Models for Gesture Recognition. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 21 (1999), September, S. 884–900. <http://dx.doi.org/http://dx.doi.org/10.1109/34.790429>. – DOI <http://dx.doi.org/10.1109/34.790429>. – ISSN 0162–8828 4.4.4
- [93] WOODWARD, A. L. ; GUAJARDO, J. J.: Infants’ understanding of the point gesture as an object-directed action. In: *Cognitive Development* 17 (2002), S. 1061–1084 5.6
- [94] WU, Y. ; LIU, Q. ; HUANG, T. S.: Robust Real-Time Human Hand Localization by Self-Organizing Color Segmentation. In: *Proceedings of the International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, IEEE Computer Society, 1999 (RATFG-RTS ’99). – ISBN 0–7695–0378–0, 161– 4.4.2



- [95] YANG, J. ; XU, Y.: Hidden Markov Model for Gesture Recognition / Robotics Institute. Version: 1994. [http://www.ri.cmu.edu/pub\\_files/pub3/yang\\_jie\\_1994\\_1/yang\\_jie\\_1994\\_1.pdf](http://www.ri.cmu.edu/pub_files/pub3/yang_jie_1994_1/yang_jie_1994_1.pdf), Abruf: 01.02.2012. 1994. – Forschungsbericht 4.4.4
- [96] ZARIT, B. D. ; SUPER, B. J. ; QUEK, F. K. H.: Comparison of Five Color Models in Skin Pixel Classification. In: *Proceedings of the International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*. Washington, DC, USA : IEEE Computer Society, 1999 (RATFG-RTS '99). – ISBN 0-7695-0378-0, 58– 4.4.2



## Eidesstattliche Erklärung

Ich, Florian Gebauer, Matrikel-Nr. 11048082, versichere hiermit, dass ich meine Masterarbeit mit dem Thema

*Gestenbasierte Interfaces Konzeption und Entwicklung unter Verwendung der Microsoft Kinect*

selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, wobei ich alle wörtlichen und sinngemäßen Zitate als solche gekennzeichnet habe. Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Mir ist bekannt, dass ich meine Masterarbeit zusammen mit dieser Erklärung fristgemäß nach Vergabe des Themas in dreifacher Ausfertigung und gebunden im Prüfungsamt der Fachhochschule Köln abzugeben oder spätestens mit dem Poststempel des Tages, an dem die Frist abläuft, zu senden habe.

Gummersbach, den 22. Februar 2012

---

FLORIAN GEBAUER



## **A. Anhang**

### **A.1. Inhalt der DVD**

Die Ordner der beiliegenden DVD beinhalten folgende Materialien:

- **Masterthesis**  
Masterthesis im PDF-Format
- **Konzeption**  
Abbildungen des konzipierten NUI in 720p Auflösung
- **Implementierung**  
Der Quellcode der prototypischen Realisierung
- **Web-Ressourcen**  
Kopien der zitierten Internet-Quellen

### **A.2. Interface Abbildungen**

Container: Personen: **A.1**

Container: Räume: **A.5**

Container: Schwarzes Brett: **A.9**

Container: Vorlesungen: **A.11**

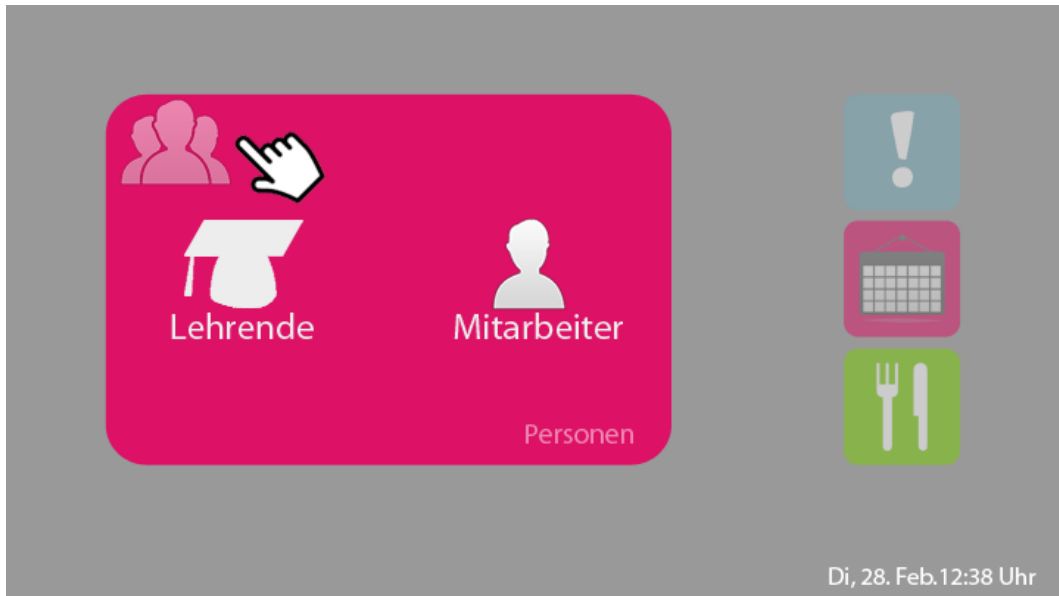


Abbildung A.1.: Selektion des Containers Personen.

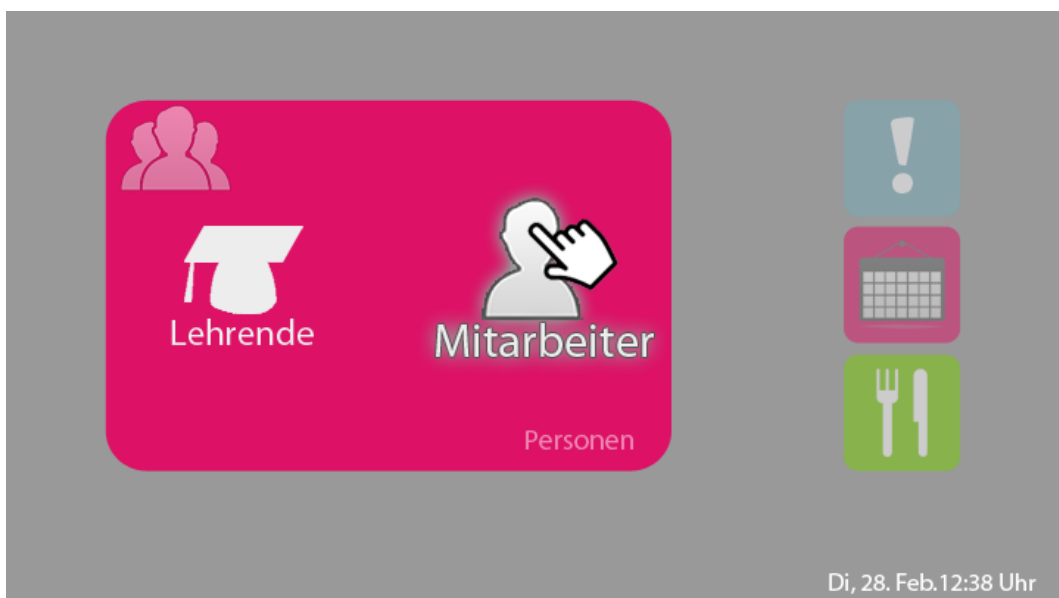


Abbildung A.2.: Bestätigung anhand einer Push-Geste.



Abbildung A.3.: Container Personen: Die Darstellung der Objekte umfasst den Namen und ein Bild.



Abbildung A.4.: Selektion eines Objektes: Weitergehende Inhalte des Objektes werden dargestellt.

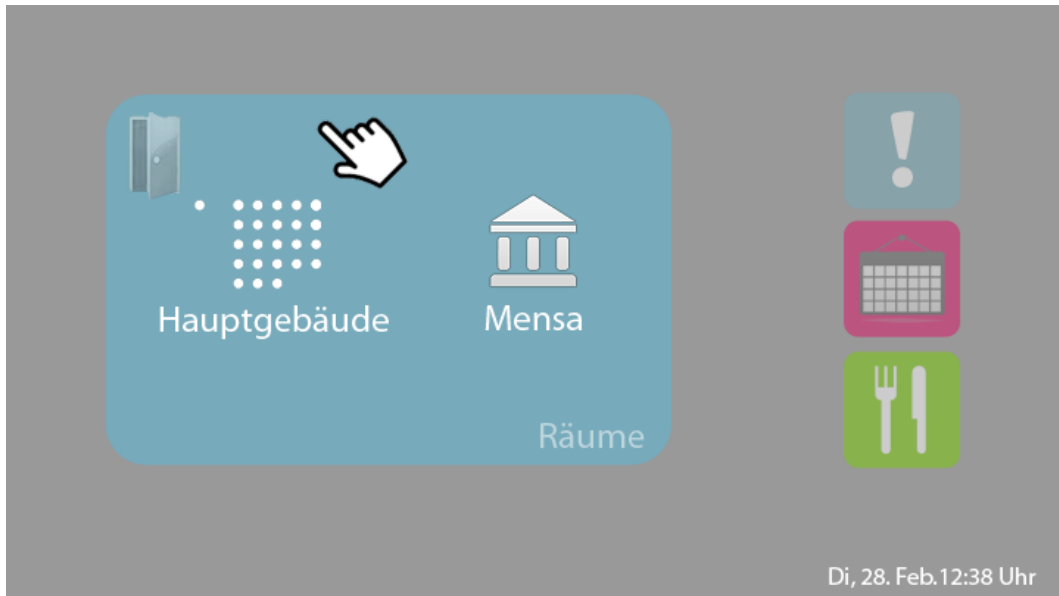


Abbildung A.5.: Selektion des Containers Räume.

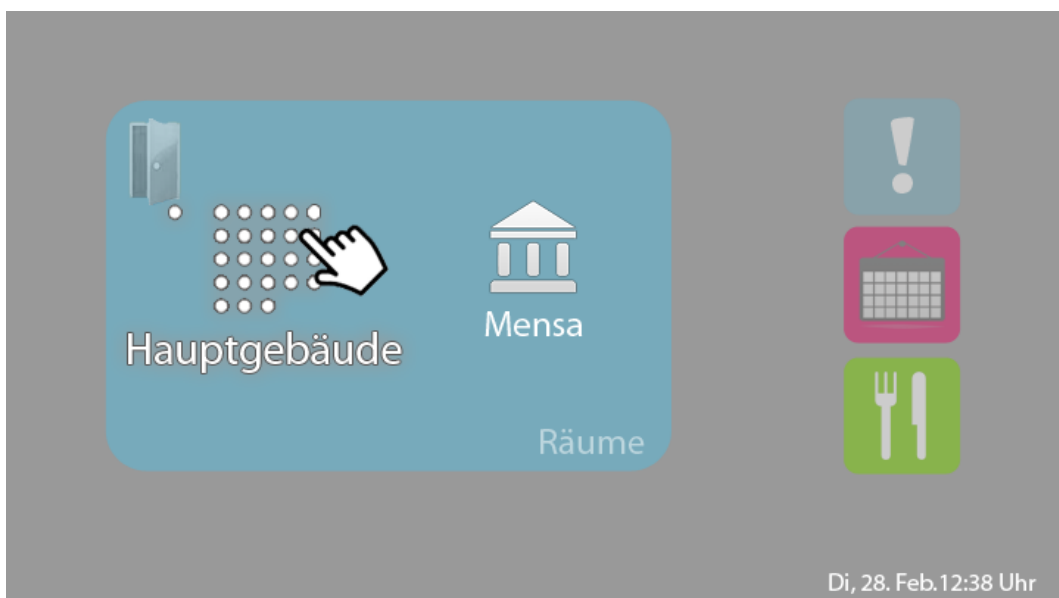


Abbildung A.6.: Bestätigung anhand einer Push-Geste.



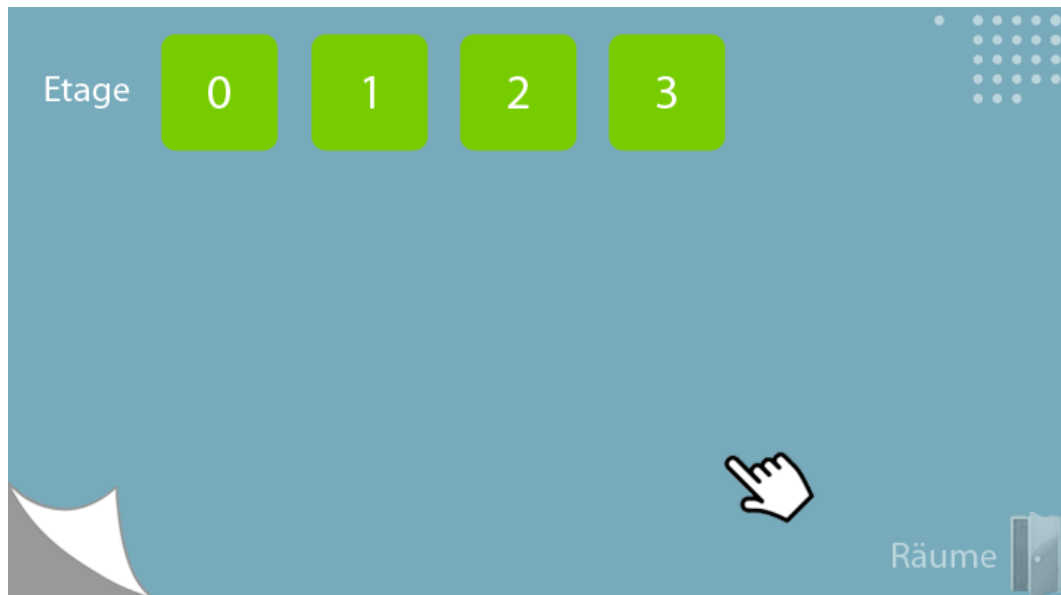


Abbildung A.7.: Container Räume: Anhand der Etagen wird eine Unterscheidung vorgenommen.

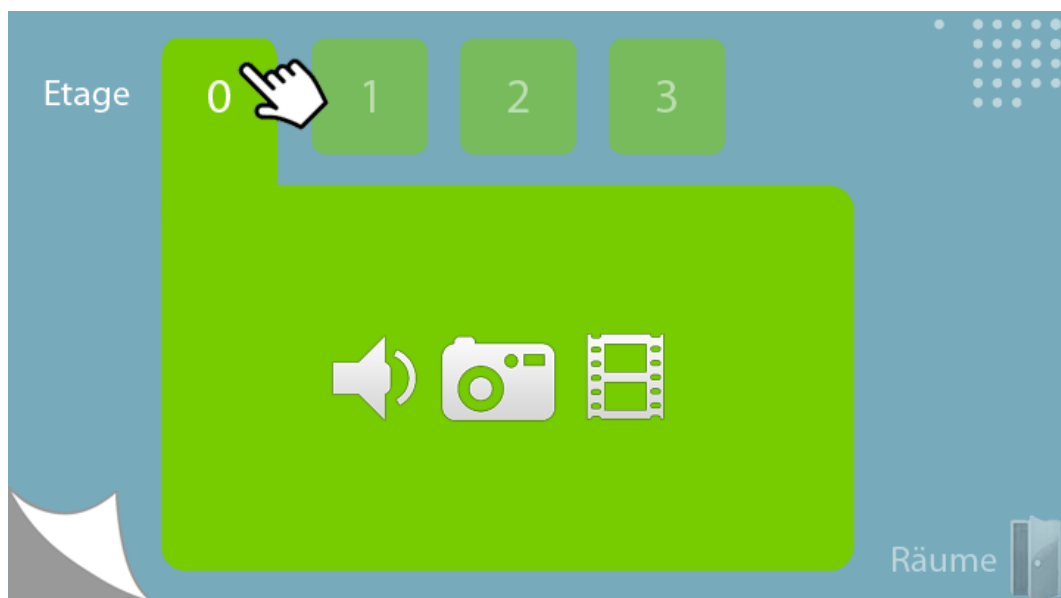


Abbildung A.8.: Container Räume: Die Darstellung der Objekte orientiert sich am Anwendungszweck. Eine angemessene Darstellung für Räume wäre z. B. ein Video, das den Weg zum Ziel beschreibt.

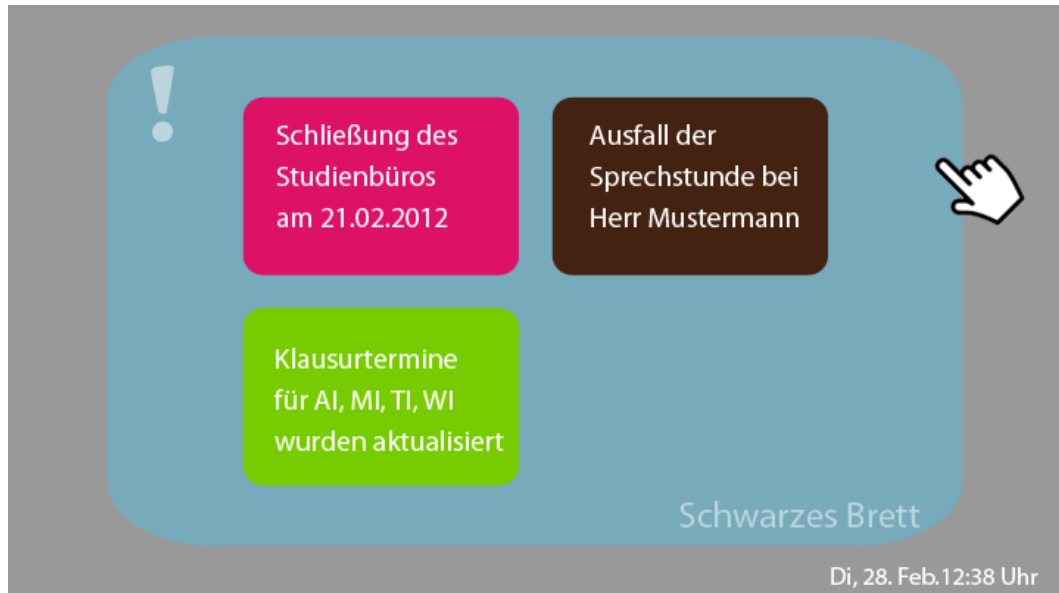


Abbildung A.9.: Selektion des Containers Schwarzes Brett. Eine Voransicht auf die hinterlegten Objekte wird angeboten.

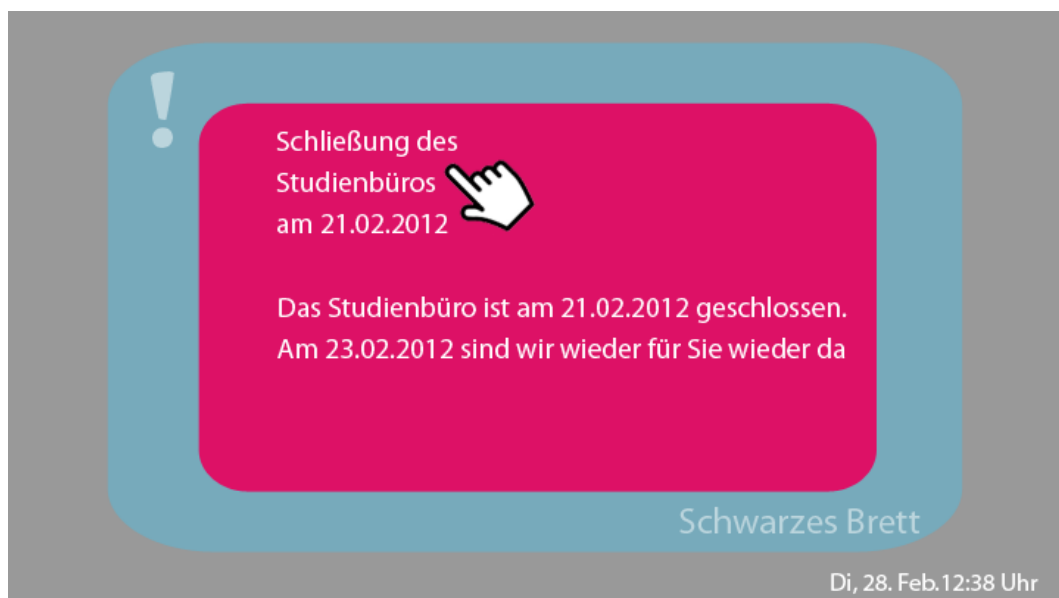


Abbildung A.10.: Selektion eines Objekts im Container Schwarzes Brett.



Abbildung A.11.: Selektion des Containers Vorlesungen.

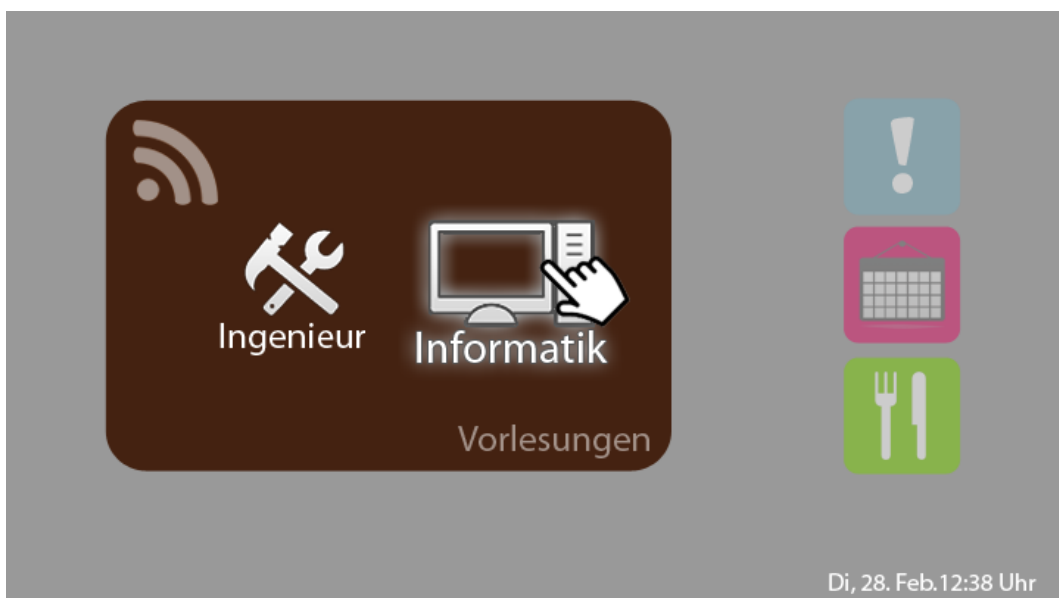


Abbildung A.12.: Bestätigung anhand einer Push-Geste.

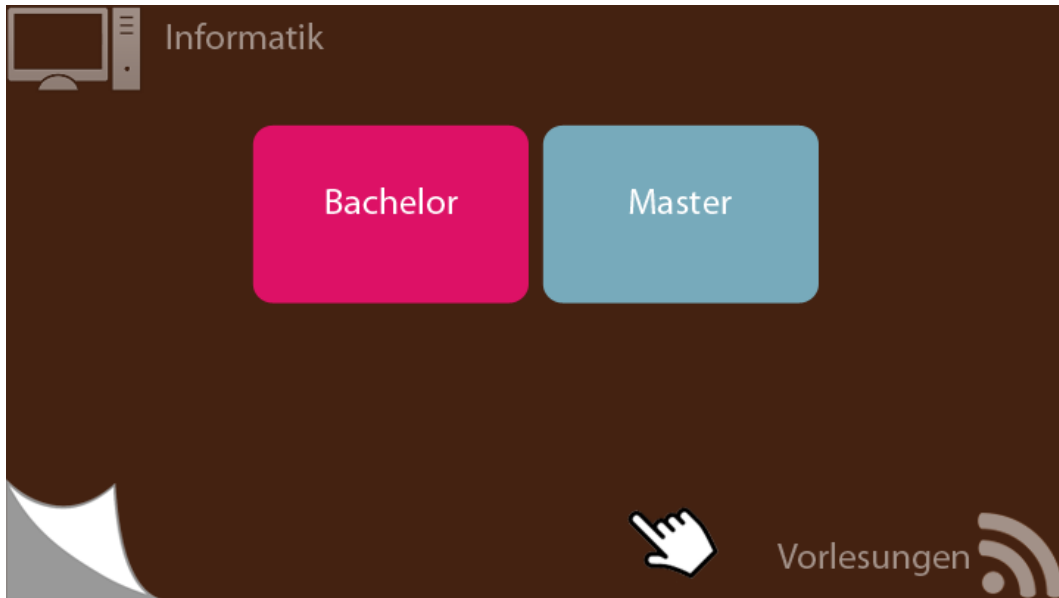


Abbildung A.13.: Selektion der Container Bachelor, Master.

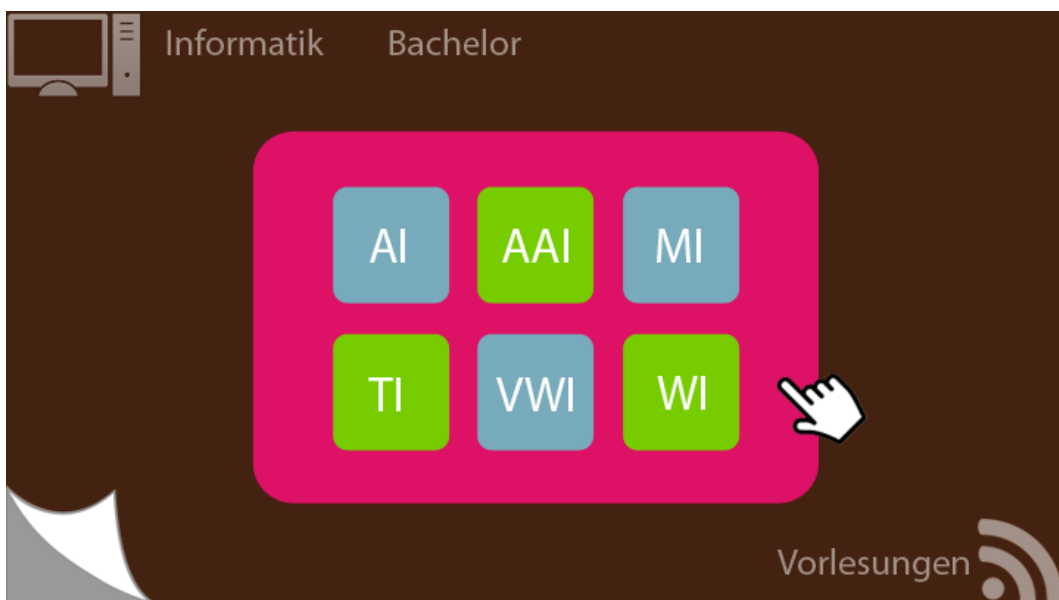


Abbildung A.14.: Selektion des Container Studiengang.

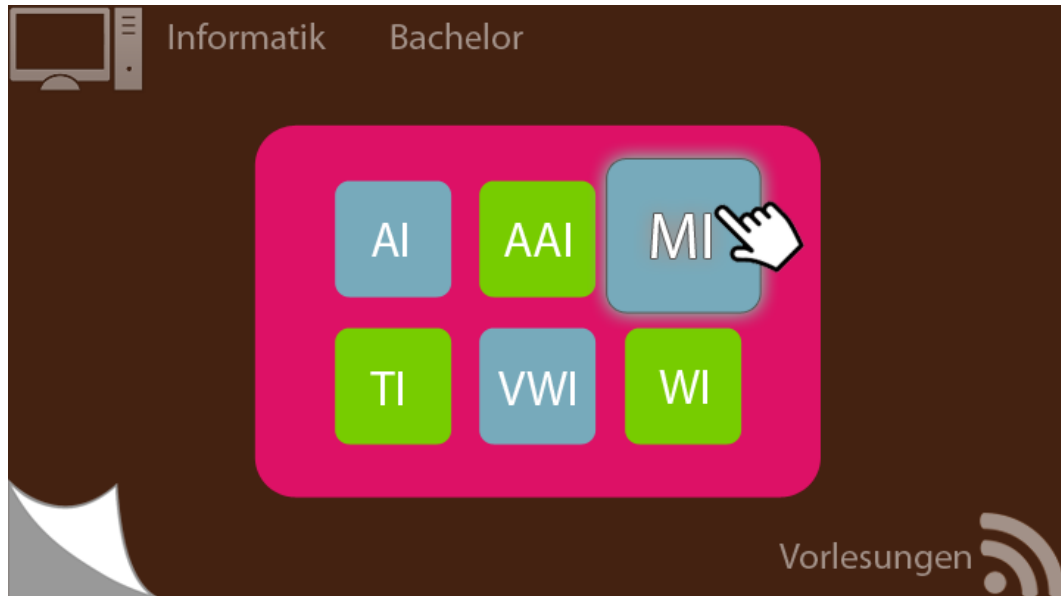


Abbildung A.15.: Bestätigung anhand einer Push-Geste.

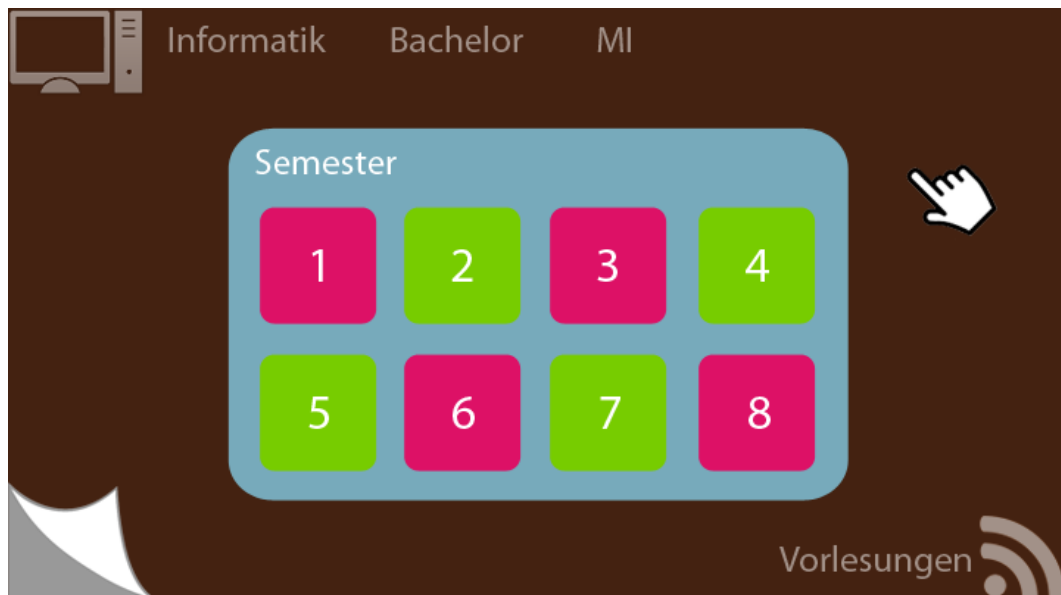


Abbildung A.16.: Selektion des Containers Semester. Daraufhin gelangt man zum Objekt, dem persönlichen Vorlesungsplan.