



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

# **Masterarbeit**

**Stefan Gehn**

**Interpretation mehrdeutiger Eingabedaten auf Infrarot-basierten  
Multitouchdisplays**

Stefan Gehn

**Interpretation mehrdeutiger Eingabedaten auf Infrarot-basierten  
Multitouchdisplays**

Masterarbeit eingereicht im Rahmen der Masterprüfung  
im Studiengang Master of Science Informatik  
am Studiendepartment Informatik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg  
Betreuender Prüfer: Prof. Dr. Kai v. Luck

Zweitgutachter: Prof. Dr. Gunter Klemke  
Eingereicht am: 24. Februar 2010

**Stefan Gehn**

**Thema der Arbeit**

Interpretation mehrdeutiger Eingabedaten auf Infrarot-basierten Multitouchdisplays

**Stichworte**

Ubiquitous Computing, HCI, Multitouch, Infrarot

**Kurzzusammenfassung**

Die vorliegende Arbeit beschreibt die Evaluation einer Infrarotgitter-basierten Multitouchhardware in Hinblick auf ihre Eignung für eine natürlich wirkende gestenbasierte Interaktion. Es wird eine Analyse der Hardware durchgeführt um die Eigenschaften der Hardware bezüglich der Eingabe zu untersuchen. Die Analyse erfolgt mit Hilfe einer Testworkbench welche die Grundlage für alle weiteren Betrachtungen bildet. Im Anschluss wird ein Verfahren vorgestellt, mit dem die während der Analyse erkannten problematischen Eigenschaften der Hardware kompensiert werden sollen. Das erarbeitete Verfahren wird im Rahmen einer Testanwendung eingesetzt und durch einen Usability-Test auf seine praktischen Auswirkungen auf die Eingabequalität für den Anwender untersucht. Abschließend werden die Ergebnisse des Tests ausgewertet sowie mögliche Verbesserungen für zukünftige Arbeiten diskutiert.

**Stefan Gehn**

**Title of the paper**

Interpretation of ambiguous input on infrared-based multitouch displays

**Keywords**

Ubiquitous Computing, HCI, Multitouch, Infrared

**Abstract**

This thesis describes the evaluation of an infrared grid-based multitouch-hardware in regard to suitability for a natural and gesture-based interaction. An analysis is conducted to determine the properties of the hardware in regard to input handling. The analysis is based on a test-workbench that also serves as a foundation for all further examinations throughout this thesis. After the analysis a method is presented which tries to compensate for the problematic hardware properties that were discovered. The developed method is used in the context of a testing-application and reviewed for its impact on practical input quality by conducting a usability test. The thesis concludes by evaluating the results of the usability test as well as giving an outlook on possible future improvements.

# Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>8</b>
1.1. Vision	9
1.2. Multitouch-Hardware	9
1.3. Ziele	10
1.4. Aufbau dieser Arbeit	10
<b>2. Grundlagen</b>	<b>12</b>
2.1. HCI und Usability	12
2.2. Seamless Interaction	13
2.3. Mentale Modelle	13
2.4. Direkte Manipulation und Interaktion	14
2.4.1. Eingabe mit Hilfsmitteln	14
2.4.2. Eingabe ohne Hilfsmittel	16
2.5. Single- und Multitoucheingabe	16
2.5.1. Resistive Touchsysteme	17
2.5.2. Kapazitive Techniken	17
2.5.3. Kamerabasierte Systeme	18
2.5.4. Akustische Systeme	20
2.5.5. Infrarotgitter-basierte Techniken	20
2.5.6. Nutzung von Infrarotgitter-basierter Technik	20
2.6. Interaktion durch Gesten	21
2.6.1. Von der Bewegung zur Geste	21
2.6.2. Gestenrepertoire für Multitoucheingabe	22
<b>3. Einsatzgebiete der Multitoucheingabe</b>	<b>28</b>
3.1. Szenarien	28
3.1.1. Organisation visueller Daten	28
3.1.2. Navigation in Datenbeständen	29
3.1.3. Wahl des Szenarios für die Implementation	29
<b>4. Analyse der Rahmenbedingungen</b>	<b>30</b>
4.1. Laboraufbau	30
4.1.1. Software	30
4.1.2. Hardware	31
4.2. Ziele der Analyse	33

4.3. Umsetzung des Analysetools . . . . .	33
4.3.1. Aufbau . . . . .	33
4.4. Ergebnisse der Analyse . . . . .	37
4.4.1. Reaktionsgeschwindigkeit . . . . .	37
4.4.2. Format der übermittelten Daten . . . . .	38
4.4.3. Genauigkeit der Daten . . . . .	38
4.4.4. Mehrdeutigkeit bei mehreren Druckpunkten . . . . .	38
4.4.5. Abschätzung von Druckpunkten auf einer Achse . . . . .	39
4.4.6. Mögliche Ansätze zur Implementation von Gesten . . . . .	40
4.5. Fazit der Analyse . . . . .	43
<b>5. Algorithmen zur Datenaufbereitung . . . . .</b>	<b>44</b>
5.1. Anforderungen an die Eingabedaten . . . . .	44
5.2. Objektidentifizierung und -verfolgung . . . . .	44
5.3. Object-Tracking durch Kombination von Achswerten . . . . .	45
5.3.1. Evaluation der Ansätze für ein Object-Tracking . . . . .	46
5.4. Tracking von Achswerten mit nachfolgendem Object-Tracking . . . . .	47
5.4.1. Ablauf . . . . .	48
5.4.2. Zuordnung alter Achswerte . . . . .	48
5.4.3. Zuordnung aufgeteilter Achswerte . . . . .	49
5.4.4. Zuordnung zusammenfallender Achswerte . . . . .	50
5.4.5. Tracking virtueller Werte . . . . .	51
5.4.6. Identifikation neuer Achswerte . . . . .	52
5.4.7. Object-Tracking . . . . .	52
5.4.8. Test des Achswert-Trackings innerhalb der Workbench . . . . .	52
<b>6. Praktischer Einsatz der Algorithmen . . . . .</b>	<b>56</b>
6.1. Testanwendung "Touchbook" . . . . .	56
6.1.1. Funktionsumfang . . . . .	56
6.1.2. Interaktionsmöglichkeiten . . . . .	57
6.2. Systemaufbau . . . . .	60
6.2.1. Architektur des Systems . . . . .	60
6.2.2. Implementation des Controllers . . . . .	60
6.2.3. Stand der Implementation & Ausblick . . . . .	65
<b>7. Usability-Test . . . . .</b>	<b>67</b>
7.1. Vorbereitung . . . . .	67
7.1.1. Ziele des Usability-Tests . . . . .	67
7.1.2. Definition der Zielgruppe . . . . .	67
7.1.3. Aufbau des Usability-Tests . . . . .	67
7.1.4. Fragen an Probanden . . . . .	69
7.2. Durchführung . . . . .	70
7.2.1. Ergebnisse für einzelne Gesten . . . . .	71

7.2.2. Ergebnisse pro Testproband . . . . .	72
7.3. Auswertung . . . . .	78
7.3.1. Vorwissen der Probanden . . . . .	78
7.3.2. Qualität der Eingabeerkennung . . . . .	78
7.3.3. Gestenspezifische Beobachtungen . . . . .	79
7.3.4. Anwendungsspezifische Beobachtungen . . . . .	79
7.3.5. Weitere Beobachtungen . . . . .	83
7.3.6. Fazit der Auswertung . . . . .	83
<b>8. Fazit</b>	<b>85</b>
8.1. Direkte Interaktion mittels Multitouch . . . . .	85
8.2. Ausblick . . . . .	86
<b>Literaturverzeichnis</b>	<b>89</b>
<b>A. State-Machine des Serial Readers</b>	<b>92</b>
<b>B. Gestenrepertoire für den Usability-Test</b>	<b>93</b>
<b>C. Aufgaben für den Usability-Test</b>	<b>94</b>
<b>D. Fragebogen zum Usability-Test</b>	<b>95</b>

# 1. Einleitung

Die Grundsteine des bei Computersystemen angewendeten Bedienkonzepts bestehend aus Fenstern, Symbolen, Menüs und Zeigegerät (WIMP-Paradigma) wurden bereits in den 1960er Jahren von Doug Engelbart durch die Erfindung der Computermaus sowie von Alan Kay durch die Umsetzung überlappender Fenster gelegt. Seit den 1980er Jahren stellt das WIMP-Paradigma sowie die damit verknüpfte Desktopmetapher mehr oder minder den Standard für Oberflächen- und Eingabedesign von Computersystemen dar.

In den letzten Jahren setzen sich verstärkt auch andere Paradigmen durch, welche die abstrakte Darstellung der etablierten Desktopsysteme teilweise abschaffen und zudem nicht mehr ausschließlich mit Hilfe von Maus und Tastatur gesteuert werden. Alternative Eingabesysteme wie Touchpads und Touchscreens finden sich immer häufiger in alltäglichen Computersystemen. Touchsysteme, die lange Zeit ausschließlich dem Schutz vor Vandalismus bei Fahrkartenterminals und Bankautomaten dienten, finden mittlerweile sowohl für Werbezwecke als auch als Eingabesystem für PDAs und Mobiltelefone vermehrt Anwendung. Mit der Einführung von Geräten wie dem Apple iPhone wurde zudem die öffentliche Wahrnehmung von touchbasierter Eingabe noch weiter verstärkt.

Die Einführung alternativer Eingabe- und Darstellungsformen lässt sich zum Einen auf die geringen Ausmaße einiger Geräte wie beispielsweise PDAs zurückführen, bei denen eine Interaktion auf Basis von Menüs und Fenstern nur schwer möglich erscheint. Zum Anderen zeigt sich in jüngster Zeit vermehrt, dass das auf Tastatur- und Mauseingabe zugeschnittene WIMP-Paradigma weder für alle Anwender noch für alle Anwendungsklassen ideal zu sein scheint. Für Benutzer, die eine Desktopanwendung zum ersten Mal einsetzen, ist die Bedienung oftmals nicht offensichtlich. Für Anwender, die dagegen regelmäßig mit einer Anwendung arbeiten, erscheint die Trennung von Aktionen und den Objekten, auf die eine Aktion angewendet wird, als hinderlich für eine flüssige Arbeitsweise.



### 1.1. Vision

An Stelle der per Maus und Tastatur gesteuerten Desktop-Metapher könnten Schnittstellen treten, die natürlichen Vorbildern nachempfunden sind und dem Benutzer daher eher vertraut sind als eine abstrakte Schnittstelle. Diese, oftmals auch als Natural User Interface oder Reality-Based Interface bezeichneten Schnittstellenformen zeichnen sich durch eine möglichst direkte, da ohne Hilfsmittel wie Computermäuse auskommende Interaktion zwischen virtuellen Objekten und dem Anwender aus.

Um nicht nur eine direkte sondern auch eine möglichst natürlich wirkende Interaktion zu ermöglichen, sollte sich diese nur geringfügig von der Interaktion mit realen Objekten unterscheiden, insbesondere wenn reale Abläufe und Objekte in einer virtuellen Darstellung nachgeahmt werden. Da Gesten ein essentieller Teil der menschlichen Interaktion darstellen, bietet es sich an, diese als Teil der Mensch-Maschine-Interaktion einzusetzen.

Die bekannteste Umsetzung einer direkten Interaktion stellt neben dem Konzept des Tangible Computing (siehe [Ishii \[1997\]](#)), die Nutzung von Touchscreens dar. Die Einführung von Multitouch-Displays ermöglicht zudem sowohl die Eingabe mit mehr als einem Finger als auch mit mehr als einer Hand und entspricht im Vergleich zur Mauseingabe eher der menschlichen Physiologie.

### 1.2. Multitouch-Hardware

Ein zentrales Hindernis für die praktische Evaluation sowie den Einsatz von Multitouch-Hardware stellt die mangelnde Verfügbarkeit der Technik dar. Oftmals befindet sich die in Publikationen referenzierte Hardware noch im Prototypenstadium und steht daher für den praktischen Einsatz oder die Evaluation nicht zur Verfügung.

Die am Markt verfügbare Hardware ist bisher auf wenige Einsatzgebiete beschränkt. Insbesondere im Bereich mobiler Geräte die nur eine kleine Eingabefläche besitzen, setzt sich die Eingabe mit mehr als einem Finger und die Nutzung intuitiver Gesten bereits durch. Im Bereich herkömmlicher Computersysteme die mit weitaus größeren Anzeigegeräten arbeiten, existieren dagegen nur wenige Lösungen. Die dort eingesetzte Hardware ist aufgrund ihrer Eigenschaften wie Größe, Lebensdauer und letztendlich auch wegen ihrer Anschaffungskosten jedoch nur für einige wenige Einsatzgebiete geeignet. Als Beispiel sind hier das Multitouch-fähige Microsoft Surface sowie die von Jeff Han (siehe [Per](#)) entwickelten Lösungen zu nennen. Beide Produkte basieren auf dem Prinzip der Rückprojektion der Bildschirmanzeige, welches eine hohe Einbautiefe nach sich zieht.

Eine Alternative könnte die Nutzung von Infrarotgitter-basierter Hardware darstellen. Diese ist von mehreren Herstellern bereits verfügbar und ist für die Singletouch-Eingabe bereits im Einsatz.

Die Nutzung dieser Technik für eine gestenbasierte Multitouch-Eingabe scheint dagegen bisher nicht öffentlich publiziert worden zu sein.

**Fazit** Die zuvor genannte Vision ist bisher auf einige wenige Einsatzgebiete begrenzt, ein Einsatz jenseits mobiler Geräte stellt momentan eher eine Ausnahme dar. Der Einsatz Infrarotgitter-basierter Hardware könnte bestehende Systeme unkompliziert um Multitouch-Fähigkeiten erweitern und daher eine alternative Eingabeform für weitere Szenarien sowie Einsatzorte darstellen.

### 1.3. Ziele

Im Rahmen dieser Arbeit sollen mehrere Ziele parallel verfolgt werden. Zum Einen soll eine Evaluation einer Infrarotgitter-basierte Hardware auf ihre Fähigkeiten in Bezug auf eine Multitouch-Eingabe erfolgen. Darauf aufbauend soll die Umsetzbarkeit eines vorher festgelegten Gestenrepertoires auf der vorliegenden Hardware geklärt werden. Die zuvor gewonnenen Erkenntnisse sollen anschließend anhand der Implementation eines praxisnahen Szenarios in Form einer Testanwendung validiert werden. Anhand eines Usability-Tests auf Grundlage der zuvor implementierten Testanwendung soll geprüft werden, in wie weit sich das in der Vision genannte Ziel einer direkten und natürlichen Interaktion umsetzen ließ.

Die vorliegende Arbeit nutzt als Zielsystem einen Tisch, in den ein 42" TFT-Monitor eingelassen ist. Die Interaktion ähnelt der Werkstattperspektive: Auf dem Tisch angezeigte Objekte können durch den Anwender manipuliert werden, vergleichbar mit zu bearbeitenden Gegenständen auf einer Werkbank.

Die Ausmaße der Anzeigefläche des eingesetzten Systems ist größer als bei normalen Computerarbeitsplätzen und insbesondere größer als bei tragbaren Geräten. Die spezifischen Probleme tragbarer Geräte wie die starke Verdeckung der Anzeige durch die Eingabe mit Hilfe von Fingern können für diese Arbeit daher vernachlässigt werden<sup>1</sup>.

### 1.4. Aufbau dieser Arbeit

Im zweiten Kapitel erfolgt zunächst eine Einführung in die Begrifflichkeiten dieser Arbeit gefolgt von einer Kategorisierung gängiger Schnittstellen für die Eingabe mit Hilfe von Händen. Die Kategorisierung von Eingabeformen wird mit einer Übersicht über die gebräuchlichsten Touchtechniken abgeschlossen. Des Weiteren wird die Wahl der in dieser Arbeit genutzten Hardware näher erläu-

---

<sup>1</sup>Zu Projekten und Forschungsergebnissen zum Einsatz von Multitouch in kleinen Geräten siehe [Baudisch \[2010\]](#)

tert. Abschließend erfolgt die Vorstellung eines Gestenrepertoires, welches die Grundlage für die später innerhalb der Testanwendung konkret umgesetzten Gesten bildet.

Innerhalb des dritten Kapitels wird die eingesetzte Touchhardware genauer analysiert. Die Analyse wird mit Hilfe einer erweiterbaren Testworkbench durchgeführt, welche sowohl alle benötigten Daten sowohl für eine Beurteilung der Hardwarefähigkeiten liefert, als auch die Grundlage für die spätere Implementation einer Gestenerkennung bildet. Im Rahmen der Analyse wird weiterhin überprüft, welche Gesten des Gestenrepertoires theoretisch auf der gegebenen Hardware umsetzbar sind.

Die Ergebnisse der Analyse werden im Rahmen des fünften Kapitels genutzt um ein Verfahren zu entwickeln, welches die benötigte Datengrundlage für alle umsetzbaren Gesten erzeugt. Die Ergebnisse dieser Verfahren werden am Ende des Kapitels im Rahmen eines informellen Tests dargestellt.

Um das zuvor erarbeitete Verfahren in einem praxisrelevanten Szenario untersuchen zu können, wird im sechsten Kapitel die Implementation einer konkreten Testanwendung vorgestellt, welche die erarbeiteten Erkenntnisse und Verfahren nutzt. Des Weiteren werden einige Besonderheiten in Bezug auf die Integration von Multitoucheingaben in die Ereignisverarbeitung aktueller Betriebssysteme und Bibliotheken diskutiert sowie die Implementation einzelner Gesten näher erläutert. Abschließend wird ein kurzer Ausblick auf mögliche zukünftige Entwicklungen der Applikation sowie zukünftiger Schnittstellen für die Multitoucheingabe auf Anwendungs- und Betriebssystemebene gegeben.

Die Testanwendung wird innerhalb des siebten Kapitels einem Usability-Test mit mehreren Testprobanden unterzogen. Zum Einen soll mit diesem Test die Eingabequalität für untrainierte Anwender geprüft werden. Zum Anderen soll der Test eine Einschätzung ermöglichen, in wie weit sich die zu Anfang skizzierte Vision mit Hilfe der vorliegenden Hardware und dem Einsatz der erarbeiteten Verfahren umsetzen ließ.

Die Arbeit schließt mit einer Zusammenfassung des erreichten Standes sowie noch offener Punkte ab. Weiterhin wird ein Ausblick auf den Bereich natürlicher, direkter Interaktion mittels Multitouch-Interfaces sowie auf die Kombination mit anderen alternativen Eingabemethoden gegeben, um die zu Anfang skizzierte Vision zu vervollständigen.

## 2. Grundlagen

An diese Stelle erfolgt eine Vorstellung der Begriffe und Felder, welche die Grundlage dieser Arbeit bilden. Es erfolgt zuerst eine Vorstellung gebräuchlicher Eingabeformen für Computersysteme sowie eine Eindordnung der Eingabe mittels Touchtechniken. Anschließend werden die technischen Eigenschaften unterschiedlicher Multitouchsysteme näher betrachtet und die Wahl der in dieser Arbeit eingesetzten Infrarottechnik erläutert.

### 2.1. HCI und Usability

Human-Computer Interaction (HCI) oder auch Mensch-Maschine-Interaktion (MMI) bezeichnet das umfangreiche Feld der Gestaltung, Evaluierung sowie Umsetzung von Interaktionsmöglichkeiten zwischen Mensch und Maschine, wobei sich HCI auf computerbasierte Maschinen beschränkt. Da keine endgültig feststehende Definition für den Begriff existiert, gibt es diverse Arbeitsdefinitionen, die sich geringfügig unterscheiden. Unter anderem wurde der Begriff in [Hewett u. a. \[1997\]](#) folgendermaßen definiert:

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

Als allgemeine Richtlinie für den Bereich Usability kann die DIN EN ISO 9241-10 (siehe [\[ISO, 1996\]](#)) angesehen werden, die Gestaltungsgrundsätze für Dialoge zwischen Mensch und Maschine beschreibt. Zumeist werden diese Grundsätze bei der Gestaltung von grafischen Oberflächen herangezogen, sie können jedoch auch als grobe Richtlinie für die Gestaltung einer Eingabeform dienen. Auf Eingabeformen bezogen sind vor allem die in der Norm genannten Punkte der Selbstbeschreibungsfähigkeit, sowie Erwartungskonformität von Interesse:

- Selbstbeschreibungsfähigkeit

Ein Dialog ist selbstbeschreibungsfähig, wenn jeder einzelne Dialogschritt durch Rückmeldung des Dialogsystems unmittelbar verständlich ist oder dem Benutzer auf Anfrage erklärt wird.

Das System muss dem Anwender ersichtlich machen, welche Eingabemöglichkeiten bestehen und wie er diese Eingabe durchführen kann. Dies kann beispielsweise visuelles Feedback oder auch eine Kurzbeschreibung der möglichen Eingabemöglichkeiten sein.

- Erwartungskonformität

Ein Dialog ist erwartungskonform, wenn er konsistent ist und den Merkmalen des Benutzers entspricht, z.B. seinen Kenntnissen aus dem Arbeitsgebiet, seiner Ausbildung und seiner Erfahrung sowie den allgemein anerkannten Konventionen.

Analog zur Erwartungskonformität von Dialogen lässt sich für eine Eingabeform sagen, dass diese für den Anwender einfacher zu erlernen ist, wenn er Kenntnisse über zuvor genutzte Systeme sowie Eingabeformen darauf übertragen kann.

## 2.2. Seamless Interaction

Im Bereich des Collaborative Workplace definiert [Ishii u. a., 1994, S. 84] den Begriff "Seamless Collaboration" auf zwei unterschiedliche Arten. Beide Definitionen sind in Teilen auch auf "Seamless Interaction" übertragbar.

Zum Einen bezeichnet der Begriff die Übertragbarkeit von realen in virtuelle Arbeitsabläufe ("continuity"). Falls diese Übertragbarkeit nicht gegeben ist, muss der Anwender bei der Nutzung des Systems neue Arbeitsabläufe erlernen, was für ihn als unbequeme Hürde während der Umstellung erscheinen kann.

Die zweite Definition in [Ishii u. a., 1994] bezieht sich dagegen auf den möglichst reibungslosen Übergang zwischen unterschiedlichen Arbeitsumgebungen und wird dort als "smooth transition" bezeichnet. Im Bereich der virtuellen Realität bezieht sich der Begriff Seamless Interaction nach [Forsberg u. a., 1997] vor allem auf den fließenden Übergang zwischen unterschiedlichen Werkzeugen.

## 2.3. Mentale Modelle

Das unter anderem in Dedre Gentner [1983] sowie Davidson u. a. [1999] beschriebene Konzept mentaler Modelle bezeichnet die Bildung einer mentalen Repräsentation eines realen oder imaginären Vorgangs aufgrund der Wahrnehmung, Vorstellungskraft sowie Interpretation durch eine Person. Die Betrachtung sowie Benutzung eines Systems führt beim Anwender zu Annahmen, die sich aus dem Feedback des Systems und zuvor gemachten Erfahrungen bilden. Mentale Modelle

können unter anderem eine Grundlage für die in 2.1 auf Seite 12 beschriebene Erwartungshaltung des Anwenders bilden. Der Umfang und die Komplexität eines mentalen Modells können variieren, zudem muss ein mentales Modell weder vollständig noch korrekt sein.

Mentale Modelle des Benutzer können bei der Entwicklung einer Mensch-Maschine-Schnittstelle als Grundlage dienen. Die Herausforderung hierbei ist es, das vom Entwickler stammende mentale Modell, welches unweigerlich während der Entwicklung entsteht, mit dem mentalen Modell des Benutzers in Einklang zu bringen. In Rogers u. a. [1992] wird angemerkt, dass eine Befragung des Anwenders zu seinem mentalen Modell dieses bereits verändern kann. Aus diesem Grund wird im Umfeld der HCI zumeist die Selbstbeobachtung während der Planungsphase einer Schnittstelle eingesetzt.

### 2.4. Direkte Manipulation und Interaktion

Eine direkte Manipulation von virtuellen Objekten ist gegeben, wenn die Interaktion ohne eine Indirektion durch Hilfsmittel erfolgt, die vom Anwender gehalten oder am Körper getragen werden müssen. Hilfsmittel können beispielsweise Mäuse, Joysticks oder auch Handschuhe sein, deren Lage oder Bewegung vom System ausgewertet wird.

Es existieren unterschiedlichste Formen der Mensch-Maschine-Interaktion. Im Folgenden soll eine Kategorisierung handbasierter Eingabemöglichkeiten vorgenommen werden um einen Vergleich sowie eine Abgrenzung zur Multitoucheingabe zu ermöglichen.

#### 2.4.1. Eingabe mit Hilfsmitteln

Für die Eingabe mit Hilfsmitteln werden an dieser Stelle drei unterschiedliche Eingabemöglichkeiten vorgestellt.

**Mausbasierte Cursorsteuerung** Die am meisten verbreitete und vermutlich auch am häufigsten genutzte Eingabeform stellt mit Sicherheit die Computermaus dar. Die Eingabe mit Hilfe einer Maus erzeugt durch die Einführung des virtuellen Mausursors eine Indirektion zwischen der Eingabe des Anwenders und der Reaktion des Computersystems. Bewegungen der Maus verändern die Position des Cursors.

**Stiftbasierte Eingabe** Eine Alternative zur Mauseingabe stellt die Eingabe mit Hilfe eines Stifts dar. Die Bildschirmoberfläche registriert die Spitze des Stifts und setzt dies in eine Eingabe um. Diese Eingabeform wird insbesondere für Geräte mit kleinem Formfaktor, wie PDA oder auch

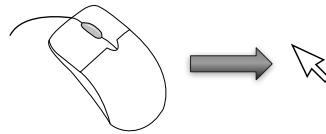


Abbildung 2.1.: Maus & gesteuerter Cursor

(Quelle: Wikimedia Commons, Souris\_schema\_svg, Cursor-design1-arrow.svg)

Laptops eingesetzt, bei denen eine Bedienung mit einem Finger nur schwer möglich ist (siehe auch [2.4.2 auf der nächsten Seite](#)).



Abbildung 2.2.: PDA mit Stift zur Eingabe

(Quelle: Wikimedia Commons, AcerN10Wiki.jpg)

**Markerbasierte Positionserkennung im Raum** Bei der Eingabe im Raum muss der Benutzer Körperextremitäten wie Hände, Füße oder Finger durch Marker für das System sichtbar machen. Dies wird beispielsweise über Reflektoren oder sich farblich stark abhebende Gegenstände wie Handschuhe ermöglicht. Als Haupteinsatzgebiet sind vor allem Szenarien zu nennen, deren Objekte drei Dimensionen aufweisen und die innerhalb eines virtuellen Raums beliebig ausgerichtet werden können. Hierzu zählen unter anderem Anwendungen im Bereich Computer Aided Design (CAD) sowie Visualisierungssysteme für Ärzte.

Ein System welches markerbasierte Positionserkennung bietet ist die PowerWall der HAW, bei der vom Nutzer ein Marker gehalten werden muss, der Infrarotlicht reflektiert. Die Reflektionen

werden von mehreren Kameras registriert, mit Hilfe der Bilddaten der Kameras wird anschließend die Position des Markers innerhalb des überwachten Bereichs errechnet.

### 2.4.2. Eingabe ohne Hilfsmittel

Eingabeformen ohne Hilfsmittel lassen sich ohne besondere Gegenstände wie Stift oder Marker bedienen. Üblicherweise erfolgt die Eingabe über die Erkennung von Hand- oder Fingerbewegungen.

Es existieren drei unterschiedliche Eingabeformen:

1. Singletoucheingabe auf einer Anzeigefläche
2. Multitouch auf einer Anzeigefläche
3. Positionserkennung im Raum ohne weitere Hilfsmittel

Die Single- sowie Multitoucheingabe wird im folgenden Abschnitt näher betrachtet. Die Positionserkennung im Raum ohne den Einsatz besonderer Hilfsmittel ähnelt der markerbasierten Positionserkennung im Raum und ist grundsätzlich für die gleichen Einsatzgebiete geeignet. Im Gegensatz zu markerbasierten Systemen ist jedoch die Erkennung und Verfolgung relevanter Objekte innerhalb des überwachten Raums ungleich komplexer. Trotz der langen Historie dieses Forschungsgebiets ist dem Autor dieser Arbeit kein System bekannt, welches unabhängig vom Einsatzort eine ausreichende Eingabeerkennung für beliebige Nutzer bietet. Für den Einsatz in praxisrelevanten Szenarien kann daher momentan davon ausgegangen werden, dass die markerlose Positionserkennung keine geeignete Eingabeform darstellt.

## 2.5. Single- und Multitoucheingabe

An dieser Stelle erfolgt eine Vorstellung von Systemen, welche die Eingabe mit einem oder mehreren Fingern oder anderen Objektarten auf einem Bildschirm ermöglichen. Da die eingesetzte Touchtechnik sowohl einen Einfluss auf die Freiheiten des Anwenders während der Eingabe als auch auf mögliche Einsatzgebiete hat, ist es notwendig, die verschiedenen Hardwareeigenschaften und das Verhalten in Bezug auf die Eingabe einschätzen zu können. Im Folgenden soll ein kurzer Überblick über die verschiedenen Hardwareansätze und deren bauartbedingte Eigenschaften gegeben werden. Ein Großteil der nachfolgend genannten Techniken steht an der HAW zur Verfügung und konnte daher vom Autor persönlich evaluiert werden.



### 2.5.1. Resistive Touchsysteme

Resistive Touchhardware basiert auf zwei elektrisch leitfähigen Schichten, die sich durch Druck berühren und dadurch einen elektrischen Kontakt bilden, der von der Sensorhardware durch Änderung des Widerstandes innerhalb des Gitters erkannt wird. Im Bereich der Singletouchsysteme kommen resistive Techniken bereits seit längerer Zeit zum Einsatz. Eine Erweiterung auf Multitouchfähigkeiten existiert unter anderem von der Firma Stantum (siehe [Sta](#)), die als Einsatzgebiet kleinere tragbare Geräte wie Telefone, Organizer und Laptops angibt. Aufgrund des technischen Aufbaus werden resistive Techniken üblicherweise nur für kleinere Bilddiagonalen eingesetzt. Materialermüdung sowie Temperaturschwankungen erfordern gegebenenfalls eine erneute Kalibrierung des Systems. Durch ihre Bauart bieten resistive Systeme ein minimales taktiles Feedback, was je nach Anwender und Einsatzzweck sowohl vor- als auch nachteilig sein kann.

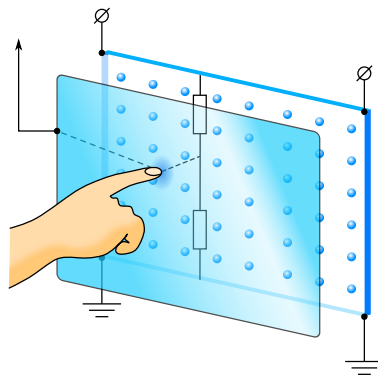


Abbildung 2.3.: Resistiver Touchscreen  
(Quelle: Wikimedia Commons: TouchScreen\_5wires.svg)

### 2.5.2. Kapazitive Techniken

Kapazitive Touchsysteme basieren auf der Messung von Änderungen eines elektrischen Feldes, welches in der Anzeigeoberfläche erzeugt wird. Um eine Änderung des elektrischen Feldes zu bewirken, muss das berührende Objekt daher elektrisch leitfähig sein. Eine Bedienung mit Handschuhen oder Stiften ist aufgrund der fehlenden elektrischen Leitfähigkeit dieser Objekte bei resistiven Systemen nicht möglich. Existierende Systeme verfügen bisher nur über kleine Bilddiagonalen. Ein Einsatz für größere Installationen wie beispielsweise für einen kompletten Tisch ist daher in der Praxis momentan nicht möglich. Das bekannteste Produkt basierend auf kapazitiver Multitouchhardware ist das iPhone der Firma Apple.

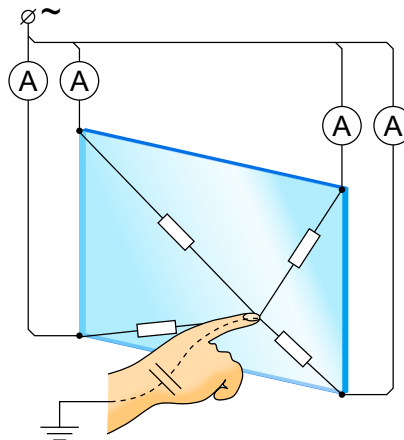


Abbildung 2.4.: Kapazitiver Touchscreen  
(Quelle: Wikimedia Commons: TouchScreen\_capacitive.svg)

### 2.5.3. Kamerabasierte Systeme

Zu den kamerabasierten Systemen zählen mehrere Techniken, die auf einer Abtastung der Touchoberfläche durch eine Kamera aufbauen. Die Kamera registriert dabei je nach Technik Infrarotlicht oder abgelenkte Laserstrahlen und ist bei allen Systemen unterhalb der Bildschirmoberfläche angebracht.

- Frustrated Total Internal Reflection (FTIR, siehe auch [Per](#))
- Diffused Illumination (DI), unter Anderem eingesetzt im Microsoft Surface (siehe [Mic](#))
- Laser Light Plane (LLP)
- Diffused Surface Illumination (DSI)

Eine Übersicht über kamerabasierte Multitouchtechniken sowie die individuellen Vor- und Nachteile jeder Technik ist in [NUI-Group \[2009\]](#) nachzulesen.

Jede der zuvor genannten Techniken benötigt eine spezielle Bildschirmoberfläche, die eine diffuse oder halbdiffuse Oberfläche oder anderweitige spezifische Eigenschaften für die Ablenkung sowie Verteilung von Lichtwellen bieten muss. Durch die eingesetzten optischen System wie Kamera und Beamer ist eine Nutzung herkömmlicher Bildschirmtechnik auf Basis von TFTs nicht möglich. Hinzu kommt der Platzbedarf unterhalb der Anzeigefläche, da dort sowohl Kamera als auch Beamer positioniert werden. Eine Anbringung an Wände ist aufgrund des Platzbedarfs

nicht ohne weiteres möglich. Einige der zuvor genannten Techniken sind empfindlich gegenüber einfallendem Infrarotlicht, da dies die Objekterkennung verfälschen oder je nach Stärke des einfallenden Lichts unmöglich machen kann. Es ist daher oftmals nötig, die Lichtverhältnisse des Raums kontrollieren zu können um einen fehlerfreien Betrieb zu ermöglichen.

Dem Autor ist momentan kein System aus der Praxis bekannt, welches auf Laser Light Plane basiert. Ein Grund hierfür könnte das Gefahrenpotential bei der Verwendung von Laserstrahlen sein. Es kann nicht garantiert werden, dass der Strahl nicht durch einen Gegenstand auf der Anzeigefläche beliebig abgelenkt wird.

### **SMART DVIT**

Von der Firma SMART Technologies wird ebenfalls ein kamerabasiertes Produkt vertrieben (siehe [Dvi \[2003\]](#)), welches sich jedoch stark von den zuvor genannten Techniken unterscheidet. Die Erkennung von Druckpunkten erfolgt von den Seiten eines Rahmens aus, der direkt oberhalb der Bildschirmoberfläche angebracht wird. Vom Formfaktor entspricht diese Technik daher dem in dieser Arbeit untersuchten IRTOUCH-Rahmen. Die zuerst nur für Singletouch-Eingabe konzipierte Technik soll laut Hersteller mittlerweile auch die Erkennung mehrerer Objekte zur gleichen Zeit ermöglichen und daher multitouchfähig sein.

Im Praxiseinsatz an der HAW hat sich gezeigt, dass die offenliegenden Kameras sehr empfindlich sind, das System ist daher nicht gegen Vandalismus gesichert und für den Einsatz in öffentlichen Räumen nur bedingt geeignet.



Abbildung 2.5.: SMART DVIT Rahmen mit Kamera

(Quelle: SMART. Copyright 2001–2008 SMART Technologies ULC. All rights reserved.)

### 2.5.4. Akustische Systeme

Bei akustischen Techniken wie dem Surface Acoustic Wave (SAW) erfolgt die Positionsbestimmung von Objekten auf der Anzeigefläche anhand von Vibrationen auf der Oberfläche. Diese wird mit Hilfe von Ultraschallwellen abgetastet, Reflektionen und Laufzeitunterschiede durch Objekte auf der Oberfläche werden von Sensoren registriert und darauf aufbauend die Objektposition bestimmt.

Ein Einsatz dieses Verfahrens im Multitouch-Umfeld ist dem Autor dieser Arbeit nicht bekannt. Ein Grund hierfür könnten die physikalischen Eigenschaften von Schallwellen und Vibrationen sein, durch die eine Ortung mehrerer Druckpunkte schwierig erscheint, da das Signal durch Phänomene wie die mehrfache Reflektion sich auf unvorhersehbare Weise verändert. Die an der HAW verfügbare Hardware auf SAW-Basis neigt zudem durch von außen zugeführte Schallwellen (lautes Sprechen nahe der Oberfläche) zu Fehlerkennungen.

### 2.5.5. Infrarotgitter-basierte Techniken

Bei Infrarotgitter-basierten Touchsystemen wird oberhalb der Anzeigefläche ein Gitter aus Infrarotstrahlen aufgespannt. Eine Unterbrechung innerhalb des Gitters wird von Sensoren erkannt und daraufhin als erkannte Toucheingabe behandelt (siehe auch [4.1.2 auf Seite 31](#)). Da das Infrarotgitter an den Seiten der Anzeigefläche erzeugt wird, ist eine Umrahmung der Anzeige nötig, die eigentliche Darstellung kann jedoch weiterhin mit Hilfe normaler TFT-Bildschirme erfolgen.

Es existieren bereits mehrere Produkte, die auf dieser Technik basieren. Im Rahmen dieser Arbeit wird der von der Firma IRTOUCH Systems (siehe [\[Irt\]](#)) hergestellte Infrarotrahmen näher betrachtet.

### 2.5.6. Nutzung von Infrarotgitter-basierter Technik

Im Vergleich zu kamerabasierten Systemen scheint die Infrarottechnik nicht oder nur in sehr geringem Ausmaß untersucht worden zu sein. Bei Recherchen zu dieser Arbeit ließen sich keinerlei dokumentierte Erkenntnisse über das Verhalten Infrarotgitter-basierter Systeme im praktischen Einsatz finden.

Weitere Gründe für die Evaluation Infrarotgitter-basierter Touchtechnik liegen in den Zielen dieser Arbeit begründet. Da einer der Schwerpunkte dieser Arbeit die Praxisnähe des Systems ist, sollen auch Eigenschaften jenseits der reinen Eingabeerkennung beachtet werden, die das Gesamtsystem für den Praxiseinsatz erst tauglich erscheinen lassen.

Eine dieser Eigenschaften sind die Ausmaße des Gesamtsystems, welche einen großen Einfluss auf mögliche Einsatzszenarien haben. Durch die geringen Ausmaße und das geringe Gewicht

im Vergleich zu anderen Techniken sowie der damit verbundenen Portabilität der Anzeige kann eine Nutzung sowohl in Form von Wandinstallationen als auch für Tischinstallationen erfolgen. Des Weiteren skaliert die Technik relativ gut mit der gewünschten Anzeigegröße, vom Hersteller der in dieser Arbeit eingesetzten Hardware sind aktuell Rahmen von 15" bis 52" verfügbar. Kapazitive und resistive Techniken sind oftmals nur mit kleineren Bildschirmdiagonalen verfügbar.

Die Infrarottechnik ist im Gegensatz zur kamerabasierten Systemen relativ unempfindlich gegenüber äußeren Einflüssen wie Lichteinfall oder Temperatur. Es wird keine Kalibrierung durch den Benutzer benötigt, da sich das Verhalten des Infrarotgitters durch die zuvor genannten Parameter kaum verändert. Dies ist insbesondere für den Einsatz in öffentlich zugänglichen Installationen von Vorteil, da potentielle Nutzer somit nicht durch eventuelle Kalibrierungsprobleme an der Nutzung des Systems gehindert werden.

Aufgrund der zuvor genannten Punkte erscheint eine Evaluation der Infrarotgitter-basierten Technik auf die Möglichkeiten einer direkten und natürlichen Interaktion unter der Verwendung von Gesten sinnvoll.

## 2.6. Interaktion durch Gesten

Wie bereits in [1.1 auf Seite 9](#) beschrieben, ermöglicht die Multitoucheingabe die Nutzung von Gesten, die sich an realen Bewegungen orientieren und somit eine möglichst natürliche Interaktion. Nachfolgend soll geklärt werden, wodurch eine Geste definiert ist und was eine einfache Manipulation von Objekten wie beispielsweise die Verschiebung eines Objekts von einer Geste unterscheidet.

### 2.6.1. Von der Bewegung zur Geste

Eine Unterscheidung zwischen Gesten und Bewegungen ist in [Kela u. a. \[2006\]](#) beschrieben. Sie basiert primär auf der Komplexität der Eingabeerkennung.

**Bewegungserkennung; Measure and Control** Direkte Umsetzung von Messwerten wie Neigung, Drehung etc.

**Gestenerkennung; Gesture Command** Kommandos bei denen der Anwender Anfang und Ende markiert. Eine Markierung kann beispielsweise das Drücken eines Knopfes während der Eingabe sein oder auch eine Zwangspause zwischen zwei Gesten. Die Gesten werden meist statisch vorgegeben oder können durch den Anwender trainiert sowie erweitert werden.

**Gestenstrom; Continuous Gesture Command** Gestenstrom bezeichnet die Erkennung aus einem permanentem Datenstrom heraus. Das explizite Markieren von Anfang und Ende einer Geste fällt weg, der Anwender kann Gesten direkt aneinander anschließend ausführen. Ein Beispiel hierfür ist die Erkennung von Schreibschrift.

**Relevante Erkennungsarten für diese Arbeit** Sowohl die Bewegungserkennung als auch die Gestenerkennung spielen im Zusammenhang mit der Multitoucheingabe eine Rolle. Die Erkennung eines Gestenstroms liegt dagegen nur bei komplexen Bewegungen mit mehreren Stati vor, wie beispielsweise bei der Schrifterkennung. Da die Eingabe von Schriftzeichen üblicherweise mit einem Stift und daher mit maximal einem Druckpunkt auskommt, ist eine Implementation im Kontext dieser Arbeit unnötig und wird nicht näher betrachtet.

### 2.6.2. Gestenrepertoire für Multitoucheingabe

Im Folgenden erfolgt eine Sammlung von sowie ein Überblick über Gesten sowie einfachen Objektmanipulationen, die in Arbeiten im Bereich HCI mittels Multitoucheingabe bereits implementiert oder anderweitig thematisiert wurden. Im späteren Verlauf dieser Arbeit ist zu klären, welche dieser Gesten auf der vorliegenden Hardware umsetzbar sind und wie diese im Anwendungskontext der Beispielapplikation eine möglichst intuitive Interaktion ermöglichen können.

In [Epps u. a. \[2006\]](#) werden Handhaltungen auf statischen Bildern typischer Desktopszenarien beobachtet. Hieraus lassen sich Einschränkungen sowie Empfehlungen ableiten, die bei der Zusammenstellung eines Gestenrepertoires hilfreich sein können. So nutzte ein Großteil der Testprobanden nur einen Finger für eine Vielzahl von Aktionen und die beidhändige Bedienung wurde primär für eine Zoomaktion verwendet. Gesten sollten demnach möglichst einfach auszuführen sein, was sich sowohl auf die Handhaltung als auch auf die Anzahl der Finger beziehen kann, die für eine Gestenerkennung zu Rate gezogen werden. Vergleichbare Beobachtungen wurden ebenfalls in [Koskinen u. a. \[2008\]](#) gemacht.

Das in [Kim u. a. \[2007, S. 393ff\]](#) beschriebene Gestenrepertoire unterscheidet sich nur geringfügig von denen anderer Arbeiten, jedoch wurde hier auch die technische Umsetzung der Gesten dokumentiert. Einen aktuellen und viele unterschiedliche Gesten umfassenden Überblick bietet [Wobbrock u. a. \[2009\]](#). Der Überblick ist ähnlich wie in [Epps u. a. \[2006\]](#) unabhängig von der eingesetzten Hardware angelegt. Vor einer Implementation muss daher geklärt werden, ob die jeweilige Hardware die gewünschten Gesten erkennen kann.

Im Folgenden sollen einige Gesten vorgestellt werden, die eine intuitive Interaktionen mit einem oder mehreren Objekten ermöglichen.

### Auswahl

Für die Auswahl von Objekten ist lediglich ein einzelner Druckpunkt nötig. Die Eingabe entspricht daher dem von klassischen Singletouch-Systemen. In der Vergangenheit wurden bereits mehrere Variationen für die Auswahl von Objekten entwickelt:

**Land-On** Vorgestellt in [Potter u. a. \[1988\]](#). Die Selektion erfolgt, sowie der Finger die Oberfläche des Touchsystems berührt. Die Auswahl eines anderen Objekts ist durch entfernen und erneutes aufsetzen des Fingers möglich.

**First-Contact** Ähnlich des Land-On-Verfahrens, nutzt die in [Potter u. a. \[1988\]](#) vorgestellte Methode den permanenten Datenstrom der Hardware aus. Eine Selektion erfolgt, sowie der Finger ein Objekt auf dem Bildschirm ganz oder teilweise berührt. Der Anwender kann daher im Vergleich zu Land-On den Finger an einer Position aufsetzen an der sich kein Objekt befindet und nachfolgend das auszuwählende Objekt anvisieren.

**Lift-Off** Die Lift-Off-Methode wurde ebenfalls in [Potter u. a. \[1988\]](#) vorgestellt. Eine Selektion erfolgt bei Lift-Off erst beim Entfernen des Fingers von der Touchoberfläche. Die Selektion erfolgt zudem mit Hilfe eines Cursors, der oberhalb des Druckpunktes eingeblendet wird. Der Benutzer kann bei Lift-Off während der Berührung die Positionierung auf dem zu selektierenden Objekt beliebig lange korrigieren.

**Selektion nach Timeout** Eine Selektion erfolgt erst, nachdem der Finger eine längere Zeit auf einer Position verweilt hat. Diese Methode dient insbesondere der Reduktion von Fehleingaben die bei Anwendungsfeldern wie beispielsweise medizinischen Apparaturen zu Gefahrensituationen führen könnten.

**SimPress** Die in [Benko u. a. \[2006\]](#) vorgestellte SimPress-Technik ermöglicht im Gegensatz zu den vorher vorgestellten Methoden eine Unterscheidung zwischen der Bewegung ("tracking") und der Selektion ("dragging"). Die Selektion wird hierbei durch eine kurzzeitige Vergrößerung der Auflagefläche des Fingers erreicht ("short rocking motion"). SimPress benötigt Hardware, welche auch die Objektgröße erfasst.

**Shadow Tracking** In [Echtler u. a. \[2008\]](#) wurde eine Technik vorgestellt, die ebenfalls eine Unterscheidung zwischen der Bewegung und der eigentlichen Selektion ermöglicht. Voraussetzung hierfür ist die Erkennung von Schatten auf der Anzeigefläche. Aus der Intensität des Schattens lässt sich die Nähe eines Objekts zur Anzeigefläche bestimmen, die Eingabeerkennung besitzt daher eine weitere Unterscheidungsmöglichkeit anhand der gelieferten Tiefeninformationen.

### Verschiebung

Die Verschiebung von Objekten hängt bei Touchsystemen von der Art der unterstützten Selektionsmethode ab. Eine Verschiebung von Objekten erfolgt zumeist mit der Selektion, falls das System nicht zwischen "tracking" und "dragging" unterscheiden kann.



Abbildung 2.6.: Gesten - Verschiebung

### Rotation

Die Rotation von Objekten auf touchbasierten Systemen ist unter Anderem in [Kruger u. a. \[2005\]](#) sowie [Liu u. a. \[2006\]](#) näher betrachtet worden. Die dabei entstandenen Methoden "Rotate and Translate" (RNT) sowie TNT dienen der Abbildung von Rotationsmittelpunkt, Rotationswinkel sowie einem Translationsvektor zur Verschiebung innerhalb einer einzelnen Geste. RNT ist auf beliebigen Singletouchsystemen umsetzbar, da es lediglich einen einzelnen Druckpunkt für die Ausführung der Aktion benötigt. Da das TNT-Verfahren spezielle Sensorhardware benötigt, die vom Anwender, ähnlich einer Maus, bewegt werden muss, wird von einer näheren Betrachtung im Rahmen dieser Arbeit abgesehen.

Bei Multitouchsystemen kann der Anwender den Mittelpunkt der Drehung sowie den Drehwinkel jeweils durch einen Finger festlegen (siehe [Kim u. a. \[2007, S. 3\]](#)). Des Weiteren ließe sich ähnlich wie beim zuvor genannten RNT-Verfahren die Verschiebung mit in die Geste integrieren, in dem man eine gemeinsame Bewegung beider Finger in die selbe Richtung als Verschiebung des Objekts ansieht.

### Skalierung

Die Skalierungsgeste ermöglicht die Skalierung von Objekten wie beispielsweise Fotos oder den Zoom innerhalb einer anzeigefüllenden Ansicht, beispielsweise bei Landkarten. Das zu skalierende Objekt wird bei dieser Geste üblicherweise durch die Position eines oder beider Finger festgelegt. Das Objekt sowie der Skalierungsfaktor werden daher gleichzeitig festgelegt, es ist keine vorher durchgeführte explizite Auswahl des Objekts nötig.



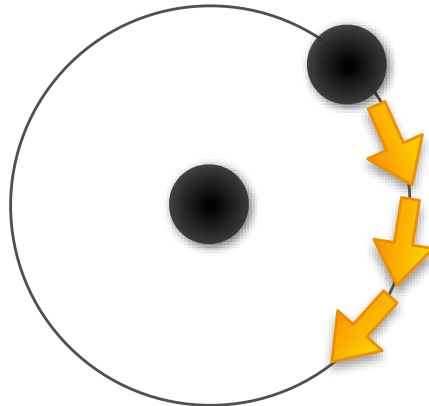


Abbildung 2.7.: Gesten - Rotation

Auf bereits vorhandenen Multitouchsystemen wie dem Apple iPhone wird die Skalierung durch Nutzung von zwei Fingern realisiert (siehe 2.8). Hierfür wird über die Zeit die Abstandsänderung der beiden Finger zueinander auf die Skalierung des zu manipulierenden Objekts umgesetzt. Ein konkreter Ansatz zur Abstandsberechnung wurde in Kim u. a. [2007, S. 3] vorgestellt. Für die Manipulation des Objekts ist unerheblich, ob diese mit einer Hand oder zwei Händen durchgeführt wird. Hinzu kommt, dass bei den meisten Systemen eine Unterscheidung von zwei Händen derselben Person technisch nicht möglich ist.



Abbildung 2.8.: Gesten - Skalierung

### **Flick and Catch**

Die in Wu u. Balakrishnan [2003] vorgestellte Flick-and-Catch-Geste ermöglicht eine Verschiebung von Objekten über eine große Strecke hinweg, insbesondere über eine Entfernung, die größer als der Aktionsradius des Anwenders ist. Die besondere Eigenschaft der Flick-Geste ist die kurzzeitige hohe Beschleunigung des Fingers auf einem Objekt. Sie ähnelt daher einer realen, physikalischen Gesetzen unterworfenen Objektinteraktion. Wie in 2.9 auf der nächsten Seite skizziert benötigt

die Geste ähnlich wie die bereits beschriebene Verschiebung von Objekten nur einen einzigen Druckpunkt und ließe sich daher auch auf einer Singletouch-Hardware umsetzen.



Abbildung 2.9.: Gesten - Flick

### **Sweep-Geste**

Die ebenfalls in [Wu u. Balakrishnan \[2003, S.6\]](#) vorgestellte Sweep-Geste ermöglicht die Manipulation mehrerer Objekte mit Hilfe einer Handkante. Alle Objekte, die während der Bewegung einer Handkante berührt werden, können so gleichzeitig verschoben oder anderweitig bearbeitet werden. Im Vergleich zu den zuvor genannten Gesten benötigt die Sweep-Geste eine Erkennung des Objekts, welches die Bildschirmoberfläche berührt. Die Hardware muss daher für diese Geste zumindest die groben Ausmaße der Berührung liefern können.

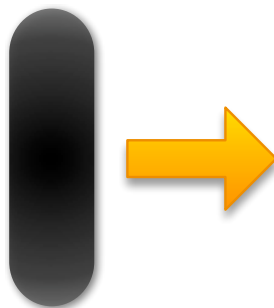


Abbildung 2.10.: Gesten - Sweep

### **Zusammenfassung**

Das vorgestellte Gestenrepertoire stellt die Grundlage sowohl für die Analyse der Hardware als auch für die spätere Implementation einiger dieser Gesten dar. Weiterhin bietet die hier dargestellte

## 2. Grundlagen

---

Auswahl von Gesten bereits einen ersten Überblick über die Möglichkeiten der touchbasierten Interaktion unter Verwendung einer multitouchfähigen Hardware.

Durch den Einsatz natürlicher Bewegungen wie dem Wischen oder der Skalierung durch zwei Finger werden Aufgaben, die bei einer mausbasierten Interaktion üblicherweise aus zwei getrennten Aktionen bestehen durch eine einzige Benutzereingabe ermöglicht. Eine Unterscheidung zwischen der Auswahl des zu manipulierenden Objekts und der darauf anzuwendenden Manipulation ist bei diesen Gesten nicht mehr nötig. Die Reduktion der Anzahl von Schritten die vom Anwender durchgeführt werden müssen um eine bestimmte Manipulation eines Objekts zu erreichen führt daher zu einer vereinfachten Eingabe.

Eine Zuordnung der vorgestellten Gesten zu konkreten Aktionen findet in der nachfolgenden Beschreibung praxisrelevanter Szenarien statt.

## 3. Einsatzgebiete der Multitoucheingabe

Um die Möglichkeiten der Touch- sowie Multitoucheingabe aufzuzeigen werden an dieser Stelle einige Szenarien vorgestellt, die von dieser Eingabeform profitieren können. Hierzu zählen vor allem Szenarien, die eine starke Bindung an eine zweidimensionale Darstellung sowie die Bearbeitung von Objekten auf zweidimensionaler Ebene aufweisen.

Ein weiterer Aspekt für nachfolgend beschriebene Szenarien ist die Realitätsnähe dieser Szenarien. Wie bereits in der Vision in Abschnitt 1.1 auf Seite 9 erläutert wurde, lässt sich eine natürliche Interaktion mit Hilfe von Gesten für den Anwender am ehesten nachvollziehen, wenn die zu manipulierenden Objekte eine reale Entsprechung besitzen und die gewählten Gesten identisch oder ähnlich den Gesten sind, die innerhalb des realen Szenarios zum Einsatz kommen. Szenarien die realen Arbeitsfeldern entstammen sind daher besonders für den Einsatz einer Multitoucheingabe geeignet.

### 3.1. Szenarien

Nachfolgend wird eine Übersicht über mögliche Szenarien gegeben, die die Grundlage für die spätere Testanwendung darstellen. Es findet zudem eine Zuordnung der im Grundlagenkapitel erwähnten Gesten zu Aktionen innerhalb des jeweiligen Szenarios statt.

#### 3.1.1. Organisation visueller Daten

Die klassische Organisation von Bildern oder Fotos auf einem Tisch lässt sich originalgetreu mit Hilfe von Multitouch in einer virtuellen Darstellung nachbilden. Zusätzlich ermöglicht die computer-gestützte Organisation auch ein Ändern der Darstellungsgröße der Bilddaten. Da Fotos mittlerweile meist digital erstellt, verwaltet und archiviert werden, ist dieses Szenario mit mausbasierter Eingabe bereits im praktischen Einsatz. Es könnte daher ein direkter Vergleich zwischen einer maus- und einer touchbasierten Eingabe erfolgen.

Übliche Aktionen in diesem Szenario sind unter Anderem die Skalierung einzelner oder aller Bilder, das Verschieben eines Bilds oder mehrerer Bilder innerhalb einer Gruppierung sowie die Rotation von einzelnen Bildern.

### 3.1.2. Navigation in Datenbeständen

Die Navigation innerhalb zweidimensionaler Datenbestände mit großen Ausmaßen, wie dies bei Landkarten oder einer Globusansicht der Erde der Fall ist, ähnelt auf den ersten Blick dem zuvor genannten Szenario. Im Unterschied zu Diesem existiert üblicherweise jedoch nur ein einzelnes großes Objekt (die Karte), dessen Anzeige vom Anwender zu manipulieren ist.

Um Karteninformationen mit wechselndem Detailgrad zu präsentieren, wird dem Anwender die Kontrolle über Orientierung, Zoomstufe sowie den sichtbaren Ausschnitt gegeben. Durch diese Interaktionsmöglichkeiten mit dem Kartenmaterial kann der Anwender für ihn interessante Bereiche genauer einsehen als es mit einer statischen Kartenansicht mit festem Maßstab möglich wäre. Als Einsatzorte für dynamische Karten kommen beispielsweise größere Gebäude, wie Einkaufszentren, Flughäfen oder auch Bahnhöfe in Betracht.

Die vermutlich am häufigsten eingesetzten Aktionen in diesem Szenario sind die Verschiebung des sichtbaren Ausschnitts sowie der Zoom aus oder in das Kartenmaterial. Beide Aktionen lassen sich mit Hilfe der Verschiebungs- sowie Zoom-Geste umsetzen. Je nach Anwendungsfall kann das Kartenmaterial außerdem drehbar sein, wobei die in [2.6.2 auf Seite 24](#) genannte Rotationsgeste diese Aktion direkt unterstützt.

### 3.1.3. Wahl des Szenarios für die Implementation

Da die benötigten Aktionen und somit auch die eingesetzten Gesten bei der Organisation zweidimensionaler Bilddaten vielfältiger als bei einer Kartennavigation sind, bietet sich eine praktische Umsetzung des zuerst genannten Szenarios an.

## 4. Analyse der Rahmenbedingungen

Um die verwendete Hardware vor der eigentlichen Implementation besser beurteilen zu können, erfolgt an dieser Stelle eine genauere Betrachtung der Hardware. Des Weiteren wird ein kurzer Überblick über die weitere Hard- und Softwareumgebung gegeben.

### 4.1. Laboraufbau

Nachfolgend wird die eingesetzte Hard- und Software vorgestellt. Es werden weiterhin alle vor den praktischen Test bekannten Eigenschaften der Multitouch-Hardware beschrieben sowie eine kurze Vorstellung des seriellen Protokolls gegeben.

#### 4.1.1. Software

Die Entwicklung der Arbeit findet primär auf dem Betriebssystem Mac OS X statt. Da bei der Wahl der Softwarekomponenten auf eine Betriebssystemunabhängigkeit geachtet wird, stellt die Auswahl des genutzten Betriebssystems für die Implementation jedoch nur eine untergeordnete Rolle dar.

Die Softwareentwicklung erfolgt in der Programmiersprache C++ mit Hilfe der portablen Klassenbibliothek Qt (siehe Qt). Diese ermöglicht eine effiziente Darstellung zweidimensionaler Objekte sowie eine freie Ausrichtung und Skalierung dieser Objekte. Die Darstellung erfolgt optional per OpenGL, wodurch die für größere Auflösungen benötigte Skalierbarkeit erreicht wird. Für die Ansteuerung der seriellen Schnittstelle wird die QExtSerialPort-Bibliothek eingesetzt (siehe Sander u. a.).

Eine Implementation in Java wurde nach einer kurzen Evaluation verworfen, da die benötigte Ansteuerung der seriellen Schnittstelle mit Hilfe der RXTX-Bibliothek (siehe Rxt) sich als zu instabil bei längerem Betrieb herausstellte und keine Alternativbibliothek zur Verfügung stand.

### 4.1.2. Hardware

Als Hardware stehen zwei infrarotbasierte Multitouchrahmen der Firma IRTOUCH zur Verfügung, die sich primär durch ihre Größe (17" sowie 42", siehe [7.1 auf Seite 68](#)) unterscheiden. Die Ansteuerung der Rahmen erfolgt per serieller Schnittstelle und erfordert keinerlei Initialisierung oder weitergehende Konfiguration. Der Rahmen sendet Daten, sobald eine Unterbrechung des Infrarotgitters registriert wird. Es werden dabei laut Hersteller bis zu sieben Unterbrechungen gleichzeitig je Achse verarbeitet.

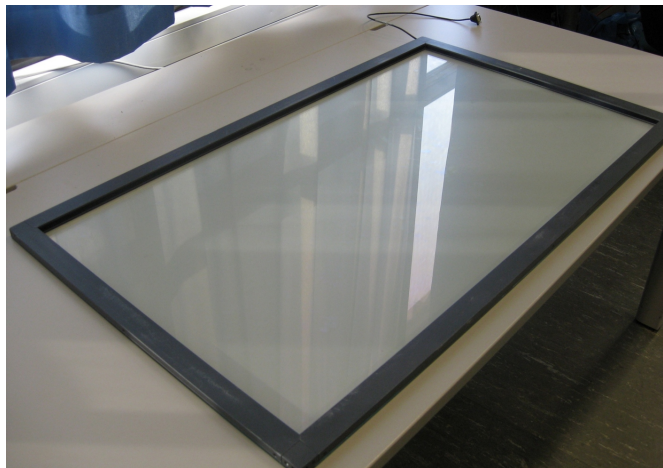


Abbildung 4.1.: IRTOUCH 42" Rahmen

### Funktionsprinzip

Die Hardware erzeugt ein feinmaschiges Gitter von Infrarotstrahlen oberhalb der Bildschirmoberfläche. Werden einer oder mehrere Strahlen von einem Objekt wie z.B. einem Finger unterbrochen, so registriert die Hardware diese Unterbrechung und überträgt den Startwert sowie die Gesamtbreite der Unterbrechung (siehe [4.2 auf der nächsten Seite](#)).

### Serielles Protokoll und Datenformat

Die Übertragung der zuvor genannten Unterbrechungen erfolgt über eine serielle Schnittstelle bei einer festen Baudrate von 38400 bps. Die Daten werden in Form von Paketen übertragen, welche Byteweise verarbeitet werden können. Ein Datenpaket besteht aus den folgenden Elementen:

1. Startwert (Ein Byte mit festem Wert `0x55`)

#### 4. Analyse der Rahmenbedingungen

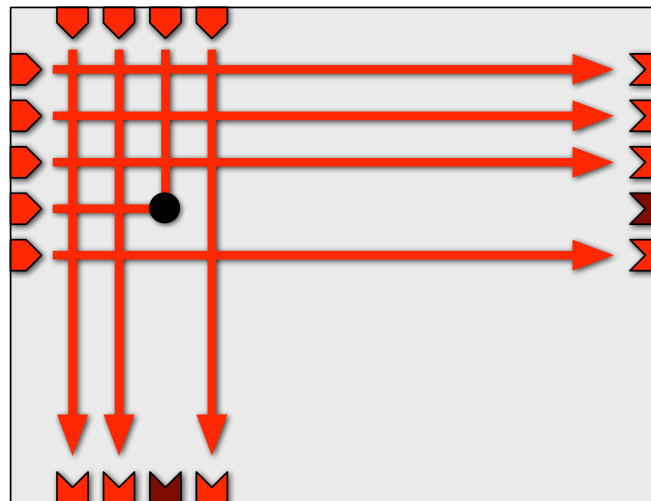


Abbildung 4.2.: Funktionsprinzip Infrarotgitter

2. Anzahl der Werte je Achse (Zwei Bytes)
3. (optional) Liste von Unterbrechungen auf der horizontalen Achse (Startposition sowie Breite, 4 Bytes pro Wert)
4. (optional) Liste von Unterbrechungen auf der vertikalen Achse (Startposition sowie Breite, 4 Bytes pro Wert)
5. Checksumme des Pakets (Summe aller vorigen Bytes dargestellt als ein Byte, der entstehende Wertüberlauf ist beabsichtigt)

Theoretisch sind bei maximal sieben Unterbrechungen pro Achse 80 Datenpakete pro Sekunde übertragbar:  $38400 \text{ bps} / (60 \text{ bytes} * 8)$ . Ein Datenpaket mit je einer Unterbrechung zur Achse ist in 4.1 exemplarisch dargestellt.

Tabelle 4.1.: Beispielpaket mit einer Unterbrechung je Achse

1 Byte	1 Byte	1 Byte	2 Bytes	2 Bytes	2 Bytes	2 Bytes	1 Byte
Start	Anz. vert.	Anz. hor.	Startpos. 1 auf X-Achse	Breite 1 auf X-Achse	Startpos. 1 auf Y-Achse	Breite 1 auf Y-Achse	Checksumme
0x55	0x1	0x1	0x10	0x08	0x0A	0x08	0x81



### 4.2. Ziele der Analyse

Da außer den in 4.1.2 auf Seite 31 bereits genannten technischen Daten, dem grundsätzlichen Funktionsprinzip der Hardware, sowie der Dokumentation des seriellen Protokolls keine weiteren Informationen von Seiten des Herstellers zur Verfügung standen, wird die Multitouch-Hardware vor einer praktischen Nutzung zuerst genauer analysiert. Das Verhalten der Hardware während der Eingabe soll genauer dokumentiert werden, um mögliche Hindernisse bei der Umsetzung von Gesten (siehe 2.6.2 auf Seite 22) frühzeitig erkennen zu können.

Für die Analyse bietet sich die Entwicklung einer Workbench an, welche die ankommenden Touchdaten visualisiert und parallel zur Visualisierung über die Zeit der Eingabe Messwerte sammelt. Durch die praktische Evaluation der Hardware kann diese sowohl quantitativ über die Visualisierung als auch qualitativ über die gesammelten Messwerte beurteilt werden. Die Workbench soll darüber hinaus auch im weiteren Projektverlauf für kontinuierliche Tests eingesetzt und erweitert werden.

### 4.3. Umsetzung des Analysetools

Nachfolgend wird die Implementation der Workbench beschrieben, welche für die anschließende Analyse benötigt wird.

#### 4.3.1. Aufbau

Das Analysetool besteht aus den drei Komponenten Adapter, Treiber sowie der eigentlichen Applikation (siehe 4.3 auf der nächsten Seite). Die Aufteilung in Komponenten ermöglicht den Austausch einzelner Elemente des Systems, z.B. bei einem Schnittstellenwechsel durch den Hardwarehersteller. Des Weiteren können Adapter und Treiber als Grundlage für die Beispielapplikation dienen. Durch die Verwendung identischer Komponenten sowohl im Analysetool als auch in der Beispielanwendung ist sichergestellt, dass das während der Analyse beobachtete Eingabeverhalten identisch zum späteren Verhalten innerhalb der Beispielapplikation ist.

Nachfolgend werden die einzelnen Komponenten sowie deren Schnittstellen näher erläutert.

#### Serial Reader

Der Serial Reader übernimmt die Kommunikation mit der Hardware über die serielle Schnittstelle sowie die Dekodierung und Verifikation ankommender Datenpakete gemäß des in 4.1.2 auf Seite 31 dokumentierten Protokolls. Datenpakete mit fehlerhafter Prüfsumme werden verworfen und

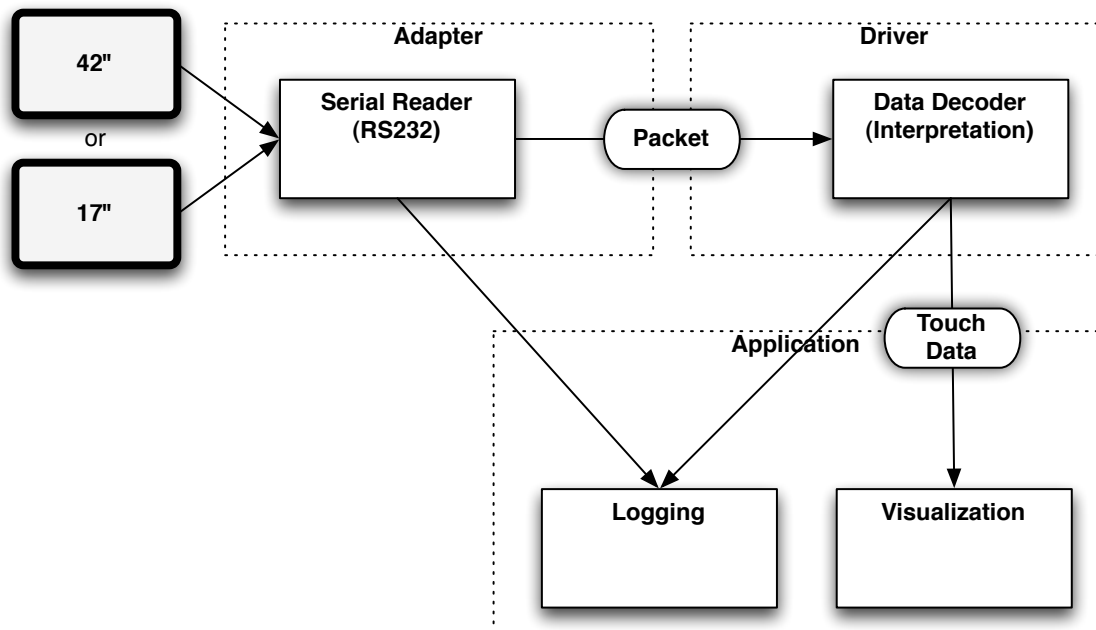


Abbildung 4.3.: Komponenten des Analysetools

nicht an nachfolgende Komponenten weitergegeben. Das Parsing der Datenpakete ist innerhalb einer State Machine umgesetzt und in [A auf Seite 92](#) skizziert. Die erhaltenen Werte werden vom Serial Reader unverändert weitergegeben, eine Skalierung oder anderweitige Interpretation der Daten erfolgt erst innerhalb des nachfolgend beschriebenen Data Decoders.

### Data Decoder

Die Testapplikation bietet bereits eine erste Erkennung von Objekten bzw. Punkten, die innerhalb des Data Decoders generiert wird. Die vom Serial Reader gelieferten Datenpakete werden mit Hilfe der erzeugten Objektinformationen annotiert und in Form von Touchdata-Strukturen an die Visualisierung weitergereicht. Durch die Annotation bleiben die Originaldaten für nachfolgende Komponenten weiterhin verfügbar.

Zum Zeitpunkt der ersten Analyse beschränkt sich die Objekterkennung auf eine einfache Kombination zu Wertepaaren. Weitergehende Interpretationen ankommender Touchpakete sollen im Verlauf dieser Arbeit innerhalb des Data Decoders umgesetzt werden. Des Weiteren wäre eine Plugin-Architektur für den Data Decoder denkbar, über die weitere Interpretationen dynamisch hinzugefügt werden könnten.

### Visualisierung

Die Visualisierungskomponente stellt die vom Data Decoder gelieferten Werte grafisch dar. Die Darstellung erfolgt in Form von Rechtecken, die den gemeldeten Unterbrechungen auf horizontaler und vertikaler Achse entsprechen. Erkannte Objekte werden in Form von roten Rechtecken visualisiert.

Gemeldete Unterbrechungen sowie vom Data Decoder erkannte Objekte werden mit einer Bemäßung versehen, die Startposition sowie Breite/Ausdehnung im Koordinatensystem der Hardware oder in Millimetern darstellen. Der Abstand zwischen Unterbrechungen auf einer Achse wird ebenfalls in Millimetern angegeben. Abbildung 4.3.1 stellt exemplarisch die Ausgabe des Analysetools bei Berührung durch zwei Finger dar.

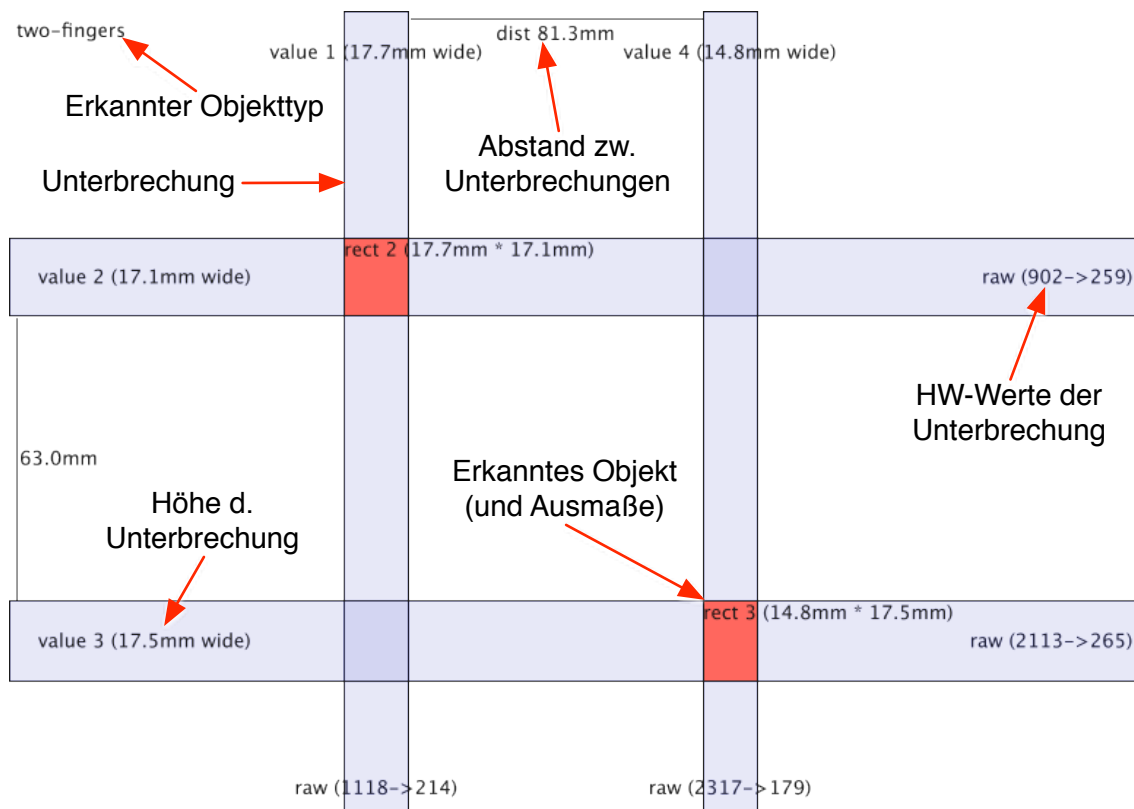


Abbildung 4.4.: Visualisierung des Analysetools

Die Visualisierung erfolgt mit Hilfe der Qt-Bibliothek und dessen Graphics View Framework. Da dieses bereits eine Skalierung zwischen unterschiedlichen Koordinatensystemen ermöglicht, wird

#### 4. Analyse der Rahmenbedingungen

---

das von der Hardware vorgegebene Koordinatensystem (0, 0 → 4095, 4095) für die Darstellung innerhalb des Frameworks übernommen.

#### Logging

Die Ausgaben stammen jeweils direkt aus einer der Komponenten. Auf eine Abspaltung in eine eigenständige Komponente wurde verzichtet, da die getätigten Ausgaben primär für die Performance-Einschätzung der Hardware sowie zur eventuellen Fehlersuche (Debugging) gedacht sind. Für die nachfolgende Analyse wurden folgende Werte während der Eingabe gemessen:

- Anzahl erhaltener Datenpakete pro Sekunde (Ausgabe jede Sekunde)
- Zeit zwischen dem Erhalt zweier Datenpakete (Ausgabe für jedes 5. Paket)

Beide Werte wurden innerhalb des Serial Readers erhoben um eine möglichst exakte Zeitmessung zu erreichen. Der Zugriff auf die serielle Schnittstelle erfolgte innerhalb eines eigenständigen Threads, wodurch die Datenannahme sowie die Zeitmessung von der grafischen Ausgabe entkoppelt wurde.

```
NEW: Value(3; 2857->67)
NEW: Value(4; 2630->223)
New TouchRect TouchRect{ 3; 3 & 4 -> QRect(2857,2630 67x223) }
delta time: 13 msec
delta time: 5 msec
~ 77.1513 packets/sec
delta time: 13 msec
delta time: 21 msec
delta time: 6 msec
delta time: 14 msec
delta time: 5 msec
delta time: 21 msec
delta time: 21 msec
delta time: 6 msec
delta time: 21 msec
delta time: 21 msec
delta time: 5 msec
delta time: 13 msec
delta time: 21 msec
delta time: 5 msec
delta time: 6 msec
delta time: 14 msec
Removing previous TouchRect TouchRect{ 3; 3 & 4 -> QRect(2810,2657 139x209) }
~ 75.9241 packets/sec
delta time: 13 msec
Removing previous TouchRect TouchRect{ 2; 1 & 2 -> QRect(1644,1653 173x152) }
```

Abbildung 4.5.: Logausgaben des Analysetools

## 4.4. Ergebnisse der Analyse

Mit Hilfe der in 4.3 auf Seite 33 beschriebenen Testapplikation wurden einige Eigenschaften der Hardware überprüft und nachfolgend dokumentiert. Entsprechend der in 4.2 auf Seite 33 genannten Ziele wurde die Hardware dabei sowohl auf ihre technische Leistungsfähigkeit als auch auf ihr Verhalten im Bezug auf die Eingabe mit mehr als einem Druckpunkt sowie mit anderen Objektarten überprüft.

### 4.4.1. Reaktionsgeschwindigkeit

Um die Reaktionsgeschwindigkeit der Hardware beurteilen zu können, wurde sowohl die durchschnittliche Anzahl der Datenpakete pro Sekunde als auch der zeitliche Abstand zwischen einzelnen Paketen betrachtet. Getestet wurde mit einem, fünf sowie zehn Fingern, die gleichzeitig den Aufsatz berührten, wobei Überlappungen der X- und Y-Achswerte vermieden wurden.

Bei Auswertung der in 4.2 zusammengefassten Ergebnisse stellt sich heraus, dass sich die beiden zur Verfügung stehenden Aufsatzgrößen signifikant in ihrer Abtastrate unterscheiden. Dies ist vermutlich durch die unterschiedliche Anzahl abzufragender IR-Dioden bei identischer Abtastgeschwindigkeit begründet.

Weiterhin nimmt die Anzahl der Pakete/s bei steigender Anzahl von Werten ab. Warum die Paketverzögerung erst bei der maximalen Anzahl von 7 Werten je Achse ansteigt und nicht linear mit der Anzahl zurückzuliefernder Werte zunimmt, konnte nicht geklärt werden. Ebenfalls unklar ist, warum dieser Effekt nur bei der 17"-Variante zu beobachten ist.

Tabelle 4.2.: Reaktionsgeschwindigkeit

Aufsatzgröße	Anz. Werte	Pakete/s	Pause zwischen Paketen (ms)
17"	2	76	10 - 20
42"	2	26	30 - 40
17"	10	76	10 - 20
42"	10	26	30 - 40
17"	14	38	20 - 30
42"	14	26	30 - 40

Wenn man die Anzahl der Pakete pro Sekunde mit der Abtastrate einer Maus vergleicht, so fällt auf, dass die Paketrate des 42"-Rahmens mit durchschnittlich 25 Paketen/s unterhalb der Rate einer PS/2- (100 Samples/s) oder USB-Maus (125 Samples/s) liegt (vgl. Chapweske [2003]). Multitouchlösungen auf Basis von FTIR sowie einer herkömmlichen Webcam besitzen üblicherweise

ebenfalls eine Abtastrate von 25 - 30 Bilder/s. Hier ist die Webcam der limitierende Faktor. Verglichen mit anderen Multitouchlösungen ist die Abtastrate daher nicht als problematisch anzusehen. Für schnelle Bewegungen wie sie mit einer Maus ausgeführt werden können, wird die gemessene Abtastrate voraussichtlich nicht ausreichen.

##### 4.4.2. Format der übermittelten Daten

Von der Hardware erkannte Unterbrechungen in vertikaler sowie horizontaler Richtung werden in zwei unabhängigen Arrays, jeweils aus Position sowie Ausdehnung der Unterbrechung bestehend, übermittelt. Innerhalb der übertragenen Daten existieren dementsprechend keine Wertepaare die den realen Druckpunkten entsprechen würden. Aufgrund dieser Tatsache werden die Werte für beide Achsen in Form von Rechtecken visualisiert, die jeweils die komplette Breite bzw. Höhe der jeweils anderen Achse einnehmen.

Es stellte sich weiterhin heraus, dass die Position der Werte innerhalb der übermittelten Datenpakete ausschließlich an ihre Position auf dem Display gebunden ist: Die Arrays für die horizontalen und vertikalen Achswerte sind immer in aufsteigender Reihenfolge sortiert. Aufgrund dieses Verhaltens ist eine Kombination von Werten zu Wertepaaren anhand der Position innerhalb der Datenpakete nicht möglich. Die Kombination von Werten zu konkreten Druckpunkten muss daher über andere Eigenschaften erfolgen.

##### 4.4.3. Genauigkeit der Daten

Werte auf einer Achse, deren Abstand nahe an oder unterhalb der Genauigkeit der Hardware liegen, neigen dazu, zu einem einzelnen Wert zusammengefasst zu werden. Da die Genauigkeit bei wenigen Millimetern liegt, sind vom Benutzer sichtbare Effekte die durch dieses Verhalten entstehen eher unwahrscheinlich, solange die Anzahl der Werte eines Datenpakets nicht als einziger Faktor für eine Unterscheidung von Zuständen oder Objekten verwendet wird.

##### 4.4.4. Mehrdeutigkeit bei mehreren Druckpunkten

Durch die fehlende Zuordnung von Werten beider Achsen, ergeben sich bei mehr als einem Druckpunkt Mehrdeutigkeiten. [Abbildung 4.6 auf der nächsten Seite](#) verdeutlicht dies anhand von zwei Druckpunkten  $P_2$  und  $P_2$ , deren Achswerte sich an insgesamt vier Stellen kreuzen. *Es ist nicht eindeutig definiert, ob die Punkte  $P_n$  oder die Punkte  $S_n$  die realen Druckpunkte auf der Touchoberfläche repräsentieren.* Weiterhin steige die Anzahl der möglichen Druckpunkte quadratisch mit der Anzahl an Unterbrechungen an.

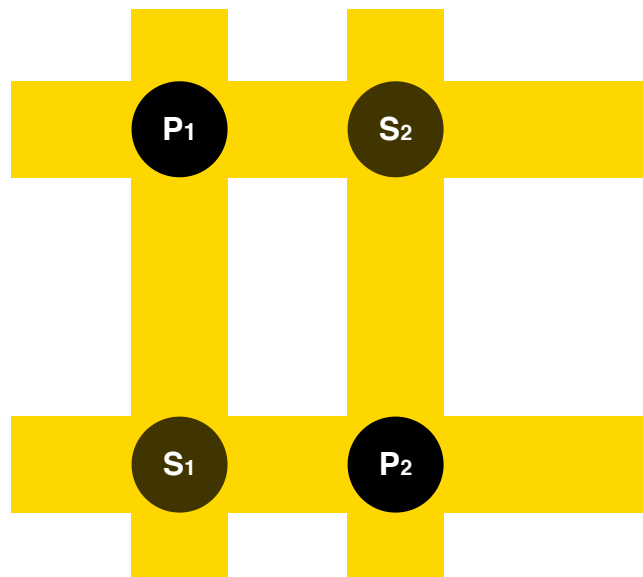


Abbildung 4.6.: Ambiguität zweier Druckpunkte ( $P_n$ : reale Druckpunkte;  $S_n$ : alternative Druckpunkte aus Softwaresicht)

Diese Eigenschaft ist nicht problematisch, solange die exakten Druckpunktdaten für die auszuführende Aktion unerheblich sind. Bei einer Objektdrehung um den Objektmittelpunkt wäre dies z.B. der Fall. Würde einer der Punkte jedoch den Drehmittelpunkt angeben, so müssten beide Punkte während der Drehung eindeutig identifizierbar bleiben. Insbesondere bei der Nutzung durch mehrere Personen ist die Kenntnis über die Position der realen Druckpunkte zwingend erforderlich, da sonst Aktionen mit Fingern unterschiedlicher Personen nicht parallel ausgeführt werden können. Eine gegenseitige Beeinflussung der Eingabedaten der Personen wäre bei fehlerhafter Erkennung der Druckpunkte nicht auszuschließen.

#### 4.4.5. Abschattung von Druckpunkten auf einer Achse

Wenn sich mehrere Druckpunkte auf gleicher Höhe auf einer Achse befinden, legt der Punkt mit der größten Ausdehnung die Gesamtausdehnung an dieser Stelle fest. Die Ausdehnung der restlichen Druckpunkte ist nicht feststellbar (siehe [4.7 auf der nächsten Seite](#), die Höhe von  $P_2$  ist in diesem Fall nicht bestimmbar, da nur die Ausdehnung von  $H_1$  bekannt ist, die der Höhe von  $P_1$  entspricht).

Dieses Verhalten ist insbesondere für sich bewegende Druckpunkte problematisch, da während der Bewegung einer der zugehörigen Achsenwerte zu verschwinden scheint, wenn er sich mit

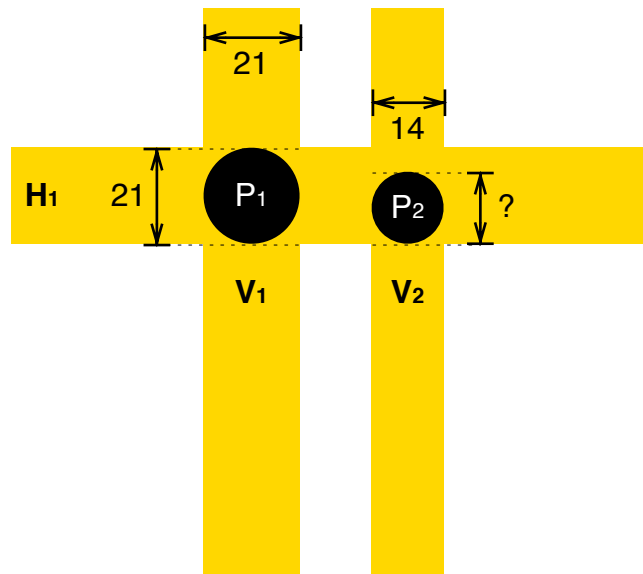


Abbildung 4.7.: Abschätzung von Druckpunkten auf der gleichen Achse

einem anderen Wert auf der gleichen Achse überschneidet. Um eine sichere Druckpunkterkennung zu ermöglichen, muss dieses Verhalten bei der Erzeugung von eindeutigen Druckpunkten berücksichtigt werden.

Eine ähnliche Situation ergibt sich bei zeitlich sehr kurzen Unterbrechungen oder falls eine Unterbrechung nicht vollständig ist (z.B. wenn einer der Finger noch nicht auf der Bildschirmoberfläche aufgesetzt wurde). In diesem Fall kann es insbesondere bei größeren Touchaufsätzen dazu kommen, dass kurzzeitig nur ein Achsenwert für den Druckpunkt übertragen wird. Der zweite Achsenwert folgt entweder in einem nachfolgenden Datenpaket oder bleibt komplett aus, falls das Objekt wieder aus dem Bereich des Infrarotgitters entfernt wird.

#### 4.4.6. Mögliche Ansätze zur Implementation von Gesten

Auf Basis der zuvor geführten Tests erfolgt an dieser Stelle eine erste Diskussion über die Umsetzbarkeit des in [2.6.2 auf Seite 22](#) genannten Gestenrepertoires. Es soll geklärt werden welche Gesten in wie weit von den Hardwareeigenschaften beeinflusst werden können und ob/welche softwaretechnischen Möglichkeiten zur Umgehung dieser Einschränkungen zur Verfügung stehen.

Da sich Eingaben mehrerer Personen bei der eingesetzten Hardware grundsätzlich durch Überlappung sowie Mehrdeutigkeit von Achswerten gegenseitig beeinflussen können, wird die Annahme getroffen, dass die Hardware nur für den Betrieb mit einem Benutzer sinnvoll einsetzbar ist.



Die nachfolgende Diskussion beschränkt sich daher auf die Betrachtung der Eingabe von maximal einem Benutzer zur Zeit.

##### **Auswahl & Verschiebung**

Um Objekte auswählen und verschieben zu können, wird die Erkennung von mindestens einem Druckpunkt benötigt. Da sich bei einem Achswert pro Achse keine Mehrdeutigkeiten ergeben, stellt eine Umsetzung auf der IRTOUCH-Hardware kein Problem dar. Bis auf das Shadow Tracking, welches eine Erkennung der Abschattung durch Gegenstände benötigt, lassen sich alle in [2.6.2 auf Seite 23](#) genannten Verfahren umsetzen.

Bei der Objektauswahl mit Hilfe einer Mausemulation muss die Position des Mausursors innerhalb des berührten Bereichs festgelegt werden. Als mögliche Position innerhalb des Druckpunktes bietet sich der Mittelpunkt an. Da eine Fingerkuppe ihren Schwerpunkt leicht oberhalb des Mittelpunkts hat, könnte die Cursorposition stattdessen leicht nach oben verschoben werden, was insbesondere bei kleineren Objekten die Auswahl für den Benutzer erleichtern könnte. Die SimPress-Eingabemethode hingegen verwendet den Mittelpunkt der Oberkante des Druckpunkts als Cursorposition, da nur diese Position durch die Änderung der Auflagefläche während eines "Klicks" stabil bleibt.

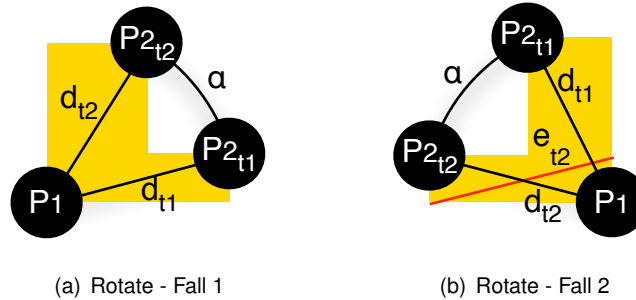
##### **Rotation & Skalierung**

Die Rotation über das RNT-Verfahren aus [Kruger u. a. \[2005\]](#) lässt sich aufgrund der Beschränkung auf einen Druckpunkt direkt umsetzen.

Für die Rotation durch zwei Finger werden eindeutige Druckpunkte benötigt, da nur auf diese Weise sowohl der Drehwinkel als auch die korrekte Drehrichtung bestimmt werden können. Eine Bestimmung der Drehrichtung allein aus dem durch zwei Druckpunkte aufgespannten Rechteck ist nicht möglich, die Drehrichtung wäre je nach Implementation in zwei der vier möglichen Quadranten umgedreht zur vom Anwender erwarteten Drehrichtung. Begründet ist dies in der Unentscheidbarkeit der verwendeten Rechtecksdiagonalen, die den Winkel der Drehung bestimmen würden. [Abbildung 4.4.6 auf der nächsten Seite](#) stellt zwei mögliche Fälle dar:

**Fall 1**  $P1$  und  $P2$  entsprechen jeweils einem Finger, wobei nur  $P2$  seine Position verändert. Der Winkel  $\alpha$ , der zu den beiden Zeitpunkten  $t1$  und  $t2$  zwischen  $d_{t1}$  und  $d_{t2}$  definiert ist, gibt sowohl die Drehrichtung als auch den Drehwinkel an. Bei Nutzung einer Rechtecksdiagonalen von der unteren linken Ecke ausgehend wären die Diagonalen identisch zu  $d_{t1}$  und  $d_{t2}$ , die Drehrichtung wäre daher ebenfalls identisch.

**Fall 2** Der Druckpunkt  $P1$  liegt in der rechten unteren Ecke des durch  $P1$  und  $P2$  aufgespannten Rechtecks. Der Winkel  $\alpha$  ist durch die Geraden  $d_{t1}$  und  $d_{t2}$  definiert. Bei gleicher Annahme über die Rechtecksdiagonalen wie im ersten Fall würde der Winkel durch  $e_{t1}$  und  $e_{t2}$  definiert sein und daher einen von  $\alpha$  abweichenden Wert besitzen, die Rechtecksdiagonalen stimmen nicht mit  $d_{t1}$  und  $d_{t2}$  überein.



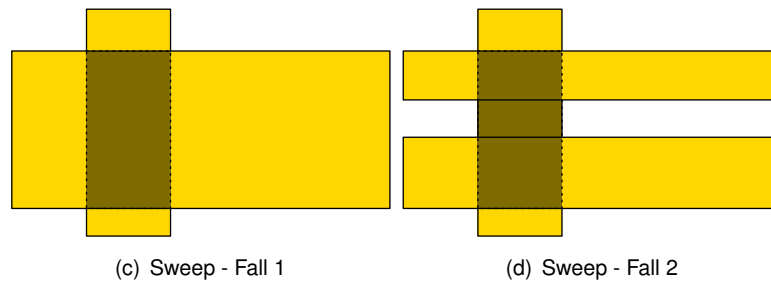
### Sweep

Das Wischen über ein Objekt lässt sich unter der Annahme, dass nur eine Geste zur Zeit erkannt werden muss umsetzen. Abbildung 4.4.6 auf der nächsten Seite) skizziert zwei mögliche Datenkonstellationen bei der eine Handkante auf der Bildschirmoberfläche aufliegt. Die Geste ist definiert durch eine schmale Unterbrechung auf der einen sowie einer breiten Unterbrechung auf der anderen Achse (Fall 1). Im Falle einer nur teilweise aufliegenden Handkante sollte zudem das umgebende Rechteck um alle Achswerte betrachtet werden und auf die Ausmaße sowie das Seitenverhältnis einer Handkante überprüft werden (Fall 2).

Die Definition einer Handkante über Breite und Höhe bzw. deren Verhältnis zueinander ist ausreichend, da sich die zu prüfenden Werte stark genug von denen einzelner Finger unterscheiden. Die durch Finger erzeugten Achswerte besitzen eine erheblich geringere Ausdehnung.

### Flick

Da sich die Flick-Geste von der Verschiebung nur durch eine hohe Beschleunigung und ein abruptes Gestenende auszeichnet, ist eine Umsetzung auf jeglicher Touchhardware möglich, die permanent Daten über die Position der einzelnen Druckpunkte liefert. Eine Implementation kann durch eine Historie der Bewegungsgeschwindigkeit erfolgen, die bei Gestenende überprüft wird. Überschreitet die Geschwindigkeit einen vorher festgelegten Wert über einen längeren Zeitraum, so wird das unter der Anfangsposition liegende Objekt beschleunigt verschoben.



### 4.5. Fazit der Analyse

Die zuvor durchgeführte Analyse der Hardware zeigt einige Hardwareeigenschaften auf, die den weiteren Designprozess beeinflussen.

**Datenformat** Das von der Hardware gelieferte Datenformat ist nicht mit Formaten anderer Multitouchtechniken vergleichbar. Es werden keine Bilddaten geliefert, eine Erkennung durch bildverarbeitende Algorithmen wie beispielsweise das gängige Blob-Tracking ist daher nicht möglich.

**Achswerte und Druckpunkte** Druckpunkte in der Form zweidimensionaler Positions- oder Rechteckkoordinaten müssen zuerst durch eine geeignete Verbindung von Achswerten beider Achsen erzeugt werden (siehe [4.4.2 auf Seite 38](#)). Es existiert grundsätzlich kein aus den Daten erkennbarer Zusammenhang zwischen horizontalen und vertikalen Achswerten.

**Tracking von Druckpunkten** Aus den vorigen beiden Punkten ergibt sich, dass zur Verfolgung einzelner Druckpunkte eine Alternative zu den ungeeigneten Blob-Tracking-Algorithmen implementiert werden muss. Weiterhin muss eine Zuordnung von horizontalen und vertikalen Achswerten zu Wertepaaren erfolgen, die die realen Druckpunkte repräsentieren. Auf Basis dieser Wertepaare kann daraufhin eine Verfolgung einzelner Druckpunkte implementiert werden.

**Genauigkeit und Fehlverhalten der Erkennung** Die Verlässlichkeit der Eingabedaten hängt zu einem großen Teil vom Benutzer ab, da simultane Druckpunkte sich gegenseitig beeinflussen, sobald sie sich auf einer der beiden Achsen überlappen. Ein Mehrbenutzerbetrieb ist aufgrund dieser Unwägbarkeit höchstwahrscheinlich nicht sinnvoll und wird daher für alle nachfolgenden Betrachtungen ausgeklammert.

## 5. Algorithmen zur Datenaufbereitung

Um die in [4.4 auf Seite 37](#) dokumentierten Eigenschaften der Hardware auszugleichen und die Voraussetzungen für eine Gestenerkennung zu schaffen, müssen die von der Hardware gelieferten Eingabedaten zuerst interpretiert werden. Im Folgenden werden einige evaluierte sowie eingesetzte Methoden zur Aufbereitung der Eingabedaten vorgestellt und, wo möglich, mit anderen Systemen verglichen.

### 5.1. Anforderungen an die Eingabedaten

Voraussetzungen für das in [2.6.2 auf Seite 22](#) vorgestellte Gestenrepertoire sind die Identifizierung von Druckpunkten und ihrer Bewegung sowie die Erkennung von Objektarten (siehe [4.4.6 auf Seite 40](#)). Die Objekterkennung wird dabei primär für die Unterscheidung zwischen einzelnen Fingern und anderen Objekten benötigt. Dies kann zum Einen zur Filterung ungültiger Eingabedaten genutzt werden, zum Anderen um Aktionen in Abhängigkeit von der jeweiligen Objektart auszuführen. Die Objekterkennung ist eine Voraussetzung für die Implementation der Sweep-Geste (vgl. [2.6.2 auf Seite 26](#)), da diese nicht mit Hilfe von Fingern durchgeführt wird.

### 5.2. Objektidentifizierung und -verfolgung

Um die Bewegung mehrerer Objekte nachvollziehen zu können, muss das System diese über den gesamten Berührungszeitraum eindeutig identifizieren können. Eindeutigkeit bei ansonsten gleichartigen Objekten lässt sich über die Vergabe eines abstrakten Identifiers für jedes erkannte Objekt erreichen.

Die wiederholte Zuordnung eines abstrakten Identifiers zu einem Objekt über mehrere Datenpakete hinweg wird nachfolgend als "Object-Tracking" bezeichnet. In anderen Multitouch-Systemen wird dieser Vorgang meist als "Point-Tracking" bezeichnet, wobei jedoch die Ausmaße der Berührung ignoriert oder nicht bis zur Applikationsebene weitergereicht werden. Insbesondere kamera-basierte Touchsystem (vgl. [2.5.3 auf Seite 18](#)) setzen Point-Tracking-Verfahren ein. Verschiedene

Ansätze hierzu sind unter Anderem in Wang u. a. [2008] sowie im Touchlib-Projekt<sup>1</sup> der NUI-Group (siehe NUI [2006]) dokumentiert.

Der Ablauf von Object-Tracking-Verfahren lässt sich in drei Abschnitte aufteilen:

1. Zuordnung von Identifiern alter Objekte zu identischen oder ähnlichen neuen Objekten
2. Vergabe neuer Identifier an verbliebene neue Objekte
3. Verwerfen bisheriger Identifier die nicht an neue Objekte vergeben wurden

Im Folgenden werden einige evaluierte Ansätze beschrieben, wie das Object-Tracking auf infrarotbasierter Hardware umgesetzt werden kann, und was im Vergleich zu anderen Systemen (siehe 2.5 auf Seite 16) gesondert zu beachten ist.

### 5.3. Object-Tracking durch Kombination von Achswerten

Die Kombination von Achswerten zu Wertepaaren stellt den einfachsten Ansatz für ein Object-Tracking dar. Dabei werden horizontale und vertikale Werte jeweils paarweise miteinander verknüpft. Die Überschneidung der verknüpften Achswerte definiert die Objektposition sowie die Ausmaße des Objekts. Erkannte Objekte werden aufgrund des von der Hardware gelieferten Datenformats immer als Rechtecke repräsentiert (siehe 5.1).

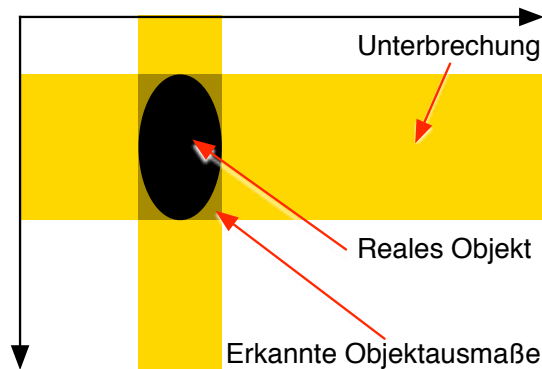


Abbildung 5.1.: Zusammenhang zwischen Achswerten, Objekten (oval) und erkannten Objektausmaßen (rechteckig)

Um das Objekt im nachfolgenden Datenpaket erneut zu erkennen wurden drei Ansätze evaluiert. Ziel war es, die Erkennung sich bewegender Finger sowohl unter hohen Bewegungsgeschwin-

<sup>1</sup><http://code.google.com/p/touchlib>

digkeiten als auch von nahe beieinander liegenden Fingern zu gewährleisten. Folgende Ansätze wurden evaluiert:

1. Umgebungssuche, bei der nahe der letzten Position im nachfolgenden Paket eine neue Überschneidung gesucht wird, die daraufhin den bisherigen Identifier erhält.
2. Suche unter Einbeziehung mehrerer voriger Positionen, durch welche die Bewegungsrichtung des Objekts abgeschätzt und der Suchbereich eingegrenzt werden kann.
3. Ansatz wie zuvor, jedoch unter der Annahme, dass die Änderung der Bewegungsgeschwindigkeit nicht linear, sondern beispielsweise logarithmisch verläuft.

### 5.3.1. Evaluation der Ansätze für ein Object-Tracking

Bei Tests der drei Ansätze innerhalb der Testworkbench stellte sich heraus, dass die Suche unter Einbeziehung der Bewegungsrichtung für die vorhandene Hardware keinen offensichtlichen Vorteil gegenüber der reinen Umgebungssuche zu bieten scheint. Der Algorithmus konnte bei einer schnellen Umkehr der Bewegungsrichtung eines Fingers diesen nicht zuverlässig verfolgen, da sich das Objekt durch die Einengung des Suchbereichs nicht erneut finden ließ. Eine Ausweitung des Suchbereichs führte dagegen schnell zu falschen Zuordnungen bei nahe beieinander liegenden Werten. Dieses Verhalten könnte unter Anderem in der niedrigen Framerate der Hardware begründet sein.

Ein Tracking unter Zuhilfenahme voriger Werte und unter der Annahme einer nicht-linearen Bewegung (Ansatz 3) führte ebenfalls zu keiner relevanten Verbesserung des Object-Trackings. Die Umgebungssuche wurde auf Basis der zu erwartenden Position innerhalb des nächsten Pakets durchgeführt. Die zu erwartenden Startwerte ( $Pos$ ) sowie Breiten ( $Ext$ ) wurden dabei durch zwei verschiedene Ansätze berechnet:

- $Pos_n = (Pos_{n-2} - Pos_{n-1})/2$   
 $Ext_n = (Ext_{n-1} + Ext_{n-1})/2$
- $Pos_n = ((Pos_{n-1} * 9) + Pos_{n-2})/10$   
 $Ext_n = ((Ext_{n-1} * 9) + Ext_{n-2})/10$

Ein vergleichbarer Ansatz auf Basis einer kubischen Interpolation ist im "Irtouch Driver" des Sparsh-Ui Projekts implementiert (siehe [Spa](#)). Dieser wurde im Rahmen dieser Arbeit jedoch nicht weiter evaluiert.

**Fazit der Evaluation** Mit Hilfe des Object-Tracking durch Wertekombination lassen sich Fingerbewegungen auf der IRTOUCH-Hardware grundsätzlich nachvollziehen. Solange sich die Finger nicht auf gleicher Höhe auf einer der beiden Achsen überschneiden ist die Erkennung ausreichend. Keiner der drei Ansätze kommt jedoch mit Überlappungen der Achswerte zurecht (siehe auch [4.4.5 auf Seite 39](#)).

Ein Achswert der durch Überlappung entsteht, könnte zur Bildung beliebig vieler Objektpositionen herangezogen werden, jedoch muss dabei für alle Objekte die gleiche Position und Ausdehnung auf der betreffenden Achse angenommen werden. Die Bewegung von Druckpunkten ist für die Achse auf der die Überlappung stattfindet während der Überlappung nicht nachvollziehbar. Bei Aufteilung einer Überlappung in zwei eigenständige Achswerte ist zudem nicht feststellbar, welcher der nun getrennt vorliegenden Achswerte welchem zuvor identifizierten Objekt zuzuordnen ist. Abbildung 5.2 verdeutlicht die drei problematischen Zeitpunkte während einer Überlappung.

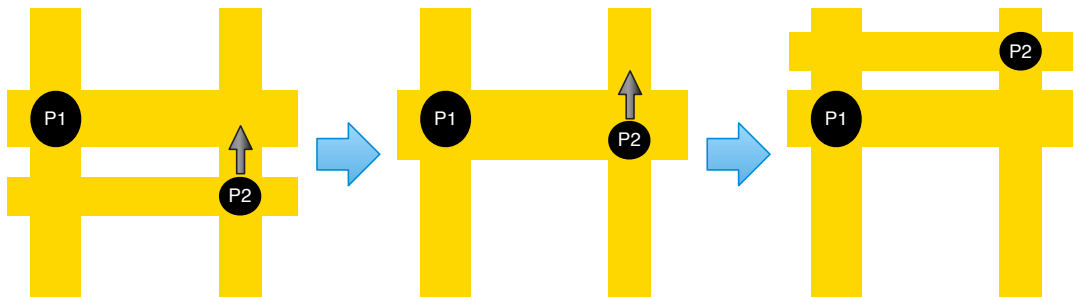


Abbildung 5.2.: Beispiel - Achswerte vor, während und nach der Überlappung

### 5.4. Tracking von Achswerten mit nachfolgendem Object-Tracking

Um das zuvor beschriebene Problem der Überlappung von Achswerten zu lösen, bietet es sich an, im ersten Verarbeitungsschritt der Eingabedaten zuerst nur die Achswerte einem Tracking zu unterziehen. Um sich überlappende Unterbrechungen sicher verfolgen zu können, müssen diese während der Überlappung weiterhin als eigenständige Werte behandelt werden. Diese Werte werden nachfolgend als virtuelle Achswerte bezeichnet, die von der Hardware gelieferten Werte werden als "reale Achswerte" bezeichnet.

### 5.4.1. Ablauf

Das Verfahren lässt sich in folgende Schritte unterteilen, die im Anschluss einzeln betrachtet werden:

1. Zuordnung alter Achswerte
2. Zuordnung aufgeteilter Achswerte
3. Zuordnung zusammenfallender Achswerte
4. Tracking virtueller Werte
5. Vergabe neuer Identifier für neue Achswerte
6. Kombination von Achswerten anhand ihrer Identifier (Object-Tracking)

Der Ablauf ist vergleichbar mit dem des reinen Object-Trackings. Zusätzlich werden in den Schritten zwei bis vier zusammenfallende und sich aufteilende reale Achswerte betrachtet, die während einer Achswertüberlappung auftreten, und in virtuelle Achswerte überführt. Jeder von der Hardware gelieferte Achswert wird innerhalb des Verfahrens exakt ein mal verwendet und steht daraufhin für die nachfolgenden Verarbeitungsschritte nicht mehr zur Verfügung.

### 5.4.2. Zuordnung alter Achswerte

Die im vorigen Datenpaket identifizierten Achswerte werden durch eine Umgebungssuche in den aktuellen Daten gesucht (siehe [5.3 auf Seite 50](#)). Die Umgebungssuche basiert auf einem Delta-wert um den der aktuelle Achswert vom vorigen Achswert maximal abweichen darf. Die Nutzung von Deltawerten ergibt sich aus zwei Annahmen:

1. Unterbrechungen bewegen sich mit einer nach oben begrenzten Geschwindigkeit
2. Unterbrechungen ändern ihre Breite nur geringfügig

Die erste Annahme ist für die fingerbasierte Eingabe nicht ideal, da Fingerbewegungen mit einer sehr hohen Geschwindigkeit durchgeführt werden können und die Geschwindigkeit je nach Anwender variieren kann. Da das Einbeziehen der aktuellen Bewegungsgeschwindigkeit jedoch bereits beim Object-Tracking zu groben Fehleinschätzungen führte (siehe [5.3 auf Seite 45](#)), scheint dies dennoch die einzig verlässliche Zuordnungsmethode darzustellen. Die zweite Annahme ist für Finger bei hinreichend großem Delta-Wert immer zutreffend, da sich die Auflagefläche von Fingern im Vergleich zu ihrer Position nur geringfügig verändert.



### Wahl des Deltawerts

Um den idealen Deltawert für die Umgebungssuche zu bestimmen, wurden zwei Ansätze kombiniert: Eine empirische Suche ("Trial-and-Error") sowie eine Validierung der gefundenen Werte basierend auf den Hardwareeigenschaften.

Aus der empirischen Suche ergab sich für den 17"-Aufsatz ein Delta-Wert von ca. 400, der in etwa 33 mm entspricht ( $380 * (337.6 \text{ mm} / 4096) \approx 31.32 \text{ mm}$ ). Bei einer Datenrate von 75 Paketen in der Sekunde kann ein Finger daher theoretisch eine Strecke von ca. 234 cm zurücklegen, ohne dass der Algorithmus dabei das Objekt "verliert". Da sich die Skalierung der Touchdaten für die horizontale und vertikale Achse unterscheiden, müssen die Deltawerte für jede Achse einzeln festgelegt werden.

Da der 42"-Rahmen nur 35 Pakete pro Sekunde liefert (siehe [4.4.1 auf Seite 37](#)), muss bei einer vergleichbaren Erkennungsqualität der Deltawert größere "Sprünge" innerhalb der Achswerte von zwei Datenpaketen tolerieren. Durch eine Vergrößerung des Deltawerts erhöht sich jedoch auch die Wahrscheinlichkeit für eine falsche Zuordnung der Achswerte. Der Deltawert wurde daher für den größeren Rahmen auf ca. 210 festgelegt, was horizontal einer Strecke von ca. 50 mm entspricht ( $213 * (960 \text{ mm} / 4096) \approx 49.92 \text{ mm}$ ) und eine Bewegung von ca. 175 cm innerhalb einer Sekunde ermöglicht.

### Algorithmus

Um einen neuen Achswert einem vorigen Achswert eindeutig zuordnen zu können, muss sich der Mittelpunkt des neuen Achswerts innerhalb des Intervalls  $(V_{center} - \delta, V_{center} + \delta)$  befinden, wobei  $V_{center}$  den Mittelpunkt des alten Achswerts bezeichnet. Des Weiteren versucht der Algorithmus den Abstand der Mittelpunkte zu minimieren, indem eine Zuordnung angestrebt wird, bei der die Mittelpunkte des alten und neuen Achswerts möglichst wenig voneinander abweichen. Durch die Minimierung wird sichergestellt, dass nahe beieinander liegende Achswerte zuverlässig erkannt werden.

Wurde ein Achswert erneut identifiziert, so wird er in den aktuellen Daten als bearbeitet gekennzeichnet um in den nachfolgenden Schritten nicht erneut berücksichtigt zu werden.

#### 5.4.3. Zuordnung aufgeteilter Achswerte

Achswerte, die sich zuvor überlappt haben und im aktuellen Datenpaket als eigenständige Achswerte auftreten, müssen anhand ihrer Eigenschaften wie Breite und Startposition zugeordnet werden.

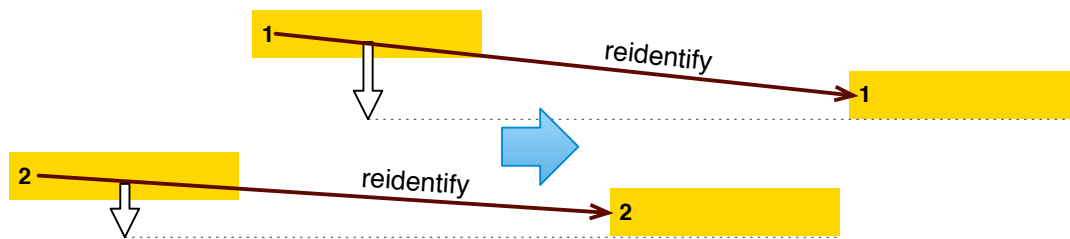


Abbildung 5.3.: Zuordnung alter Achswerte zu aktuellen Achswerten

Hierfür wird eine Umgebungssuche durchgeführt, bei der Start- und Endposition eines vorigen Achswertes, der durch eine Überlappung entstanden ist, ähnlich den Start- und Endpositionen zweier neuer Achswerte entspricht. Das Ähnlichkeitskriterium ist hier ebenfalls ein Deltawert, der jedoch erheblich kleiner als der in 5.4.2 auf Seite 48 verwendete Wert ist. Der Deltawert wurde kleiner gewählt, da es sonst zu einer Verwechslung mit Achswerten kommen kann, die sich nahe eines überlappten Achswerts befinden. Abbildung 5.4 skizziert den Übergang zweier virtueller Achswerte innerhalb einer Überlappung in zwei eigenständige Achswerte.

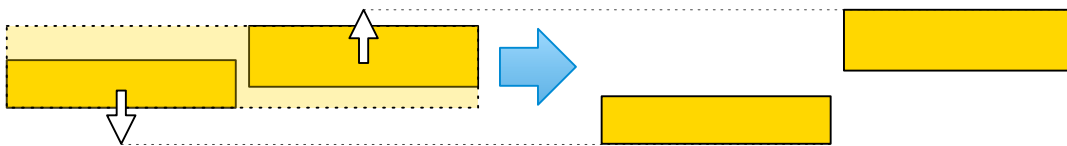


Abbildung 5.4.: Übergang von überlappten zu getrennten Achswerten

#### 5.4.4. Zuordnung zusammenfallender Achswerte

Analog zum vorigen Schritt muss eine Erkennung von Achswerten erfolgen, die im letzten Datenpaket getrennt vorlagen und im aktuellen Datenpaket aufgrund einer Achswertüberlappung als einzelner Achswert sichtbar sind. Abbildung 5.5 auf der nächsten Seite verdeutlicht diesen Fall.

Es erfolgt eine Umgebungssuche, bei der zwei nah beieinander liegende vorige Achswerte eine vergleichbare Startposition und Ausdehnung besitzen wie einer der aktuellen Achswerte. Der für die Zuordnung aufgeteilter Achswerte verwendete Deltawert wird hier erneut als Ähnlichkeitskriterium verwendet.

Ein gefundener Achswert auf den die zuvor genannten Eigenschaften zutreffen, wird in zwei virtuelle Achswerte überführt, wobei der Abstand der erzeugten virtuellen Achswerte Null beträgt. Das Größenverhältnis der beiden vorigen Achswerte wird für die neu erzeugten virtuellen Achs-

werte übernommen. Die so entstandenen virtuellen Achswerte repräsentieren während der Überlappung den realen Achswert auf dessen Basis sie erzeugt wurden.

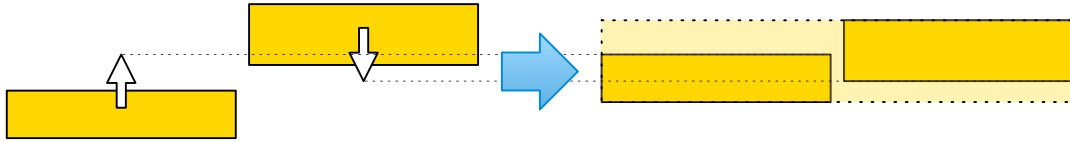


Abbildung 5.5.: Übergang von getrennten zu überlappenden Achswerten

### 5.4.5. Tracking virtueller Werte

Das Tracking virtueller Werte dient der Unterscheidung vormals getrennt vorliegender Achswerte während einer Überlappung. Die zu diesem Zweck im vorigen Schritt erzeugten virtuellen Achswerte werden in nachfolgenden Paketen als zusammenhängender Achswert gesucht und an dessen Startposition und Ausdehnung angepasst.

Es erfolgt eine gleichmäßige Anpassung der Anfangspositionen aller virtuellen Achswerte innerhalb der Grenzen des zugehörigen realen Achswerts. Ziel ist es, dass die virtuellen Achswerte den kompletten Bereich des realen Achswerts ausfüllen und dabei niemals darüber hinausragen, da sie den realen Achswert in den aufbereiteten Daten ersetzen. Weiterhin wird die Ausdehnung der einzelnen virtuellen Achswerte an Änderungen der Ausdehnung des realen Achswerts angepasst. Die Änderungen von Position und Ausdehnung des realen Achswerts werden zu gleichen Teilen auf die zugehörigen virtuellen Achswerte aufgeteilt.

Um während einer Überlappung von Achswerten feststellen zu können, ob sich die Objekte aneinander vorbei bewegt haben, wird die Annahme getroffen, dass für diesen Vorgang der reale Achswert kurzzeitig die Ausdehnung einer der virtuellen Achswerte einnimmt. Konkret handelt es sich dabei um die Ausdehnung des größten virtuellen Achswerts an dieser Stelle, da das zugehörige Objekt alle kleineren Objekte abschattet. Bei Eintreten dieses Falls werden die Positionen der Achswerte vertauscht um die erkannte Bewegung in den virtuellen Daten wiederzugeben.

Um durch Schwankungen der Ausdehnung der Überlappung nicht einen wiederholten Reihenfolgewechsel innerhalb der virtuellen Achswerte auszulösen, wird zusätzlich eine Low- und High-Watermark eingeführt. Der zuvor genannte Positionstausch wird erst durchgeführt, sobald der reale Achswert seine Minimalausdehnung (Low-Watermark) bereits wieder um 20% übersteigt (High-Watermark).

#### 5.4.6. Identifikation neuer Achswerte

Für alle bisher ungenutzten realen Achswerte des aktuellen Datenpakets wird angenommen, dass sie seit der Verarbeitung des vorigen Datenpakets neu hinzugekommen sind. Ihnen wird daher ein neuer Identifier zugeordnet.

#### 5.4.7. Object-Tracking

Im Anschluss an die Aufbereitung der Achswerte müssen diese analog zum reinen Object-Tracking (vgl. 5.3 auf Seite 45) zu Achswertpaaren verknüpft werden, um für eine Gestenerkennung eindeutige Objekte bereitzustellen. Das zuvor durchgeführte Achswert-Tracking erlaubt eine Vereinfachung des Object-Trackings, da Objekte nun lediglich durch zwei Identifier der jeweiligen Achswerte definiert sind. Die Suche der zugehörigen Achswerte zu einem Objekt anhand ihrer Position entfällt daher.

Desweiteren wird eine Zuordnung eines einzelnen Achswerts zu mehreren Objekten nicht mehr benötigt, da Achswerte die durch Überlappungen entstanden sind nun in Form virtueller Achswerte vorliegen. Jeder dieser virtuellen Werte kann daher exakt einem erkannten Objekt zugeordnet werden.

Einige der Probleme des Object-Trackings bleiben auch nach Einführung eindeutiger Achswerte bestehen. So ist weiterhin eine eindeutige Zuordnung bei mehr als zwei neu hinzukommenden Achswerten nicht möglich. Da die Zuordnung von Achswerten zu Objekten ausschließlich über Änderungen der Anzahl der Achswerte erfolgen kann, lässt sich diese Einschränkung des Infrarotgittersystems nicht beseitigen.

#### 5.4.8. Test des Achswert-Trackings innerhalb der Workbench

Um die zuvor vorgestellten Algorithmen für das Achswert- sowie Object-Tracking zu testen, wurde innerhalb der Testworkbench die Anzeige der virtuellen Achswerte zusätzlich zu den realen Achswerten eingeführt. Anhand der Anzeige konnte so nachvollzogen werden, dass die Summe der virtuellen Achswerte in ihrer Position sowie Ausdehnung identisch zu den ersetzten realen Achswerten waren.

Getestet wurde insbesondere, in wie weit sich das Verhalten des Object-Tracking durch die gesonderte Behandlung von sich überlappenden Achswerten verändert und ob durch ein zuverlässigeres Achswert-Tracking auch die Zuverlässigkeit des daran anschließenden Object-Trackings verbessert werden kann. Die Zuverlässigkeit beider Trackingverfahren zeigt sich anhand gleichbleibender Identifier für sich bewegende Achswerte sowie erkannte Objekte. Abbildung 5.6 auf Seite 54 stellt exemplarisch vier Zeitpunkte eines Testlaufs dar, bei dem sich zwei Finger auf dem

Display befunden haben, wobei der rechte Finger während des Tests am linken Finger vorbei bewegt wurde (vgl. [4.3.1 auf Seite 35](#) für eine Legende aller dargestellten Elemente).

### Ergebnis des informellen Tests

Ein informeller Test mit Hilfe der Workbench zeigte, dass die virtuellen Achswerte bei etwa 75% aller Überlappungsszenarien mit den realen Fingerpositionen übereinstimmten. Der Identifier, welcher einer Fingerposition vor Eintritt in eine Achswertüberlappung zugewiesen wurde, blieb sowohl während als auch nach Ende der Überlappung bestehen. Die verbliebenen Fälle, in denen das Achswert-Tracking nicht ordnungsgemäß funktionierte, ließen sich auf drei Problemfälle eingrenzen:

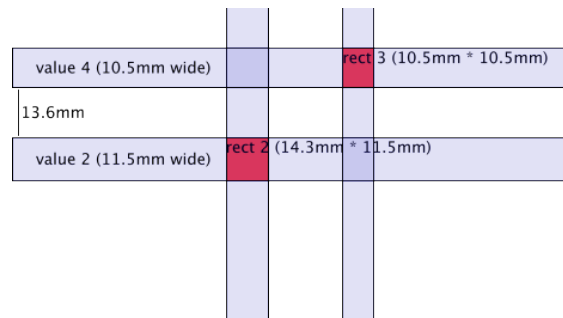
- Die Bewegungsgeschwindigkeit des Fingers lag außerhalb der durch die Deltawerte definierten Toleranzgrenzen. Der Achswert befand sich bei Eintritt oder Verlassen der Überlappung außerhalb des Suchradius.
- Einer der realen Achswerte änderte während der Überlappung seine Ausdehnung ungewöhnlich stark, wodurch er beim Verlassen der Überlappung nicht wiedererkannt wurde. Das Verhalten begründet sich durch die Annahme, dass die Ausdehnung eines Achswerts während einer Überlappung weitgehend identisch bleibt.
- Die realen Achswerte änderten nahe des Ein- oder Austritts in eine Überlappung ihre Breite übermäßig stark.
- Der Wechsel zwischen überlappten und getrennten Achswerten fand aufgrund schwankender Achswertpositionen in aufeinanderfolgenden Datenpaketen wiederholt statt.

**Gelöste Probleme** Eine Überlappung von Achswerten führt durch Einführung des Achswert-Trackings nicht mehr zwangsläufig zum Verlust eines erkannten Druckpunkts. Durch den Algorithmus wird die Erkennung der Bewegung zweier Finger aneinander vorbei ermöglicht. Die maximale Bewegungsgeschwindigkeit ist maßgeblich durch die Wahl der Deltawerte sowie durch die Erkennungsgeschwindigkeit der Hardware begrenzt.

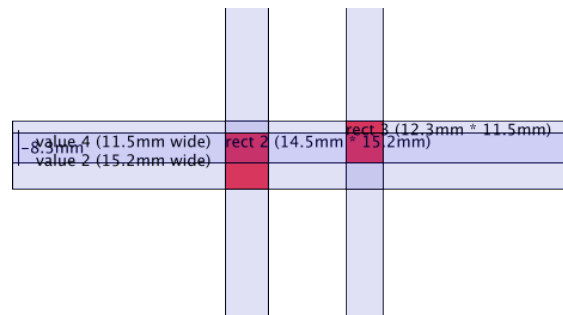
**Offene Punkte** Die Mehrdeutigkeit bei gleichzeitig auftretenden Druckpunkten (siehe [4.4.4 auf Seite 38](#)) bleibt auch nach Anwendung zuvor genannter Algorithmen bestehen. Theoretisch ließe sich das Problem durch eine Kamera oder weitere Sensoren lösen, die während einer Überlappung von Achswerten die fehlenden Informationen über einzelne Objekte auf der Oberfläche liefern. Die Kombination unterschiedlicher Sensorarten (Kamera sowie Infrarotsensoren) zieht jedoch neue

## 5. Algorithmen zur Datenaufbereitung

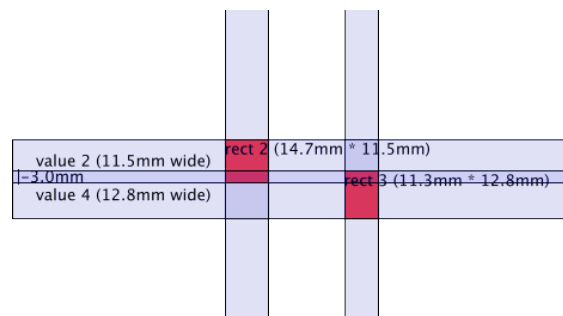
---



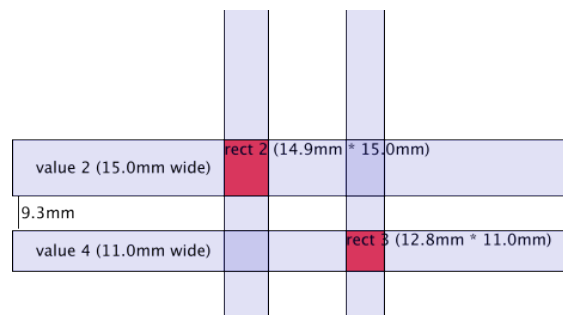
(a) Vor der Überlappung



(b) Rechter Finger oberhalb des linken Fingers



(c) Rechter Finger unterhalb des linken Fingers



(d) Nach der Überlappung

Abbildung 5.6.: Vier Zeitpunkte einer Achswertüberlappung

Probleme nach sich. Die Datenformate der Sensoren unterscheiden sich stark voneinander, die Informationen der Sensoren müssten daher zuerst in eine vereinheitlichte Form überführt werden. Da dieser Ansatz eine Erweiterung der Hardware darstellt, wird er im Rahmen dieser Arbeit nicht weiter verfolgt.

Das vorgestellte Verfahren des Achswert-Trackings ließe sich in Hinsicht auf die Erkennung von Anfang und Ende einer Überlappung eventuell in seiner Zuordnungsgenauigkeit zwischen zwei Paketen weiter verbessern. So könnte eine Suche zusammenfallender oder erneut getrennt auftretender Achswerte analog zu dem in [5.4.2 auf Seite 48](#) verwendeten Minimumkriterium erfolgen. Hierfür müsste zuerst eine Liste aller möglichen Konfigurationen erstellt werden, die jeweils aus den Zuordnungen von alten zu neuen Achswerten bestehen. Daraufhin müsste die ideale Konfiguration gefunden werden, beispielsweise durch Minimierung der Summe aller Positionsänderungen oder durch eine Gewichtung jeder Achswertveränderung anhand einer Gauss-Verteilung. Da die Bewegung von Objekten auf der Touchoberfläche selten linear erfolgt ist bei diesem Ansatz jedoch nicht sichergestellt, dass dadurch das Erkennungsergebnis signifikant verbessert wird. Auch dieser Ansatz würde eine Annahme über die Bewegungsgeschwindigkeit der Objekte treffen. Wie sich bereits in [5.3.1 auf Seite 46](#) gezeigt hat, ist diese Annahme vom Benutzer leicht außer Kraft zu setzen, der Ansatz erscheint daher wenig Erfolg versprechend.

## 6. Praktischer Einsatz der Algorithmen

Um die in Abschnitt 5 auf Seite 44 vorgestellten Ansätze zu testen, erfolgt eine Implementation im Rahmen eines realen Anwendungsszenarios (vgl. Abschnitt 3 auf Seite 28). Als Testanwendung bietet sich die digitale Nachbildung eines Fotoalbums an, da sich in diesem Anwendungskontext ein Großteil des in 2.6.2 auf Seite 22 vorgestellten Gestenrepertoires sinnvoll nutzen lässt. Die Testanwendung wird anschließend innerhalb des Usability-Tests in Hinblick auf die Intuitivität und Qualität der Eingabeerkennung untersucht.

### 6.1. Testanwendung “Touchbook”

Die Testanwendung bildet ein reales Fotoalbum nach, in dem Fotos auf den Seiten beliebig angeordnet und mit Hilfe von Fotoecken befestigt werden können. Im Gegensatz zu anderen digitalen Fotoalben wird dabei auch das explizite Befestigen von Fotos auf einer Seite unterstützt. Im Vergleich zu einem normalen Fotoalbum ist bei der softwaregestützten Version die Vergrößerung und Verkleinerung von Fotos sowie ein Hinzufügen weiterer Seiten möglich. Des Weiteren kann ein Foto beliebig oft verwendet werden und daher auf mehreren Seiten vorkommen.

Die im Vergleich zu anderen digitalen Fotoalben starke Orientierung an einem realen Fotobuch wurde gewählt um eventuell bestehende mentale Modelle (siehe 2.3 auf Seite 13) des Anwenders anzusprechen. Im Kontext der Testanwendung soll die Ausnutzung vorhandener mentaler Modelle über die Verwendung von Büchern im Allgemeinen und Fotobüchern im Speziellen die Benutzung der Anwendung erleichtern.

#### 6.1.1. Funktionsumfang

Das Fotobuch unterstützt die folgenden Funktionen:

- Scrollbare Fotoliste
- Ablage von Fotos auf Fotobuchseiten per Drag-and-Drop
- Rotation und Skalierung von Fotos



## 6. Praktischer Einsatz der Algorithmen

- Fixieren von Fotos auf Seiten
- Hinzufügen von Seiten
- Blättern zwischen den Seiten
- Entfernen einzelner nicht fixierter Fotos
- Entfernen aller Fotos auf einer Seite

Alle Funktionen sind sowohl per Mauseingabe als auch über entsprechende Touchgesten ausführbar um einen direkten Vergleich der beiden Eingabemethoden zu ermöglichen. Einen Überblick über den grafischen Aufbau der Testanwendung sowie über den Hauptteil der zur Verfügung stehenden Interaktionsmöglichkeiten bietet Abb. 6.1.

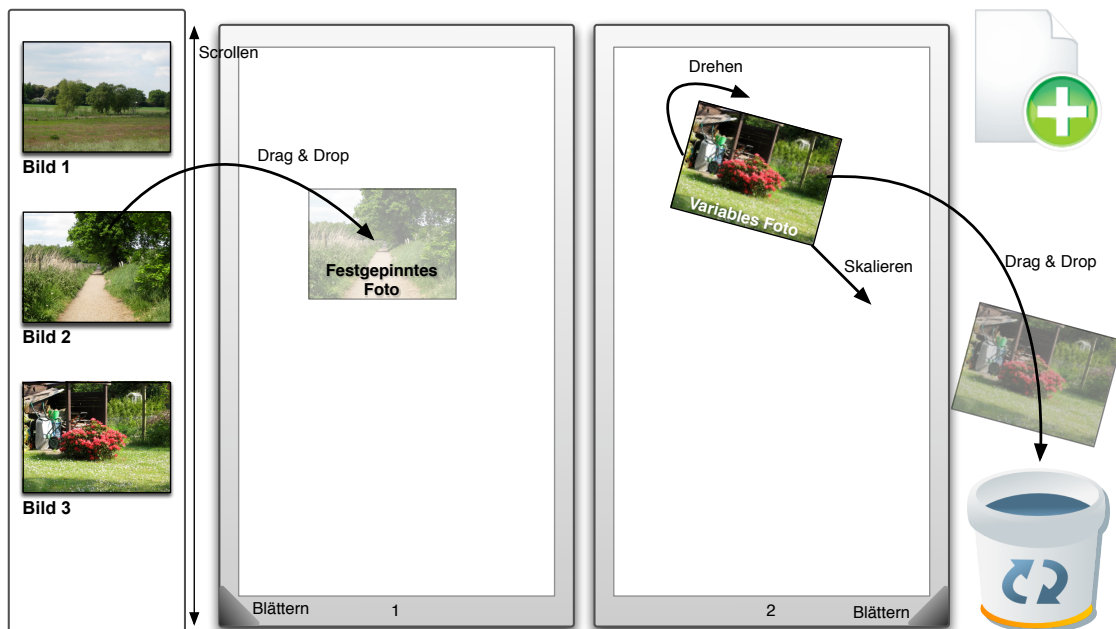


Abbildung 6.1.: GUI-Prototyp der Testanwendung "Fotobuch"

### 6.1.2. Interaktionsmöglichkeiten

Nachfolgend wird ein Überblick über die in der Testanwendung umgesetzten Interaktionsmöglichkeiten gegeben. Die Unterschiede zwischen Maus- und Toucheingabe innerhalb der Testanwendung werden beschrieben, wobei das Hauptaugenmerk auf der Toucheingabe liegt. Des Weiteren

wird der Zusammenhang zwischen dem in [2.6.2 auf Seite 22](#) vorgestellten Gestenrepertoire und konkreten Aktionen innerhalb der Testanwendung geknüpft.

### **Auswahl und Bewegung von Fotos**

Die Auswahl bzw. Selektion entspricht der Mausbedienung mit gedrückter linker Maustaste. Die Auswahl eines Fotos erfolgt per Toucheingabe sowie das System einen einzelnen Finger registriert (initiale Berührung). Das Verhalten entspricht in etwa der in [2.6.2 auf Seite 23](#) vorgestellten First-Contact-Methode. Alternativ hätte sich die Implementation der SimPress-Methode angeboten. Da viele der im praktischen Einsatz befindliche Singletouch-Systeme jedoch First-Contact implementieren, wird angenommen, dass dies bei ausreichend großen Objekten eine hinreichende Auswahlmöglichkeit der Objekte darstellt. Zudem wird in der Testanwendung die durch SimPress eingeführte Unterscheidung zwischen dem "hover" und "press"-Status nicht benötigt.

Die Verschiebung eines Fotos erfolgt automatisch nach der initialen Berührung mit einem einzelnen Finger. Die Verschiebung stoppt, so wie mehr als ein Finger oder kein Objekt vom System erkannt wird.

### **Weitere Elemente mit Singletouch-Interaktion**

Einige Oberflächenelemente besitzen in Hinblick auf die Touchbedienung ein, im Vergleich zu einem ausschließlich mausbasierten System verändertes Verhalten:

Die Fotoliste unterscheidet nach initialer Berührung die Bewegungsrichtung. Bei einer primär horizontalen Bewegung wird das zuerst berührte Foto aus der Liste dupliziert damit der Anwender es anschließend auf eine der Buchseiten ablegen kann. Bei einer vorwiegend vertikalen Bewegung wird dagegen eine Verschiebung des Listeninhalts (Scrolling) durchgeführt. Die getroffene Unterscheidung bleibt bis zum Entfernen des Fingers bestehen.

Das Umblättern der Albumseiten erfolgt über die Seitenecken. Diese müssen, vergleichbar mit dem Umblättern einer echten Buchseite, zur Buchmitte gezogen werden, woraufhin das eigentliche Umblättern stattfindet <sup>1</sup>. Das Verhalten soll zum Einen das virtuelle Fotoalbum realistischer erscheinen lassen, zum Anderen ist dadurch gewährleistet, dass der Anwender nicht unbeabsichtigt zwischen den Seiten wechselt. Ein unbeabsichtigter Wechsel ist bei der Toucheingabe eher zu erwarten als bei einer Mauseingabe, da bereits eine leichte Berührung der Bildschirmoberfläche einen normalen Button aktivieren würde.

---

<sup>1</sup>Aufgrund des technischen Aufwands wurde auf eine zugehörige Animation, die das realistische Umblättern einer Seite darstellt verzichtet

### **Rotation und Skalierung von Fotos**

Fotos die nicht auf einer Seite festgepinnt sind, können mit Hilfe von zwei Fingern gleichzeitig skaliert sowie gedreht werden. Mindestens einer der Finger muss sich für die Rotation und Skalierung auf dem zu drehenden Foto befinden, der zweite Finger kann dagegen auch außerhalb des Fotos sein. Das Verhalten basiert auf der Annahme, dass die meisten Anwender nur einen der beiden Finger während der Rotation bewegen und während einer Skalierung immer mindestens einer der Finger das zu skalierende Objekt berührt. Alternativ lassen sich Fotos per gedrückter linker Maustaste über die rechte untere Ecke sowohl drehen als auch skalieren.

### **Fixieren von Fotos**

Fotos lassen sich auf einer Albumseite fixieren. Ein Foto wird fixiert bzw. die Fixierung wird aufgehoben, sowie zwei Finger sich auf dem Foto befinden und sich für einen kurzen Zeitraum kaum oder gar nicht bewegen. Die Finger müssen sich dafür, im Gegensatz zur zuvor beschriebenen Rotation und Skalierung, beide auf dem zu fixierenden Foto befinden. Das Verhalten dient der Nachahmung von den bei realen Fotoalben gebräuchlichen Klebeecken und ermöglicht weiterhin eine Unterscheidung zwischen einem lose herumliegenden Foto und einem bereits für eine bestimmte Seite vorgesehenen und daher fixierten Foto. Eine Fixierung per Maus erfolgt bei einem Doppelklick mit der linken Maustaste auf das zu fixierende Foto.

### **Leeren einer Seite**

Das seitliche Wischen mit einer Handkante über eine Seite des Fotoalbums führt zur Entfernung aller auf der Seite festgepinnten Fotos (vgl. [2.6.2 auf Seite 26](#)). Die Aktion wird ausschließlich bei Nutzung einer Handkante ausgeführt. Es muss dabei über mindestens 60% der Seitenfläche gewischt werden, um die Löschung auszuführen. Das Verhalten ist dem realen verschieben mehrerer herumliegender Fotos auf einer glatten Fläche nachempfunden, ermöglicht im Gegensatz zu einem realen Fotoalbum jedoch das Entfernen bereits fixierter Fotos. Damit der Anwender eine Fotoseite nicht unbeabsichtigt löscht, wurde zusätzlich die Mindestüberdeckung eingeführt, unterhalb derer die Wischgeste ignoriert wird. Zusätzlich wird während des Wischvorgangs visuelles Feedback durch ein auf der bisher überstrichenen Fläche angezeigtes halbtransparentes Rechteck gegeben. Bei mausbasierter Interaktion ist das Löschen der Seite über ein Kontextmenü erreichbar, welches sich per rechter Maustaste öffnen lässt.

## 6.2. Systemaufbau

Aufbau der Testanwendung, Komponentendarstellung sowie ein Überblick über die Anbindung von Single- sowie Multitouch-Events an das Eventhandling der Qt-Klassenbibliothek.

### 6.2.1. Architektur des Systems

Die Testanwendung besitzt eine grafische Oberfläche, daher ist der Einsatz des Model-View-Controller-Patterns (MVC) naheliegend. Des Weiteren basieren große Teile der verwendeten Qt-Bibliothek auf dem MVC-Pattern, insbesondere das für die grafische Darstellung eingesetzte "Graphics Scene Framework". Die grafischen Objekte werden dabei in einer hierarchischen Struktur innerhalb der sog. Scene verwaltet und durch die View dargestellt. Die bereits im Analysetool eingesetzten Komponenten für die Hardwareanbindung ("Adapter") und Dateninterpretationen ("Driver") werden innerhalb der Testapplikation wiederverwendet.

Einen Überblick über alle Komponenten der Testanwendung zeigt [6.2 auf der nächsten Seite](#). Nachfolgend werden die innerhalb des Applikationsteils hinzugekommenen Komponenten näher erläutert. Dabei wird insbesondere auf die Implementation des Controllers eingegangen, welcher die Integration von Toucheingabedaten in die vorhandene Ereignisverarbeitung ermöglicht.

### 6.2.2. Implementation des Controllers

Nachfolgend wird die innerhalb der Testapplikation umgesetzte Integration der Toucheingabe näher betrachtet (siehe Abb. [6.3 auf der nächsten Seite](#)). Dabei wird zuerst auf das Konzept sowie den Ablauf der Ereignisbehandlung eingegangen. Anschließend wird die Implementation konkreter Gesten erläutert.

#### Ereignisbehandlung und -propagierung

Der Controller in aktuellen Fenstersystemen dient primär der Generierung von Ereignissen auf Basis von Eingabedaten die von Treibern des Betriebssystems zur Verfügung gestellt werden. Nachfolgend werden die Eigenschaften der Ereignisbehandlung sowie die Integration der Multitouch-Eingabedaten innerhalb der Testanwendung beschrieben.

Im Folgenden soll ein Überblick über die Eventverarbeitung innerhalb der Qt-Bibliothek gegeben werden. Des Weiteren werden Gemeinsamkeiten und Unterschiede zu anderen ereignisbasierten Systemen genannt, um eine Grundlage für die anschließende Integration der Multitouch-Eingabedaten zu schaffen.

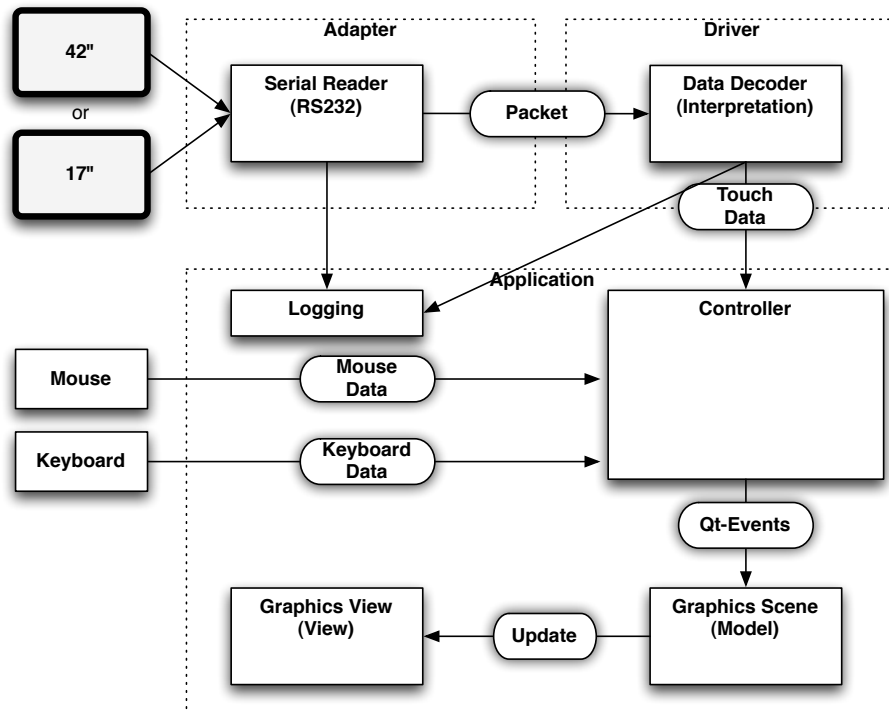


Abbildung 6.2.: Übersicht über Komponenten der Testanwendung

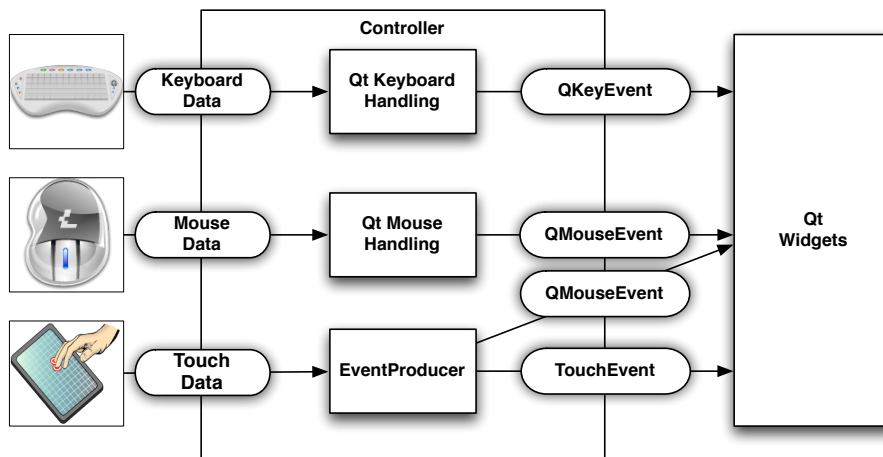


Abbildung 6.3.: Aufbau des Controllers

**Grundlage von Fenstersystemen** Fenstersysteme bestehen aus Toplevel-Fenstern sowie aus grafischen Elementen (auch als Controls bzw. Widgets bezeichnet), die innerhalb von Toplevel-Fenstern hierarchisch angeordnet sind. Die gesamte Oberfläche kann als eine Baumstruktur angesehen werden, an deren Wurzel der Desktop (bzw. das Desktop-Fenster) steht.

**Propagierung von Ereignissen** Durch Eingabegeräte ausgelöste Ereignisse werden in Fenstersystemen an das Element propagiert, welches den Eingabefokus besitzt (Tastaturereignis) oder sich an der aktuellen Eingabeposition befindet (Mausereignis). Von dort ausgehend wird bestimmt, ob das Element dieses Ereignis verarbeitet oder ignoriert. Wird das Ereignis ignoriert, so wird es dem Elternelement zugestellt und erneut geprüft, ob dieses Element das Ereignis verarbeiten kann (Propagierung an Elternelemente endet spätestens beim Toplevel-Fenster).

**Ereignisbehandlung in Qt** Qt wendet diese Ereignispropagierung innerhalb der eigenen Widgethierarchie an und dehnt sie auch auf das enthaltene "Graphics Scene Framework" aus<sup>2</sup>.

### Integration der Multitouch-Eingabedaten

Da gängige Betriebssysteme bisher keine direkte Unterstützung für Multitouch-Eingabedaten bieten, wird die Verarbeitung dieser Eingabedaten bis auf Weiteres in die Anwendung verlagert. Die Touchdaten werden dabei in die bestehende Ereignisbehandlung integriert.

Folgende Anforderungen muss die Integration erfüllen:

- Die Interaktion mit gewöhnlichen Widgets sollte weiterhin möglich sein, um bestehende Oberflächenelemente weiterhin nutzen zu können.
- Multitouch-Eingaben sollten analog zur Behandlung von Maus- und Tastatureingaben in die Ereignisverarbeitung integriert werden.

Die erste Anforderung lässt sich durch eine Emulation der Mauseingabe umsetzen. Die zweite Anforderung erfordert eine Erweiterung der Ereignisverarbeitung um Ereignisse, die durch die Multitoucheingabe mit mehr als einem Finger oder anderen Gegenständen ausgelöst werden. Beide Anforderungen werden innerhalb der Testanwendung mit Hilfe der EventProducer-Komponente umgesetzt.

---

<sup>2</sup>Mit Ausnahme von Toplevel-Fenstern sind Qt-Widgets seit Qt 4.4 keine Fenster in der Fensterverwaltung des Betriebssystems mehr. Qt verwaltet Inhalte von Toplevel-Fenstern ohne BS-Mittel

**Mausemulation** In Oberflächenelementen wird die mausbasierte Interaktion über die drei Ereignisarten "Mousebutton Down", "Mousebutton Up" sowie "Mouse Move" realisiert. Diese werden bei der Eingabe mit einem einzelnen Finger durch die EventProducer-Komponente erzeugt. Weiterhin implementiert die Komponente die für Mausereignisse benötigte Mousegrab-Logik: Ein Widget, welches das initiale "Mousebutton Down"-Ereignis erhalten hat, erhält von diesem Zeitpunkt an "Mouse Move"-Ereignisse bis der betreffende Mausknopf respektive Finger losgelassen oder entfernt wurde. Abschließend wird ein "Mousebutton Up"-Ereignis gesendet, wodurch der Mousegrab aufgehoben wird. Eine Erzeugung von Mausereignissen erfolgt ausschließlich, während ein einzelner Finger erkannt wurde. Alle anderen erkannten Touchobjekte werden in Form von Multitouchereignissen an Widgets versendet.

**Multitouchereignisse** Die EventProducer-Komponente erzeugt Multitouchereignisse, sobald mehr als ein Finger oder ein beliebiges anderes Objekt wie z.B. eine Handkante erkannt wird. Im Gegensatz zu der Bedienung mit einem Finger ergibt sich bei Eingaben mit größeren oder mehreren Objekten die Problematik, das Ereignis eindeutig einem einzelnen Widget zuzuordnen. Um die fehlende Eindeutigkeit bei Multitouchereignissen zu lösen können zwei unterschiedliche Ansätze gewählt werden:

1. Jedes Touchobjekt wird eigenständig behandelt und als Ereignis versendet. Die Zustellung kann an das Widget unter dem Mittelpunkt des jeweiligen Touchobjekts erfolgen.
2. Alle Touchobjekte werden gemeinsam behandelt und als Liste innerhalb eines Ereignis versendet.

Bei der ersten Methode können bei großen Touchobjekten mehrere Widgets in Frage kommen, an die das Ereignis zugestellt werden kann. Eine eindeutige Zuordnung von Ereignissen ist mit der ersten Methode aufgrund der großen Auflageflächen von Fingern sowie anderen Objekten nicht möglich. Für die eindeutige Zuordnung von Multitouchereignissen zu einem bestimmten Widget wäre ein Mechanismus ähnlich eines Mousegrabs vonnöten, der es ermöglicht, dass Widgets auch Ereignisse außerhalb ihrer Koordinaten erhalten, falls sie zuvor durch den Grabmechanismus ausgewählt wurden.

Aufgrund der zuvor genannten Probleme der ersten Methode wurde innerhalb der Testanwendung die zweite Methode gewählt. Hierdurch kann die Wahl des zu manipulierenden Objekts innerhalb der Anwendung getroffen werden und wird nicht vom Controller anhand einer vorgegebenen Logik festgelegt. Im Falle der Testanwendung ermöglicht dies die Zuordnung von Zwei-Finger-Gesten zu einem darunter befindlichen Foto, solange einer der beiden Finger das Foto berührt.

## Behandlung von Multitouchereignissen

**Unterscheidung zwischen einzelnen Gesten** Alle Gesten werden in der aktuellen Implementation ausschließlich anhand der Anzahl von Unterbrechungen, Touchobjekten sowie ggf. den Ausmaßen der Touchobjekte unterschieden. Da mit wachsender Anzahl von Unterbrechungen die Erkennung von Touchobjekten Fehler aufweisen kann, werden die Ausmaße der Touchobjekte ausschließlich für die Erkennung von ein und zwei Fingern genutzt. Die Erkennung von horizontalen und vertikalen Handkanten basiert dagegen auf der Gesamtbreite und -höhe aller Unterbrechungen, da beispielsweise eine vertikal aufgelegte Handkante mehr als eine vertikalen Unterbrechung hervorrufen kann (vgl. [4.4.6 auf Seite 42](#)).

**Rotation & Skalierung** Eine Rotation und Skalierung von Fotos erfolgt, sowie zwei Finger erkannt wurden. Die Erkennung basiert auf dem vom Data Decoder erkannten Objekttyp ("two-fingers"). Die Änderungen der Fingerpositionen werden anhand der Identifier der zwei Touchobjekte festgestellt. Durch die Nutzung der Identifier ist sichergestellt, dass die Delta-Werte nur aus den beiden gefundenen Touchobjekten errechnet werden.

Die für die Drehung und Skalierung benötigten Delta-Werte werden zwischen den aktuellen und den vorigen Positionen der Touchobjekte berechnet. Der Skalierungsfaktor errechnet sich aus der Änderung des Abstand der Mittelpunkte beider Touchobjekte. Der Rotationswinkel ist bestimmt durch den Winkel zwischen den beiden Geraden, die zwischen beiden Touchobjekten gezogen werden können.

**Fotofixierung** Die Fixierung von Fotos erfolgt gleichzeitig mit der Berechnung der Parameter für Rotation & Skalierung. Eine Fixierung erfolgt, wenn sowohl Drehwinkel als auch der Skalierungsfaktor für mehr als 300 Millisekunden nahezu unverändert geblieben sind. Eine exakte Prüfung auf  $0^\circ$  Drehung sowie einen gleichbleibenden Skalierungsfaktor ist aufgrund der leicht schwankenden Eingabewerte nicht möglich, daher wird die Prüfung anhand von Schwellwerten durchgeführt. Diese liegen bei 2 Pixeln Abstandsänderung zwischen den Fingern sowie bei  $\pm 0.4^\circ$  Winkeländerung.

**Sweep** Die Erkennung einer vertikalen Handkante erfolgt ebenfalls durch den Data Decoder (siehe [4.4.6 auf Seite 42](#)). Die Geste wird innerhalb des Models anhand des erkannten Objekttyps ("vertical-hand-side") aktiviert. Da die Seitenlöschung erst erfolgen soll, wenn mindestens 60 Prozent der Seitenbreite überstrichen wurden, wird während der Geste die insgesamt überstrichene Fläche errechnet und die Breite dieser Fläche mit der Seitenbreite verglichen. Die überstrichene Fläche ergibt sich über die Zeit aus der Vereinigungsmenge aller umgebenden Rechtecke für alle Achswerte (siehe [Abb. 6.4 auf der nächsten Seite](#)).



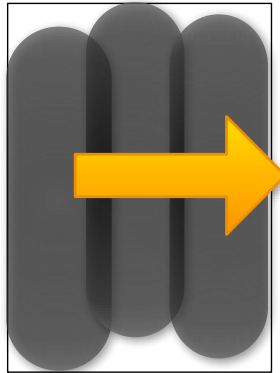


Abbildung 6.4.: Sweep Geste - Vereinigungsmenge der überstrichenen Fläche

### 6.2.3. Stand der Implementation & Ausblick

**Erkennung von Fehleingaben** Die Auswertung der Eingabedaten auf Basis von Achswerten anstatt anhand von erkannten Touchobjekten bietet momentan keine Möglichkeit, eventuelle Fehleingaben zu erkennen. Fehleingaben können beispielsweise durch in das Infrarotgitter ragende Ärmel oder ähnliche Objekte ausgelöst werden. Da ungewollte Unterbrechungen die Eingabe des Anwenders durch die entstehenden Achswertüberlappungen stark beeinflussen können, ist nicht mit einer ausreichend sicheren Lösung für die Ausblendung solcher Fehleingaben zu rechnen. Eine möglicher Ansatz um dem Anwender das Verhalten des Systems plausibler erscheinen zu lassen, wäre eine Visualisierung von fehlerhaften Eingabedaten. Als fehlerhaft könnten beispielsweise Achswerte gelten, deren Ausmaße keinem bekannten Objekt (Finger, Handkante) entsprechen oder deren Ausmaße weit überschreiten. An der Position der unerwarteten Achswerte könnte ein grafischer Effekt dargestellt werden, der die vom Anwender zu entfernende Unterbrechung hervorhebt.

**Objekt- und Gestenerkennung** Die Treiberkomponente welche die interpretierten Touchdaten liefert, unterscheidet bisher nur einige wenige Objektarten anhand der Größe der erkannten Touchobjekte. Eine Erkennung auf Basis von Wahrscheinlichkeiten für die Objektarten sowie eine Erkennung von Druckpunktschwerpunkten auf der Oberfläche, wie sie bei Fingern mehrerer Hände vorkommen können, würde für den Einsatz mit mehr als einem Benutzer sowie für die Implementation eines größeren und daher auch komplexeren Gestenrepertoires benötigt werden. Ein Ansatz für die Erkennung und Unterscheidung von Gesten könnte die Nutzung von Hidden-Markov-Models darstellen. Diese ermöglichen, entsprechendes Training vorausgesetzt, die Erkennung von ges-

tentypischen Bewegungsmustern und wären flexibler als die in der Testanwendung umgesetzte explizite Fallunterscheidung für einzelne Gesten.

**Mehrbenutzereinsatz** Die Erkennung von Objektarten sowie die Erkennung einzelner Gestenparameter innerhalb der Testanwendung ist zum Teil nicht für den Mehrbenutzerbetrieb ausgelegt. Der Einsatz mit mehr als einer Person war allerdings auch nicht Ziel der Testanwendung, da diese Nutzungsart bereits in Abschnitt 4.5 auf Seite 43 ausgeklammert wurde.

**Schnittstelle zwischen Hardware und Anwendung** Bei einer Weiterentwicklung des Systems würde sich die Nutzung einer generalisierten Schnittstelle zwischen Hardware und Anwendung anbieten um die Testanwendung auch auf anderer Multitouch-Hardware nutzen zu können. Des Weiteren würde eine einheitliche Schnittstelle die Verwendung der IRTOUCH-Hardware mit bestehenden Anwendungen ermöglichen. Aktuell kämen als Schnittstellen unter Anderem die Multitouchunterstützung von Windows 7 (siehe Win [2010]), das Sparsh-Ui Projekt (siehe Spa) sowie die neu eingeführte Unterstützung von Multitoucheingabedaten und Gesten in Qt 4.6 (siehe QtA [2009]) in Frage.

Die in Windows 7 für die Multitouchunterstützung neu eingeführten Ereignisarten WM\_GESTURE und WM\_TOUCH erlauben entweder eine Erkennung vordefinierter Gesten oder die Verarbeitung einzelner Touchpunkte. Voraussetzung für die Generierung dieser Ereignisse durch das Betriebssystem ist jedoch ein Hardwaretreiber. Dieser könnte auf der "Adapter"-Komponente basieren, der Aufwand für die Implementation eines betriebssystemspezifischen Treibers wäre jedoch ungleich höher als die direkte Anbindung an die Hardware auf Applikationsebene.

Das Protokoll des TouchAdapters<sup>3</sup> in Sparsh-Ui ist auf Punkte ohne Ausdehnung beschränkt. Gesten die nicht mit Fingern ausgeführt werden, wären mit dieser Schnittstelle daher nicht umsetzbar.

Da die Testanwendung bereits auf der Qt-Bibliothek basiert und die in Qt 4.6 neu eingeführten Ereignisarten für Multitoucheingabedaten sowie für Gesten die komplette Funktionalität der Testanwendung abdecken, würde sich eine Portierung auf diese Schnittstelle anbieten. Sowohl die Übertragung der Ausmaße eines Druckpunkts als auch die Definition eigener Gesten wie beispielsweise der Sweep-Geste sind möglich.

---

<sup>3</sup>API-Dokumentation siehe [http://code.google.com/p/sparsh-ui/wiki/SparshUI\\_Device\\_Adapter](http://code.google.com/p/sparsh-ui/wiki/SparshUI_Device_Adapter)

## 7. Usability-Test

Nachfolgend wird die Vorbereitung sowie Durchführung des Usability-Tests mit Hilfe der zuvor beschriebenen Testanwendung "Fotobuch" beschrieben. Abschließend werden die durchgeführten Tests ausgewertet sowie bewertet.

### 7.1. Vorbereitung

#### 7.1.1. Ziele des Usability-Tests

Hauptziel ist die praktische Überprüfung des implementierten Gestenrepertoires auf seine Funktionalität. Es soll geprüft werden, in wie weit sich die analysierten Hardwarebeschränkungen auf die Eingabe auswirken und in welchem Umfang die eingesetzten Algorithmen die Hardwarebeschränkungen mindern können. Letzteres wird durch die Beobachtung der Fehlerrate während der Drehung von Objekten erreicht werden.

#### 7.1.2. Definition der Zielgruppe

Da der Fokus des Usability-Tests auf der Qualität der technischen Umsetzung lag, wurden keine besonderen Anforderungen an die Auswahl der Testprobanden gestellt. Idealerweise sollte eine Gleichverteilung männlicher und weiblicher Testprobanden bestehen, um eventuell unterschiedliche Herangehensweisen bei der Bedienung einer Touchoberfläche in die Auswertung einzubeziehen. Die Anzahl der Testprobanden sollte zehn Probanden nicht übersteigen. Pro Proband muss mit etwa einer Stunde Aufwand pro Testlauf und Auswertung gerechnet werden. Des Weiteren ist bei mehr Probanden nicht von einer signifikanten Variation der Testergebnisse auszugehen.

#### 7.1.3. Aufbau des Usability-Tests

**Technischer Aufbau** Da reale Fotobücher üblicherweise zum Einkleben von Fotos auf einem Tisch liegend bearbeitet werden, wird der Test mit einem in einen Tisch eingelassenen Bildschirm durchgeführt (siehe Abb. [7.1 auf der nächsten Seite](#)). Die Höhe des zur Verfügung stehenden Tisches ermöglicht ausschließlich eine Bedienung im Stehen, daher sollte der einzelne Testdurchlauf

nur wenige Minuten dauern. Für längere Tests wäre ein Aufbau mit einem niedrigeren Tisch denkbar, vor dem der Proband sitzen könnte. Dies würde eine längere Nutzung ohne die ermüdende Armhaltung einer stehenden Bedienung ermöglichen.



Abbildung 7.1.: Tisch mit eingelassenem Bildschirm und IRTOUCH-Rahmen

Die Testläufe werden innerhalb des Usability-Labors an der HAW-Hamburg durchgeführt. Dies ermöglicht eine Aufnahme aller Aktionen auf dem Tisch durch sechs Kameras sowie einen Mitschnitt des Bildschirmgeschehens. Des Weiteren werden alle Toucheingabedaten für die spätere Auswertung in Form binärer Dumpdateien aufgezeichnet. Der Testablauf kann dadurch im Nachhinein aus diesen Dumps beliebig oft nachgestellt werden um eventuelle Besonderheiten während der Tests auf Ebene der Eingabedaten zu untersuchen. Insbesondere bei Fehleingaben soll so geklärt werden, welches Benutzerverhalten zu der fehlerhaften Erkennung geführt hat.

**Zweiteiliger Test** Die Tests werden in zwei Benutzergruppen durchgeführt. Eine Gruppe bedient die Testanwendung zuerst nur mit klassischer Mauseingabe gefolgt von der Multitoucheingabe, bei der anderen Gruppe wird die Reihenfolge umgedreht. Dadurch ist sichergestellt, dass beide Probandengruppen die Unterschiede der Eingabemethoden einschätzen und kommentieren können, ohne dass sich durch die Testreihenfolge ein Vor- oder Nachteil für eine bestimmte Eingabemethode ergibt. Des Weiteren erlaubt dieser Aufbau den direkten Vergleich zwischen Maus- und Multitoucheingabe.

**Vorstellung des Gestenrepertoires** Da die Auswahl des Gestenrepertoires eher nachrangig sein soll und die technische Implementation auf der Hardware im Vordergrund steht, steht das

Gestenrepertoire in Form einer Piktogrammliste (siehe Anhang B auf Seite 93) für den Testprobanden zur Verfügung. Die Liste wird dem Probanden vor dem Test auf einem Zettel ausgehändigt. Die Assoziation der skizzierten Gesten zu einzelnen Aktionen innerhalb der Testanwendung soll weiterhin vom Testprobanden durchgeführt werden.

**Aufgabenstellung für die Probanden** Die von den Probanden durchzuführenden Aufgaben sollen den Einsatz des gesamten Gestenrepertoires erfordern. Die an die Testprobanden ausgehändigte Aufgabenstellung ist in Anhang C auf Seite 94 nachzulesen. Jeder Aufgabe erfordert den Einsatz einer oder mehrerer Gesten (siehe Kap. 2.6.2 auf Seite 22). Die Zuordnung von Aufgaben zu Gesten ist in Tabelle 7.1 dargestellt.

Tabelle 7.1.: Zuordnung von Testaufgaben zu Gesten

Aufgabe	Geste(n)
Ablage von Fotos	Selektion, Verschiebung
Drehung & Skalierung von Fotos	Rotation, Skalierung
Fixierung von Fotos	Rotation, Skalierung (ohne Bewegung)
Seite leeren	Sweep
Neue Seite anlegen	Selektion
Seite wechseln	Selektion, Verschiebung

### 7.1.4. Fragen an Probanden

Vor jedem Testdurchlauf soll geklärt werden, wie groß das Vorwissen im Bereich der Touch- und Multitoucheingabe ist, um einschätzen zu können, ob der Testproband bereits eine konkrete Vorstellung über Gesten und die damit assoziierten Aktionen besitzt. Ein mögliches Vorwissen kann das Verhalten des Testprobanden sowohl positiv als auch negativ beeinflussen, da sich bei der Nutzung anderer Touchtechniken bereits ein mentales Modell über die Nutzung dieser Technik gebildet hat. Dieses kann sowohl Auswirkungen auf die Art der ausgeführten Gesten, die Nutzung und Haltung von Händen und Fingern sowie auf die Ausführungsgeschwindigkeit einzelner Gesten haben. Je nach Intensität der bisherigen Nutzung von Touchtechniken kann auch bereits eine Verknüpfung einzelner Gesten zu Aktionen im Bewusstsein des Anwenders vorhanden sein.

Nach Durchführung des Tests soll ein informelles Feedback gegeben werden, welches die Interpretation und Auswertung der während des Tests angefallenen Daten unterstützen soll. Ein Schwerpunkt hierbei soll die subjektive Einschätzung des Probanden sein. Es soll geklärt werden,

in welchem Maße die Probleme bei der Gesten- und Eingabeerkennung vom Probanden wahrgenommen werden. Einen groben Leitfaden für das abschließende Feedback bietet der in Anhang D auf Seite 95 befindliche Fragebogen.

### 7.2. Durchführung

Der Usability-Test wurde in den Räumlichkeiten des Usability-Labors der HAW durchgeführt. Der in Abb. 7.1 auf Seite 68 dargestellte Tisch wurde dafür in der Mitte des in Abb. 7.2 gezeigten Raumes aufgestellt. Die Testläufe wurden aus sechs unterschiedlichen Blickwinkeln von Kameras sowie durch die in 7.2 sichtbaren Mikrofone aufgezeichnet.



Abbildung 7.2.: Usability Labor an der HAW

Tests der Mauseingabe wurden nur mit zwei der insgesamt sechs Testkandidaten durchgeführt. Grund hierfür waren vor allem die während der Tests auftretenden Probleme bei der Touchbedienung wodurch ein direkter Vergleich zur Mauseingabe wenig sinnvoll erschien. Die Mauseingabe wurde daher nur exemplarisch überprüft. Es stellte sich außerdem während der Tests heraus, dass die zum Mac Pro gehörende Mighty Mouse für die Probanden eine eher ungewohnte Bedienung bot. Dies könnte unter Anderem im Fehlen eigenständiger Maustasten begründet gewesen sein. Es musste statt einer einzelnen Maustaste die gesamte Mausoberfläche heruntergedrückt werden um einen Mausklick auszulösen.

### 7.2.1. Ergebnisse für einzelne Gesten

Um eine Übersicht über die Ergebnisse für einzelne Gesten zu ermöglichen, werden diese im Folgenden zusammengefasst beschrieben. Im Anschluss wird auf die einzelnen Testverläufe detaillierter eingegangen.

**Ablage von Fotos** Die Ablage von Fotos von der Fotoliste ausgehend auf eine der beiden Fotobuchseiten stellte für keinen der Testprobanden ein Problem dar. Die einzige Ausnahme bildeten Fehlerkennungen während zweier Testläufe, die jedoch durch unbeabsichtigte Bewegungen der parallel angeschlossenen Maus oder einen in das Infrarotgitter hineinragenden Ärmel zu erklären waren.

**Drehung & Skalierung von Fotos** Mehrere Probanden benötigten einen Hinweis auf die Nutzung von zwei Fingern, es wurde mehrfach eine Drehung mit nur einem Finger versucht. Die symbolische Darstellung der zur Verfügung stehenden Gesten wurde daher entweder nicht richtig interpretiert oder der Zusammenhang zwischen Geste und Aktion war für den jeweiligen Probanden nicht ersichtlich.

Die Rotation verlief bei den meisten Probanden nicht wie gewünscht. Nur wenige Versuche der Probanden führten zu einer korrekten Drehrichtung bei Überlappung der Finger auf einer Achse.

Dies ist zum Einen damit zu begründen, dass die Nutzer die Geste sehr schnell ausführten, wodurch der Algorithmus die Überlappung nicht korrekt ausgleichen konnte. Zum Anderen wurden beide Finger oftmals gleichzeitig sowie auf gleicher Höhe auf einer Achse aufgesetzt. Insbesondere Letzteres erschwerte die Bedienung maßgeblich, da dieses Eingabeverhalten von der Implementation nicht ausgeglichen werden konnte (siehe Abschnitt [5.4.8 auf Seite 53](#)).

**Fixierung von Fotos** Die Fixierung einzelner Fotos verlief bei den meisten Probanden erst nach Hilfestellung fehlerfrei. Den Probanden war das Konzept des Fixierens von Fotos auf einer Fotoseite nicht klar. Weiterhin gab es unter den Probanden keine eindeutige Assoziation zwischen einer bestimmten Geste und der Fixierungsaktion. Es wurden daher mehrere Gesten für die Fixierung von den Probanden ausprobiert:

- Längeres Halten eines einzelnen Fingers auf dem Foto
- Doppeltes Antippen des Fotos

**Seite leeren** Beim Löschen aller fixierten Fotos auf einer Fotobuchseite wurden von den Probanden zwei unterschiedliche Ansätze gewählt:

1. Manuelles Löschen aller Fotos auf der Seite per Drag-and-Drop auf das Papierkorbsymbol
2. Löschen per Sweep-Geste

Das manuelle Löschen funktionierte bei allen Probanden fehlerfrei. Die Löschung über die Sweep-Geste funktionierte aus technischer Sicht einwandfrei, jedoch wurde die Assoziation zwischen der Geste und der Aktion nicht bei allen Probanden automatisch geknüpft. Des Weiteren führten nicht-fixierte Fotos, die auf einer Seite lagen zu Verwirrung, da diese nach der Seitenleerung weiterhin vorhanden waren.

**Neue Seite anlegen** Das Hinzufügen einer weiteren Seite über den entsprechenden Knopf war erwartungsgemäß für keinen der Probanden ein Problem. Lediglich die eventuell oberhalb der neuen Seite angezeigten nicht-fixierten Fotos führten bei einem der Testläufe zu leichter Verwirrung, da für den Probanden nicht eindeutig ersichtlich war, ob eine neue Seite erstellt worden war oder ob noch die vorige Seite angezeigt wird.

**Seite wechseln** Der Seitenwechsel über die aktiven Seitenecken war nur für einen Probanden nicht offensichtlich. Für die restlichen Probanden schien die Anlehnung an das Umblättern einer normalen Buchseite intuitiv und verständlich zu sein. Die Geste für das Umblättern wurde von mehreren Probanden explizit als intuitiv beschrieben, zwei Probanden wünschten sich allerdings ein weitreichenderes visuelles Feedback, wie beispielsweise Pfeile, die bei Berührung der Ecke die benötigte Bewegungsrichtung anzeigen.

### 7.2.2. Ergebnisse pro Testproband

Die einzelnen Testverläufe werden an dieser Stelle genauer dargestellt, wobei auf die Besonderheiten des jeweiligen Verlaufs näher eingegangen wird. Um Wiederholungen zu vermeiden werden einzelne Aktionen bei Bedarf im Text nur in Stichpunktform erwähnt.

#### Proband 1

**Ablage von Fotos** Zu Anfang funktionierte die Ablage von Fotos aus der Fotoliste nicht. Der Proband versuchte daraufhin durch schnelles Tippen eine Geste ähnlich des Doppelklicks mit einer Maus auszuführen, vermutlich um ein Bild explizit aus der Fotoliste zu holen. Aus den aufgezeichneten Eingabedaten ergab sich, dass die ebenfalls an das System angeschlossene Maus sich zu Anfang bewegt haben muss, wodurch die Emulation von Mausereignissen gestört wurde. Nach einigen Versuchen war die Ablage mehrerer Fotos dann erfolgreich.



**Drehung & Skalierung** Die Skalierung von Fotos versuchte der Proband sowohl mit einem Finger, ohne jedoch die für die Mausbedienung vorgesehene rechte untere Ecke zu nutzen, als auch mit doppeltem Tippen auf das Objekt. Erst ein Hinweis auf die Möglichkeit zwei Finger zu nutzen half dem Probanden die Skalierung durchzuführen.

Während der Rotation war die Drehrichtung vorwiegend entgegengesetzt der gewünschten Richtung. Der Proband setzte den zweiten Finger zumeist auf gleicher Höhe mit dem ersten Finger auf, oftmals auch fast gleichzeitig. Größere Rotationswinkel wurden vom Probanden für nachfolgende Fotos vermieden.

**Fixierung von Fotos** Das Fixieren der Fotos funktionierte erst nach dem Hinweis, dass hierfür ebenfalls zwei Finger benötigt werden. Der Proband berührte daraufhin automatisch zwei gegenüberliegende Ecken eines Fotos mit jeweils einem Finger.

**Seite leeren** Das Vorgehen zum Leeren einer Fotobuchseite war dem Probanden unklar. Der Proband war sich unsicher mit welcher Geste die Seite zu leeren ist, er schaute sich daher das Gestenrepertoire erneut an. Nach Entfixierung eines Fotos strich der Proband mit einer Handkante darüber, wodurch das noch fixierte Foto auf derselben Seite entfernt wurde, das eben gelöste Foto jedoch bestehen blieb. Der Zusammenhang zwischen dem Leeren einer Seite und dem Fixierstatus war dem Probanden offensichtlich nicht klar, erwartet wurde ein Löschen des losen Fotos.

**Neue Seite anlegen** Das Hinzufügen einer Seite über den entsprechenden Knopf sowie die Ablage weiterer Fotos stellte kein Problem dar.

**Seite wechseln** Die aktive Ecke für den Seitenwechsel wurde schnell gefunden, der erste Versuch war jedoch nur ein einfaches Antippen der Seitenecke. Da keine Reaktion bis auf die Farbänderung der Ecke erfolgte, entschied sich der Proband zum ziehen der Ecke in Richtung der Buchmitte, welches die gewünschte Aktion hervorrief. Eine weitere Hilfestellung war nicht nötig.

### **Proband 2**

Im zweiten Testlauf wurde sowohl die Touch- als auch die Mausbedienung getestet.

#### **Touchbedienung**

**Drehung & Skalierung von Fotos** Der Proband nutzte zwei Finger an gegenüberliegenden Ecken zum skalieren, skalierte die Bilder jedoch so, dass sie ziemlich exakt die gesamte

Seite ausfüllen. Angedacht für den Test war jedoch eine lockere und daher unterschiedlich gedrehte und skalierte Anordnung der einzelnen Fotos. Die Rotation eines Fotos erfolgte sehr langsam, so dass das Foto irrtümlicherweise fixiert wurde. Der Proband erkannte diesen Zustand jedoch als "fixiert", die Darstellung erschien daher als eindeutig, auch wenn die Aktion für ihn unerwartet war. Eine Entfixierung von Fotos wurde nach dem Hinweis "zwei Finger versuchen" durch eine Zoomgeste angestrebt und führte daher nicht zum Erfolg. Es wurde daher auf die nachfolgende Aufgabe verwiesen.

**Seite leeren** Die Seite wurde durch manuelles Drag-and-Drop der Fotos auf das Papierkorbsymbol geleert, da die Fotos zuvor vom Probanden nicht fixiert worden waren.

**Neue Seite anlegen** Die Erstellung einer neuen Seite verlief unproblematisch.

**Seite wechseln** Das Umblättern über die Seitenecke gelang dem Probanden sofort ohne weitere Hilfestellung. Aus dem Ergebnis der vorigen Aufgabe ergab sich, dass das lose Foto weiterhin sichtbar blieb.

**Mausbedienung** Sowohl die Ablage sowie das Entfernen von Fotos als auch die Rotation sowie Skalierung stellten für den Probanden keinerlei Problem dar. Die Fixierung von Fotos gelang erst nach dem Hinweis auf einen Doppelklick. Der Proband versuchte durch Gedrückthalten der linken Maustaste eine Fixierung, ähnlich der zuvor still stehenden Finger auf dem Foto.

**Feedback des Probanden** Der Testproband merkte an, dass sich die Bedienung für ihn intuitiv anfühlt. Die Drehung von Fotos wurde ebenfalls als positiv beschrieben, allerdings waren die beobachteten Drehwinkel während des Tests immer geringer als  $45^\circ$ , so dass Probleme während der Überlappung der Achswerte nicht zum tragen kamen. Die Skalierung von Fotos wünschte sich der Anwender von der Ecke aus, in der sich der stillstehende Finger befand. Die Skalierung hatte in der Testanwendung als Mittelpunkt immer das Zentrum des Fotos.

### **Proband 3**

#### **Mausbedienung**

**Ablage von Fotos** Kein Problem.

**Drehung & Skalierung von Fotos** Kein Problem. Der Proband merkte jedoch an, dass die Apple Mighty Mouse für ihn ungewohnt ist.

**Fixierung von Fotos** Der Proband findet eigenständig den Doppelklick zur Fotofixierung.

**Seite leeren** Der Proband leert die Seite durch manuelles Drag-and-Drop der Fotos auf das Papierkorbsymbol. Das Kontextmenü wird nicht verwendet.

**Neue Seite anlegen** Der entsprechende Knopf zum anlegen einer neuen Seite wird sofort gefunden. Der Proband ist jedoch verwirrt durch nicht fixierte Fotos der ersten Seite die nun auf der dritten Seite zu liegen scheinen. Ein Hinweis darauf, dass die Fotos nicht fixiert waren, klärt die Situation auf.

**Seite wechseln** Die Seitenecken werden vom Probanden erkannt und richtig genutzt.

### **Touchbedienung**

**Drehung & Skalierung** Die Drehung wird vom Probanden sehr schnell durchgeführt, dadurch kommt es vermehrt zu Erkennungsproblemen der Druckpunkte sowie zu einer daraus resultierenden Drehung entgegen der gewünschten Richtung. Ein auf Bildschirmgröße hochskaliertes Foto wird fälschlicherweise durch Überlappung mit dem Papierkorbsymbol gelöscht. Das Verhalten verwirrt den Probanden.

**Fixieren von Fotos** Der Proband nutzt nach Hinweis die beiden oberen Fotoecken, wodurch keine Erkennung beider Finger möglich ist. Da bei erneuten Versuchen keine andere Positionierung der Finger gewählt wird, wurde die Aufgabe übersprungen.

**Seitenwechsel** Der Proband kennt das Verhalten für den Seitenwechsel bereits von der Mausbedienung, benötigt jedoch drei Anläufe, da die Ecke vom Finger anscheinend nicht getroffen wird.

**Löschen von Seiteninhalten** Das Löschen funktioniert nach zweimaligem Anlauf, der Proband hat jedoch ebenfalls die Löschung der lose herumliegenden Fotos erwartet.

### **Proband 4**

**Ablage von Fotos** Kein Problem

**Drehung & Skalierung von Fotos** Die Drehung eines kleinen Fotos erweist sich als problematisch da eine permanente Überlappung der Achswerte beider Finger besteht. Der Proband merkt an, dass die Kombination von Rotation und Skalierung eine ungewollte Größenänderung des Fotos verursacht. Auch bei diesem Probanden kommt es in Folge langsamer Drehung und Skalierung zu ungewollten Fixierungen von Fotos.

**Fixierung von Fotos** Ein Hinweis auf die Fixierung nach dem erfolglosen Löschen einer Seite hilft dem Probanden nicht, da er die zugehörige Fixierungsgeste nicht eigenständig findet, er wendet stattdessen eine Geste ähnlich eines "Doppelklicks" an. Ein Hinweis auf die Verwendung von zwei Fingern ist ebenfalls nicht ausreichend, da der Proband die Finger nur für sehr kurze Zeit auf dem Foto belässt. Nach Hinweis auf die Länge der Berührung funktioniert das Fixieren und daraufhin auch das nachfolgende Löschen.

**Seite leeren** Die Leerung ist beim ersten Versuch erfolgreich, die zugehörige Geste wird eigenständig gefunden. Da keines der vorher abgelegten Fotos fixiert war wurde auch kein Foto entfernt.

**Neue Seite anlegen** Der Button wird vom Probanden gefunden, er lässt sich jedoch von den losen Fotos die auf der neuen Seite liegen verwirren und legt daher eine weitere Seite an.

**Seite wechseln** Die aktive Seitenecke wird vom Probanden erkannt, der erste Versuch ist jedoch nur ein Antippen und kein seitliches ziehen der Ecke. Letzteres wird beim dritten Versuch vom Probanden ausprobiert und funktioniert wie erwartet.

**Feedback des Probanden** Der Proband bezeichnete die Interaktion mit dem System insgesamt als "flüssig". Insbesondere die Geste zum umblättern von Seiten gefiel ihm. Das Fixieren von Fotos bezeichnete er hingegen als unintuitiv.

### Proband 5

Während des Testdurchlaufs führte ein in das Infrarotgitter hineinragender Ärmel des Probanden mehrmals zu Fehlerkennungen.

**Ablage von Fotos** Kein Problem.

**Drehung & Skalierung von Fotos** Der Proband ignoriert die Information über das Gestenrepertoire und nutzt stattdessen die rechte untere Fotoecke für Rotation und Skalierung und hat aufgrund der geringen Größe der Ecke Probleme diese mit dem Finger zu treffen.

**Fixierung von Fotos** Es wird ein Hinweis auf die Fotofixierung mit Hilfe von zwei Fingern gegeben. Der Proband berührt nur kurzzeitig das Foto mit beiden Fingern und versucht durch doppeltes Antippen eine Fixierung zu erreichen. Weiterhin bemerkt er währenddessen, dass zwei Finger für die Rotation und Skalierung verwendet werden können.

**Seite leeren** Die Aufgabe wurde vom Probanden ausgelassen.

**Neue Seite anlegen** Kein Problem.

**Seite wechseln** Der Proband erkannte die aktive Seitenecke, tippte jedoch nur kurzzeitig darauf. Ein Hinweis auf ein "ziehen der Ecke" half ihm ebenfalls nicht bei der Lösung der Aufgabe. Der Proband versuchte daraufhin die Seitenecke mit zwei Fingern zu manipulieren, obwohl die Ecke offensichtlich zu klein für diese Art der Bedienung ist.

**Feedback des Probanden** Es wurde angemerkt, dass mehr Feedback für die aktiven Seitenecken zum Umblättern hilfreich gewesen wäre. Als Beispiel nannte der Proband Pfeile, welche die benötigte Bewegungsrichtung während der Gestendurchführung anzeigen.

### **Proband 6**

Der Proband ist aktiver Nutzer eines Apple iPhones und daher bereits durch die kapazitive Multi-touchhardware dieses Geräts vortrainiert.

**Ablage von Fotos** Kein Problem

**Drehung & Skalierung von Fotos** Der Proband nutzte sofort zwei Finger, die Skalierung war ihm persönlich jedoch zu schnell. Die Rotation erfolgte zumeist in die entgegengesetzte Richtung. Grund hierfür war, dass der Proband beide Finger gleichzeitig aufsetzte und beide Finger während der Rotation und Skalierung gleichzeitig bewegte. Aufgrund der zu Anfang schnellen Skalierung verringerte der Proband die Eingabegeschwindigkeit stark, wodurch es mehrmals zu einer ungewollten Fixierung von Fotos kam.

**Fixierung von Fotos** Da die Fixierung bereits ungewollt ausgeführt wurde, wurde dieser Punkt nicht mehr explizit betrachtet.

**Seite leeren** Der Proband versuchte die Wischgeste auf einer Seite mit losen Fotos und wechselte daraufhin auf die Nutzung des Papierkorbsymbols und ein Drag-and-Drop einzelner Fotos über.

**Neue Seite anlegen** Kein Problem

**Seite wechseln** Der Proband tippte zuerst die Seitenecke nur kurz an, nutzte nach dem ersten Misserfolg dann aber ebenfalls das seitliche rüberziehen der Ecke.

**Feedback des Probanden** Die Skalierung von Fotos war dem Probanden wie bereits angemerkt zu sensibel. Weiterhin merkte er während des Tests eine fehlende Intuitivität beim Umblättern an, konnte jedoch keine genauere Angaben zu dieser Aussage machen.

### 7.3. Auswertung

Die Auswertung erfolgt in mehreren Teilen, wobei die Eigenschaften der Hardware, die Eingabeerkennung sowie die Anwendungslogik getrennt voneinander betrachtet werden. Abschließend werden die Beobachtungen nochmals zusammengefasst betrachtet und bezogen auf die in Abschnitt 2 auf Seite 12 genannten Gestaltungsgrundsätze diskutiert.

#### 7.3.1. Vorwissen der Probanden

Alle Probanden nutzen gelegentlich Geräte mit Singletoucheingabe wie beispielsweise Fahrkartenschalter. Nur einer der Probanden nutzt regelmäßig ein Gerät mit Multitoucheingabe (Apple iPhone).

Während der Tests konnte oftmals eine starke Fixierung auf die Singletoucheingabe beobachtet werden. Dies lässt sich zum Einen über die bisherige Nutzung reiner Singletouchsysteme erklären, zum Anderen wurde bereits bei der Beobachtung von Gesten in Epps u. a. [2006] festgestellt, dass die Probanden die Mehrheit der Aktionen nur mit Hilfe eines einzelnen Fingers ausführen.

#### 7.3.2. Qualität der Eingabeerkennung

Insbesondere während der Rotation von Fotos traten diverse Probleme auf. Ein Großteil der Testprobanden setzte für die Rotation beide Finger gleichzeitig sowie auf gleicher Höhe auf. Durch dieses Verhalten war die Erkennung einzelner Druckpunkte jedoch nicht mehr möglich (siehe auch Abschnitt 4.4.4 auf Seite 38 und 5.4.8 auf Seite 52), daher führte die nachfolgende Kreisbewegung einer der Finger nicht zur gewünschten Rotation des darunter liegenden Fotos. Hinzu kam, dass die Geschwindigkeit der Rotation bei vielen Anwendern oberhalb der Toleranzgrenze des Algorithmus lag, wodurch eine verlässliche Zuordnung der Druckpunkte nicht mehr möglich war. Da die in 5.4 auf Seite 47 beschriebenen Deltawerte sich nicht beliebig erhöhen lassen, bliebe für eine Verbesserung der Erkennung bei hohen Eingabegeschwindigkeiten einzig die Erhöhung der Datenrate der Hardware.

Innerhalb der Anwendung offenbarten sich die zuvor genannten Probleme durch einen Wechsel der Rotationsrichtung nach einer Achswertüberlappung. Fast alle Probanden reagierten auf dieses Fehlverhalten mit dem Entfernen beider Finger und einer Wiederholung der Geste. Da sich das Verhalten der Probanden zwischen während der wiederholten Ausführung jedoch nur geringfügig veränderte, musste üblicherweise ein Hinweis auf die Grundstellung beider Finger gegeben werden ("nicht gleichzeitig und auf unterschiedlicher Höhe aufsetzen"). Ohne diese Hinweise war den Probanden nicht ersichtlich, aus welchem Grund die Eingabe nicht die gewünschte Reaktion hervorrief und wie sie den Problemfall umgehen konnten.

Ein weiteres Hindernis ergab sich durch ungewollte Unterbrechungen des Infrarotgitters. So konnten beispielsweise in das Gitter hineinragende Ärmel die Erkennung eines einzelnen Fingers bereits unmöglich machen, da ein Großteil der horizontalen Achse durch den Gegenstand abgeschattet war. Ein Ansatz zur Verbesserung des Systemverhaltens während ungewollter Abschattungen wurde in [6.2.3 auf Seite 65](#) bereits angesprochen.

### 7.3.3. Gestenspezifische Beobachtungen

**Fixierung von Fotos** Da die Geste zur Fixierung sich nur durch Ausbleiben einer Fingerbewegung von der Geste zur Rotation und Skalierung unterscheidet, kam es während der Tests desöfteren zu ungewollten Fixierungen einzelner Fotos.

Das unterschiedliche Verhalten zwischen der Maus- und Toucheingabe beim Fixieren von Fotos erschwert für Anwender eventuell den Wechsel zwischen den Eingabearten. Anwender die zuerst die Mausbedienung verwendet haben, werden vermutlich zur Fixierung Fotos doppelt antippen, im umgedrehten Fall hätte der Anwender dagegen keinerlei Equivalent zur Bedienung mit zwei Fingern.

Das mentale Modell sowie die daraus entstehende Erwartungshaltung des Anwenders unterscheiden sich für diese Geste je nach verwendeter Eingabeform. Im Vergleich zur Fixierung ist die Geste zur Verschiebung von Fotos bei beiden Eingabearten nahezu identisch, für den Anwender wäre ein Wechsel der Eingabeart daher unerheblich. Da sich Maus- und Toucheingabe in ihren Gesten stark voneinander unterschieden können, sollte für produktive Systeme vom gleichzeitigen Einsatz beider Techniken durch eine einzige Benutzergruppe abgesehen werden.

**Sweep-Geste** Die Erkennung der Sweep-Geste war unproblematisch und funktionierte bei den Probanden die sie eingesetzt haben fehlerfrei, die glatte Glasoberfläche der eingesetzten Hardware erlaubte jedoch keine flüssige Bewegung der Handkante. Bei einigen Probanden haftete die Handoberfläche stark auf der Glasoberfläche. Die Geste ist daher vermutlich eher für raue oder aufgeraute Oberflächen geeignet, auf denen eine Handkante seitlich bewegt werden kann, ohne auf der Oberfläche haften zu bleiben.

### 7.3.4. Anwendungsspezifische Beobachtungen

#### Fixierung von Fotos

Die explizite Unterscheidung zwischen fixierten und beweglichen Fotos erhöht offensichtlich die Anwendungskomplexität für den Anwender. Den Probanden war unklar, dass Fotos expliziert fixiert

werden müssen, wie sich fixierte von nicht fixierten Fotos unterscheiden und wie ein Wechsel zwischen den beiden Modi vollzogen werden kann.

Für das fehlende Verständnis des Konzepts können mehrere Faktoren verantwortlich sein, deren Auswirkungen sowie mögliche Lösungsansätze nachfolgend diskutiert werden.

**Visuelle Präsentation** Die visuelle Unterscheidung zwischen fixierten und unfixierten Fotos ist entweder nicht auffällig genug oder die dafür vorgesehene farbige Umrahmung führte nicht zur gewünschten Assoziation beim Anwender. Einige Probanden gaben dies auch in ihrem Feedback explizit an. Eine mögliche Lösung wäre die Einführung realistisch wirkender Schatten für lose Fotos sowie eine Darstellung von Fotoecken für fixierte Fotos. Zusätzlich würde sich eine Animation während der Fixierungsgeste anbieten, die während des Stillhaltens beider Finger den Übergang zwischen beiden Modi visualisiert und dem Anwender so ein zeitnahes Feedback darüber liefert, dass seine momentane Eingabe eine Aktion nach sich ziehen wird.

**Eigenschaften der Fixierungsgeste** Wie bereits in [7.3.3 auf der vorherigen Seite](#) beschrieben, waren die Parameter für die Fixierungsgeste ungünstig gewählt. Als Alternative zum Fixieren mit zwei Fingern würde sich das von mehreren Probanden angewendete doppelte Antippen von Fotos anbieten. Diese Geste wäre weit weniger realistisch als die Nutzung zweier Finger, die das Andrücken eines Fotos auf der Fotobuchseite repräsentieren, jedoch ist die Analogie zur Mausbedienung beim doppelten Antippen ungleich höher. Eventuell würde sich auch die Implementation beider Gesten anbieten, da auf diese Weise sowohl Anwender mit starkem Bezug zur Mauseingabe wie auch Anwender mit dem Wissen über reale Fotobücher ihr jeweiliges Wissen anwenden können.

**Verhalten loser Fotos** Das Verhalten der Fotos innerhalb der Testanwendung unterscheidet sich zu stark vom realen Vorbild. So bleiben nicht fixierte Fotos die auf einer Seite liegen trotz Umblätterns weiterhin an gleicher Stelle liegen. Eine reale Buchseite hätte beim Umblättern zu einer Verschiebung aller losen Fotos geführt. Des Weiteren würde bei einem realen Fotobuch die Anordnung loser Fotos vom Anwender eher um das Fotobuch herum erfolgen um einen Überblick über die zur Verfügung stehenden Fotos zu erhalten. Eine Ablage auf dem Fotobuch hingegen ginge im Allgemeinen mit einer Fixierung des Fotos einher. Die fehlende Nähe zum realen Vorbild könnte dazu geführt haben, dass ein eventuell vorhandenes mentales Modell über die Nutzung von Fotobüchern während der Interaktion überhaupt nicht zum Tragen kam.



**Implizites Fixieren anstatt explizitem Fixieren** Anwendungen wie beispielsweise Vektorgrafikeditoren sehen die Zielposition einer Objektverschiebung automatisch als feste Position des verschobenen Objekts an und benötigen keine explizite Fixierung. Falls dem Anwender dieses Konzept bereits bekannt ist, wird er das Verhalten bei vergleichbaren Anwendungstypen ebenfalls erwarten. Für diese Anwenderklasse wäre die Erwartungskonformität (vgl. Abschnitt 2.1 auf Seite 12) der Testanwendung daher nicht erfüllt.

Um die durch die explizite Fixierung entstandenen Verständnisprobleme zu mindern, würde sich die Einführung einer impliziten Fixierung anbieten. Fotos, die auf einer Seite abgelegt wurden, sollten automatisch auf der jeweiligen Seite fixiert werden. Die Fixierung ließe sich durch die längere Berührung mit einem oder mehreren Fingern lösen. Dieses Verhalten würde die versehentliche Verschiebung einzelner Fotos weiterhin vermeiden, dem Anwender aber die explizite Modumschaltung zwischen fixierten und losen Fotos abnehmen. Zusätzlich würde die Mehrdeutigkeit der Sweep-Geste aufgelöst werden, da sich auf einer Fotobuchseite ausschließlich fixierte Fotos befinden können.

Eine besondere Geste zum Lösen einer Fixierung wäre bei diesem Ansatz nicht zwingend nötig, da jede andere Geste bei längerer Ausführung das jeweilige Foto lösen könnte. So würde eine Drehung um einen Mindestwinkel oder eine Verschiebung über eine Mindestentfernung das Foto automatisch lösen, eine kurze Berührung würde dagegen das Foto an seiner bisherigen Stelle belassen.

### Seitenwechsel

In Abschnitt 7.2.1 auf Seite 71 wurde bereits angedeutet, dass das visuelle Feedback während des Seitenwechsels nicht für alle Probanden ausreichend war. Da die Mehrheit der Probanden beim ersten Versuch die Seitenecken mit einem normalen Button verwechselten und diesen nur kurzzeitig antippten, erscheint eine Erweiterung des visuellen Feedbacks sinnvoll.

Es würde sich anbieten, einen Pfeil einzublenden, sobald eine der Seitenecken berührt wird, um einen Hinweis auf die benötigte seitliche Bewegung zu geben. Dieser könnte während der Bewegung die Position parallel zur Bewegung des Fingers ändern, um zu visualisieren, wie lange die Seitwärtsbewegung fortgesetzt werden muss.

Alternativ könnte eine Art "Overlay"-Anleitung eingeführt werden, welche die diversen Interaktionsmöglichkeiten innerhalb der Oberfläche zusammengefasst darstellt. Dieser Ansatz wäre jedoch nur für Oberflächen mit einer beschränkten Anzahl von Interaktionsmöglichkeiten umsetzbar. Bei komplexeren Oberflächen würde dies schnell in einer mehrseitigen Anleitung ähnlich eines Hilfedokuments münden.

### **Aktive Fotoecke**

Die Anzeige der aktiven Fotoecke für die Skalierung und Rotation per Maus führte bei einem der Probanden dazu, dass er diesen vergleichsweise kleinen Button für die Skalierung mit einem Finger nutzte. Da der Button nur für die Mauseingabe vorgesehen ist, sollte er bei einer reinen Touchbedienung entfernt werden. Generell kann daher davon ausgegangen werden, dass Oberflächenelemente die nur für eine bestimmte Eingabeform vorgesehen sind auch nur dann sichtbar sein sollten, wenn sie für das aktuelle Eingabegerät geeignet sind. Falls ein System unterschiedliche Eingabegeräte wie Touch- und Mauseingabe unterstützen muss, so sollte das verwendete Eingabegerät einstellbar sein oder falls möglich durch automatische Hardwareerkennung automatisch erkannt werden.

### **Skalierungsursprung von Fotos**

Das Zentrum für die Skalierung von Fotos war innerhalb der Testanwendung auf die Fotomitte festgelegt. Einer der Testprobanden merkte jedoch an, dass er sich gewünscht hätte, dass bei der Skalierung der still stehende der beiden Finger das Skalierungszentrum festlegt. So könnte etwa ein Foto an der linken oberen Ecke festgehalten und von dort ausgehend skaliert werden. Der Vorteil dieser Methode im Vergleich zum Verhalten in der Testanwendung wäre in der Kontrolle der Ausrichtung mehrerer Fotos zueinander zu sehen. Auf diese Weise könnten Fotos beispielsweise auf den verbliebenen freien Platz zwischen anderen Fotos skaliert werden, ohne dabei durch die Skalierung umliegende Bilder zu überlappen.

### **Interaktion mit dem Papierkorbsymbol**

Die Desktop-Metapher des Papierkorbs, auf den man Objekte zum Löschen fallenlassen kann war einem Großteil der Probanden geläufig. Das Verhalten des Papierkorbs war allerdings in manchen Situationen unpassend, da dieser bei mehr als 50% Überlappung durch ein Foto dieses bereits löschte. Dabei war es unerheblich, ob die Hälfte des Fotos oder die Hälfte des Papierkorbs überlappt waren. Bei sehr großen Fotos führte dieses Verhalten während der Testläufe zwei Mal zu einem Fehlerfall, bei dem das Foto den Papierkorb unbeabsichtigt überlappte und daraufhin gelöscht wurde.

Eine mögliche Lösung zur Vermeidung unbeabsichtigter Löschung bei Überlappung des Papierkorbsymbols wäre eine Abstandsprüfung der Mittelpunkte von Foto und Symbol, wobei eine Löschung erst bei einem nahezu identischen Mittelpunkt erfolgt. Des Weiteren sollte die Darstellung des Fotos bei einer bevorstehenden Löschung weiter verbessert werden. Die implementierte Änderung der Rahmenfarbe des Fotos waren für die Probanden im Test offensichtlich nicht aus-

reichend. Ein über dem Foto halbtransparent eingeblendetes Symbol, welches den Löschvorgang darstellt, könnte hier Abhilfe schaffen.

### 7.3.5. Weitere Beobachtungen

Das Wort "seitenfüllend" innerhalb des Aufgabentextes war für mehrere Probanden missverständlich, daher war die von den Probanden gewählte Anordnung der Fotos oftmals eher steril, von der Rotation wurde nur wenig Gebrauch gemacht. Der Text hätte die Rotation der Fotos stärker begünstigen müssen um die Probleme der Rotation noch intensiver testen zu können. Aus diesem Grund wurde während der Testläufe vermehrt der Hinweis gegeben, die Fotos möglichst natürlich wirkend anzuordnen um die Probanden eher zur Nutzung der Rotation zu verleiten.

Das abschließende Feedback der Probanden war trotz der aufgetretenen Probleme größtenteils positiv. Die direkte Eingabe wurde von den Probanden die die Mauseingabe getestet haben bevorzugt. Konkrete Gründe konnten für diese Präferenz jedoch mit Ausnahme der "natürlich" sowie "interessanter" wirkenden Eingabe nicht genannt werden.

Die Bevorzugung direkter Eingabe kann unterschiedliche Gründe haben. So kann bereits die Tatsache, dass die Touch- sowie Multitoucheingabe für viele Anwender nach wie vor ein Novum darstellt, einen Anreiz für die Nutzung dieser Eingabe bieten. Des Weiteren wirkt eine direkte Interaktion trotz eventueller Beschränkung auf Singletouch eventuell intuitiver und damit für den Anwender einladender, auch wenn eine Mauseingabe effizienter für die auszuführende Aufgabe wäre. Interessant wäre in diesem Zusammenhang, in wie weit sich das Meinungsbild von Anwendern direkter Eingabemethoden bei längerer Nutzung ändert.

### 7.3.6. Fazit der Auswertung

#### Eingabeerkennung und Algorithmen

Das Hauptproblem der eingesetzten Hardware, die Erkennung von Druckpunkten während der Überlappung von Achswerten, konnte für die Praxis ansatzweise gelöst werden. Die hohe Eingabegeschwindigkeit der Anwender sowie die niedrige Datenrate des 42"-Rahmens ermöglichten während des Usability-Tests jedoch keine ausreichende Erkennung von Fingern deren Achswerte sich zeitweilig überlappten.

Um die Erkennung eindeutiger Druckpunkte für den 42"-Rahmen zu verbessern, müsste Die Abtast- sowie Datenrate der Hardware mindestens identisch zur Datenrate des 17"-Rahmens sein. Je höher die Abtastrate ausfällt, desto exakter können Anfang und Ende einer Überlappung mit Hilfe des Algorithmus nachverfolgt werden. Eine exaktere Druckpunkterkennung wiederum würde

die Zuverlässigkeit der Erkennung von Gesten erhöhen, welche auf der unabhängigen Bewegung zweier Finger aufbauen.

Die große Mehrheit der Testprobanden löste die vom Algorithmus nicht ausgleichbaren Problemfälle (siehe Abschnitt [7.3.2 auf Seite 78](#)) aus, wodurch eine sichere Erkennung der anschließenden Fingerbewegungen verhindert wurde. Diese Erkennungsprobleme lassen sich durch eine Erhöhung der Datenrate nicht beseitigen. Es wären stattdessen weitreichende Änderungen am Konzept der Hardware notwendig. So müsste eine Beobachtung von Druckpunkten aus beliebigen Winkeln vom Rahmen aus erfolgen. Ein möglicher Ansatz hierfür wäre die Kombination mit kamerabasierten Systemen. Dies wäre jedoch mit einem erhöhten Aufwand für die gemeinsame Verarbeitung und Analyse von Infrarot- und Bilddaten verbunden (siehe Abschnitt [5.4.8 auf Seite 53](#)). Bei Beibehaltung des Hardwarekonzepts müssten Anwender vor der Benutzung des Systems eine Einweisung in die für die Eingabe relevanten Hardwareeigenschaften erhalten, um die zuvor beschriebenen Problemfälle während der Eingabe zu vermeiden.

Die Erkennung von Objektarten war für das gewählte Gestenrepertoire ausreichend. Die Unterscheidung, zwischen einem und mehreren Fingern sowie einer Handkante funktionierte im Usability-Test fehlerfrei und führte während des Tests zu keinen sichtbaren Problemen. Für eine begrenzte Anzahl unterschiedlicher Objektarten kann daher davon ausgegangen werden, dass die Objekterkennung anhand charakteristischer Größen oder dem Seitenverhältnis in der Praxis verlässlich arbeitet (siehe auch [4.3.1 auf Seite 34](#) sowie [4.4.6 auf Seite 42](#)).

### **Testanwendung**

Da die Anlehnung der Testanwendung an ein reales Fotobuch nicht vollständig von den Testprobanden nachvollzogen werden konnte, würden sich zwei Änderungen für eine Weiterentwicklung der Testanwendung anbieten:

Um die Unterscheidung zwischen losen und auf einer Seite fixierten Fotos für den Anwender zu vereinfachen, sollten sich die beiden Zustände visuell stärker voneinander abheben. Ein Wechsel zwischen den Modi sollte zudem mit einer entsprechenden Animation kenntlich gemacht werden, falls dafür eine längere Eingabe vonnöten ist.

Des Weiteren sollte ein Wechsel des konzeptionellen Modells für die Zustände der Fotos erfolgen. Direkt auf dem Fotobuch positionierte Fotos sollten grundsätzlich sofort auf einer der angezeigten Seiten fixiert werden. Lose Fotos existieren bei diesem Konzept nur noch während der Bearbeitung des Fotos oder, bei einer Änderung des Oberflächendesigns, bei einer Ablage außerhalb Fotobuchs. Diese Änderung würde sowohl die Erwartungskonformität in Bezug auf Anwender anderer computergestützter Grafiksysteme wie beispielsweise CAD-Anwendungen erhöhen, als auch die Mehrdeutigkeit der Fixierungs- sowie Rotationsgeste vermeiden.

## 8. Fazit

In dieser Arbeit wurde eine Infrarotgitter-basierte Touchhardware evaluiert. Ziel war es zu überprüfen, in wie weit die gewählte Hardware die in der Vision angesprochene direkte und natürlich wirkende Interaktion innerhalb eines praxisnahen Szenarios ermöglicht.

Es erfolgte eine Analyse der Hardware in Bezug auf ihr Verhalten während der Benutzereingabe gefolgt von einer Implementation eines Systems, welches die für das zuvor festgelegte Gestenrepertoire benötigten Eingabeparameter liefert. Hierfür wurden unter Anderem die in Kapitel 5 beschriebenen Verfahren genutzt, insbesondere um die Grundlage für Gesten auf der Basis zweier Finger zu schaffen.

Mit Hilfe der zuvor erstellten Komponenten wurde eine Testanwendung aufgebaut, welche ein virtuelles Fotobuch darstellt, auf dem der Benutzer Fotos beliebig anordnen und fixieren kann. Die Testanwendung wurde anschließend innerhalb des in Kapitel 7 beschriebenen Usability-Tests auf ihre Praxistauglichkeit sowie auf die zuvor genannten Eigenschaften der Vision hin untersucht.

### 8.1. Direkte Interaktion mittels Multitouch

Die positiven Rückmeldungen der Probanden während des Usability-Tests zeigen, dass der Einsatz einer direkten Interaktion als Ersatz für eine mausbasierte Eingabe für das gewählte Szenario erfolgversprechend erscheint. Eine intuitive Interaktion per Multitoucheingabe erscheint daher generell trotz eventueller Hardwaredefizite möglich zu sein. Es ist jedoch darauf zu achten, dass jede aktuell verfügbare Multitouch-Technik, einschließlich der hier Evaluierten, ihre Stärken und Schwächen hat. Für den jeweiligen Einsatzzweck sollte zuvor geklärt werden, welche Anforderungen von der Hardware erfüllt werden müssen, um für den Anwender eine intuitive Eingabe zu ermöglichen.

Gesten die über die Nutzung eines einzelnen Fingers hinausgehen, können für den Anwender natürlicher und daher intuitiver wirken als eine Mauseingabe. Es muss dabei sichergestellt werden, dass die Geste der Erwartungshaltung des Anwenders genügt und ihm die Geste im aktuellen Kontext plausibel erscheint. Dies kann zum Einen dadurch erreicht werden, dass die Anwendung einem bereits bekannten System nachempfunden ist, wodurch gegebenenfalls ein bestehendes mentales Modell beim Nutzer angesprochen werden kann. Der Nutzer kann daraufhin sein vor-

handenes Wissen auf die Bedienung der Anwendung übertragen. Falls die Erwartungshaltung durch eine vorherige Computernutzung geprägt ist, kann es unter Umständen auch sinnvoll sein, die Toucheingabe durch mausähnliche Gesten wie beispielsweise doppeltes Antippen zu ermöglichen. Da im Usability-Test erkennbar war (siehe Abschnitt 7.3.3 auf Seite 79), dass verschiedene Anwender unterschiedliche Erwartungen sowie Vorkenntnisse besitzen, kann es daher für eine universell einsetzbare Multitoucheingabe sinnvoll sein, für eine Aktion mehrere Gesten zuzulassen, um den unterschiedlichen Anwendern und ihrem jeweiligen Vorwissen gerecht zu werden.

Das visuelle Feedback des Systems stellt eine weitere wichtige Eigenschaft dar, die in direktem Zusammenhang mit der Erwartungshaltung des Anwenders steht. Bei einer direkten Interaktion ist ein permanentes Feedback für den Anwender von Vorteil, da er so schneller ein Verständnis für das System entwickeln kann, als wenn das System während der Ausführung einer Geste keine Rückmeldungen gäbe. Für die Implementation von Gesten sowie einer direkten Manipulationen von Objekten per Multitouch sollte daher bereits während der Ausführung einer Geste eine visuelle Andeutung der am Ende ausgeführten Aktion gegeben werden. Für die Testanwendung bedeutet dies beispielsweise, dass ein Seitenwechsel bereits bei der ersten Berührung einer Seitenecke visuell angekündigt werden sollte.

## 8.2. Ausblick

Als Abschluss dieser Arbeit wird an dieser Stelle ein kurzer Ausblick auf weitere Entwicklungen im Bereich der infrarotbasierten Multitoucheingabe gegeben, sowie eine Einschätzung der zukünftigen Herausforderungen des Ubiquitous Computing und der Rolle natürlicher Interaktion innerhalb dieses Forschungsfeldes.

**Infrarotbasierter Multitouch** Das in dieser Arbeit beschriebene System könnte bei einer Weiterentwicklung von einem Konzept profitieren, welches aus dem Bereich des "Active Vision" entlehnt ist. Die Interpretation der Eingabedaten könnte hierbei in Abhängigkeit der momentan im Anwendungskontext gültigen Gesten erfolgen. Irrelevante, da im aktuellen Eingabekontext nicht zulässige, Bewegungen und Gesten könnten somit frühzeitig während der Interpretation der Eingabedaten ignoriert werden. So könnte beispielsweise die Rotation in Abhängigkeit des aktuellen Eingabekontexts nur in eine Richtung möglich sein, die Richtung der Drehung wäre für diesen Fall unerheblich und könnte daher während der Interpretation der Eingabedaten vernachlässigt werden. Die Herausforderung bei diesem Ansatz wäre die Einführung eines Feedbacks von der Anwendungsebene aus, welches der üblicherweise starken Trennung zwischen Treiberebene und Anwendungsebene entgegen läuft.

Die Einführung neuer Hardwarekonzepte auf Basis von Infrarot wie dem von der Firma Citron entwickelten und kurz vor Fertigstellung dieser Arbeit vorgestellten DreamTouch (siehe Cit [2009]) verspricht für die nahe Zukunft eine verbesserte Erkennung mehrerer Objekte unter Beibehaltung des Formfaktors der Hardware. Weiterhin wird aktuell an der HAW ebenfalls eine multitouchfähige Hardware auf Infrarotbasis entwickelt, deren theoretische Möglichkeiten bei der Erkennung und Reaktionsgeschwindigkeit weit über denen der in dieser Arbeit verwendeten Hardware liegen. Für zukünftige Entwicklungen wäre zu evaluieren, welche der in dieser Arbeit diskutierten Hardwareeigenschaften bei neueren Hardwarekonzepten weiterhin zutreffen und ob die in dieser Arbeit dokumentierten Verfahren eine Verbesserung der Eingabeerkennung ermöglichen.

**Multitouch & Ubiquitous Computing** Das Thema Ubiquitous Computing gewinnt aufgrund der zunehmenden Verbreitung computergestützter Systeme im alltäglichen Leben immer mehr an Bedeutung. Computersysteme sind mittlerweile allgegenwärtig und begleiten den Menschen in allen Lebenslagen. Unter dieser Voraussetzung spielt das Design der Mensch-Maschine-Interaktion bei diesen Systemen verstärkt eine Rolle, da im Gegensatz zu normalen Computerarbeitsplätzen die Interaktion mit allgegenwärtigen Systemen nicht unbedingt mit der gesamten Aufmerksamkeit des Benutzers erfolgt, das System wird stattdessen "nebenbei" bedient.

Die Multitoucheingabe stellt neben der sprachbasierten sowie räumlich gestenbasierten Interaktion eine Eingabeform dar, die Ubiquitous Computing ermöglicht. Allen genannten Eingabeformen ist gemeinsam, dass sie neue Freiheiten für den Anwender bieten, die Verarbeitung der Eingabedaten gleichzeitig jedoch komplexer als eine herkömmliche Mauseingabe ist. Die Herausforderung in der Zukunft liegt daher darin, die Komplexität dieser neuen Eingabeformen sowohl für den Anwender als auch für Entwickler handhabbar zu machen. Um dies zu erreichen bedarf es auf vielen Ebenen weiterer Forschung und Entwicklung. Zum Einen kann die Definition einheitlicher Begrifflichkeiten, wie beispielsweise das von George [2009] vorgestellte Konzept von Objekten, Containern, Gesten und Manipulationen (OCGM) als Basis für den weiteren Diskurs dienen. Zum Anderen sollten allgemeingültige Gestenrepertoires erarbeitet werden, welche sich in möglichst vielen Szenarien einsetzen lassen. Auch die mögliche Übertragung realer Gesten in virtuelle Arbeitsabläufe bietet sich weiterhin an. Die in dieser Arbeit zusammengetragenen Gesten zeigen nur einen kleinen Ausschnitt möglicher Bewegungsmuster, die sich für eine natürlich wirkende Interaktion eignen könnten. Weiterhin wäre eine Festlegung einheitlicher visueller Elemente, welche sich mit Hilfe von Gesten aus einem Gestenrepertoire manipulieren lassen, denkbar um als Richtlinie für die Entwicklung praktischer Anwendungen zu dienen.

Auch die Kombination von Spracherkennung, Multitoucheingabe sowie einer Gestenerkennung im Raum stellt eine interessante Erweiterung der Eingabemöglichkeiten des Anwenders dar und

könnte gleichzeitig die Auflösung eventueller Doppeldeutigkeiten während der Eingabe durch eine Kombination mehrerer Eingabemodalitäten ermöglichen. Die Möglichkeiten dieser multimodalen Interaktion sowie der Auflösung von Mehrdeutigkeiten sind nach wie vor wenig in praxisnahen Szenarien zum Einsatz gekommen, es besteht also auch hier viel Freiraum für weitere Arbeiten.

Weiterhin könnte die Kombination der Multitoucheingabe mit weiteren Modalitäten wie beispielsweise Sensoren für eine Ortung von Anwendern in der Umgebung von Geräten sowie ihrer momentanen Ausrichtung zu diesen Geräten einen weiteren Schritt sowohl zu sogenannten Disappearing Interfaces als auch zu Ambient Computing im Allgemeinen darstellen.

Ein Aspekt der in dieser Arbeit bewusst ausgeklammert wurde, stellt die explizite Wahl einer untrainierten Zielgruppe dar. Für Kinder sowie Jugendliche ist das Erlernen neuartiger Interaktionsmöglichkeiten oftmals keinerlei Hindernis, wie sich bereits am Erfolg von Eingabegeräten wie dem digitalen Joypad in den 1980er Jahren sowie der Wii Remote zeigt. Es wäre zu prüfen ob und wie sich vollkommen neuartige Interaktionsformen ohne Rückgriffe auf bestehende Paradigmen, Gesten oder Visualisierungen für diese eher selten beachtete Zielgruppe eignen könnten.



# Literaturverzeichnis

- [Irt ] *IRTOUCH Systems Co. Ltd.* Website. <http://www.irtouch.com>, Abruf: 2010-02-17
- [Mic ] *Microsoft Surface Website.* <http://www.microsoft.com/surface/>, Abruf: 2010-02-19
- [Per ] *Perceptive Pixel Website.* <http://www.perceptivepixel.com>, Abruf: 2010-02-19
- [Qt ] *Qt cross-platform application and UI framework.* <http://qt.nokia.com>, Abruf: 2010-02-19
- [Rxt ] *RXTX serial and parallel I/O libraries supporting Sun's CommAPI.* <http://www.rxtx.org>,  
Abruf: 2010-02-19
- [Spa ] *Sparsh-Ui.* <http://code.google.com/p/sparsh-ui/>, Abruf: 2010-02-19
- [Sta ] *Stantum - Unlimited Multi-touch.* <http://www.stantum.com/spip.php?article75>, Ab-  
ruf: 2010-02-17
- [ISO 1996] Norm DIN EN ISO 9241-10 1996. *Ergonomische Anforderungen für Bürotätigkeiten mit  
Bildschirmgeräten - Teil 10: Grundsätze der Dialoggestaltung*
- [Dvi 2003] *DViT (Digital-Vision Touch) Technology.* Website. <http://smarttech.com/DViT/>.  
Version: 2003, Abruf: 2010-02-08
- [NUI 2006] *Natural User Interface Group.* Website. <http://www.nuigroup.com>. Version: 2006,  
Abruf: 2010-02-19
- [Cit 2009] *Citron Homepage.* Website. <http://www.citron.de>. Version: 2009, Abruf: 2010-02-  
17
- [QtA 2009] *Qt 4.6 Reference Documentation.* Website. <http://qt.nokia.com/doc/4.6/>.  
Version: 2009, Abruf: 2010-02-05
- [Win 2010] *Windows Touch API.* Website. [http://msdn.microsoft.com/en-us/library/  
dd562197%28VS.85%29.aspx](http://msdn.microsoft.com/en-us/library/dd562197%28VS.85%29.aspx). Version: 2010, Abruf: 2010-02-05
- [Baudisch 2010] BAUDISCH, Patrick: *Patrick Baudisch's Homepage.* Website. [http://www.  
patrickbaudisch.com](http://www.patrickbaudisch.com). Version: 2010, Abruf: 2010-02-19
- [Benko u. a. 2006] BENKO, Hrvoje ; WILSON, Andrew D. ; BAUDISCH, Patrick: Precise selection  
techniques for multi-touch screens. In: *CHI '06: Proceedings of the SIGCHI conference on  
Human Factors in computing systems*, ACM Press, 2006. – ISBN 1–59593–372–7, S. 1263–  
1272

- [Chapweske 2003] CHAPWESKE, Adam: *The PS/2 Mouse Interface*. <http://www.computer-engineering.org/ps2mouse/>. Version: April 2003, Abruf: 2010-02-19
- [Davidson u. a. 1999] DAVIDSON, Mary J. ; DOVE, Laura ; WELTZ, Julie: *Mental Models and Usability*. Website. <http://www.lauradove.info/reports/mental%20models.htm>. Version: November 1999, Abruf: 2010-02-19
- [Dedre Gentner 1983] DEDRE GENTNER, Albert L. S.: *Mental Models*. Lawrence Erlbaum Associates, Inc., 1983. – ISBN 0–89859–242–9
- [Echtler u. a. 2008] ECHTLER, Florian ; HUBER, Manuel ; KLINKER, Gudrun: Shadow tracking on multi-touch tables. In: *AVI '08: Proceedings of the working conference on Advanced visual interfaces*. New York, NY, USA : ACM, 2008. – ISBN 1–978–60558–141–5, S. 388–391
- [Epps u. a. 2006] EPPS, Julien ; LICHMAN, Serge ; WU, Mike: A study of hand shape use in tabletop gesture interaction. In: *CHI '06: CHI '06 extended abstracts on Human factors in computing systems*. New York, NY, USA : ACM, 2006. – ISBN 1–59593–298–4, S. 748–753
- [Forsberg u. a. 1997] FORSBERG, A.S. ; LAVIOLA, Jr. J.J. ; MARKOSIAN, L. ; ZELEZNIK, R.C.: Seamless interaction in virtual reality. In: *Computer Graphics and Applications, IEEE* 17 (1997), November, Nr. 6, S. 6–9. <http://dx.doi.org/10.1109/38.626956>. – DOI 10.1109/38.626956. – ISSN 0272–1716
- [George 2009] GEORGE, Ron: *OCGM is the replacement for WIMP*. Website. <http://blog.rongear.com/design/ocgm-pronounced-occams-razor-is-the-replacement-for-wimp/>. Version: Dezember 2009, Abruf: 2010-02-19
- [Hewett u. a. 1997] HEWETT, Thomas T. ; BAECKER, Ronald ; CARD, Stuart ; CAREY, Tom ; GASEN, Jean ; MANTEI, Marilyn ; PERLMAN, Gary ; STRONG, Gary ; VERPLANK, William: *Curricula for Human-Computer Interaction*. <http://old.sigchi.org/cdg/>. Version: April 1997, Abruf: 2010-02-19
- [Ishii 1997] ISHII, Hiroshi: *Tangible Media Group*. Website. <http://tangible.media.mit.edu>. Version: 1997, Abruf: 2010-02-19
- [Ishii u. a. 1994] ISHII, Hiroshi ; KOBAYASHI, Minoru ; ARITA, Kazuho: Iterative design of seamless collaboration media. In: *Commun. ACM* 37 (1994), Nr. 8, S. 83–97. <http://dx.doi.org/10.1145/179606.179687>. – DOI 10.1145/179606.179687. – ISSN 0001–0782
- [Kela u. a. 2006] KELA, Juha ; KORPIPÄÄ, Panu ; MÄNTYJÄRVI, Jani ; KALLIO, Sanna ; SAVINO, Giuseppe ; JOZZO, Luca ; MARCA, Di: Accelerometer-based gesture control for a design environment. In: *Personal Ubiquitous Comput.* 10 (2006), Nr. 5, S. 285–299. <http://dx.doi.org/10.1007/s00779-005-0033-8>. – DOI 10.1007/s00779-005-0033-8. – ISSN 1617–4909
- [Kim u. a. 2007] KIM, Jangwoon ; PARK, Jaewan ; KIM, HyungKwan ; LEE, Chilwoo: HCI(Human Computer Interaction) Using Multi-touch Tabletop Display. In: *Communications, Computers and*

- Signal Processing, 2007. PacRim 2007. IEEE Pacific Rim Conference on.* Victoria, BC, Canada, : IEEE Computer Society, August 2007. – ISBN 978–1–4244–1189–4, S. 391–394
- [Koskinen u. a. 2008] KOSKINEN, Hanna Maria K. ; LAARNI, Jari O. ; HONKAMAA, Petri M.: Hands-on the process control: users preferences and associations on hand movements. In: *CHI '08: CHI '08 extended abstracts on Human factors in computing systems.* New York, NY, USA : ACM, 2008. – ISBN 978–1–60558–012–X, S. 3063–3068
- [Kruger u. a. 2005] KRUGER, Russell ; CARPENDALE, Sheelagh ; SCOTT, Stacey D. ; TANG, Anthony: Fluid integration of rotation and translation. In: *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems.* New York, NY, USA : ACM, 2005. – ISBN 1–58113–998–5, S. 601–610
- [Liu u. a. 2006] LIU, Jun ; PINELLE, David ; SALLAM, Samer ; SUBRAMANIAN, Sriram ; GUTWIN, Carl: TNT: improved rotation and translation on digital tables. In: *GI '06: Proceedings of Graphics Interface 2006.* Toronto, Ont., Canada, Canada : Canadian Information Processing Society, 2006. – ISBN 1–56881–308–2, S. 25–32
- [NUI-Group 2009] NUI-GROUP: *Multi-Touch Technologies.* Website. [http://nuicode.com/attachments/download/115/Multi-Touch\\_Technologies\\_v1.01.pdf](http://nuicode.com/attachments/download/115/Multi-Touch_Technologies_v1.01.pdf). Version: Mai 2009, Abruf: 2010-02-19
- [Potter u. a. 1988] POTTER, R. L. ; WELDON, L. J. ; SHNEIDERMAN, B.: Improving the accuracy of touch screens: an experimental evaluation of three strategies. In: *CHI '88: Proceedings of the SIGCHI conference on Human factors in computing systems,* ACM Press, 1988. – ISBN 0–201–14237–6, S. 27–32
- [Rogers u. a. 1992] ROGERS, Yvonne ; RUTHERFORD, Andrew ; BIBBY, Peter A.: *Models in the Mind: Theory, Perspective and Application.* Academic Press Inc., 1992. – 330 S. – ISBN 0125929706
- [Sander u. a. ] SANDER, Stefan ; POLICHT, Michal ; FOSDICK, Brandon: *QextSerialPort.* Website. <http://code.google.com/p/qextserialport/>, Abruf: 2010-02-19
- [Wang u. a. 2008] WANG, Feng ; REN, Xiangshi ; LIU, Zhen: A Robust Blob Recognition and Tracking Method in Vision-Based Multi-touch Technique. In: *Parallel and Distributed Processing with Applications, 2008. ISPA '08. International Symposium on,* 2008, S. 971–974
- [Wobbrock u. a. 2009] WOB BROCK, Jacob O. ; MORRIS, Meredith R. ; WILSON, Andrew D.: User-defined gestures for surface computing. In: *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems.* New York, NY, USA : ACM, 2009. – ISBN 978–1–60558–246–7, S. 1083–1092
- [Wu u. Balakrishnan 2003] WU, Mike ; BALAKRISHNAN, Ravin: Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. In: *UIST '03: Proceedings of the 16th annual ACM symposium on User interface software and technology,* ACM Press, 2003. – ISBN 1–58113–636–6, S. 193–202

## A. State-Machine des Serial Readers

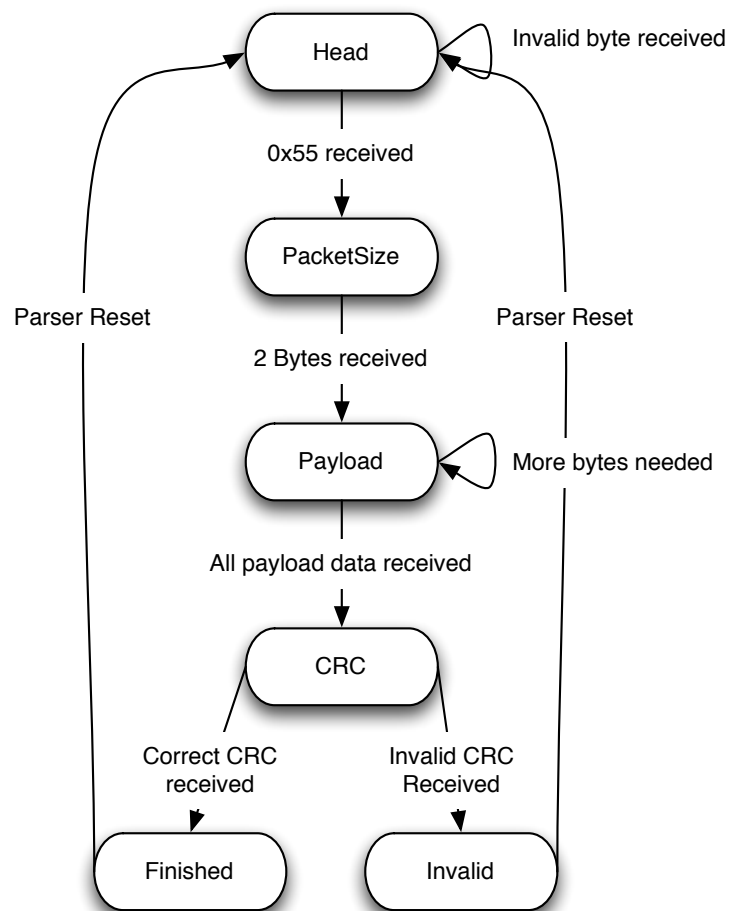


Abbildung A.1.: State-Machine des Serial Readers

## B. Gestenrepertoire für den Usability-Test

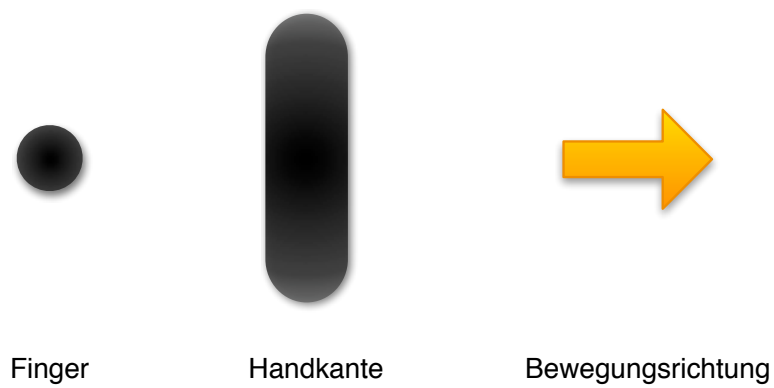


Abbildung B.1.: Legende

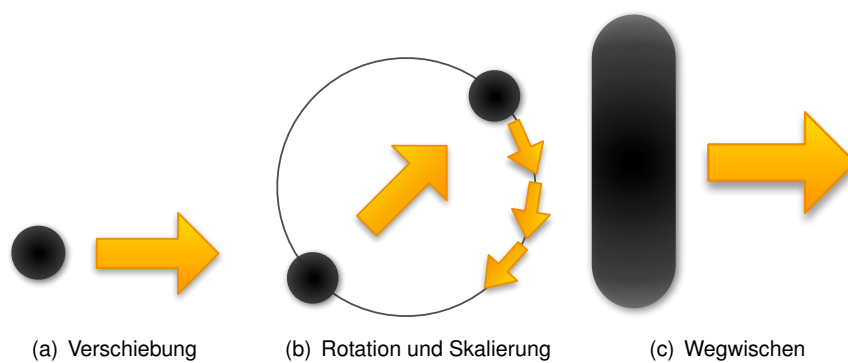


Abbildung B.2.: Unterstütztes Gestenrepertoire

## **C. Aufgaben für den Usability-Test**

1. Legen Sie vier Fotos auf den ersten beiden Fotobuchseiten ab
2. Ordnen Sie die vier Fotos möglichst seitenfüllend per Rotation und Skalierung an
3. Leeren Sie eine der beiden Fotobuchseiten
4. Legen Sie eine dritte Fotobuchseite an
5. Legen Sie ein Foto auf der dritten Seite ab
6. Wechseln Sie zurück auf die ersten beiden Seiten

## D. Fragebogen zum Usability-Test

### Fragen vor dem Test

**Nutzen Sie Computersysteme mit (Single-)Toucheingabe?**

- Ja, regelmäßig
- Ja, gelegentlich
- Nein

**Nutzen Sie Computersysteme mit Multitouch-Eingabe?**

- Ja, regelmäßig
- Ja, gelegentlich
- Nein

### Fragen nach dem Test

- Was fanden Sie gut?
- Was fanden Sie weniger gelungen?
- Einschätzung des praktischen Nutzens von Multitouch für eine Anwendung wie das Foto-  
buch
- Eigene Vorstellung von Gesten für vorgegebene Aktionen (falls Erwartungen innerhalb des  
Tests offensichtlich nicht erfüllt wurden, Hinweis auf unpassende Wahl der Gesten)

*Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §22(4) bzw. §24(4) ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.*

Hamburg, 24. Februar 2010 Stefan Gehn