

Institut für Visualisierung und Interaktive Systeme

Universität Stuttgart
Pfaffenwaldring 5a
D-70569 Stuttgart

Fachstudie Nr. 195

Performance Analyse zweier Toolkits für interaktive projizierte Displays

Dieter Daniel Türk, Merve Yildiz, Manuel Lorenz

Studiengang:	Softwaretechnik
Prüfer/in:	Prof. Dr. Albrecht Schmidt
Betreuer/in:	M.A. Oliver Korn, Dipl.-Inf. Markus Funk
Beginn am:	4. April 2014
Beendet am:	4. Oktober 2014
CR-Nummer:	H.5.2.

Kurzfassung

Mit der Microsoft Kinect wurden Tiefenkameras zu günstigen Preisen für die Allgemeinheit verfügbar. Kombiniert man eine Kinect mit einem Beamer, kann man interaktive projizierte Displays bauen. Das Beste daran ist, dass man dafür keine fest installierten Displays braucht. So können Displays aus jeder beliebigen Oberfläche, egal welcher Form und Struktur, einen Multitouch-Display erzeugen. Zurzeit werden Toolkits entwickelt, die eine bequeme Möglichkeit bieten, interaktive Inhalte für diese Displays bereitzustellen. Aus diesem Grund werden in dieser Fachstudie zwei solche Toolkits untersucht und miteinander verglichen. In einer Nutzerstudie werden mittels eines Fitts' Law Tasks grundlegende Performanz Indikatoren erfasst und ausgewertet. Bei der Auswertung wird nicht nur die Performanz der Toolkits gegenübergestellt, sondern die Ergebnisse werden auch mit der subjektiven Beurteilung der Studienteilnehmer verglichen. Ziel der Fachstudie ist es, jeweils die Vor- und Nachteile beider Toolkits darzustellen.

Abstract

With the release of the Microsoft Kinect, depth cameras became available at reasonable prices to the public. Interactive projected displays can be build by combining a Kinect with a projector and transform any surface no matter what form and structure into a multi touch display. The best thing about it, is the fact that no installed display is needed. Currently, toolkits are developed to provide a convenient way to provide interactive content for these displays. This study paper examines and compares the performance of two such toolkits against each other. Therefore a Fitts' Law task is implemented to help collecting basic performance indicators through a user study. Not only the comparison of the performance of the toolkits is evaluated, but also the subjective participant opinion to each toolkit. The aim of this study paper is to respectively present the advantages and disadvantages of both toolkits.

Inhaltsverzeichnis

1	Einleitung	7
1.1	motionEAP	7
1.2	Ziel der Fachstudie	8
1.3	Inhalt des Dokuments	9
2	Themenverwandte Arbeiten	11
2.1	Tiefenkameras als Touchsensor	11
2.2	The Everywhere Displays	12
2.3	dSensingNI	13
2.4	TouchLight	14
2.5	OmniTouch	15
2.6	AMP-D	15
2.7	UbiDisplays	16
2.8	Fitts' Law	17
3	System	19
3.1	Hardware & Software	19
3.2	Toolkits	20
3.3	Testanwendung	21
4	Nutzerstudie	25
4.1	Design	25
4.2	Apparatur	25
4.3	Teilnehmer	26
4.4	Vorgehen	26
4.5	Ergebnis	27
4.6	Auswertung	29
5	Fazit und Ausblick	33
6	Begriffslexikon	35
	Literaturverzeichnis	39

Abbildungsverzeichnis

1.1	Überblick motionEAP. Quelle: [FKS14b]	8
2.1	Grenzbereiche $d_{(min)}$ und $d_{(max)}$ werden verwendet um Touch Events bei Berühren der Fläche dsurface voneinander zu unterscheiden. Quelle: [Wil10]	11
2.2	Aufbau des Everyday Display projector. Mittels eines drehbaren Spiegels werden die Inhalte im Raum verteilt. Quelle: [Pin01]	12
2.3	Aufbau einer Tiefenkamera mit einer Interaktionsfläche für die Verwendung mit dSensingNI. Quelle: [KNF12]	13
2.4	TouchLight System bestehend aus einem DNP HoloScreen, zwei Kameras, einen Projektor und die dazugehörigen Filter. Quelle: [Wil04]	14
2.5	Steuerung eines Zahlenfeldes mit Omnitouch. Quelle: [HBW11]	15
2.6	Aufbau des Ambient Mobile Pervasive Display Prototyps. Quelle: [WSDR14]	16
2.7	Die Oberfläche des UbiDisplays Toolkits bei der Kalibrierung der Interaktionsfläche. Quelle: [HA12]	16
3.1	Die Abbildung zeigt den Aufbau einer Microsoft Kinect Tiefenkamera [Mic14]	20
3.2	Beispiel .csv eines Teilnehmers mit einem Spiel und 14 geklickten Kreisen.	22
3.3	Diagramm der Anwendungsarchitektur.	23
4.1	Die bei der Nutzerstudie verwendete zufällige Permutationsreihenfolge.	27
4.2	Ergebnisse der durchgeführten Studie.	28
4.3	Überblick der Auswertungsdiagramme aller Durchläufe.	30
4.4	Überblick der Auswertungsdiagramme für die subjektiven Eindrücke der Nutzer.	31

1 Einleitung

Die Interaktion zwischen Mensch und Computer erfolgte lange Zeit nur über die klassische Kombination aus Maus und Tastatur. Da Computer im Alltag eine wichtigere Rolle als je zuvor spielen, sind sie in vielen Bereichen gar nicht mehr wegzudenken. Das führt auch dazu, dass immer neue Interaktionstechnologien entwickelt werden. Betrachtet man die Entwicklungen der letzten 15 Jahre im Bereich Mensch-Computer-Interaktion stellt man fest, dass sehr viele Benutzungsschnittstellen und Formen der Interaktion sowohl im Privatbereich als auch in der Industrie dazu gekommen sind. Touchscreens beispielsweise haben die klassischen Displays die nur Informationen dargestellt haben revolutioniert. Egal ob im Smartphone, Fahrkartenautomat oder in der Supermarktkasse eingebaut, machen sie die Verwendung einer klassischen Tastatur überflüssig. Sie existieren, um die Interaktion zwischen Mensch und Computer im Alltag zu erleichtern. Der einzige Haken an dieser Technologie ist der, dass man dafür tatsächlich installierte Bildschirme benötigt und die Bedienoberfläche flach sein muss. Das kann vor allem in Anwendungsbereichen problematisch werden, in denen der Einbau solcher Hardware nicht möglich ist oder einfach nicht sinnvoll erscheint, da zum Beispiel die Oberfläche beschädigt werden könnte. Was wäre aber, wenn jede erdenkliche Oberfläche sich in einem Multi-Touch-Bildschirm verwandeln ließe? Diese Art von Displays sind bereits vorhanden und werden als interaktive projizierte Displays bezeichnet. Sie gehören zum Gebiet des Ubiquitären Computing, einen Begriff welcher von dem Wissenschaftler Mark Weiser [Wei91] geprägt wurde. Damit bezeichnet er die Allgegenwärtigkeit von Informationstechnik und Computer, die in prinzipiell alle Alltagsgegenstände von der industriellen Produktion bis in den privaten Alltag eindringen. Der Grund weshalb diese Technologien im Alltag noch nicht zu sehen sind liegt daran, dass sie sehr teuer in der Umsetzung und umständlich in der Installation sind. Sie sind nur auf bestimmte Anwendungsumgebungen zugeschnitten. Glücklicherweise werden zur Zeit Toolkits entwickelt, die in der Kombination mit einem Videoprojektor und einer Tiefenkamera (beispielsweise die Microsoft Kinect für Windows) die Möglichkeit bieten, interaktive Inhalte sehr einfach und günstig auf Oberflächen zu projizieren und damit zu interagieren. Auch wenn das Sichtfeld eingeschränkt und die Technologie nicht so präzise und effizient wie kapazitive Touchscreens ist, entstehen dennoch auch viele Vorteile. Der größte Vorteil ist, dass man keine Bildschirme mehr braucht. Jede Oberfläche, egal welcher Form und Struktur, kann in einen Multitouch-Display verwandelt werden. Mithilfe des Videoprojektors werden die Inhalte auf der Oberfläche dargestellt während die Tiefenkamera für die Erfassung der Berührungspunkte zuständig ist.

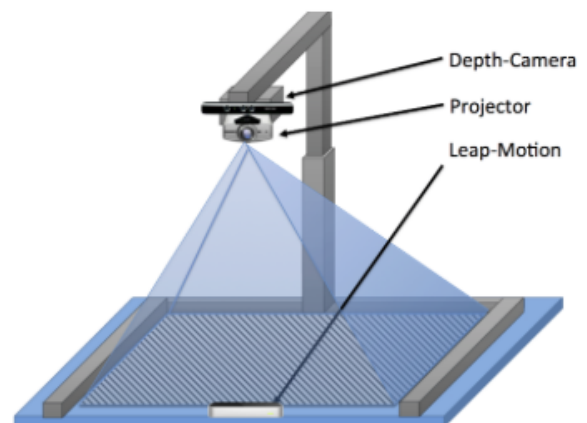
1.1 motionEAP

motionEAP [FKS14b] ist ein System zur Effizienzsteigerung und Assistenz von Produktionsprozessen in Unternehmen auf Basis von Bewegungserkennung und Projektion. Ziel des motionEAP [FKS14a]

ist die anwenderorientierte Konzeption, prototypische Umsetzung und Evaluation eines neuartigen prozessorientierten Assistenzsystems für Produktionsprozesse. Beispielsweise könnte das System dazu verwendet werden, klassische Arbeitsplatzsysteme (Abbildung: 1.1)in der Industrie mit sensorgestützten Bewegungserkennung für Echtzeit-Feedback, In-situ-Projektion und barrierefreies Informationsdesign aufzurüsten. Durch diese Hilfssysteme (Abbildung: 1.1) kann die Produktivität gesteigert werden und physisch oder psychisch eingeschränkte Arbeiter können komplexere Aufgaben erledigen. Die Anpassungsfähigkeit und die Umweltwahrnehmung des Systems erlauben das Entwickeln neuer Arbeitstechniken. Die Gamification [Kor12] des Arbeitsplatzes könnte beispielsweise die Mitarbeiter mit Echtzeitinformationen über die Qualität der gerade verrichteten Arbeit im Vergleich zu anderen Mitarbeiter versorgen. Das sofortige Feedback hilft die Qualität der verrichteten Arbeit und somit die Qualität des hergestellten Produkts zu verbessern.



(a) Arbeitsplatzsystem aus der Industrie



(b) Aufbau des verwendeten motionEAP Hilfssystems

Abbildung 1.1: Überblick motionEAP. Quelle: [FKS14b]

1.2 Ziel der Fachstudie

Ziel dieser Studie ist es zwei dieser Toolkits miteinander zu vergleichen um ihre Performanz auszuwerten. Dafür wird ein einfaches Fitts' Law Task entwickelt, welches bestimmte Performance-Indikatoren ermittelt die anschließend ausgewertet werden. Um die Transparenz zwischen den beiden Toolkits zu bewahren, wird der Fitts' Law Task in einer unabhängigen Software implementiert. Die Touch-Events werden durch eine Netzwerkkomponente von den jeweiligen Toolkits zum Task übertragen.

1.3 Inhalt des Dokuments

Der Inhalt dieser Fachstudie ist in fünf verschiedene Kapiteln gegliedert. Nach der Einführung im ersten Kapitel folgt im darauffolgenden ein Überblick über themenverwandte Arbeiten. Im dritten Kapitel werden die Architektur, genauso wie alle für die Studie relevanten Spezifikation beschrieben. Das vierte Kapitel beinhaltet den Aufbau und Durchführung der Studie gefolgt von der Auswertung der Ergebnisse. Zum Schluss folgt noch das Fazit und es wird ein Ausblick auf zukünftige Entwicklungen gegeben.

2 Themenverwandte Arbeiten

Es gibt eine Vielzahl von Entwicklungen und Forschungsrichtungen die sich mit interaktiven projizierten Displays beschäftigen. Im folgenden Abschnitt werden einige wichtige Arbeiten auf diesem Gebiet nahe gebracht um einen Überblick darüber zu schaffen.

2.1 Tiefenkameras als Touchsensor

Andrew D. Wilson veröffentlichte in „Using a Depth Camera as a Touch Sensor“[Wil10] die Idee, eine Tiefenkamera zu verwenden um interaktive Displays zu projizieren. Als eines der wichtigsten Vorteile nennt er nicht nur die Tatsache, dass die Touchoberfläche ohne Touchsensoren auskommt, sondern sogar dass unebene Flächen als Touchscreens verwendet werden können. Außerdem bieten Tiefenkameras die Möglichkeit Körperteile wie Arme, Hände oder Finger auf unterschiedlicher Weise auszuwerten. Bestimmte Berührungen von einem oder unterschiedlichen Usern stammen werden beispielsweise auch erkannt und unterschieden. Zum Experimentieren wurde eine Microsoft Kinect-Kamera verwendet, welche über eine Tischplatte montiert wurde. Bei diesem System werden Touch Events erkannt, wenn ein Finger in einem zuvor festgelegten Wertebereich von oben „eintaucht“. Dabei muss der Finger einen festgelegten Grenzbereich überschreiten, sodass ein Touch Event ausgelöst wird. Bei den Versuchen wurden verschiedene Abstände zwischen Kamera und Tischplatte getestet. Man erlangte die Feststellung, dass die Anzahl der Tiefeninformationen bei ansteigender Entfernung zwischen Kamerasensor und Berührfläche abnimmt. Auch wenn diese Technologie nicht an die Performanz der meisten herkömmlichen Touch-Technologien reicht, ist sie doch reif genug um in vielen Bereichen eingesetzt werden zu können.

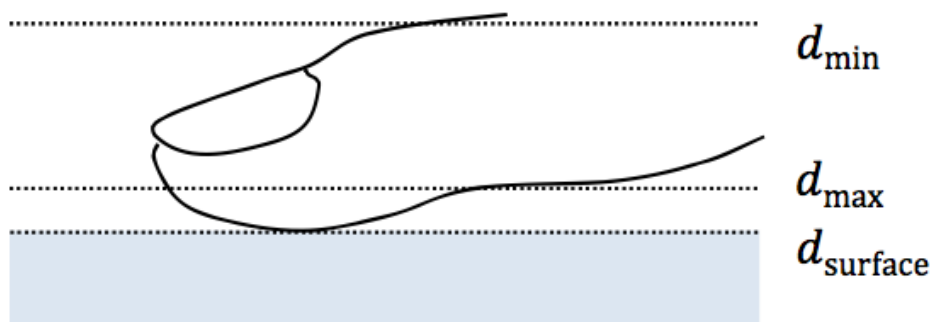


Abbildung 2.1: Grenzbereiche d_{\min} und d_{\max} werden verwendet um Touch Events bei Berührungen der Fläche d_{surface} voneinander zu unterscheiden. Quelle: [Wil10]

2.2 The Everywhere Displays

Claudio Pinhanez schaffte es in seinem Projekt „Everywhere Display projector [Pin01] mittels eines Projektors und eines computergesteuerten drehbaren Spiegels interaktive Inhalte in einem Raum zu projizieren. Bei der Überlegung wurde sehr viel Wert darauf gelegt, dass keine speziellen Geräte verwendet werden und die projizierten Displays sollten auch ohne Verkabelung auskommen. Das System erreichte eine Abdeckung von 60° vertikal und 230° horizontal und machte somit die Nutzung einer großen Raumfläche als Projektionsfläche möglich 2.2. Um Verzerrungen des Bildes zu vermeiden mussten die Flächen, auf denen der Inhalt dargestellt werden sollte vordefiniert werden. Anschließend wurde ein Muster auf die Fläche projiziert und mittels dieser Muster wurde die Verzerrung des Bildes vor der Projektion auf dem Computer entzerrt. So wurde sichergestellt, dass die darstellbaren Inhalte auch lesbar waren. Da der Projektor nicht rechtwinklig auf die projizierte Fläche ausgerichtet ist, geht durch die Entzerrung des Bildes jedoch ein großer Teil der Auflösung verloren. Die Idee hinter dem Projekt war dieses Gerät für neue Spielkonzepte zu nutzen, in denen die Spieler nicht mehr gezwungen sind klassische Eingabegeräte oder spezielle Brillen zu nutzen, sondern mit ihrer Umgebung interagieren können. Auch wenn dieses Konzept aufgrund der nötigen Kalibrierung der Verzerrung und der möglichen Unlesbarkeit durch unterschiedlicher Lichtverhältnissen im Raum leider nicht ausgereift ist, kann man dennoch hoffen, dass die Entwicklung leistungsfähigerer Projektionstechnologien dieses Konzept alltagstauglich macht.

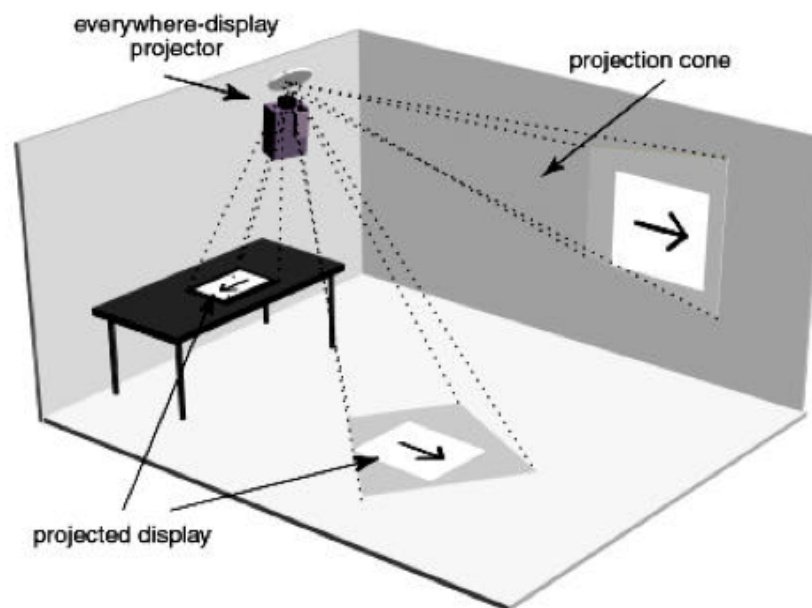


Abbildung 2.2: Aufbau des Everyday Display projector. Mittels eines drehbaren Spiegels werden die Inhalte im Raum verteilt. Quelle: [Pin01]

2.3 dSensingNI

Das Framework dSensingNI (Depth Sensing Natural Interaction) [KNF12] ermöglicht moderne Interaktionsmöglichkeiten mittels Multitouch und Tangible Interaction, also eine anfassbare Benutzerschnittstelle, die im Grunde die Bedienung einer Maschine durch physische Objekte erlaubt 2.3. Das Framework unterstützt im Zusammenhang mit einer Tiefenkamera das Erkennen und Verfolgen von Finger- und Handbewegungen. Zusätzlich erkennt dSensingNI das Packen, Stapeln und Gruppieren von Objekten. Das Ziel bei der Entwicklung von dSensingNI war es, Anwendungsentwicklern die Möglichkeit zu geben, ohne großen Aufwand Techniken für Tangible Interaction zu entwickeln. Um das zu ermöglichen, haben die Entwickler sehr hohe Anforderungen an das Framework gestellt. Neben dem Erkennen von Fingerberührungen auf willkürlichen Oberflächen in Echtzeit, sollte das Tool auch die Fähigkeit besitzen, Gesten über bzw. vor einem Bildschirm zu erkennen. Ferner sollte das Tool Umgebungsverdeckungen minimieren, um eine gute Erkennung von Interaktionen zu gewährleisten. Das Framework wurde als Standalone-Software entwickelt und ist für Windows-Betriebssysteme verfügbar. Das Framework ist in der Lage angeschlossene Tiefenkameras automatisch zu erkennen und sich damit zu verbinden. Der Benutzer kann dann das Tiefenbild nach seinem Wunsch anhand Parameteränderungen anpassen.



Abbildung 2.3: Aufbau einer Tiefenkamera mit einer Interaktionsfläche für die Verwendung mit dSensingNI. Quelle: [KNF12]

2.4 TouchLight

TouchLight [Wil04] ist ein System, welches mithilfe von zwei Kameras und einem Projektor ein interaktives Bild auf ein Stück transparentes Acrylglas projiziert. Dabei werden einfache Bildverarbeitungstechniken verwendet, um Bilder von zwei sich hinter einem DNP-Holoscreen befindlichen Kameras, die mit IR-Passfilter versehen sind, zu kombinieren, um es anschließend darauf zu projizieren. Das sonst durchsichtige Glas verwandelt sich dadurch in eine hochauflösende Oberfläche, die touchbasierte Interaktionen erlaubt. Um Berührungen erkennen zu können, nutzt TouchLight eine der Kameras, um Bilder von den mit der Oberfläche in Kontakt kommenden Objekten zu machen, um sie anschließend in Touch-Events zu verwandeln. Damit erlaubt TouchLight reale Umgebungen mit projizierten Grafiken zu kombinieren bzw. zu ergänzen. Ein mögliches Szenario wäre beispielsweise die Darstellung von Wetterinformationen (Temperatur, Windgeschwindigkeit, Regenwahrscheinlichkeit, etc.) auf einem Fenster. So kann man das echte Wetter draußen betrachten und bekommt gleichzeitig zusätzliche Informationen mit. Diese Möglichkeiten unterscheidet TouchLight von allen bisher bekannten Technologien dieser Art.

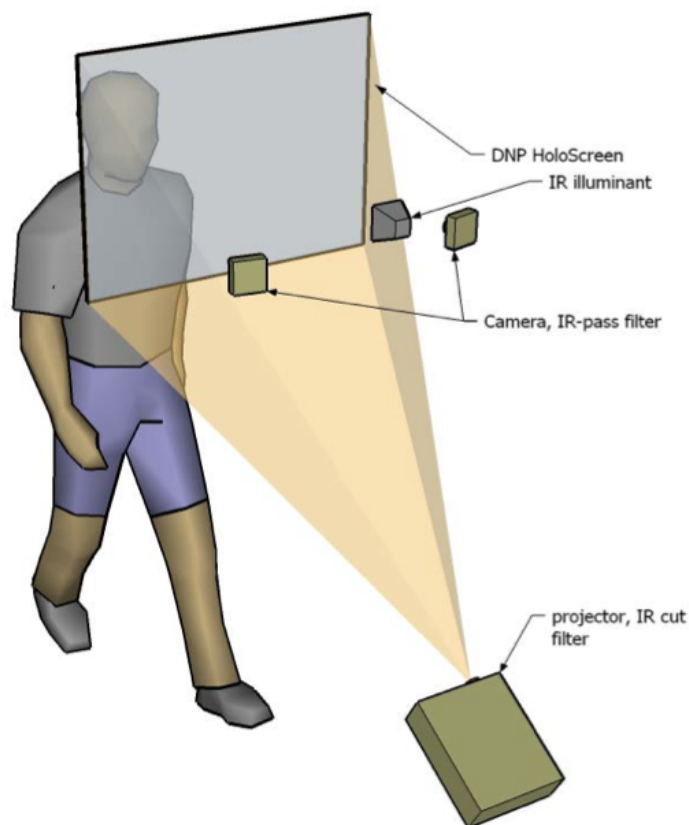


Abbildung 2.4: TouchLight System bestehend aus einem DNP HoloScreen, zwei Kameras, einen Projektor und die dazugehörigen Filter. Quelle: [Wil04]

2.5 OmniTouch

Ein futuristisches tragbares System das interaktive projizierte Inhalte auf willkürliche Oberflächen erlaubt ist das so genannte Omnitouch [HBW11] das von den Wissenschaftlern von Microsoft Research entwickelt wurde. Omnitouch ist ein Prototyp bestehend aus einer tragbaren Tiefensensorkamera der Firma PrimeSense und einem Microvision ShowWX+ Picoprojektor, die an einem Metallgerüst befestigt sind und auf der Schulter des Benutzers getragen werden. Das System erlaubt dem Benutzer Inhalte beispielsweise auf seiner Handfläche darzustellen und damit zu interagieren. Die Ansteuerung der darstellbaren Inhalte erfolgt wie bei einem herkömmlichen Touchscreen. Mittels X- und Y-Achsen auf einer 2D-Schnittstelle wird die Fingergeste entweder als „gedrückt“ oder als „schweben“ interpretiert. Auch wenn OmniTouch noch in den Kinderschuhen steckt, sind die Einsatzmöglichkeiten solcher Systeme nahezu grenzenlos und spätere Entwicklungen können dazu führen dass solche Systeme noch kompakter und leistungsfähiger werden.



Abbildung 2.5: Steuerung eines Zahlenfeldes mit Omnitouch. Quelle: [HBW11]

2.6 AMP-D

Der Prototyp „Ambient Mobile Pervasive Display“ (AMP-D) [WSDR14] von den Ulmer Doktoranden Christian Winkler ist ein tragbares System welches interaktive Inhalte nicht nur auf der Handfläche des Benutzers sondern auch auf dem Boden projiziert. Das System besteht aus einer Tiefenkamera und einem Orientierungssensor die an einer Teleskopstange befestigt und mit einem mobilen Endgerät verbunden sind (siehe Abbildung 2.5). Über eine Servosteuerung ist das System in der Lage automatisch auf die Handfläche oder auf dem Boden zu fokussieren. Im Vergleich zu anderen mobilen Geräten hat der Benutzer das Fenster zu der virtuellen Welt direkt vor sich und muss nicht beispielsweise erst das Smartphone aus der Tasche rausholen und einschalten. Als zukünftiges Ziel gilt es den Prototyp zu verkleinern und Gewicht zu sparen um die Mobilität zu verbessern.

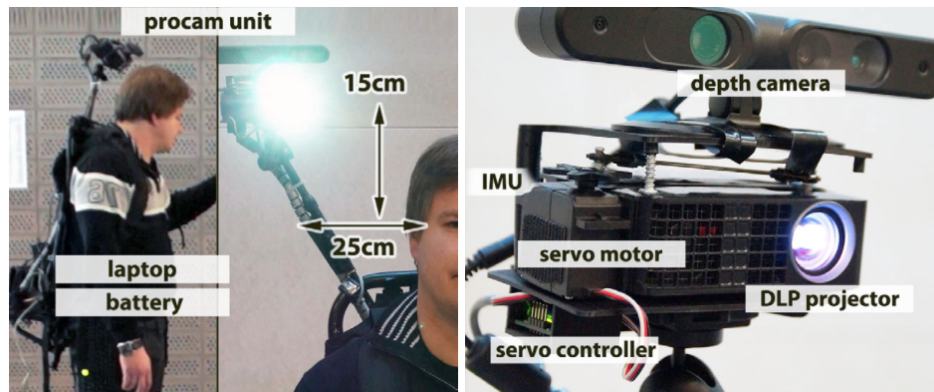


Abbildung 2.6: Aufbau des Ambient Mobile Pervasive Display Prototyps. Quelle: [WSDR14]

2.7 UbiDisplays

Unter dem Titel „Toolkit Support for Interactive Projected Displays“ [HA12] haben John Hardy und Jason Alexander das sogenannte UbiDisplays Toolkit veröffentlicht. Dieses Toolkit wurde dazu entwickelt um Benutzern die Möglichkeit zu geben interaktive Inhalte mit wenig Aufwand bereitzustellen. Das Open-Source-Toolkit bietet zusammen mit einem Projektor und einer Kinect die Möglichkeit Webinhalte im Raum zu projizieren und damit zu interagieren. Die Benutzeroberfläche erlaubt außerdem eine sehr einfache Kalibrierung der Tiefenkamera 2.7. So kann jeder, der über eine Kinect-Kamera und einen Projektor verfügt und Kenntnisse zum Bauen einer Webseite besitzt, einfach und kostengünstig interaktive projizierbare Displays herstellen. Zusätzlich kann man aus verschiedenen Interaktionsarten wählen und sogar – da das Toolkit Open-Source ist – eigene Interaktionsmethoden entwickeln.

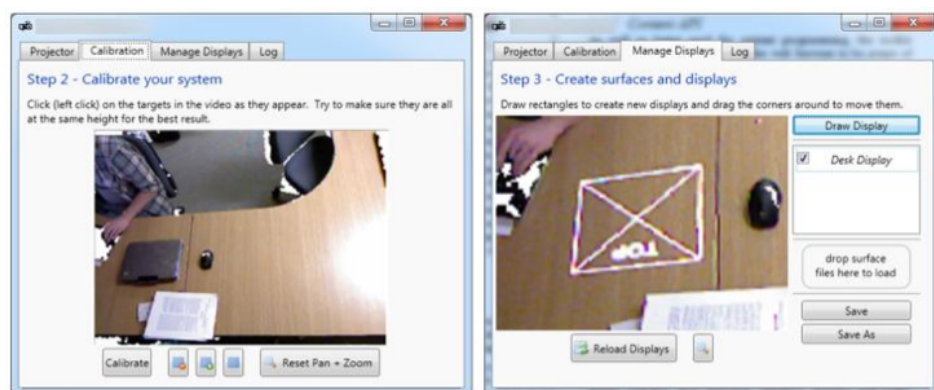


Abbildung 2.7: Die Oberfläche des UbiDisplays Toolkits bei der Kalibrierung der Interaktionsfläche. Quelle: [HA12]

2.8 Fitts' Law

Fitts' Law (Fitts' Gesetz, manchmal auch Fitts'sches Gesetz) [Fit54] spielt im Bereich Mensch-Computer-Interaktion eine wichtige Rolle. Das Gesetz wird dazu verwendet, um vorherzusagen, welche Geschwindigkeit für eine gezielte schnelle Bewegung von einem bestimmten Start zu einem bestimmten Ziel benötigt wird. In unserer Studie wird das Gesetz dazu verwendet, um die Geschwindigkeit beim Berühren der schwarzen Punkte auf der projizierten Fläche zu messen. Die benötigte Zeit für die Start-Ziel-Bewegung lässt sich als Funktion von Abstand und Zielgröße modellieren:

$$MT = a + b * \log(1 + \frac{A}{W})$$

MT (movement time):	für die Aufgabe benötigte Zeit
a (intercept):	Verzögerung bis die Bewegung beginnt
b (slope):	Beschleunigung und Gerätespezifische Änderung der Bewegung
A (amplitude):	Distanz vom Start und Ziel
W (width):	Breite des Ziels gemessen in der Bewegungsrichtung (bei uns spielt die Richtung keine Rolle, da die Ziele Kreise sind)

Der Aufbau der Formel kann so erklärt werden, dass große und nahe Ziele schneller erreicht werden als kleine und weite Ziele. Außerdem wird bei Fitts' Law der Term $\log(1 + \frac{A}{W})$ zu Index of Difficulty (ID) zusammengefasst. Der ID ist dazu da, um die Schwierigkeit einer Aufgabe zu werten. Der Wert der ID erhöht sich immer um 1, wenn die Amplitude A verdoppelt oder die Breite W halbiert wird.

3 System

Dieses Kapitel beschreibt das System das eingesetzt wurde. Neben der Testumgebung wird auch die Architektur sowie der Zugriff auf die beiden Toolkits erläutert. Des Weiteren wird die Testanwendung, die bei der Studie zum Einsatz kam, erklärt.

Im Vergleich zu den konventionellen Touch-Bildschirmen müssen interaktive projizierte Displays nicht fest installiert sein und die Berühroberfläche muss nicht zwangsläufig flach sein. Während der Videoprojektor für die Darstellung der Inhalte zuständig ist, hilft die Tiefenkamera mit der Erkennung und Verarbeitung der Touch-Events. Außerdem hat man dank der Tiefenkamera beispielsweise die Möglichkeit Informationen über die Form der Benutzerhand zu verarbeiten und kann somit erkennen ob gleichzeitige Berührungen von der einer oder mehreren Personen getätigt wurden. Im folgenden Abschnitt werden die benötigten Komponenten für die Durchführung der Studie dargestellt.

3.1 Hardware & Software

Im nachfolgenden Kapitel wird die verwendete Hard- und Software beschrieben. Das im hier zum Einsatz kommende motionEAP System ist im Kapitel 1.1 bereits erläutert. Hier wird nur der Aufbau des motionEAP Tisches erklärt.

3.1.1 Microsoft Kinect für Windows

Tiefenkameras wurden ursprünglich mit dem Ziel entwickelt, natürliche Interaktionen zwischen Mensch und Computer zu gewährleisten. Für den Nahbereich sind Modelle von Herstellern wie PrimeSense, Microsoft und Asus verfügbar. Alle sind nahezu baugleich und basieren auf der PrimeSense 3D-Sensor-Technologie. Zu den Kamerakomponenten gehören neben einer RGB-Kamera auch eine Infrarotkamera und ein Infrarotsensor. Der synchronisierte Ausgabestream von Tiefe und Farbinformation wird in die räumlichen Informationen umgesetzt. Der Infrarot-Projektor sendet im nahen Infrarotbereich ein für das menschliche Auge nicht sichtbares, codiertes Punktmuster aus. Ein CMOS-Sensor empfängt das von der Szene reflektierte Bild und berechnet aufgrund des Kameraabstandes über die Parallaxen korrespondierender Punkte die Tiefenmatrix mit VGA-Auflösung, der ein RGB-Bild zugeordnet werden kann. Die Berechnungen erfolgen mit parallelen Algorithmen auf einem Chip in der PrimeSense SoC Technologie, wodurch der Hostrechner erheblich entlastet ist.

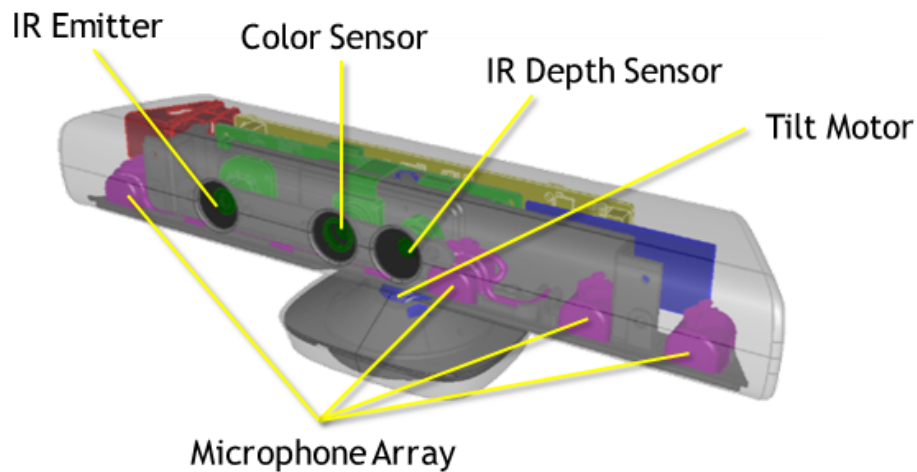


Abbildung 3.1: Die Abbildung zeigt den Aufbau einer Microsoft Kinect Tiefenkamera [Mic14]

3.1.2 Microsoft Kinect SDK

Mit der Microsoft Kinect SDK (Software Development Kit) haben Entwickler die Möglichkeit, Anwendungen für die Kinect-Tiefenkamera zu entwickeln. Microsoft erlaubt damit den vollen Zugriff auf alle verbauten Sensoren, inklusive auf den Infrarot-Sensor und das Mikrophon. Als Hardwareanforderungen werden mindestens ein DualCore-CPU mit 2,66 GHz und 2GB Arbeitsspeicher angegeben.

3.1.3 motionEAP

Der eingesetzte motionEAP Tisch ist aus mehreren Bestandteilen zusammengesetzt. Es besteht aus ein herkömmlicher Acer Beamer mit einer Auflösung von 1280x800 und aus einer Microsoft Kinect die für die Aufnahme der Tiefenbilder zuständig ist. Die Komponenten sind mittels eines Rahmens aus Profilen befestigt, sodass unterschiedliche Einstellungen an der Konfiguration möglich sind. Außerdem wird ein Computer zur Konfiguration und Steuerung der Software verwendet. Der genaue Aufbau des motionEAP-Tisches ist in der Abbildung 1.1 zu sehen.

3.2 Toolkits

In diesem Kapitel werden die beiden zu untersuchenden Toolkits näher erläutert. Das UbiDisplay Toolkit wurde bereits in 2.7 erklärt.

3.2.1 C# Toolkit

Das C# Toolkit wurde im Rahmen des in Kapitel 1.1 Projekts entwickelt. Es basiert auf dem von John Hardy entwickelten UbiDisplays Toolkit. Es nutzt zur Erkennung der Oberfläche eine Microsoft Kinect

und zur Darstellung der Inhalte einen handelsüblichen Projektor. Anders als das UbiDisplays Toolkit wurde es in C# implementiert und benötigt den darzustellenden Inhalt als C# Code. Derzeit wird das Toolkit als Library eingebunden und durch den Start des jeweiligen C# Programms mit ausgeführt. Für die Verwendung des Toolkits ist eine .ubi Konfigurationsfile notwendig. Da es derzeit keine Möglichkeit gibt diese direkt zu erstellen, werden zuerst die Displays mit dem UbiDisplays Toolkit erstellt und anschließend als .ubi gespeichert. Somit kann das C# Toolkit die .ubi Konfigurationsfile verwenden. Außerdem ist es dem Entwickler einfach möglich, diverse Konfigurationen zu erstellen und für das Toolkit zu nutzen. Die weitere Handhabung des C# Toolkits ist ansonsten identisch mit dem UbiDisplays, es sind keine weiteren Einstellungen notwendig. Alle Berechnungen der Touch-Events werden vom Toolkit selber erhoben. Die Nutzung der Daten ist uneingeschränkt durch die Möglichkeit, dass man C# als Programmiersprache verwendet. Das so erstellte interaktive projizierte Display lässt sich entweder aus der Entwicklungsumgebung starten oder aber auch durch die erstellte .exe.

3.3 Testanwendung

Um die Performance der Toolkits unter realen Bedingungen zu testen, wurde ein einfacher Fitts' Law Task in der Programmiersprache Java implementiert. Die Software besteht aus mehreren Komponenten, die in den folgenden Abschnitten erläutert werden und in der Abbildung 3.3 grafisch dargestellt sind.

Serverkomponente

Die Serverkomponente der Software ist zum Einen aus dem UDP-Server und zum Anderen aus dem eigentlichen Fitts' Law Task aufgebaut. Die Aufgabe des UDP-Servers besteht lediglich darin, die gesendeten Koordinaten der Touch-Events entgegenzunehmen und diese dem Fitts' Law Task zur Verfügung zu stellen. Auch wenn es zuverlässigere Transportprotokolle gibt, wurde UDP genommen, da es eine sehr schnelle Übertragung bietet und dadurch eine geringe Latenz im Netzwerk erreicht wird. Die Latenzzeiten lagen stets unter 10ms wodurch eine Verfälschung der Daten auszuschließen ist. Die Art des Fitts' Law Tasks ist eine Abwandlung des „Schlag-den-Maulwurf“ Spiels. Dabei gilt es, die auftauchenden Kreise möglichst schnell wahrzunehmen und zu berühren. Die vom UDP-Server empfangenen Koordinaten werden intern zu einem Klick-Event umgesetzt, wodurch nur Standardfunktionen von Java genutzt werden und keine zusätzlichen Bibliotheken notwendig sind. Die Generierung der auftauchenden Kreise wurde randomisiert implementiert. Dazu wird eine vordefinierte Maximal- und Minimalgröße der Kreise angelegt die nie unter- oder überschritten wird. Ebenso wird der Mindestabstand zweier aufeinander folgenden Kreisen festgelegt. Des Weiteren kann noch die Anzahl der Durchgänge pro Spiel definiert werden. Hierbei bezieht sich die Anzahl der Durchgänge auf die Anzahl der angeklickten Kreise bevor das Spiel beendet wird. In unsere Testumgebung werden folgende Standardwerte verwendet:

- Maximalgröße Kreis: 180px Durchmesser
- Minimalgröße Kreis: 60px Durchmesser

- Minimaler Abstand: 3x Durchmesser des Aktuellen Startkreises.
- Anzahl Durchgänge: 100

Zum Erzeugen der Daten für die Studie erstellt die Software für jeden Teilnehmer eine .csv Datei. Die Abbildung 3.2 zeigt ein Ausschnitt aus einer Beispieldatei. Diese Datei enthält Informationen über alle im Fitts' Law Task durchgeführten Interaktionen mit der Oberfläche. Die Auswertung der Daten wurde dadurch dass die Teilnehmerzahl im niedrigen zweistelligen Bereich liegt nicht automatisiert. Da auf eine Oberfläche für die Serverkomponente verzichtet wurde, wird das Spiel stets aus der Entwicklungsumgebung Eclipse gestartet. Dabei wird die Taskoberfläche, sofern möglich, auf dem zweiten angeschlossenen Monitor angezeigt. Da sie undecorated gestaltet ist, kann sie nur über das sich in der Leiste befindende Icon oder durch Terminieren des zugehörigen Prozesses beendet werden. Wird ein Spiel vollständig durchlaufen, endet es automatisch und die Spieloberfläche schließt sich nachdem das Ende bestätigt wurde.

Game	1	
Distance	Time	Width
434	2630	118
245	796	122
111	735	82
171	749	72
137	2028	138
327	723	146
75	1099	154
555	69	60
734	6603	66
454	1256	80
563	1547	96
359	1155	84
274	787	80
477	1130	130

Abbildung 3.2: Beispiel .csv eines Teilnehmers mit einem Spiel und 14 geklickten Kreisen.

UDP Client UbiDisplays

Für das UbiDisplays Toolkit wurde die Kommunikation mit der Serverkomponente mit node.js gelöst. Dazu greift eine HTML-Seite mit eingebettetem JavaScript-Code die Touch-Events ab und übergibt die Koordinaten an node.js. Die Serverkomponente von node.js sendet wiederum die erhaltenen Koordinaten per UDP an den Gameserver. Dabei werden die Koordinaten weder verarbeitet noch geglättet. Sie sind die direkte Ausgabe des Toolkits um die Verwässerung der Daten auszuschließen.

UDP Client C# Toolkit

Durch die Möglichkeit C# zu benutzen konnte für das C# Toolkit der UDP-Client direkt implementiert werden. Dazu wurde eine Testanwendung BlankTouch angelegt, die ein Touch-Event mit seinen Koordinaten an den spezifizierten Server sendet. Ebenso wie beim UDP-Client von UbiDisplays werden auch hier die Koordinaten nicht verarbeitet oder geglättet.

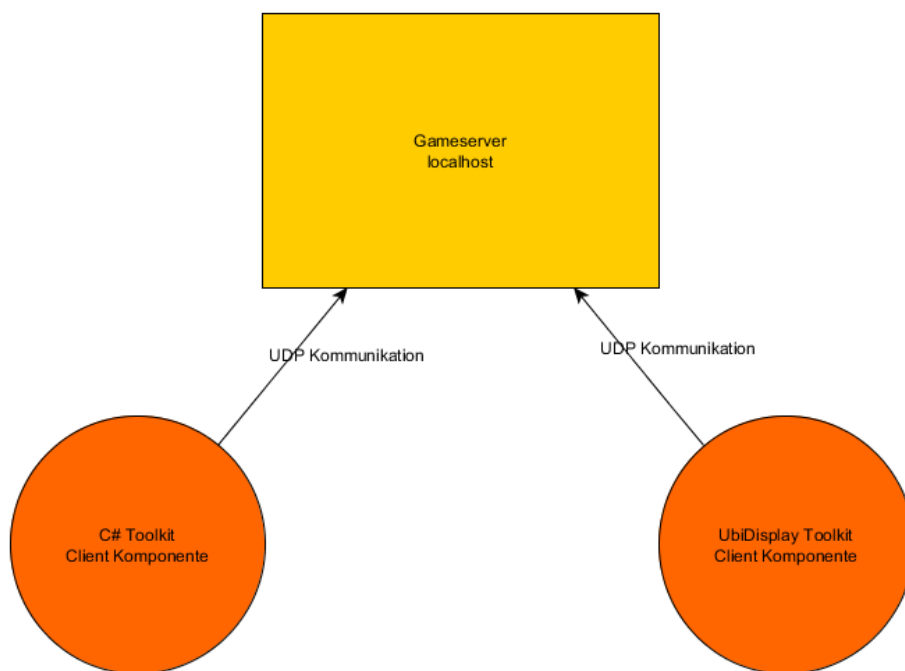


Abbildung 3.3: Diagramm der Anwendungsarchitektur.

4 Nutzerstudie

Der wichtigste Bestandteil dieser Fachstudie ist die Analyse und der Vergleich der Toolkits mittels einer Nutzerstudie. In diesem Kapitel werden der Aufbau und die Durchführung der Studie beschrieben genauso wie die Auswertung der aufgenommenen Daten und der ausgefüllten Fragebögen. Um die Performance der beiden Toolkits unter realen Bedingungen zu testen, wurde ein einfacher Fitts' Law Task implementiert. Damit werden verschiedene Messwerte während der Benutzung aufgezeichnet.

4.1 Design

Bei dieser Studie wurde das "Repeated Measures Design" angewandt. Man vergleicht hier das gleiche Subjekt unter verschiedenen Konditionen. In diesem Fall ist das jeweils eingesetzte Toolkit und das einmal mit „Wischen“ und einmal mit „Klicken“. Durch dieses Design benötigt die Studie eine geringere Anzahl an Teilnehmern um repräsentativ zu sein. Daraus folgt, dass das Design sehr effizient ist: Man kann viele Durchläufe, dadurch dass man weniger Probanden „einlernen“ muss, auch schneller durchführen. Man konnte außerdem durch Beobachten der Studenten erkennen, ob sie gegen Ende der Studie durch das Lernen des Prinzips anders auf die letzten Durchläufe reagierten oder ein Gefühl dafür bekamen, ob oder dass sich die zwei „Wischen“ (bzw. die zwei „Klicken“) voneinander unterschieden und welches besser war. Das eingesetzte Toolkit und die beiden Eingabemethoden sind auch die unabhängigen Variablen der Studie. Eine der abhängigen Variablen sind zum einen die MT von Fitts' Law 2.8 zum anderen die "Total Completion Time" der Aufgaben.

4.2 Apparatur

Für die Studie wurde ein Tisch als „Leinwand“ genommen. An diesem wurden senkrechte Profile angebracht. Ganz oben wurde die Microsoft Kinect befestigt: Die Kinect strahlte somit auf den Tisch herab und man konnte somit einen viereckigen „aktiven“ Bereich erkennen. Die Steuerung der Kinect lief durch einen Laptop (i5-Prozessor, 8GB RAM). Bei den Durchläufen wurde zweimal das C# Toolkit genommen, zweimal das UbiDisplay Toolkit. Jeweils unterschied man dann zwischen „Wischen“ und „Klicken“. Beim „Wischen“ berührte man das Display durchgehend bis zum letzten Punkt, das heißt man verband die Punkte mit dem Finger ohne diesen zu Heben und „wischte“ von Punkt zu Punkt. Beim „Klicken“ berührte man die Punkte nacheinander und hob dazwischen den Finger an. Den vier Testdurchläufen wurden Zahlen als Bezeichnung zugeordnet, somit entstand folgende Benennung:

1: C# Toolkit – Wischen

2: C# Toolkit – Klicken

3: UbiDisplay Toolkit – Wischen

4: UbiDisplay Toolkit – Klicken

4.3 Teilnehmer

Angefragt wurde durch Ansprechen von Studenten auf dem Campus, Einladungen per Email und auch der Weiterleitung an Studenten durch Tutoren. Studenten die teilnehmen wollten, haben sich in den Teilnahmeverwaltungsdienst Doodle eingetragen. Die sich schließlich ergebenden 32 Teilnehmer setzten sich aus 26 männlichen und 6 weiblichen Probanden zusammen. Das Durchschnittsalter lag bei 22,5 Jahren. Der jüngste Teilnehmer war 19 Jahre alt, der älteste Teilnehmer hingegen 30 Jahre alt. Mit Ausnahme eines Studenten haben alle Probanden angegeben, bereits Erfahrung mit Touchscreens gesammelt zu haben. Bei der Rekrutierung der Probanden wurden Studenten verschiedener Fachrichtungen eingeladen. Die eindeutige Mehrheit der Studenten studieren jedoch Informatik oder Softwaretechnik. Lediglich 4 Teilnehmer gaben etwas anderes an: Einer studiert Musikmanagement, einer Umweltschutztechnik und 2 studieren Elektrotechnik und Informationstechnik. Die meisten Probanden verwendeten ihre rechte Hand für das Spiel.

4.4 Vorgehen

Damit die Messwerte so authentisch wie möglich blieben, wurde viel Wert darauf gelegt, äußere Einflussfaktoren so gut wie möglich zu minimieren. Die Probanden wurden gebeten ihre subjektive Meinung über die Nutzung des Systems in einem Fragebogen anzukreuzen. Die Probanden kamen jeweils zur vereinbarten Uhrzeit zur Studie und bekamen eine grobe Information darüber, um was für ein Spiel es sich handelt. Sie erfuhren, dass sie Punkte nacheinander deaktivieren sollten und dass sie vier Durchgänge machen würden und dass es noch einmal genauer erklärt wird. Daraufhin lasen und unterschrieben sie die Einverständniserklärung siehe Anhang 6.

Die Studenten beantworteten dann generelle Fragen zu sich: Wie alt der Student war, welches Geschlecht er hat, welcher Studiengang besucht wird und welche Hand verwendet wird um das Spiel zu spielen. Es wurde außerdem erfragt, ob der Proband bereits Erfahrungen mit Touchscreens hatte. Anschließend bekamen die Studenten etwa 15 Sekunden Zeit, um ein Gefühl für die vorbereitete Anlage zu bekommen und zu wissen, was von ihnen verlangt wurde, wenn man ankündigt dass sie „Wischen“ oder „Klicken“ sollten.

In welcher Reihenfolge die Teilnehmenden die Testdurchläufe durchführen würden, wurde durch eine zufällige Permutation 4.1 festgelegt. Die Teilnehmer führten nacheinander jeweils eine der Zeilen durch. Diese acht Reihenfolgen wurden bis zum Ende der Studie wiederholt durchgeführt. So konnte sichergestellt werden, dass eventuelles Lernen des Prinzips die Ergebnisse der Durchläufe gegen Ende nicht verfälschten.

Auf dem projizierten Display erschienen nacheinander 100 Punkte, die sie durch Berühren deaktivieren sollten sodass der nächste Punkt erschien. Wichtig war zu beachten, dass der Winkel der Hand zur

Oberfläche nicht zu groß aber auch nicht zu klein wurde, sodass die Position des Fingers optimal übertragen werden konnte.

Nach jeder Runde wurde eine weitere Frage des Fragebogens ausgefüllt. Hierbei wurde die Reaktionszeit erfragt: Welches Gefühl hatte der Proband, wie da System auf ihn reagiert und seine „Eingabe“ (also das Berühren der Punkte mit dem Finger) registriert und verarbeitet hat? Nach dem letzten Durchlauf wurden noch folgende zwei generelle Fragen beantwortet: Wie zufrieden waren die Probanden mit dem System? Würden sie ein interaktives Display einem konventionellen in Zukunft vorziehen? Nach vollständigem Ausfüllen des Fragebogens wurden die Probanden entlassen.

1	2	3	4
2	1	4	3
3	4	1	2
4	3	2	1

1	2	3	4
2	4	1	3
3	1	4	2
4	3	2	1

Abbildung 4.1: Die bei der Nutzerstudie verwendete zufällige Permutationsreihenfolge.

4.5 Ergebnis

4.5.1 Objektive Ergebnisse

Zur Auswertung der Daten wurde von der Software für jedes Event ein Datensatz erhoben. Alle Datensätze eines Versuchs wurden zusammengefasst. Für die weitere Verwendungen und Berechnung von Fitts' Law wurden die Datensätze wie folgt aufbereitet: Dazu wurde zuerst die Standardabweichung der Zeit über alle Datensätze ermittelt, und aus den Datensätzen jene bereinigt, die größer als die dreifache Standardabweichung waren und auch jene, die kleiner als 100ms waren. Diese Werte haben wir als Fehler der Software interpretiert und finden keine Anwendung bei der Ermittlung von Fitts' Law. Des Weiteren wurde der Index of Difficulty ermittelt und zusätzlich gespeichert. Für die Berechnung wurde MS Excel verwendet. Um die beiden Konstanten des Geräts zu ermitteln, wurde nun über die Zeit und den Index of Difficulty eine lineare Regression durchgeführt. Die Ergebnisse wurden in Diagrammen festgehalten, welche die Regressionsgleichung sowie Regressionsgerade beinhalten. Die Gerade wird als Schwarze Linie dargestellt. Die Gleichung ist in allen Diagrammen oben links. Durch die Berechnung der Konstanten von Fitts' Law lassen sich die Verfahren und Toolkits vergleichen. Hierbei sind Werte die näher bei Null liegen besser. Für die Geschwindigkeit des Gerätes gilt eine geringere Steigung als schneller und somit besser. Die Ergebnisse sind in der Abbildung 4.2 dargestellt. Sie sind direkt dem vergleichenden Toolkits gegenübergestellt.

	C# klicken	UbiDisplay klicken	C# wischen	UbiDisplay wischen
a	833,05	1184,7	840,5	915,18
b	21,117x	60,54x	36,796x	22,515x

Abbildung 4.2: Ergebnisse der durchgeführten Studie.

4.5.2 Subjektive Ergebnisse

Auf die Frage, wie zufrieden die Probanden mit der Bedienung waren antworteten 13 dass sie unzufrieden waren, 7 fanden es neutral und 11 waren zufrieden. Nur ein Proband war sehr zufrieden. Hier ist eine leichte Tendenz zur Unzufriedenheit feststellbar. Ob die Studenten das interaktive Touchscreen Prinzip dem konventionellen vorziehen würden, beantwortete die Mehrheit mit einer negativen Antwort. Komplette auf der Seite des konventionellen Touchscreen Prinzips waren 9 Probanden, 9 weitere tendierten auch eher dazu, 9 Studenten waren teils für das Eine und teils für das Andere und lediglich vier tendierten zum interaktiven Touchscreen Prinzip. Nur ein Student würde das Interaktive vorziehen.

4.6 Auswertung

Zur Auswertung wurde die Erforschende Datenanalyse nach J. W. Tukey angewendet [Tuk77]. Die Tabelle 4.2 stellt die ausgewerteten Ergebnisse der einzelnen Toolkits in der jeweiligen Disziplin dar. Hierbei fällt auf dass, das C# Toolkit im direkten Vergleich "klicken" mit dem UbiDisplays in der Variable **a** um Faktor 3 besser abschneidet. Auch ist eine deutliche Diskrepanz in der Variablen **b** festzustellen.

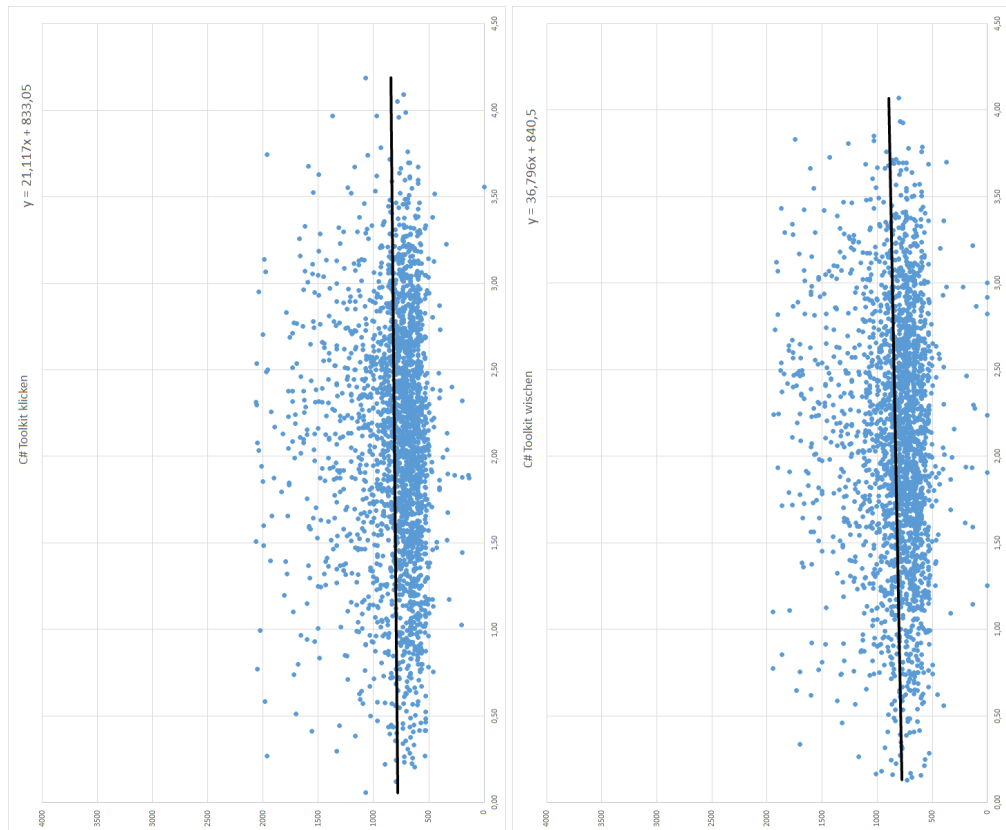
Die Unterschiede sind im Task Wischen nicht mehr so ausgeprägt wie im Task klicken. Hier liegen C# Toolkit und UbiDisplays Toolkit gleich auf. Die leichte Schwäche in **a** gleicht das C# Toolkit in der Variablen **b** wieder aus.

Betrachtet man weiterhin die Standardabweichungen der jeweiligen Toolkits ist eine deutliche Ausprägung in Richtung C# Toolkit zu erkennen. Somit arbeitet das C# Toolkit genauer als das UbiDisplays Toolkit.

Im Gesamtvergleich liegt das C# Toolkit vor dem UbiDisplays Toolkit. Im Bezug auf die subjektiven Eindrücke die in Abbildung 4.4 zu sehen sind, liegt im Bereich des Klickens auch das C# Toolkit vorne. Hierbei befand die Mehrheit der Nutzer das C# Toolkit als schnell. Nur wenige erachteten das C# Toolkit als langsam. Beim UbiDisplays Toolkit hingegen zeigt sich dass die Probanden zu gleichenteilen das UbiDisplays Toolkit langsam, angemessen und schnell befanden.

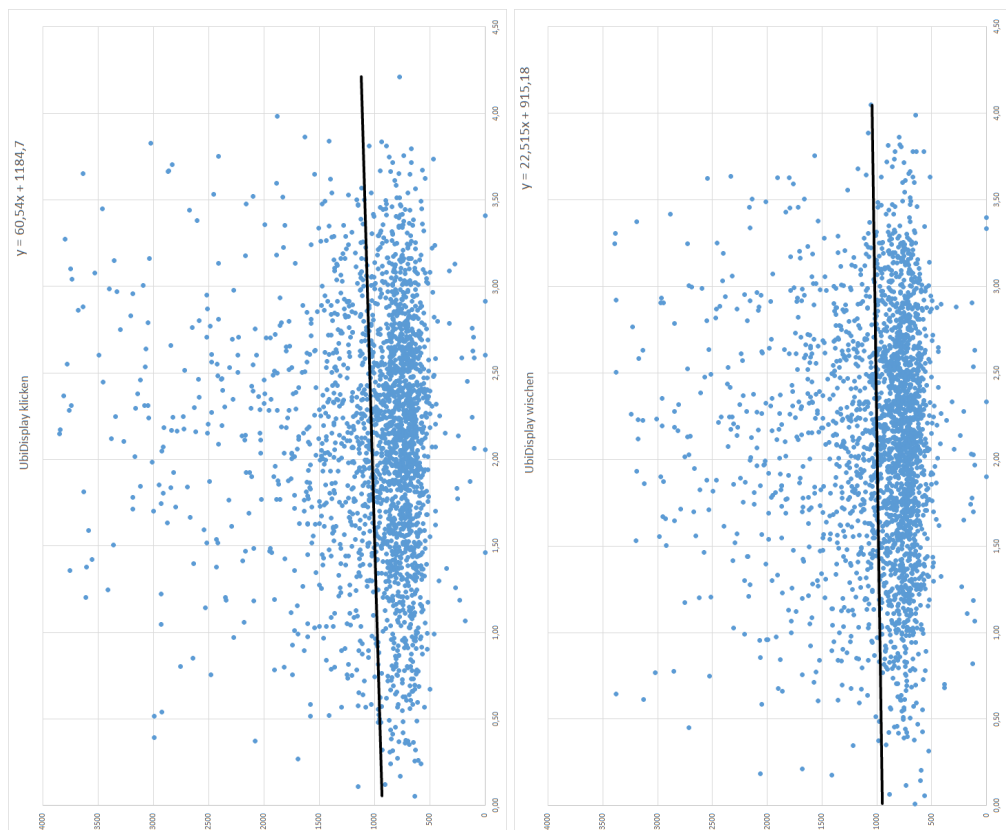
Im Bereich des Wischen liegt bei den subjektiven Eindrücken das C# Toolkit auch mit der Mehrheit bei schnell vor dem UbiDisplays Toolkit, welches eher als angemessen empfunden wurde.

Somit zeichnet sich das C# Toolkit in beiden Task als das besser ab. Wobei die Tendenzen beim klicken wesentlich deutlicher sind als beim Wischen.



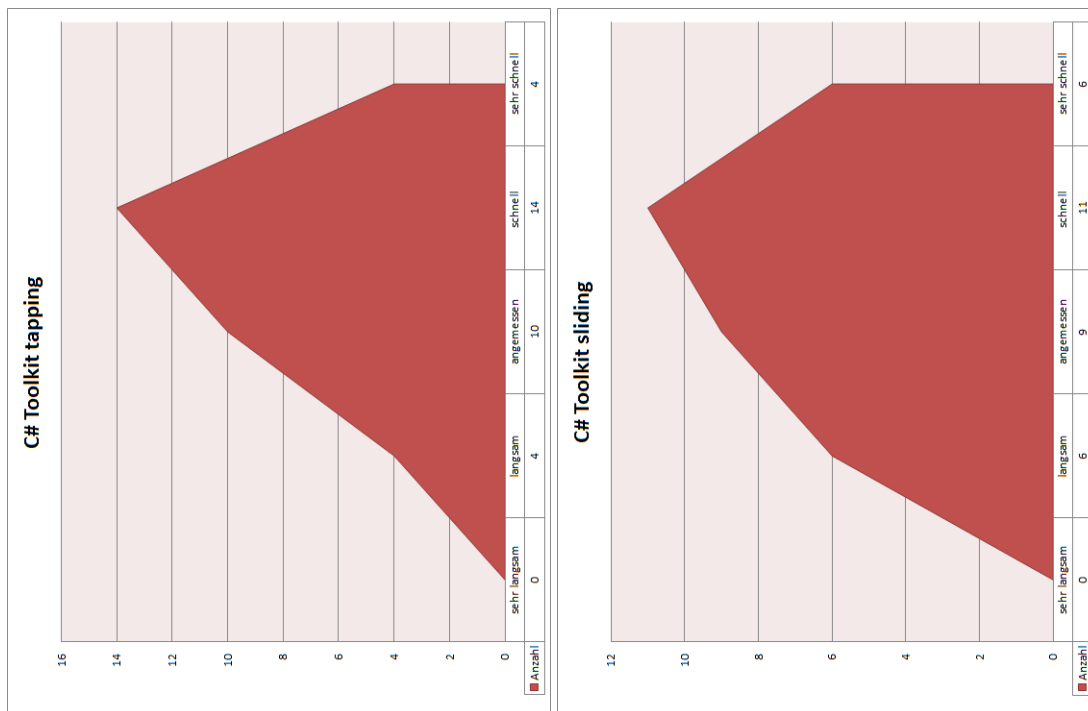
(a) Auswertung des Fitts Law Test für C Toolkits „Klicken“.

(b) Auswertung des Fitts Law Test für C Toolkits „Wischen“.



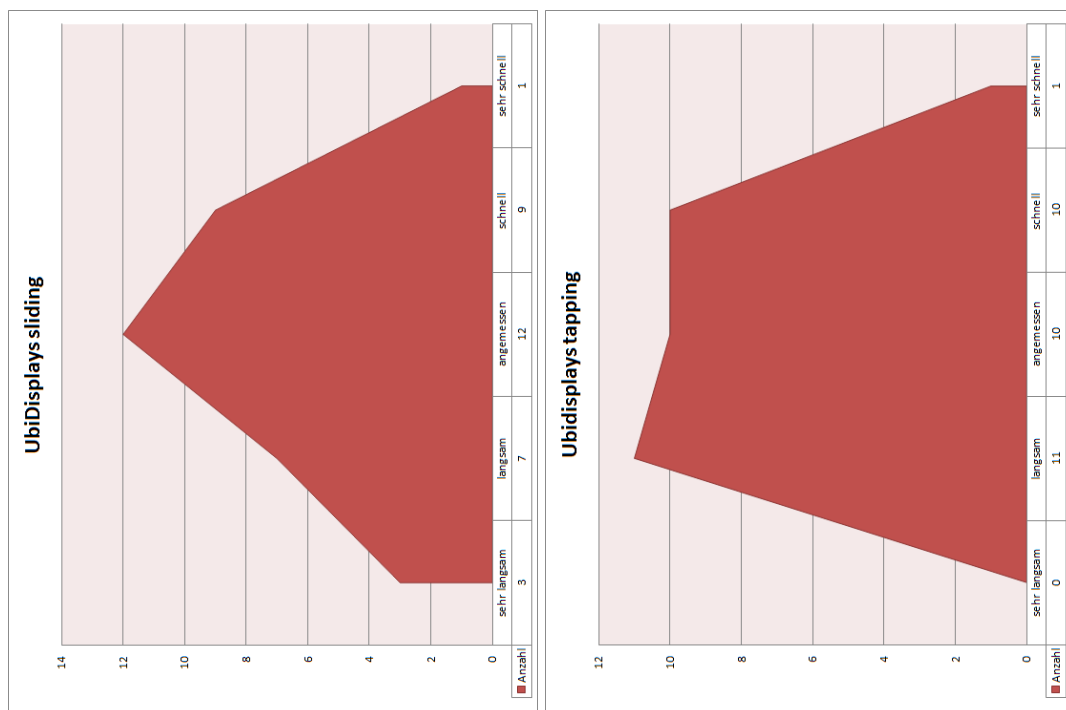
(c) Auswertung des Fitts Law Test für UbiDisplayss „Klicken“.

(d) Auswertung des Fitts Law Test für UbiDisplayss „Wischen“.



(a) Auswertung der subjektiven Eindrücke für C Toolkits „Klicken“.

(b) Auswertung des subjektiven Eindrücke für C Toolkits „Wischn“.



(c) Auswertung des subjektiven Eindrücke für UbiDisplay „Klicken“.

(d) Auswertung des subjektiven Eindrücke für UbiDisplay „Wischn“.

Abbildung 4.4: Überblick der Auswertungsdiagramme für die subjektiven Eindrücke der Nutzer.

5 Fazit und Ausblick

Ziel dieser Fachstudie war es zwei Toolkits für interaktive projizierte Displays miteinander zu vergleichen indem bestimmte Performanzindikatoren ermittelt und ausgewertet wurden. Für die Erfassung der Indikatoren wurde ein einfaches aus mehreren Komponenten bestehenden Fitts' Law Task implementiert. Um eine repräsentative Erfassung und Auswertung zu erreichen wurde der Vergleich der Toolkits mittels einer Nutzerstudie durchgeführt. Dabei mussten die 32 Probanden nach einer kurzen Einführung den Fitts' Law Task vier Mal hintereinander absolvieren und anschließend ein Fragebogen über ihre subjektive Einschätzung der Bedienung ausfüllen. Nach der Nutzerstudie wurden alle Ergebnisse analysiert und nach dem Ausschließen aller Fehler und äußeren Faktoren ausgewertet. Im Allgemeinen ist zu sehen, dass das C# Toolkit-„Klicken“ deutlich genauer und schneller im Vergleich zum UbiDisplays-„Klicken“ reagiert. Beim „Wischen“ ist der Unterschied zwischen der beiden Toolkits nicht so groß, dennoch auch hier kann man sagen, dass das C# Toolkit tendenziell um 9% schneller als das UbiDisplays Toolkit ist. Auch was die Standardabweichung und die Genauigkeit betrifft, schneidet das C# Toolkit besser ab.

Bei der Evaluation der subjektiven Einschätzung war tendentiell eine Unzufriedenheit der Teilnehmer bei der Bedienung festzustellen. Die Mehrheit der Teilnehmer würde das interaktive Touchscreen Prinzip in dieser Phase noch nicht dem Konventionellen vorziehen.

Ausblick

In Anbetracht der steigenden Nachfrage von immer größeren Touchdisplays werden sich auch interaktive projizierte Displays weiterentwickeln. Einige Projekte zielen bereits auf den kommerziellen Markt ab, wie beispielsweise das Ubi Interactive [Int14]. Weiterhin sind stets bessere Tiefenkameras verfügbar, welche die Genauigkeit und Geschwindigkeit immer weiter anheben. Da diese immerzu kostengünstiger werden, sinken die Anschaffungskosten. So ist seit Mitte 2014 die Weiterentwicklung der MS Kinect verfügbar: Die Kinect One. In naher Zukunft werden interaktive projizierte Displays Einzug in Konferenzräume haben und täglich bei Meetings eingesetzt werden können. Da es sich bisher noch nicht abzeichnet dass der Aufbau des Systems flexibler wird, sollten diese Systeme jedoch in solchen Räumen effektiv eingebaut werden, sodass sie optimal genutzt werden können.

6 Begriffslexikon

Ubiquitous Displays

Mit Ubiquitous Displays werden interaktive Bildschirme beschrieben die klassische Formen der Interaktion durch physische Komponenten erweitern oder sogar ersetzen.

Open-Source-Toolkit

Open-Source kann man mit quelloffen übersetzen. Diese Bezeichnung wird in der Regel für freie Software verwendet. Open-Source-Toolkit bedeutet dass der Quellcode dieser Software für die Weiterentwicklung von Dritten offen steht.

UbiDisplays

Toolkit für Bereitstellen von interaktiven Inhalten für interaktive projizierbare Displays unter Verwendung eines Projektors und einer Microsoft Kinect.

Microsoft Kinect

Tiefenkamera der Firma Microsoft unter Verwendung der PrimeSense-Technologie. Wurde zuerst für die Steuerung der Videospielekonsole Xbox 360 entwickelt, später auch für die Windows-Betriebssysteme.

Toolkit

Ist eine Sammlung von Tools und Schnittstellen um Entwicklern das Erstellen von Anwendungen zu erleichtern indem Standardfunktionen zur Verfügung gestellt werden.

Touchevent

Ein Ereignis das durch das Berühren einer Fläche mit dem Finger oder mit der Handfläche ausgelöst wird.

Tangible Interaction

Unter „Tangible Interaction“ versteht man die Greifbarkeit und physische Manifestation einer Benutzerschnittstelle und die Betonung der physischen Interaktion. Der Tastsinn spielt beim Greifen oder Berühren von Objekten eine wichtige Rolle.

Framework

Ein Framework kann wie ein Programmgerüst verstanden werden. Es stellt für den Programmierer einen Rahmen zur Verfügung um ihn bei der Entwicklung von Anwendungen zu unterstützen.

Standalone-Software

Eine Anwendung die überall eingesetzt werden kann und unabhängig von externen Programmbibliotheken funktioniert.

DNP-Holoscreen

Innovativer transparenter Bildschirm der Firma DNP, der über einen aktiven Hochkontrastfilter verfügt und die Möglichkeit bietet einfallendes Licht zu absorbieren und gleichzeitig das projizierte Bild zu reflektieren.



Consent Form

DESCRIPTION: You are invited to participate in a **research study** on **interaction with a projected display**.

TIME INVOLVEMENT: Your participation will take approximately **30 minutes**.

DATA COLLECTION: For this study you will be asked to play a game on the projected surface. Data about your interaction with the system will be logged on the computer. Also, you will need to fill in questionnaires.

RISKS AND BENEFITS: No risk associated with this study. The collected data is securely stored. We do guarantee no data misuse and privacy is completely preserved. Your decision whether or not to participate in this study will not affect your grade in school.

PROCEDURE:

1. Introduction to the study
2. System explanation and test rules description
3. (Perform a test run to accommodate to the surface)
4. Perform the first of four runs
5. Fill in the evaluation form till question 3
6. Perform the second run
7. Repeat until fourth run is completed
8. After filling in the evaluation right the study ends.

PAYMENTS: You will receive Hanuta hazelnut candy ☺!

PARTICIPANT'S RIGHTS: If you have read this form and have decided to participate in this project, please understand your **participation is voluntary** and you have the **right to withdraw your consent or discontinue participation at any time without penalty or loss of benefits to which you are otherwise entitled. The alternative is not to participate.** You have the right to refuse to answer particular questions. The results of this research study may be presented at scientific or professional meetings or published in scientific journals. Your identity is not disclosed unless we directly inform and ask for your permission.

CONTACT INFORMATION: If you have any questions, concerns or complaints about this research, its procedures, risks and benefits, contact following persons:

Markus Funk (markus.funk@vis.uni-stuttgart.de)

Oliver Korn (Oliver.Korn@vis.uni-stuttgart.de).

By signing this document I confirm that I agree to the terms and conditions.

Name: _____ Signature, Date: _____

Literaturverzeichnis

- [Fit54] P. M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of experimental psychology*, 47(6):381, 1954. (Zitiert auf Seite 17)
- [FKS14a] M. Funk, O. Korn, A. Schmidt. Assistive augmentation at the manual assembly workplace using in-situ projection. *Proceeding of the chi workshop on assistive augmentation*, 2014. (Zitiert auf Seite 7)
- [FKS14b] M. Funk, O. Korn, A. Schmidt. An augmented workplace for enabling user-defined tangibles. In *CHI'14 Extended Abstracts on Human Factors in Computing Systems*, S. 1285–1290. ACM, 2014. (Zitiert auf den Seiten 6, 7 und 8)
- [HA12] J. Hardy, J. Alexander. Toolkit support for interactive projected displays. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia*, S. 42. ACM, 2012. (Zitiert auf den Seiten 6 und 16)
- [HBW11] C. Harrison, H. Benko, A. D. Wilson. OmniTouch: wearable multitouch interaction everywhere. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, S. 441–450. ACM, 2011. (Zitiert auf den Seiten 6 und 15)
- [Int14] U. Interactive. Ubi Interactive. <http://www.ubi-interactive.com/>, 2014. [Online; accessed 24-Sep-2014]. (Zitiert auf Seite 33)
- [KNF12] F. Klompaker, K. Nebe, A. Fast. dSensingNI: a framework for advanced tangible interaction using a depth camera. In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction*, S. 217–224. ACM, 2012. (Zitiert auf den Seiten 6 und 13)
- [Kor12] O. Korn. Industrial playgrounds: how gamification helps to enrich work for elderly or impaired persons in production. In *Proceedings of the 4th ACM SIGCHI symposium on Engineering interactive computing systems*, S. 313–316. ACM, 2012. (Zitiert auf Seite 8)
- [Mic14] Microsoft. Kinect Tiefenkamera. <http://msdn.microsoft.com/en-us/library/jj131033.aspx>, 2014. [Online; accessed 24-Sep-2014]. (Zitiert auf den Seiten 6 und 20)
- [Pin01] C. Pinhanez. The everywhere displays projector: A device to create ubiquitous graphical interfaces. In *UbiComp 2001: Ubiquitous Computing*, S. 315–331. Springer, 2001. (Zitiert auf den Seiten 6 und 12)
- [Tuk77] J. W. Tukey. Exploratory data analysis. 1977. (Zitiert auf Seite 29)

- [Wei91] M. Weiser. The computer for the 21st century. *Scientific american*, 265(3):94–104, 1991. (Zitiert auf Seite 7)
- [Wil04] A. D. Wilson. TouchLight: an imaging touch screen and display for gesture-based interaction. In *Proceedings of the 6th international conference on Multimodal interfaces*, S. 69–76. ACM, 2004. (Zitiert auf den Seiten 6 und 14)
- [Wil10] A. D. Wilson. Using a depth camera as a touch sensor. In *ACM international conference on interactive tabletops and surfaces*, S. 69–72. ACM, 2010. (Zitiert auf den Seiten 6 und 11)
- [WSDR14] C. Winkler, J. Seifert, D. Dobbelsstein, E. Rukzio. Pervasive information through constant personal projection: the ambient mobile pervasive display (AMP-D). In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, S. 4117–4126. ACM, 2014. (Zitiert auf den Seiten 6, 15 und 16)

Alle URLs wurden zuletzt am 29. 09. 2014 geprüft.

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Stuttgart, den 04.10.2014

Ort, Datum, Unterschrift

Stuttgart, den 04.10.2014

Ort, Datum, Unterschrift

Stuttgart, den 04.10.2014

Ort, Datum, Unterschrift