

Bauhaus-Universität Weimar
Fakultät Medien
Studiengang Medieninformatik

Mehrbenutzer Multi-Touch-Interaktion mit einem immersiven 3D Bildschirmtisch

Bachelorarbeit

Sebastian Stickert
Geboren am 21.04.1992 in Rostock

Matrikelnummer 110882

1. Gutachter: Prof. Dr. Bernd Fröhlich
2. Gutachter: Junior-Prof. Dr. Florian Echtler

Datum der Abgabe: 08.05.2015

Erklärung

Hiermit versichere ich, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Weimar, den 08.05.2015

.....
Sebastian Stickert

Zusammenfassung

Von einer breiten Masse der Bevölkerung werden Geräte wie Tablets, Smartphones, oder sonstige Touchscreens täglich für unterschiedliche Aufgaben genutzt. Hierbei haben sich verschiedene Gesten zur Kontrolle von zweidimensionalen Inhalten bereits etabliert. Durch diese Erfahrungen erwarten Nutzer intuitiv Eingaben an der Oberfläche eines Bildschirmisches durchführen zu können. Auch stereoskopische Bildinhalte sind der Öffentlichkeit, beispielsweise durch 3D Kinofilme, oder 3D Fernseher längst bekannt. Für die Verbindung dieser beiden Techniken gibt es bislang jedoch keine vollständig akzeptierte Lösung.

Im Rahmen dieser Arbeit wurde ein System zur Multi-Touch basierten Navigation in dreidimensionalen, virtuellen Szenen implementiert. Es wird hierzu eine Reihe von existierenden Manipulationstechniken diskutiert. Die Verwendung des dreidimensionalen Raumes zur Visualisierung von Applikationsinhalten erweitert die für Transformationen parametrisierbaren Freiheitsgrade. Folglich lässt sich eine Steigerung der Komplexität bei Touch Gesten gegenüber bekannten 2D Interaktionen beobachten. Während des Arbeitsprozesses wurde *Levelling* entwickelt um den Nutzer implizit beim Umgang mit der Applikation zu unterstützen. Hierbei werden durch den Anwender Referenzpunkte auf dem Modell definiert und anhand der Technik auf der Bildfläche gegeben.

Eine weitere Herausforderung resultiert aus dem Umgang mit stereoskopischen Bildinhalten. Demnach führt das Eindringen der Hand in virtuelle Geometrie oberhalb der Projektionsebene zur Beeinträchtigung der Tiefenwahrnehmung. Die *See-Through* Technik ist entstanden um vom Arm des Nutzers durchdrungene Modellteile freizuschneiden. Auf diese Weise kann dem beschriebenen Wahrnehmungskonflikt entgegengewirkt werden.

Inhaltsverzeichnis

1 Einführung	3
1.1 Mehrbenutzer VR-Systeme	3
1.2 Multi-Touch Interaktion mit dreidimensionalen Szenen	3
1.3 Ziel der Arbeit	4
2 Anforderungsanalyse	6
2.1 Mehrbenutzer Touch-Eingaben	6
2.2 Tiefenwahrnehmung durch Stereo- und Bewegungsparalaxe	7
2.3 Interaktionsziele	7
2.4 Explizite und implizite Kontrolle der involvierten Freiheitsgrade	7
2.5 Visuelle Rückmeldung	8
3 Finger und Hand Tracking auf Grundlage von Maximally Stable Extremal Regions	9
3.1 Maximally Stable Extremal Regions	10
3.2 Technische Voraussetzungen	11
3.3 Implementierung	12
3.3.1 TUIO Touch Protokoll	13
3.3.2 Umgang mit Jittering	14
3.4 Vorteile und Limitierungen	14
4 3D Interaktion mit Multitoucheingaben	16
4.1 Sticky Tools	16
4.2 Balloon Selection	17
4.3 Two Axis Valuator	18
4.4 Diskussion	20
5 Wahrnehmungskonflikte und Lösungsansätze	22
5.1 Frame Cancellation	22
5.2 Touch Interaktion mit stereoskopischen Visualisierungen	24
5.3 Diskussion	26

6 Explizite Multi-Touch 3D Navigation	28
6.1 Rotation, Translation und Skalierung im Bildraum	28
6.2 3D Translation	30
6.3 3D Rotation	32
6.4 Vorteile und Limitierungen	33
7 Levelling: Implizite Multi-Touch 3D Navigation	35
7.1 Definition und Interaktionsziele	35
7.2 Depth-Levelling	38
7.3 Rotation-Levelling	39
7.4 Vorteile und Limitierungen	40
8 Wechsel zwischen Interaktionstechniken	42
8.1 Evaluierung der Bildschirmkontakte	42
8.2 Auswahl durch Menüelemente	45
8.3 Diskussion	46
9 Virtuelles Freischneiden der Hand während Berührungseingaben	48
9.1 Ansatz	48
9.2 Implementierung	50
9.3 Vorteile und Limitierungen	50
10 Abbildung von Multi-Touch Techniken in der Applikationsstruktur	52
10.1 Multi-Touch Input Pipeline	52
10.2 Touch Navigation	54
10.3 Repräsentation im Szenegraph	56
10.4 Diskussion	56
11 Diskussion	58
11.1 Anforderungsevaluierung	58
11.2 Hypothesen	60
12 Fazit und zukünftige Arbeiten	62
Literaturverzeichnis	64

Kapitel 1

Einführung

1.1 Mehrbenutzer VR-Systeme

Aktuelle Entwicklungen am Lehrstuhl für virtuelle Systeme ermöglichen die Darstellung von komplexen 3D Scans. Im Rahmen des EU-Forschungsprojekts 3D Pitoti werden beispielsweise umfangreiche Datensätze in verschiedenen Auflösungsstufen, zur Betrachtung von prähistorischen Felsgravuren in Val Camonica zur interaktiven Visualisierung, angeboten.

Um hierbei Oberflächenstruktur im Submillimeterbereich, sowie deren nähere Umgebung kollaborativ analysieren zu können, werden verschiedene Mehrbenutzer Projektionssysteme genutzt. Eine Projektionswand wird verwendet um Inhalte im 1:1 Maßstab anzuzeigen, während ein Bildschirmtisch gleichzeitig den Überblick über das gesamte Tal liefern kann. Beide Systeme begünstigen die Betrachtung stark vergrößerter Oberflächendetails.

Die Verteilung einer Applikation über mehrere Bildschirme wird nach Seyed et al. [Seyed et al., 2013] auch als *Multi-Display-Umgebung* bezeichnet. In dieser Umgebung können Nutzer in Gruppen variabler Größe, oder allein, mit den Angeboteten Darstellungen interagieren. Des Weiteren können Anwender frei zwischen verschiedenen Bildschirmen wechseln.

1.2 Multi-Touch Interaktion mit dreidimensionalen Szenen

Stereoskopisches Rendering führt zu positiver, negativer, oder null Parallaxe bei der Darstellung von Bildinhalten. Das heißt virtuelle Modelle erscheinen für die Wahrnehmung unterhalb (Positiv-Parallaxe), oberhalb (Negativ-

Parallaxe), oder auf (Null-Parallaxe) der Tischoberfläche. Bekannte 2D Touch Interaktionen basieren auf direktem Kontakt mit den Objekten der virtuellen Darstellung. Es entsteht eine Bindung die das Gefühl erzeugt Objekte festzuhalten. Dieser Effekt kann auf direkte Weise nur bei null Parallaxe auf eine stereoskopische Anwendung übertragen werden [Bruder et al., 2013]. Außerdem ruft das Greifen der Hand in negativ parallaxe Bildarealen Störungen der Tiefenwahrnehmung hervor [de la Rivière et al., 2010].

Des Weiteren hat die Integration einer 3D Szene Auswirkungen auf die Freiheitsgrade der Manipulation. Hierbei wird die 2D Translation um die Verschiebung entlang der z-Achse erweitert. Rotation erfolgt im 3D Raum unter Berücksichtigung dreier Achsen. Im zweidimensionalen Raum wird hierfür lediglich eine Achse benötigt. Da die Eingabeparameter eines Berührungs punktes der Bildschirmfläche zweidimensional sind, ergibt sich eine Herausforderung für die Entwicklung einer geeigneten 3D Manipulationsschnittstelle [Martinet and Grisoni, 2012].

1.3 Ziel der Arbeit

Die Verwendung von Projektionstischen zur Darstellung virtueller Bildinhalte begünstigt die Einbindung von taktilen Eingaben. Durch blickabhängige Renderingverfahren werden kollaborative Anwendungen leicht realisierbar. Ziel dieser Arbeit ist es, eine Mehrbenutzer-Schnittstelle zur touch-basierten Applikationsnavigation mit dreidimensionalen Inhalten zu entwickeln. Hierbei soll es mehreren Nutzern möglich sein gemeinsam mit den dargebotenen virtuellen Welten umzugehen. Im Kontext des 3D Pitoti EU Forschungsprojekts soll das entwickelte System die Exploration skalierbarer Oberflächenscans effektiv ermöglichen.

Vor diesem Hintergrund werden diverse Anforderungen definiert, welche in Kapitel 2 näher beschrieben werden. Die zugrunde liegende Finger und Hand Erkennung basiert auf einer im Rahmen einer Masterabschlussarbeit entstandenen Verfahrenstechnik [Ewerling et al., 2012]. Kapitel 3 gibt Einblicke in den hierfür verwendeten *Maximally Stable Extremal Regions* Algorithmus nach Matas et al. [Matas et al., 2004]. Außerdem werden hardwareseitige Voraussetzungen, sowie die Anbindung an das, für die Multi-Touch Navigation verwendete Framework Avango geschildert [Avango,]. Zur Implementierung geeigneter Touch Techniken gilt es verwandte Arbeiten in diesem Bereich zu analysieren und zu evaluieren. Die Auseinandersetzung mit diesem Thema wird in Kapitel 4 geschildert. Außerdem lassen sich in der Literatur diverse Hinweise auf

durch Interaktion mit stereoskopischen Projektionen entstehende Wahrnehmungskonflikte finden. In Kapitel 5 werden diese näher beleuchtet und bestehende Lösungsvorschläge diskutiert. Durch Manipulation des virtuellen Viewing Setups können Navigationsaufgaben in der Applikation realisiert werden. Hierzu sind Transformationen im dreidimensionalen Raum mit verschiedenen Freiheitsgraden zu bestimmen und anzuwenden. Kapitel 6 stellt die im Rahmen dieser Arbeit entstandenen Multi-Touch Gesten vor, welche dem Nutzer durch verschiedene Eingabeabläufe die Parametrisierung aller Freiheitsgrade explizit ermöglichen. Implizite Navigationsansätze erweitern diese Strategien durch zusätzliche Manipulation der Applikation. Diese werden aus den ausgearbeiteten Interaktionszielen des Nutzers abgeleitet und können indirekt durch den Anwender gesteuert werden. Eine Konkretisierung dieses Konzepts durch die Einführung der *Levelling*-Technik erfolgt in Kapitel 7. Um die zur Navigation verwendeten Gesten zu unterscheiden ist die Gliederung in verschiedene Eingabemodi erforderlich. Diese gilt es zu definieren und durch geeignete Mechanismen für den Nutzer auswählbar zu machen. Kapitel 8 erklärt wie der Wechsel zwischen Interaktionstechniken bewerkstelligt wird. Wie in Kapitel 5 beschrieben wird, resultieren aus dem Eindringen physischer Objekte in Applikationsgeometrie Konflikte für die Wahrnehmung der Nutzer. Durch Kapitel 9 wird die *See-Through* Technik zur Abschwächung dieser Probleme bei Touch Eingaben eingeführt und diskutiert. In Kapitel 10 wird die Implementierung der erarbeiteten Konzepte in das Applikationssystem beschrieben. Im Anschluss daran erfolgt in Kapitel 11 eine Diskussion zur Erfüllung der durch die Anforderungsanalyse definierten Vorgaben. Des Weiteren werden, entlang der im Arbeitsprozess aufgetretenen Beobachtungen, Hypothesen über das entwickelte System abgeleitet. Abschließend fasst Kapitel 12 die Beiträge dieser Bachelorarbeit zusammen und gibt einen Ausblick auf mögliche zukünftige Arbeiten.

Kapitel 2

Anforderungsanalyse

Zur Erreichung des in Abschnitt 1.3 beschriebenen Ziels der Arbeit wurden einige Anforderungen definiert. Jene Eckpfeiler der Entwicklung sollen in diesem Kapitel näher beleuchtet werden.

Abschnitt 2.1 geht hierbei auf die Benutzung des Systems durch mehrere Nutzer ein. Des Weiteren wird in Abschnitt 2.2 der Umgang mit Inhalten in Stereoparallaxe analysiert. Abschnitt 2.3 bezieht die Interaktionsziele bei der Nutzung des Systems in die Betrachtung ein. Durch Abschnitt 2.4 soll die Kontrolle der bei der Manipulation involvierten Freiheitsgrade an verschiedene Anforderungen gebunden werden. Zuletzt wird in Abschnitt 2.5 die Relevanz visueller Rückmeldung bei der Interaktion fokussiert.

2.1 Mehrbenutzer Touch-Eingaben

Anforderung 1 *Fehleingaben durch gleichzeitiges Interagieren verschiedener Nutzer mit dem Bildschirmtisch sollen vermieden werden.*

Die meisten physischen Interaktionswerkzeuge können von nicht mehr als einem Nutzer bedient werden. Selbst wenn die Auswirkung der Manipulation die ganze Gruppe betrifft, so obliegt der Umgang mit dem Werkzeug einer Person. Im Gegensatz dazu, steht der Projektionstisch allen Nutzern gleichermaßen zur Verfügung. Da eine Vielzahl verschiedener Fingerpositionen auf dem Touchtisch registriert werden können besteht die Möglichkeit, dass der Tisch zur gleichen Zeit von mehreren Personen Eingaben erhält. Diese Eingabekombinationen können leicht zu Manipulationen der Applikation führen, welche vom einzelnen Nutzer nicht beabsichtigt sind und somit vermieden werden sollten.

2.2 Tiefenwahrnehmung durch Stereo- und Bewegungsparalaxe

Anforderung 2 *Konflikten der Tiefenwahrnehmung durch Berührung negativ-parallaxer Bildareale sollte entgegen gewirkt werden.*

Durch Stereoskopische Rendering Verfahren wird die Illusion der dreidimensionalen Lage virtueller Objekte vermittelt. Während die Projektion an eine zweidimensionale Bildebene gebunden ist, erscheinen Applikationsinhalte oftmals abseits der Bildebene. Abschnitt 5.2 beschreibt einige Konflikte der Tiefenwahrnehmung, die bei der taktilen Interaktion mit stereoskopisch visualisierten Modellen entstehen können. Es gilt geeignete Techniken zu finden, welche dem Nutzer dennoch ein immersives virtuelles Erlebnis gewährleisten.

2.3 Interaktionsziele

Anforderung 3 *Manipulationen der Applikation sollen basierend auf den Interaktionszielen der Nutzer bestimmt werden.*

Interaktionsziele variieren anhand des Szenarios der Anwendung. Das System wurde vor dem Hintergrund des EU Forschungsprojekts 3D-Pitoti entwickelt. Es soll demnach die interaktive Exploration eines 3D gescannten Tals unterstützen. In diesem Kontext gilt es geeignete Navigationstechniken abzuleiten, welche dem Nutzer die Betrachtung aller dargebotenen Detailstufen des Modells ermöglichen.

2.4 Explizite und implizite Kontrolle der involvierten Freiheitsgrade

Anforderung 4 *Das System soll auch für unerfahrene Nutzer leicht benutzbar sein.*

Der zweidimensionale Interaktionsraum für Berührungseingaben ist gegenüber dem dreidimensionalen Applikationsraum unterdefiniert. Dieser Zusammenhang äußert sich in den Freiheitsgraden, welche für die verfügbaren Transformationen parametrisiert werden müssen. Demnach ist beispielsweise die Translation im zweidimensionalen Raum durch x- und y-Verschiebung definiert. Im dreidimensionalen Raum hingegen kommt die z-Komponente als Freiheitsgrad hinzu. Die Bewegung eines Punktes im zweidimensionalen Raum kann somit

lediglich zwei der drei Freiheitsgrade der 3D Translation variabel parametrieren. Die Unterbestimmung kann im Kontext der Berührungseingabe durch zusätzliche Kontaktpunkte gelöst werden. Hierdurch wird die Komplexität resultierender Gesten verglichen zu bekannten 2D Touch Techniken gesteigert. Nach Anforderung 4 sollte jedoch auch dem ungeübten Nutzer ein Umgang mit dem System nach kurzer Einarbeitung möglich sein.

Anforderung 5 *Es soll dem Nutzer möglich sein alle für das jeweilige Interaktionsziel erforderlichen Freiheitsgrade getrennt voneinander zu bedienen.*

Für die Detailbetrachtung der durch die Visualisierung dargebotenen Inhalte, ist eine präzise Kontrolle einzelner Parameter bei der Transformation unerlässlich. Das System sollte deshalb nicht nur Modi zur Kontrolle aller Freiheitsgrade bereitstellen, sondern auch die unabhängige Kontrolle einzelner ermöglichen.

Anforderung 6 *Der Nutzer soll bei der Erreichung von Interaktionszielen implizit vom System unterstützt werden.*

Wie in Anforderung 4 beschrieben, soll das System einer intuitiven Bedienung unterliegen. Hierzu gilt es Techniken zu entwickeln, welche durch den Nutzer gesteuerte Manipulationsvorgänge sinnvoll erweitern. Diese Ansätze sollen die ausgearbeiteten Interaktionsziele berücksichtigen und automatisch zur Erreichung dieser beitragen.

2.5 Visuelle Rückmeldung

Anforderung 7 *Der Umgang mit der Multi-Touch Navigation soll mit visuellem Output unterstützt werden.*

Durch einen hohen Grad der Immersion virtueller Umgebungen, verschwimmt der Bezug zu physikalischen Gegebenheiten im Arbeitsbereich. Stereoskopische Visualisierung verstärkt diesen Zusammenhang. Demnach wird beispielsweise die Sichtbarkeit der Projektionsfläche erschwert. Da diese als gleichzeitiges Eingabegerät essentiell für den Umgang mit der Anwendung ist, kann visuelle Rückmeldung helfen die Benutzung des Systems zu erleichtern. Hierzu sollen effektive Visualisierungen für Interaktionsvorgänge angeboten werden, um diese verständlich zu machen und eine leichte Eingewöhnung in die Applikation zu ermöglichen.

Kapitel 3

Finger und Hand Tracking auf Grundlage von Maximally Stable Extremal Regions

Im Rahmen dieser Arbeit wird ein Bildschirmtisch zur Projektion dreidimensionaler Inhalte verwendet, welcher Multi-Touch Eingaben erlaubt. Gegenüber handgehaltenen Geräten, wie Smartphones, oder Tablets, bietet dieser eine weit größere Interaktionsfläche. Diese begünstigt, wie in Abschnitt 1.1 beschrieben, die kollaborative Arbeit in einem Mehrbenutzerszenario. Die meisten multi-touch gesteuerten Eingabegeräte ordnen alle erkannten Fingerpositionen und deren Bewegung derselben Geste zu. Bei gleichzeitiger Bedienung durch mehrere Nutzer, können nach diesem Ansatz leicht Eingabekonflikte entstehen. Um dem Problem entgegenzuwirken ist ein System erforderlich, welches Aussagen über die hierarchische Zuordnung der Eingabepositionen treffen kann.

Hierzu wird in dieser Arbeit die von Ewerling et al. entwickelte Finger und Hand Tracking Implementierung, auf Grundlage des Maximally Stable Extremal Regions (MSER) Algorithmus verwendet [Matas et al., 2004] [Ewerling et al., 2012]. Der theoretische Ansatz dieser Lösung wird in Abschnitt 3.1 vorgestellt. Abschnitt 3.2 definiert die technischen Voraussetzungen zur Umsetzung des Multi-Touch Tisches, gefolgt von einem Überblick über die softwareseitige Integration des MSER Trackings in das genutzte Applikationsframework in Abschnitt 3.3. Zuletzt werden die Vorteile und Limitierungen des Systems in Abschnitt 3.4 diskutiert.



Abbildung 1: Das von der Infrarot Kamera eingefangene Rohbild zur Toucherkennung.

3.1 Maximally Stable Extremal Regions

Maximally Stable Extremal Regions ist ein Ansatz entwickelt von Matas et al. zur Auswertung von Bildreihen bei der photogrammetrischen Stereo-Rekonstruktion [Matas et al., 2004]. Hierbei sollen vor allem eigenschaftsbezogene Korrespondenzen zwischen Stereobildpaaren mit unterschiedlichem Blickpunkt gefunden werden. Ewerling et al. nutzen diese Erkenntnisse zur Erkennung von Fingerspitzen auf dem Bildschirm [Ewerling et al., 2012]. Es erfolgt dabei außerdem eine Zuordnung von Fingern zu Händen durch den *MSER Component Tree*. Dieses Verfahren ist motiviert durch die visuelle Wahrnehmung des Hand- und Armschattens auf den Bildern, welche durch Tracking mit diffuser Infrarotbeleuchtung erfasst werden (siehe Abbildung 1).

Extremal Regions (ER) werden als geschlossene Gruppe von Pixeln bezeichnet, welche entweder eine höhere oder eine niedrigere Farbintensität als umliegende Bildpunkte aufweisen. Die ER R_i ist eine Maximally Stable Extremal Region, wenn R_i eine anderen ER (R_{i-1}) umschließt und es eine umschließende ER (R_{i+1}) für R_i gibt. Es muss folglich $R_{i-1} \subset R_i \subset R_{i+1}$ gelten. Außerdem muss das Stabilitätskriterium ($s_\Delta(i)$) für i ein lokales Minimum erreichen. Folgende Formel beschreibt die Berechnung von $s_\Delta(i)$:

$$s_\Delta(i) = \frac{|R_{i+\Delta}| - |R_{i-\Delta}|}{|R_i|}$$

, wobei Δ eine durch den Anwender definierte Konstante ist.

Ewerling et al. nutzen einen erweiterten Ansatz zur Erkennung der MSER in linearer Zeit, nach dem Vorbild von Nistér et al. [Nistér and Stewénius, 2008]. Bei der direkten Verwendung der Methode von Nistér et al. ergibt sich eine hierarchische Struktur zur Beschreibung aller gefundenen MSER, basierend auf dem Stabilitätskriterium. Genannte Struktur wird als *Component Tree* bezeichnet. Die Konstruktion dieses Component Trees sei jedoch nach Ewerling et al. unvollständig im Hinblick auf für die Beschreibung der Finger-Hand-Relation wichtige ER. Der erweiterte Algorithmus wurde daher entwickelt, um alle verfügbaren ER zu analysieren.

Anhand der ER im Component Tree erfolgt die Ableitung der Fingerpositionen und deren Handzuordnung. Hierfür definieren Ewerling et al. zwei grundlegende Regeln:

1. Die dunkelste und gleichzeitig größte gefundene ER ist der Wurzelknoten des Component Trees
2. Für alle Kind-Eltern Beziehungen in der Struktur gilt, dass die Kindknoten kleiner sind und eine höhere Farbintensität aufweisen

Wie in Abbildung 2 zu erkennen ist, kreieren Objekte in naher Distanz zum Bildschirm eine hellere ER als weiter entfernte. Nach den gelisteten Kriterien können folglich nur Blattknoten des Baumes Fingerspitzen sein. Um eine robuste Analyse zu erreichen, werden vermeidliche Kandidaten in der untersten Ebene des Baumes auf typische visuelle Eigenschaften von Fingeraufsetzpunkten getestet. Hierzu zählen das Prüfen auf runde Form und eine definierte Durchschnittsgröße der erkannten Fingerregion. Die Zuordnung der gefundenen Fingerspitzen zu Händen erfolgt über ein hierarchisches Clustering der Knoten im Component Tree.

3.2 Technische Voraussetzungen

Für die technische Umsetzung des in Abschnitt 3.1 beschriebenen MSER Systems im Kontext der Multi-Touch Erkennung wird in unserem Labor ein rückseitiges, diffuses Infrarot-Beleuchtungssetup genutzt. Hierbei wird eine Infrarot emittierende Lichtquelle unterhalb des Bildschirms angebracht. Diese bestrahlt eine matte und lichtdurchlässige Projektionsfläche. Das von dieser



Abbildung 2: Die Helligkeit naher Objekte ist im Infrarot Bild höher als bei weiter entfernten.

Ebene reflektierte Licht wird durch den Spiegel gelenkt und von einer Infrarotkamera zur weiteren Verarbeitung aufgenommen. Abbildung 3 visualisiert das beschriebene Setup.

Die Genauigkeit des Systems kann durch infrarotes Umgebungslicht, wie beispielsweise Anteile der Sonnenstrahlung, oder andere im Arbeitsraum befindliche Trackingsysteme, gestört werden. Ewerling et al. schlagen daher eine hochfrequente Modulation der Beleuchtung nach Moeller und Kerne vor [Ewerling et al., 2012, Moeller and Kerne, 2012]. Nicht-uniforme Lichtintensität entlang der Tischfläche kann ebenfalls zu Problemen bei der Bildanalyse führen. Diese können zwar durch Filterverfahren abgeschwächt, jedoch nicht vollständig beseitigt werden. Eine gleichmäßige Ausleuchtung der Projektionsebene ist demnach ausschlaggebend für eine akkurate Evaluation der Kamerabilder.

3.3 Implementierung

Es existiert eine Anbindung des MSER Algorithmus zur Touch Erkennung, an das am Lehrstuhl entstandene Applikationsframework Avango [Avango,]. Die analysierten Eingabedaten werden basierend auf dem TUIO Protokoll an die Anwendung gesendet [TUIO,].

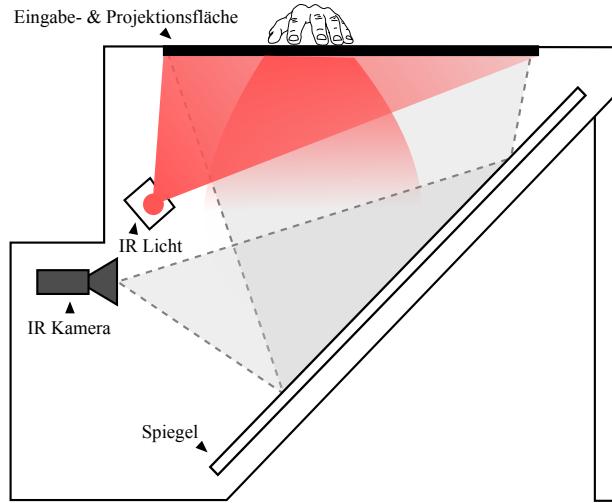


Abbildung 3: Das im Labor verwendete Hardware Setup zum Tracking von Touch Eingaben.

In Abschnitt 3.3.1 wird eine Übersicht über die mit dem TUIO Protokoll bereitgestellten Datenstrukturen gegeben. Abschnitt 3.3.2 beschreibt das Auftreten von Jittering bei der Eingabeerkennung und wie mit diesem Problem umgegangen werden kann.

3.3.1 TUIO Touch Protokoll

TUIO ist ein von Kaltenbrunner et al. entwickeltes Protokoll, welches speziell für den Umgang mit berührbaren Tabletop Eingabeschnittstellen kreiert wurde [Kaltenbrunner et al., 2005]. Nach der Auswertung der Eingabepositionen durch den MSER Algorithmus, füllt die Protokollkommunikation verschiedene Stations im Avango Daemon mit den ermittelten Daten. Es existieren zwei Stationstypen. Der Erste hält Informationen über einen jeweiligen Finger. Dies beinhaltet eine normalisierte Fingerposition und die zugewiesene *SessionID*. Ein weiterer Stationstyp ist für die Verwaltung handspezifischer Details verantwortlich. Hierzu gehören die *SessionID* der Hand, eine Liste von *FingerSessionIDs* der zugewiesenen Finger, sowie eine zweidimensionale Orientierung des Armes. Letztere ist durch die Auswertung der handumschließenden Ellipse gegeben.

Eine Abbildung dieser Struktur ist nach dem Feldcontainer Konzept, welches Avango bereitstellt, in der Applikation umgesetzt. Für jeden der Stationstypen existiert demnach eine Klasse, welche durch Update der Station aktualisierte

Eingabewerte für die Anwendung verfügbar macht. Die nachfolgende Abbildung XX zeigt den Aufbau der Klassen *TUIOFinger* und *TUIOHand*.

3.3.2 Umgang mit Jittering

Die Verfolgung der Bewegung von Eingabepunkten basiert auf der Auswertung von Bilddaten. Hierbei ist die Präzision der Positionsermittlung an die Genauigkeit des MSER Algorithmus, sowie die Auflösung der Bilder gebunden. Infolgedessen können, selbst bei Auflegen eines Fingers ohne Positionsverschiebung, variierende Eingabepositionen entstehen. In einer Visualisierung der Eingabeposition äußert sich dieser Zusammenhang als Zittern (Jittering) der Darstellungsgeometrie.

Jittering kann bei Anwendung der Eingabe auf die Manipulation einer virtuellen Szene ungewollte Auswirkungen hervorrufen. Bindet man eine Szene beispielsweise an die relative Positionsänderung einer durch Jittering beeinträchtigten Eingabe, so überträgt sich das Zittern auf die Szene, was als störend für die Wahrnehmung empfunden werden kann. Die Initiierung einer Bewegung sollte folglich weitestgehend dem Anwender überlassen sein.

Um Jittering zu vermindern wird vor der Übermittlung von Daten durch das TUIO Protokoll ein Filter auf die Eingabepositionen angewandt. Hierzu wird der von Casiez et al. vorgestellte *1€ Filter* verwendet [Casiez et al., 2012]. Dieser Ansatz überzeugt vor allem durch seine niedrige Rechenintensität, sowie die geringe Latenzanfälligkeit bei schnellen Bewegungen.

3.4 Vorteile und Limitierungen

Das MSER-System dient zur Erkennung von relationalen Eingabedaten. Es können somit Aussagen über die hierarchische Zuordnung einzelner Fingerpositionen zu aufgelegten Händen gemacht werden. Dieser Zusammenhang ist essenziell für die im Rahmen dieser Arbeit entstandenen Techniken.

Im Umgang mit der Toucherkennung wurden im Verlauf der Programmierarbeiten jedoch immer wieder verschiedene Probleme deutlich. Demnach wird eine Finger-Hand Zuweisung zwar unterstützt, nicht konsequent stabil gehalten. Die Variation der Höhe der Handfläche über dem Tisch kann beispielsweise zum Auflösen der Verbindung einzelner Teile der Wurzel ER einer Hand führen. In diesem Fall stellt sich die Eingabe einer Hand mit fünf aufgelegten Fingern, als mehrere Einzelne Hände mit einem Finger dar.

Das gleichzeitige Aufsetzen zweier Hände eines Nutzers wird nur dann problemfrei erkannt, wenn zwei separate ER für die Arme entstehen. Beugt sich der Anwender über die Tischplatte um mit beiden Händen in weit entfernten Bereichen zu interagieren, kann leicht eine Verschmelzung der ER um beide Arme entstehen. Dies beeinträchtigt eine kollisionsfreie Handerkennung weiter. Dementgegen ist die Erkennung und Verfolgung einzelner Eingabepunkte relativ stabil. Lediglich schnelle Bewegungen über die Tischfläche sind hierbei problematisch und führen zum Verlust von Fingerpositionen.

Kapitel 4

3D Interaktion mit Multitoucheingaben

Die Interaktion mit Multi-Touch Eingaben wurde in einigen Veröffentlichungen thematisiert. Hierbei haben sich unterschiedliche Strategien zur Kontrolle der Freiheitsgrade bei der Manipulation von dreidimensionalen Szenen bewährt. Die im Rahmen dieser Arbeit entstandenen Ansätze zur Umsetzung einer Multi-Touch basierten 3D Navigation stützen sich auf diese Erkenntnisse.

Dieses Kapitel stellt die in diesem Kontext wichtigsten verwandten Arbeiten vor. Hierzu beschreibt Abschnitt 4.1 ein System zur 6DOF Interaktion mit Multi-Touch Tischen. In Abschnitt 4.2 wird ein Ansatz zur Selektion von virtuellen Objekten, durch Touch Eingaben erläutert. Abschnitt 4.3 erklärt die Umsetzung einer expliziten Geste zur Steuerung der 3D Rotation. Abschließend erfolgt in Abschnitt 4.4 eine Diskussion genannter Techniken.

4.1 Sticky Tools

Hancock et al. stellen ein System zur Steuerung von sechs Freiheitsgraden der dreidimensionalen Objekttransformation durch Multi-Touch Eingaben vor, welches sie Sticky Tools nennen [Hancock et al., 2009]. Sie stützen sich dabei auf bekannte Techniken welche als *force-based* bezeichnet werden. Hierbei soll das Gefühl erzeugt werden, die virtuelle Geometrie wie in der realen Welt durch direktes Anfassen zu manipulieren.

Die x- und y-Translation, uniforme Skalierung, sowie z-Rotation mit zwei Fingern hat sich im Alltag, durch den Umgang mit zweidimensionalen Touch-Anwendungen, bewährt. Es wird hierbei ein Initialkontakt mit der Geometrie bestimmt. Während der Bewegung der Finger, bleibt dieser durch Anwen-



Abbildung 4: Durchführung der Opposable Thumbs Rotation im von Hancock et al. entwickelten System Sticky Tools [Hancock et al., 2009]. Screenshots sind dem zugehörigen Video entnommen und durch eigene Visualisierungen erweitert worden [Hancock, 2011].

dung der genannten Transformationen, erhalten. Dieses Ankleben der Objekte an die Berührungspunkte auf dem Bildschirm bezeichnen Hancock et al. als *Sticky Fingers*-Technik [Hancock et al., 2007, Hancock et al., 2009].

In dreidimensionalen Szenen wird die Translation um die z-Komponente erweitert. Hancock et al. nutzen für die Visualisierung der virtuellen Szene eine zweidimensionale Darstellung. Durch die perspektivische Verzerrung wächst der projizierte Abstand zweier Punkte auf einem Objekt, je mehr sich die Geometrie der Projektionsebene nähert. Sticky Tools simuliert den physikalischen Umgang mit nicht-elastischen Objekten. Aus diesem Grund wird auf eine Geste zur Skalierung verzichtet. Folglich wird die 2D Skalierungsgeste zur z-Translation eines Objekts genutzt.

Neben der Rotation um die Hochachse ermöglicht die Manipulation in dreidimensionalen Szenen das Drehen von Objekten um Achsen auf der Bildfläche. Hancock et al. definieren letztere als *flip*-Rotation. Um das Sticky Fingers Kriterium zu erhalten, wird diese Art der Rotation durch beidhändige Interaktion gesteuert. Hierbei werden durch das Aufsetzen zweier Finger einer Hand die direkten Kontaktpunkte auf der Geometrie festgelegt. Diese spannen einen Richtungsvektor auf, welcher die Rotationsachse definiert. Die Bewegung eines aufgesetzten Fingers der zweiten Hand im rechten Winkel zur Rotationsachse, legt den Grad der Flip-Rotation, sowie die Richtung der Drehung fest (siehe Abbildung 4).

4.2 Balloon Selection

Balloon Selection ist eine von Benko und Feiner [Benko and Feiner, 2007] entwickelte Technik zur Multi-Touch basierten Selektion von virtuellen Objekten in dreidimensionalen Szenen. Dieser Ansatz ist inspiriert vom Umgang mit ei-

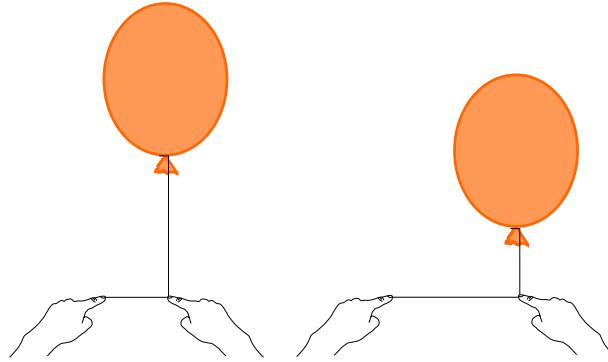


Abbildung 5: Aufstiegsveränderung eines mit Helium gefüllten Ballons.

nem Helium Ballon. Demnach wird das Ende der Leine eines solchen Ballons mit einem Finger auf den Boden gehalten. Ein zweiter Finger fixiert die Leine an einem anderen Punkt auf dieser Ebene. Die Länge des fixierten Stückes bestimmt die Aufstiegsdistanz des Ballons. Folglich kann diese Distanz, sowie die Position des Ballons über dem Boden, durch das Verschieben der Handpositionen variiert werden. Abbildung 5 visualisiert diese Metapher.

Bei der Anwendung dieser Idee für die Selektion, wird eine Kugel als Selektionsobjekt genutzt. Die abgeleitete Geste wird durch das Aufsetzen zweier Finger in unmittelbarer Nähe zueinander initiiert. Hierbei dient der zuerst aufgelegte Finger als Ankerpunkt (nach Benko und Feiner *anchor*). Der zweite Finger (nach Benko und Feiner *stretching finger*) wird zur Festlegung der Leinenlänge verwendet. Hierzu bewegt der Nutzer seine Hand vom Ankerpunkt weg. Die Interaktionsleine wächst bis diese Bewegung sich in entgegen gesetzte Richtung umkehrt. Ab diesem Punkt ist die Länge der Leine fest und die Kugel steigt um die Länge der Distanzverkürzung zwischen den beiden Fingern orthogonal zur Tischemebene nach oben. Der Ankerpunkt bestimmt durch Positionsveränderung die x- und y- Position des Ballons über der Interaktionsfläche.

4.3 Two Axis Valuator

Rousset et al. beschreiben eine Erweiterung von Scheurich und Stuerzlingers *Two Axis Valuator* (im Folgenden TAV genannt) [Scheurich and Stuerzlinger, 2013] [Rousset et al., 2014]. Der von Elisabeth Rousset et al. entwickelte TAV+ ist ein Modus zur einhändigen 3D Rotation durch die Verwendung von zwei Fingern.

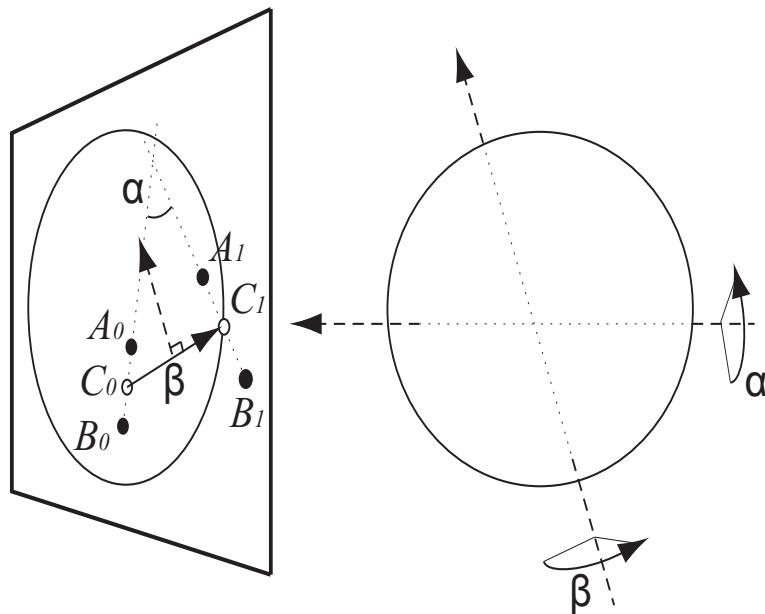


Abbildung 6: Berechnung der dreidimensionalen Rotation TAV+ nach Rousset et al. [Rousset et al., 2014]. Hierbei sind A_i und B_i die zwei zur Interaktion verwendeten Finger. C_i ergibt sich als die Zentrumsposition zwischen A_i und B_i . α und β sind die durch das Verfahren ermittelten Rotationswinkel. Diese Abbildung entstammt vollständig einer Veröffentlichung von Rousset et al [Rousset et al., 2014].

Für die Bestimmung der Rotation entlang einer beliebigen Achse auf der Bildebene, wird die Bewegung der Zentrumsposition zwischen den aufgesetzten Fingern verfolgt. Die Rotation wird hierbei um die Achse, welche rechtwinklig zur Bewegungsrichtung steht und ebenfalls in der Bildebene liegt, durchgeführt. Als Rotationszentrum dient der Schwerpunkt des zu manipulierenden Objekts. Drehrichtung und Rotationsgrad werden von der Distanz und Richtung der Translation des Fingerzentrums abgeleitet. Zusätzlich kann die Rotation um die Achse zwischen Fingerzentrum und Rotationszentrum kontrolliert werden. Dies ist durch Orientierungsänderung des Vektors zwischen den Fingern möglich. Der verwendete Winkel entspricht dem zwischen dem Vektor vor der Bewegung und demselben danach. Abbildung 6 illustriert dieses Verfahren.

4.4 Diskussion

Der direkte Umgang mit negativ-parallaxen Inhalten zeigt sich bei Touch-Interaktion mit stereoskopischen Visualisierungen als problematisch. Benko und Feiner kommen zu dem Schluss, dass Baloon Selection ein unter verschiedenen Umständen geeigneter Ansatz zur Selektion von über einer Interaktionsoberfläche liegenden virtuellen Objekten ist [Benko and Feiner, 2007]. Eine Umkehrung der Selektionsrichtung könnte das Verfahren auch für die 3D Multi-Touch Interaktion sinnvoll einsetzbar machen. Die Selektion von virtuellen Inhalten wird im Kontext dieser Arbeit nicht betrachtet. Jedoch ist eine explizite Höhenanpassung der Navigation, basierend auf den Erkenntnissen von Benko und Feiner, entstanden. Diese wird in Abschnitt 6.2 näher erläutert.

Sticky Tools zeigt sich durch seine physikalisch motivierte Entwicklung als intuitiv und nutzerfreundlich für die 3D Objektmanipulation. Aus diesem Grund fließen einige der von Hancock et al. entwickelten Konzepte maßgeblich in die entstandene 3D Multi-Touch Navigation ein. Sticky Tools ersetzt die Funktionalität der herkömmlichen 2D Skalierungsgeste mit einer expliziten Höhenanpassung. Durch die perspektivische zweidimensionale Projektion geht hierbei visuell die direkte Verbindung zu den Kontaktpunkten auf der Geometrie nicht verloren. Stereoskopisches Rendering kann diese Illusion jedoch nicht gewährleisten. Zum Erhalt der Orthogonalität ist in diesem Anwendungsszenario eine uniforme Skalierung unabdingbar.

Der Umgang mit Objekten ist bei Sticky Tools auf geringe Entfernung zur Bildecke begrenzt. Durch das Aufsetzen zweier Finger auf die Touch-Fläche werden Referenzpunkte auf der Geometrie festgelegt. Aus diesen Punkten wird eine Achse für die flip-Rotation abgeleitet. Weisen die Referenzpunkte einen Höhenunterschied auf, so wirkt sich dieser auf die hervorgerufene Drehung aus. Auf nahe Distanz zu manipulierenden Inhalten, ist die Höhendifferenz gut abschätzbar. Im Umgang mit komplexen Miniaturwelten und bei möglichen weiten Entfernungen zu Objekten, wird diese Abschätzung zunehmend schwerer. Es ist in diesem Kontext von einer Beeinträchtigung der Nutzbarkeit der flip-Rotation auszugehen.

TAV+ wird von Rousset et al. als 3D Rotationstechnik, welche für unerfahrene Nutzer einfach bedienbar ist, bewertet. Im direkten Vergleich mit anderen 3D Rotationstechniken schneidet TAV+ gut ab. Hierbei wird der Ansatz im Szenario einfacher Observationsaufgaben einzelner Objekte mit festen Positionen untersucht. Die Wahl des Rotationszentrums im Schwerpunkt der Geometrie erscheint in diesem Zusammenhang als sinnvoll. Bei der Navigation wird

der Blickpunkt auf die Szene und somit alle virtuellen Objekte fortlaufend geändert. In diesem Fall wäre die Bedienung der TAV+ Rotation mit festem Rotationsreferenzpunkt sicherlich wenig nützlich. Abschnitt 6.3 beschreibt wie der von Rousset et al. entwickelte Ansatz abgeändert wurde, um eine effektive 3D Rotation der Navigation zu erreichen.

Kapitel 5

Wahrnehmungskonflikte und Lösungsansätze

Die Darstellung virtueller Szenen durch stereoskopisches Rendering erzeugt die Illusion, dass projizierte Inhalte unter, auf oder über der Bildschirmfläche liegen. Eine konfliktfreie Wahrnehmung der Visualisierung kann durch verschiedene Faktoren eingeschränkt werden. Multi-Touch Eingaben wirken sich durch direkten Kontakt mit dem Projektionssystemen zusätzlich auf bestehende Probleme aus. In diesem Kapitel werden einige der hervorgerufenen Wahrnehmungskonflikte vorgestellt. Zusätzlich werden bestehende Lösungsansätze aufgezeigt.

Abschnitt 5.1 führt hierzu in die Thematik *Frame Cancellation* ein. In Abschnitt 5.2 werden Probleme und Lösungsbestrebungen beschrieben, welche aufgrund von Touch Interaktion mit stereoskopischen Visualisierungen entstehen. Das Kapitel wird in Abschnitt 5.3 durch eine Diskussion dieser verwandten Arbeiten abgeschlossen.

5.1 Frame Cancellation

Frame Cancellation ist ein Problem, welches in den frühen Jahren der stereoskopischen Filmproduktion von Valyus und Asher erstmals öffentlich benannt wurde [Valyus and Asher, 1966]. Der Wahrnehmungskonflikt entsteht durch eine auftretende Widersprüchlichkeit des Tiefeneindrucks an den Begrenzungen der Projektionsfläche. Bei der Darstellung negativ-parallaxer Inhalte vermittelt die Disparität der Geometrie demnach, dass sich die Darstellung vor dem Bildschirm befindet. Gleichzeitig überlagert dieser durch seine physikalische Begrenzung das Objekt.

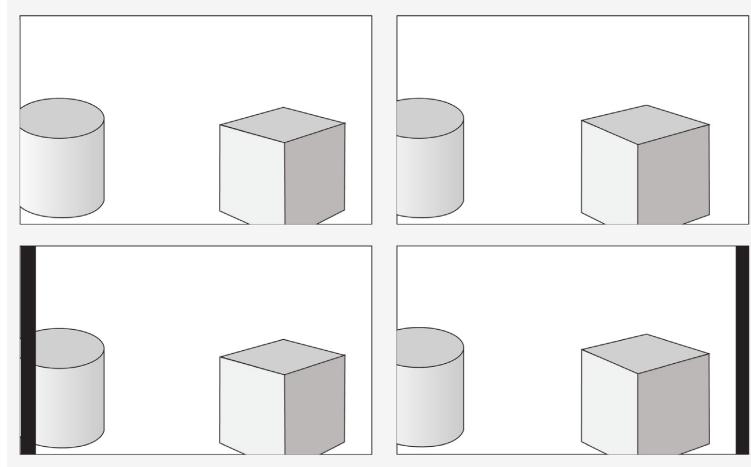


Abbildung 7: Die oberen Abbildungen zeigen die Bildausschnitte der Augen vor Anwendung der *Black-Bands* Technik nach Autodesk [Autodesk, 2008]. In den darunter liegenden Abschnitten ist die Veränderung dieser nach der Verwendung des Ansatzes zu sehen. Diese Abbildung entstammt vollständig einer Veröffentlichung von Autodesk [Autodesk, 2008].

Wartell beschreibt eine durch *Frame Cancellation* entstehende schwache Tiefeinwahrnehmung der Szene [Wartell, 2001]. Nach Lipton resultiert außerdem ein unangenehmer visueller Eindruck durch die Schwierigkeit die Bilder der beiden Augen zu vereinigen [Lipton, 2007].

Zum Umgang mit diesem Problem, schlägt Autodesk die Verwendung von *Black Bands* vor [Autodesk, 2008]. Nach diesem Ansatz werden an den äußeren Begrenzungen des Bildes, der jeweiligen Augen schwarze Balken eingeschoben. Abbildung 7 zeigt, dass ohne Anwendung der Technik im Bild des rechten Auges ein kleinerer Ausschnitt der Geometrie zu sehen ist als im rechten. Dieser fehlende Ausschnitt im Bild des linken Auges führt zum beschriebenen Konflikt. Durch Einschieben der *Black Bands* soll die konfliktfreie Verschmelzung der von beiden Augen wahrgenommenen Bilder ermöglicht werden.

Ardouin et al. schlagen die Verwendung von Clipping zur Begrenzung des Blick Frustums vor [Ardouin et al., 2011]. Der Schnittbereich der Frusta beider Augen wird nach Ardouin *Stereo Compatible Volume* genannt [Ardouin et al., 2011]. In diesem Bereich sei eine konfliktfreie Visualisierung negativ-parallaxer Szeneninhalte sicher. Das Abschneiden der Geometrie außerhalb des Stereo Compatible Volumes (nach Ardouin *Stereo Compatible Volume Clipping*) würde demnach *Frame Cancellation* vermeiden. Abbildung 8

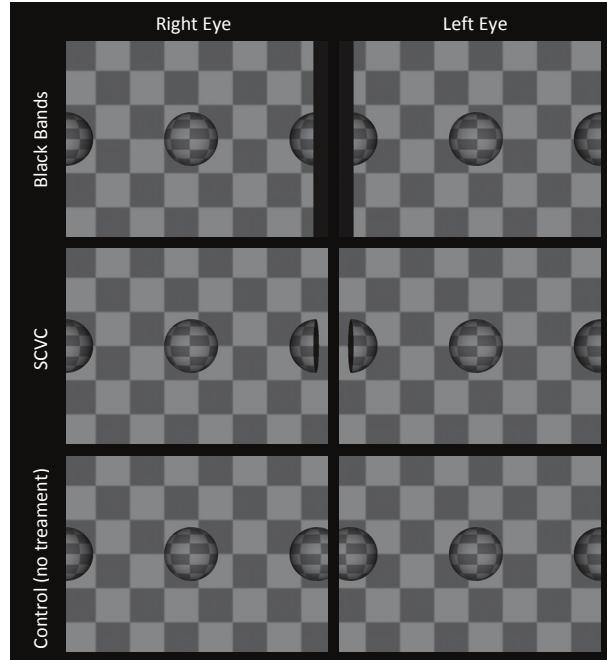


Abbildung 8: Gegenüberstellung der Techniken *Stereo Compatibile Volume Clipping* nach Arduin et al [Arduin et al., 2011] und *Black-Bands* nach Autodesk [Autodesk, 2008]. Zum Vergleich wird eine unveränderte Darstellung der Bildausschnitte gezeigt. Diese Abbildung entstammt vollständig einer Veröffentlichung von Arduin [Arduin et al., 2011].

zeigt eine Gegenüberstellung der Technik von Arduin et al. mit *Black-Bands*.

5.2 Touch Interaktion mit stereoskopischen Visualisierungen

Bei der Touch Interaktion ist die Selektion und Manipulation von virtuellen Objekten auf und abseits der Projektionsebene ausschließlich durch Eingaben auf der Tischemebene zu erreichen. Daraus leitet sich eine Wahrnehmungsdiskrepanz ab [Valkov et al., 2011, Bruder et al., 2013]. Demnach ist die Wahl des Fokuspunktes für die Beobachtung des Interaktionsvorgangs doppeldeutig. Ein Objekt in Negativ-Parallaxe liegt oberhalb der Tischemebene, sodass bei Fokussierung von selbigem die Blickrichtung der beiden Augen über der Projektionsfläche konvergiert. Die Wahrnehmung der interagierenden Hand auf dem Bildschirm würde folglich verschwimmen. Umgekehrt würde die Wahr-



Abbildung 9: Wahrnehmungsdiskrepanz durch die Wahl des Fokuspunkts. Diese Abbildung entstammt vollständig einer Veröffentlichung von Bruder et al. [Bruder et al., 2013].

nehmung des parallaxen Inhalts bei Fokussierung der Hand verschwimmen. Abbildung 9 verdeutlicht diesen Zusammenhang.

Giesler et al. entwickeln mit ihrer Technik *Void Shadows* einen Ansatz um die Touch Selektion von nicht auf der Bildfläche liegenden Objekten zu erleichtern [Giesler et al., 2014]. Hierzu wird eine zweidimensionale Repräsentation aller Geometrie als Schatten auf die Bildebene projiziert. Berührungen eines Schattens werden auf die jeweilig referenzierte Geometrie übertragen.

Ein weiteres Problem ergibt sich durch das Eindringen der Hand in über dem Tisch liegende Geometrie. Berührt der Anwender die Tischebene zur Interaktion, durchstößt er dabei die auf dem Pfad der Bewegung liegenden negativ-parallaxen Objekte. Während dem Nutzer durch die Disparität der Darstellung eine geometrische Lage der Inhalte oberhalb der Hand suggeriert wird, bleiben diese verdeckt bis die Hand des Nutzers die Eingabefläche verlässt. Folglich wird die Tiefenwahrnehmung durch die widersprüchliche Lage des projizierten Inhalts in Relation zur Hand des Nutzers gestört. Hierdurch wird der visuelle Eindruck wie bei der *Frame Cancellation* (siehe Abschnitt 5.1) beeinflusst.

Um dieser Problematik entgegenzuwirken verfolgt das von de la Rivière et al. entwickelte System die Bewegung der Hand Nutzer über dem Tisch [de la Rivière et al., 2010]. Nähert sich diese der Tischfläche, wird die Szene entlang der Bildschirmnormalen in Blickrichtung des Nutzers verschoben, bis alle virtuellen Inhalte auf oder unter der Projektionsfläche liegen. Sie verhindern somit mögliche Kollisionen der Hand mit allen dargestellten Objekten.

5.3 Diskussion

Stereoskopische Projektionssysteme erzeugen durch physikalisch motiviertes Rendering die Illusion einer tatsächlichen Dreidimensionalität der dargebotenen Inhalte. Die in den Abschnitten 5.1 und 5.2 vorgestellten Konzepte zeigen jedoch, dass die Form der Darstellung, sowie der direkte Umgang mit projizierten Inhalten zu verschiedenen Wahrnehmungskonflikte führen können. Es entsteht dadurch eine Herausforderung für die nutzerfreundliche Zusammenführung mit einem an die zweidimensionale Tischfläche gebundenen Multi-Touch System.

Für die Lösung des *Frame Cancellation* Problems wurde eine Reihe von Ansätzen vorgestellt, welche als sinnvolle Lösungen des Problems zu sehen sind (siehe Abschnitt 5.1). Die Klärung der von Valkov et al. beschrieben Wahrnehmungsdiskrepanz (siehe Abschnitt 5.2) bei Fokussierung von negativ-parallaxen Objekten in Relation zur auf dem Tisch aufgelegten Hand bleibt fraglich.

In einer Studie vergleichen Bruder et al. die Präzision der 3D Selektion bei Verwendung von 2D Touch Eingaben mit der freien Auswahl im dreidimensionalen Raum (*3D mid-air* Selektion) [Bruder et al., 2013]. Sie kommen zu dem Schluss, dass durch *3D mid-air* Selektion eine höhere Genauigkeit mit schnelleren Interaktionszeiten für Objekte, welche weiter als 10cm abseits der Bildebene liegen, gegeben ist.

Die fehlende Effizienz von 2D Touch Eingaben gegenüber *3D mid-air* Techniken könnte durch Ansätze wie *Void Shadows* verbessert werden. Giesler et al. belegen, dass ihre Technik die Präzision und Interaktionszeit, beim Umgang mit vom Bildschirm entfernten Objekten, deutlich verbessert [Giesler et al., 2014]. Das Anwendungsszenario beschränkt sich bei ihrer Arbeit auf den Umgang mit positiv-parallaxen Inhalten und sollte für Inhalte oberhalb der Tischoberfläche erweitert werden.

Die im System von de la Rivière et al. integrierte Funktionalität beseitigt das Auftreten von Wahrnehmungskonflikten durch Eingreifen in virtuelle Modelle [de la Rivière et al., 2010]. Folgende Kritikpunkte scheinen jedoch naheliegend:

- Der eigentliche Wahrnehmungskonflikt wird nicht gelöst. Man vermeidet lediglich die Konfrontation.
- Die Translation der Szene ohne direkte Eingabe des Anwenders ist fraglich für die Nutzerfreundlichkeit des Systems.

- Das System erlaubt nicht den Umgang mit der Szene in jeder geometrischen Lage

Kapitel 6

Explizite Multi-Touch 3D Navigation

In Kapitel 4 werden einige bekannte Ansätze zur Interaktion durch Multi-Touch Eingaben vorgestellt. Diese werden im Rahmen dieser Arbeit als explizite 3D Multi-Touch Techniken bezeichnet. Wir definieren explizite Multi-Touch Navigation als Strategie, mit welcher verschiedene Freiheitsgrade der Manipulation durch Nutzereingaben direkt steuerbar sind.

In diesem Kapitel wird beschrieben wie anhand dieser Techniken Gesten für die, im Rahmen dieser Arbeit entstandene Applikation zur touch- basierten Navigation, abgeleitet wurden. Abschnitt 6.1 erläutert die Integration der Rotation, Translation und Skalierung im Bildraum, welche im Folgenden mit RTS abgekürzt wird. Abschnitt 6.2 zeigt wie alle Freiheitsgrade der dreidimensionalen Translation in einer Navigationsgeste bedient werden können. In Abschnitt 6.3 wird ein Ansatz zur Steuerung aller Freiheitsgrade der dreidimensionalen Rotation vorgestellt. Abschließend werden in Abschnitt 6.4 die vorgestellten Techniken gegenübergestellt und auf ihre Vorteile und Limitierungen untersucht.

6.1 Rotation, Translation und Skalierung im Bildraum

Zur Umsetzung der RTS-Technik dienen die Erkenntnisse von Hancock et al. [Hancock et al., 2007, Hancock et al., 2009]. Nach dieser Strategie soll zu jedem Zeitpunkt der Interaktion eine orthogonale Verbindung zwischen der Eingabeposition auf dem Bildschirm und der darunter liegenden Geometrie bestehen.

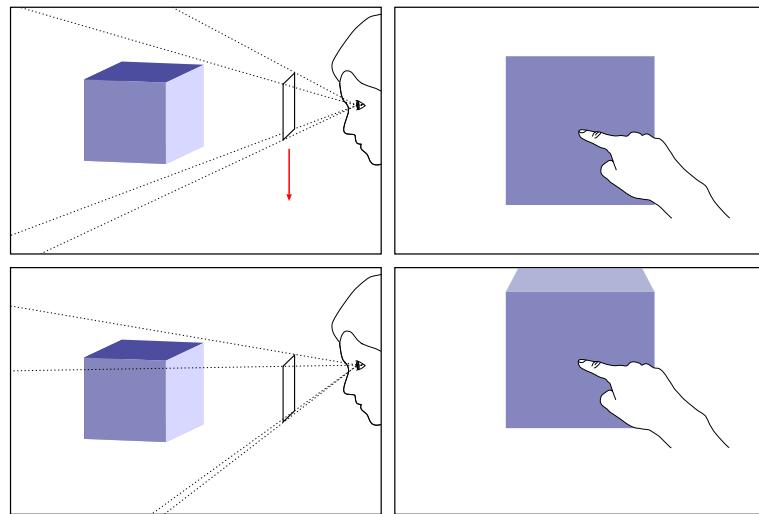


Abbildung 10: Translation der virtuellen Bildfläche und die Auswirkung auf die Darstellung.

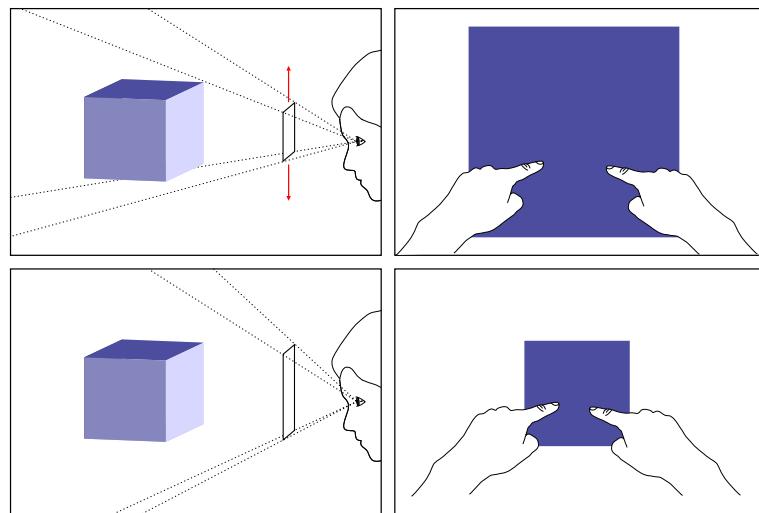


Abbildung 11: Skalierung der virtuellen Bildfläche und die Auswirkung auf die Darstellung.

Wird zur Berechnung der Manipulation ein Kontaktpunkt auf dem Bildschirm verfolgt, können mit diesem Ansatz x- und y-Translation der Navigation gesteuert werden. Hierzu ergibt sich die Verschiebung des Viewing-Setups, aus der relativen Bewegung des Kontaktpunkts auf der Projektionsfläche. Die Abbildungen 10 und 11 visualisieren die Auswirkung von Translation und Skalierung des virtuellen Bildschirms.

Wir definieren $P_1(t) = (x_1, y_1)$ als die Position des Kontaktpunkts P_1 auf der Bildebene, zu einer gegebenen Zeit t . $P_1(t') = (x_1', y_1')$ sei die Position von P_1 zu einer späteren Zeit t' . Die relative Translation $T_r = (x_r, y_r)$ berechnet sich durch $T_r = P_1(t) - P_1(t')$. Um diese Bewegung auszugleichen muss die Navigation um $T = (x_t, y_t) = (-x_r, -y_r)$ verschoben werden.

Mit der Verwendung eines zweiten Kontaktpunkts P_2 auf dem Bildschirm, wird diese Transformation um die Rotation entlang der Bildschirmnormalen, sowie die symmetrische Skalierung erweitert. Als Referenzpunkt hierfür dient P_1 .

Sei $P_2(t) = (x_2, y_2)$ die Position des Kontaktpunkts P_2 auf der Bildebene, zu einer gegebenen Zeit t und $P_2(t') = (x_2', y_2')$ die Position von P_1 zu einer späteren Zeit t' , so ergibt sich $V = P_2(t) - P_1(t)$, als Richtungsvektor zwischen den Kontaktpunkten vor der Bewegung. $V' = P_2(t') - P_1(t')$ ist folglich der Richtungsvektor nach der Bewegung. Verändert sich die Distanz der Kontaktpunkte auf der Tischfläche, so muss sich die Distanz der Angriffspunkte auf der Geometrie in gleichem Maße ändern. Die Größenrelation der dargestellten Geometrie kann durch Skalierung der virtuellen Bildebene beeinflusst werden. Der Skalierungsfaktor S , bestimmt sich nach diesem Zusammenhang aus dem Verhältnis der Distanzen zwischen den Kontaktpunkten zu den Zeiten t und t' . Folglich gilt $S = |V|/|V'|$.

Für die Rotation dienen ebenfalls die Vektoren V und V' zur Berechnung. Als Achse dient die Bildschirmnormale N . Diese kann leicht durch $N = ||V|| \times ||V'||$ bestimmt werden. Der Winkel α für die Transformation ist durch $\alpha = \arccos(||V|| * ||V'||)$ gegeben.

6.2 3D Translation

Die 3D Translation ist ein von den Erkenntnissen der Balloon Selection nach Benko und Feiner [Benko and Feiner, 2007] abgeleiteter Ansatz. Durch Berühren der Tischfläche mit einer Hand wird eine direkte Verbindung mit

der darunter liegenden Geometrie hergestellt. Diese zur Tischfläche orthogonale Beziehung soll zu jedem Zeitpunkt der Navigation in diesem Modus gewahrt bleiben. Es leitet sich demnach für die Interaktion mit nur einem Eingabepunkt eine zweidimensionale Translation, nach den in Abschnitt 6.1 beschriebenen Zusammenhängen, ab.

Durch die Verfolgung der Bewegung zweier Eingabepositionen, kann eine dreidimensionale Translation abgebildet werden. Hierzu dient einer der verfolgten Kontakte auf dem Tisch als Primäreingabe. Die Bewegung dieser Position bestimmt weiterhin die x- und y-Translation im Bildraum. Der zweite Kontakt wird im Folgenden als Sekundäreingabe bezeichnet. Die Einordnung in Primär- und Sekundäreingabe kann durch Auswertung der Startzeit der jeweiligen Eingabe festgelegt werden. Durch Distanzveränderung der Sekundäreingabe zur Primäreingabe kann die z-Verschiebung gesteuert werden. Zur Berechnung dieser bieten sich zwei verschiedene Verfahren an.

Wir definieren $P_1(t) = (x_1, y_1)$ und $P_2(t) = (x_2, y_2)$ als die Positionen der Primäreingabe P_1 und der Sekundäreingabe P_2 auf der Projektionsfläche, zu einer gegebenen Zeit t . $P_1(t') = (x_1', y_1')$ und $P_2(t') = (x_2', y_2')$ seien die Position von P_1 und P_2 zu einer späteren Zeit t' . Weiterhin werden die Richtungsvektoren $V = P_2(t) - P_1(t)$ und $V' = P_2(t') - P_1(t')$ bestimmt. z_t ist die zu ermittelnde z-Translation.

Das erste Verfahren nutzt die Differenz zwischen $|V|$ und $|V'|$ zur Bestimmung der z-Translation. Bei Verlängerung der Distanz zwischen den Eingabepunkten, soll der Abstand zur darunter liegenden Geometrie in gleichem Maße abnehmen. Bei Verkürzung der Strecke zwischen P_1 und P_2 wächst die Distanz zur Geometrie um den Betrag der Differenz. Für ein Viewing Setup, mit Blickrichtung entlang der negativen z-Achse, ergibt sich:

$$z_t = |V| - |V'|$$

Der entstehende Effekt lässt sich mit der Anwendung eines Flaschenzugs vergleichen werden. Im zweiten Verfahren wird das Verhältnis zwischen $|V|$ und $|V'|$ zur Bestimmung der z-Translation genutzt. Verdoppelt sich beispielsweise die Strecke zwischen P_1 und P_2 , so halbiert sich die Distanz zwischen P_1 und dem Geometrieauftreffpunkt (P_g). G ergibt sich als Richtungsvektor zwischen P_1 und P_g

$$z_t = |G| - |G| * \frac{|V|}{|V'|}$$

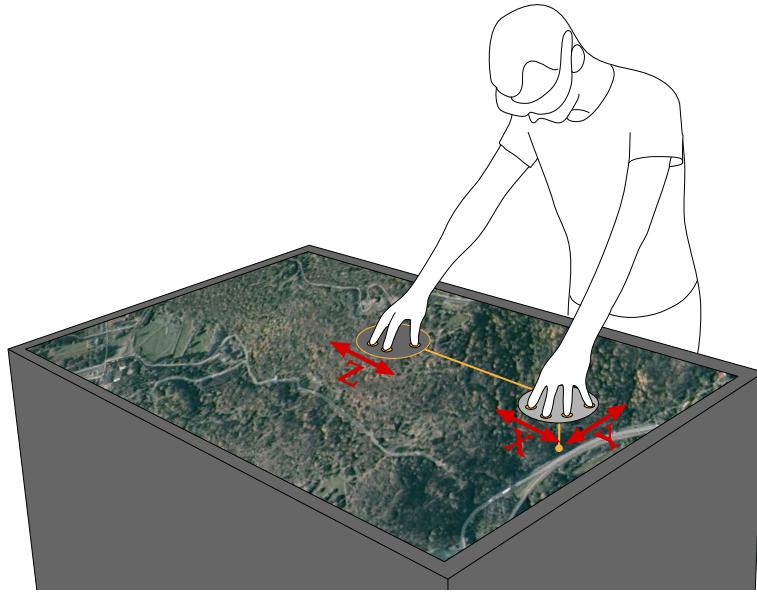


Abbildung 12: Interaktion im 3D Translationsmodus auf Grundlage der Erkenntnisse von Benko und Feiner [Benko and Feiner, 2007]. Die roten Pfeile sowie deren Beschriftung sind nicht Teil der Interaktionsvisualisierung

Für eine sinnvolle Einbindung beider Ansätze in die Applikation wird die skalierungs-basierte Berechnung von z_t nur angewandt wenn $|G| - |G| * \frac{|V|}{|V'|} > |V| - |V'|$ gilt.

Abbildung 12 visualisiert den Umgang mit diesem Interaktionsmodus. Hierbei wird dem Nutzer eine Visualisierung des manipulierten Auftreffpunktes auf der Geometrie dargestellt, sowie die Verbindungs linie zur darüber liegenden Handposition. Die Gerade zwischen den Eingabepositionen wird ebenfalls visuell referenziert.

6.3 3D Rotation

Ziel dieser Technik ist die Kontrolle aller Freiheitsgrade, welche für die Rotation im dreidimensionalen Raum benötigt werden. Hierzu wurde der in Abschnitt 4.3 vorgestellte Ansatz von Rousset et al. [Rousset et al., 2014] implementiert.

Für die Bestimmung der Manipulation wird die Hand eines Nutzers mit genau zwei aufgesetzten Fingern verfolgt. Initiiert der Nutzer die Geste durch

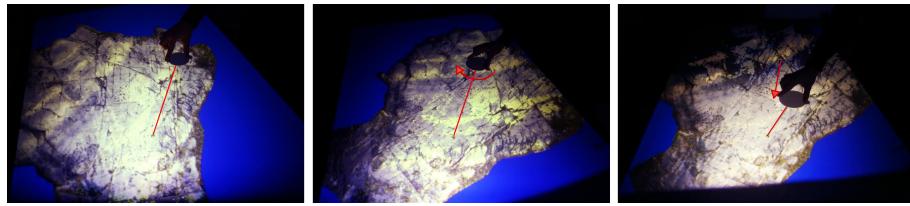


Abbildung 13: Interaktion mit dem 3D Rotationsmodus nach Rousset et al. [Rousset et al., 2014].

Aufsetzen der Hand, so wird ein direkter Angriffspunkt auf der Geometrie mit orthogonaler Verbindungsgeraden zur Bildschirmfläche berechnet. Bis zur Beendung der Geste durch Anheben der Hand erfolgt eine Rotation um diesen Punkt.

Gegeben sind $P_h(t)$ und $P_h(t')$ als Positionen des Mittelpunkts der Geraden zwischen den zwei aufgesetzten Fingern zu den Zeiten t und t' . P_g sei der beschriebene Referenzpunkt auf der Geometrie. Zuletzt sind $V_f(t)$ und $V_f(t')$, als zeitabhängigen Richtungsvektoren zwischen den Fingern, gegeben. Es werden für die Interaktion zwei Rotationen getrennt voneinander berechnet.

Die Richtungsvektoren $V_{hg}(t)$ und $V_{hg}(t')$ ergeben sich aus der Verbindung zwischen $P_h(t)$ mit P_g und $P_h(t')$ mit P_g . Die Normale auf die von den Vektoren aufgespannte Ebene bildet die Achse der ersten Rotation. Der Winkel leitet sich aus dem Winkel zwischen $V_{hg}(t)$ und $V_{hg}(t')$ ab. $V_{hg}(t')$ ist außerdem die Achse für eine zweite Rotation, deren Maß durch den Winkel zwischen $V_f(t)$ und $V_f(t')$ bestimmt wird. Abbildung 13 stellt das Verfahren an einem Beispiel dar.

6.4 Vorteile und Limitierungen

Durch die Verwendung der RTS-Technik (siehe Abschnitt 6.1) sind eine Vielzahl verschiedener Transformationen im dreidimensionalen Raum gleichzeitig und auch getrennt voneinander zu bedienen. Der anhaltende und direkte Kontakt mit der Geometrie vermittelt dem Nutzer das Gefühl die Szene zu greifen, was zu einer intuitiven Einarbeitung in den Umgang mit dem System führt. Die Technik ist jedoch begrenzt auf dieselben Freiheitsgrade, welche auch im zweidimensionalen Raum verwendet werden. Sie bietet daher nicht die Möglichkeit, die Navigation in jeden durch die drei Dimensionen gegebenen Zustand zu bewegen. Dieser Nachteil spiegelt sich bei den übrigen in diesem Kapitel vorgestellten Techniken noch stärker wieder. Demzufolge ist durch 3D Trans-

lation und 3D Rotation jeweils nur eine Form der Transformation steuerbar. Im Kontext ihrer Anwendung zeigen sich beide Techniken ebenfalls als nutzerfreundliche Strategien zur Manipulationen der von ihnen bestimmten Freiheitsgrade.

Die Flaschenzugstrategie bei der 3D Translation weist durch ihr direktes Mapping eine hohe Präzision und ein leicht Verständliches Interaktionskonzept auf. Durch die Maße des Tisches und die Reichweite des menschlichen Armes ist der Bewegungsrahmen für die Interaktion eingeschränkt. Ein weit entferntes Objekt in die Nähe der Projektionsebene zu bringen, erfordert somit das wiederholte Anheben und erneute Aufsetzen der Hand. Aus diesem Grund wirkt die Flaschenzugstrategie, bei der Arbeit mit weit entfernten Objekten, ungeeignet. Die Translationsberechnung durch das Verhältnis von Eingabepunkt- und Geometrieabstand kann hingegen effektiv für grobe Interaktionen mit weit entfernten Objekten genutzt werden. Kleine Bewegungen auf der Tischfläche führen zu einer zunehmenden Auswirkung, je weiter das berührte Objekt vom Tisch entfernt liegt. Dementgegen ist die Auswirkung von weitreichenden Bewegungen mit Geometrieelementen in Tischebene gering. Somit entstehen leicht Missverständnisse bei der Nutzung mit bildschirmnaher Geometrie.

Die vorgestellte 3D Rotationstechnik kann den Erhalt der Orthogonalität, zwischen dem Geometrie-Eingabe-Vektor und der Bildfläche, nicht gewährleisten. Das hebt die Metapher der direkten Berührung auf, welche sich als nutzerfreundlich erwiesen hat. Wie in Kapitel 5 beschrieben ist 3D Multi-Touch Interaktion effektiv für Objekte in Null-Parallaxe verwendbar. Bei der 3D Rotation können geringe Eingabeänderungen zu starken Anpassungen des Rotationswinkels führen, wenn die Distanz zwischen Eingabeposition und Rotationsreferenzpunkt gering ist. Befindet sich der Referenzpunkt auf der Tischfläche ist gar keine Berechnung der Rotation mehr möglich. Des Weiteren wird durch den Übergang zwischen Positiv- und Negativ-Parallaxe die Steuerung der Rotation invertiert.

Kapitel 7

Levelling: Implizite Multi-Touch 3D Navigation

Kapitel 4 zeigt, dass die gleichzeitige Steuerung aller Freiheitsgrade einer 3D Manipulation durch Multi-Touch Eingaben noch immer eine Herausforderung ist. In Kapitel 6 werden Lösungsansätze vorgestellt, mit welchen einzelne Freiheitsgrade der Interaktion explizit manipuliert werden können. Ein modulares System zur Koordination dieser Techniken, wäre ein Ansatz mit allen Freiheitsgraden explizit umzugehen. Die Bedienbarkeit eines solchen Systems könnte jedoch schnell komplex werden.

Dieses Kapitel beschreibt und evaluiert eine im Rahmen dieser Arbeit entstandene Navigationstechnik namens *Levelling*. Hierzu wird in Abschnitt 7.1 eine Definition zu *impliziter* Navigationstechnik, sowie eine Erklärung zum Interaktionsziel von *Levelling*, gegeben. Im darauf folgenden Abschnitt 7.2 wird die Funktionsweise von *Depth-Levelling* beschrieben, während in Abschnitt 7.3 auf den erweiterten Ansatz *Rotation-Levelling* eingegangen wird. Abschließend wird in Abschnitt 7.4 *Levelling* als Navigationstechnik diskutiert.

7.1 Definition und Interaktionsziele

Wir definieren implizite Navigationstechniken als zusätzliche Transformationen zur Erreichung eines Interaktionsziels, welche der Anwendung expliziter Techniken beigelegt werden und nicht getrennt von diesen bedienbar sind.

Bruder et al. [Bruder et al., 2013] beschreibt, dass eine effektive Interaktion mit dreidimensionalen Inhalten vor allem möglich ist, wenn Objekte in Null-Parallaxe liegen. *Levelling* ist ein Ansatz zur impliziten Steuerung der Navigation. Ziel der Technik ist es, nutzerdefinierte Applikationsinhalte auf Tischebene zu bewegen und darauf zu ebnen. Es wird dabei sowohl die Distanz der

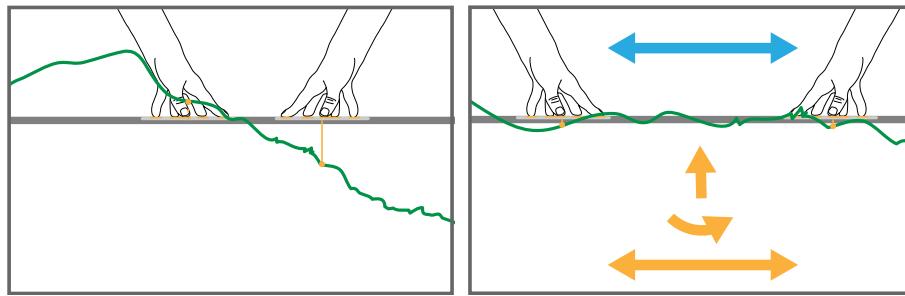


Abbildung 14: Visualisierung des *Levelling* Interaktionskonzepts. Die berührte Geometrie wird im Nahfeld des Tisches geebnet.

Geometrie zur Bildebene verringert, sowie die Orientierung angepasst. Hierzu legt der Anwender durch Berühren des Projektionstisches mit beiden Händen zwei Auftreffpunkte auf der Geometrie fest, welche durch die *Levelling* Technik schrittweise näher an die Bildschirmfläche geführt werden. Da sich beide Punkte auf unterschiedlicher Höhe befinden können, ist außerdem eine Rotation nötig um beide Punkte auf die Bildebene zu führen. *Levelling* ist demzufolge ein zweistufiges Verfahren, dessen einzelne Manipulationsschritte durch die von ihnen hervorgerufene Transformation benannt sind. Abbildung 14 zeigt die Auswirkung von *Levelling* auf eine beispielhafte Visualisierung.

Die Punkte auf der Geometrie werden durch einen Schnitttest ausgehend vom jeweiligen Eingabepunkt auf dem Tisch und entlang der Bildschirmnormalen ermittelt. Bei der Schnittberechnung ist wichtig, dass negativ parallaxe Modellbereiche nicht übersehen werden. Folglich muss die z-Koordinate des Startpunkts für den Schnittstrahl¹ auf die Höhe der *Near-Clipping-Plane* angehoben werden. Abbildung 15 veranschaulicht diesen Zusammenhang.

Wir definieren *Touch-Kontakte* als Strukturen, welche sowohl den Eingabepunkt auf dem Bildschirm, als auch den zugehörigen Geometrieschnittpunkt beinhaltet. Die Orthogonalität der Verbindungsgeraden von Bildschirm- und Geometriepunkt eines *Touch-Kontakts* soll während der Interaktion mit dem Tisch erhalten bleiben. Um dies zu erreichen, muss *Levelling* der Anwendung von RTS im Bildraum (siehe 6.1) beigefügt werden. Wir nennen diesen Modus Rotation, Translation und Skalierung im Bildraum, erweitert durch *Levelling* (RTS+L).

¹im Screenspace

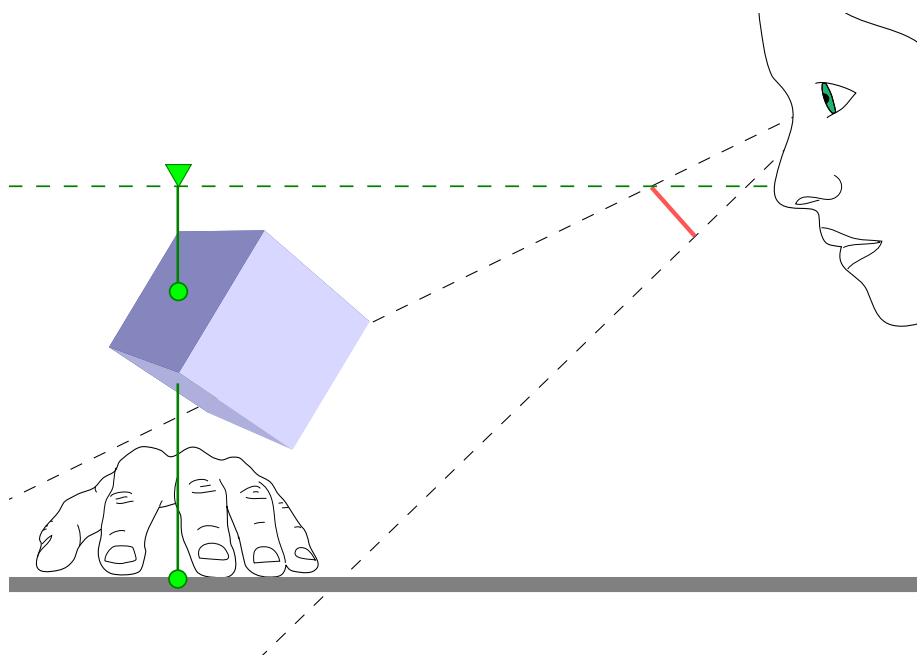


Abbildung 15: Anhebung des Startpunkts (grünes Dreieck) auf Höhe der *Near Clipping Plane* (rot) für den Schnitttest (grün) bei der *Levelling* Interaktion.

7.2 Depth-Levelling

Depth-Levelling wird durch das Eingeben zweier *Touch-Kontakte* initiiert. Nach der Schnittberechnung erfolgt die Einordnung dieser in *Primär-* und *Sekundärkontakt*. Der Geometrieschnittpunkt des *Primärkontakte*s liegt näher an der Tischemebene oder ist weiter über der dieser als der des *Sekundärkontakte*s. Ziel des *Depth-Levellings* ist es, den Geometrierefenzpunkt des *Primärkontakte*s durch Translation auf die Projektionsebene zu bringen. Um dies zu erreichen wird das Viewing Setup schrittweise entlang der Bildschirmnormalen verschoben.

Die Schrittweite der Translation wird bei RTS+L durch einen Faktor festgelegt welcher aus der Relation zwischen den *Touch-Kontakten* abgeleitet ist. Ist die Geometrie bereits in relativer Bildschirmnähe, bietet sich die Differenz der Distanz zwischen den Eingabepunkten seit der letzten Interaktion an. Demnach ist die Levelling-Distanz (t_L) gegeben durch:

$$t_L = |(|V_{S_1S_2}|) - (|V_{S_1S_2'}|)|$$

,wobei $V_{S_1S_2}$ der Vektor zwischen den Bildschirmpunkten der Touch-Kontakte vor der Eingabe ist und $V_{S_1S_2'}$ danach.

Für eine größere Translationsdistanz bei weit von der Bildfläche entfernten Objekten, kann stattdessen das Verhältnis der Länge zwischen $V_{S_1S_2}$ und $V_{S_1S_2'}$ zur Berechnung von t_L genutzt werden.

$$t_L = \left| |G| - |G| * \frac{|V_{S_1S_2}|}{|V_{S_1S_2'}|} \right|$$

,wobei G der Richtungsvektor zwischen dem Geometrieschnittpunkt des *Primärkontakte*s und dessen Punkt auf dem Bildschirm ist.

Sollte die nach einem der genannten Ansätze ermittelte *Levelling*-Distanz größer oder gleich der Länge des Richtungsvektors zwischen dem Geometrie- und dem Bildschirmpunkt des *Primärkontakte*s sein, so wird stattdessen eine Translation um diesen Vektor angewandt. Nach dieser Bewegung befindet sich der Geometrierefenzpunkt des *Primärkontakte*s auf Tischemebene und das *Depth-Levelling* ist beendet.

7.3 Rotation-Levelling

Das *Rotation-Levelling* schließt sich an die Durchführung des *Depth-Levellings* an. Dieser Vorgang dient der Heranführung des Geometrieschnittpunkts des Sekundärkontakts an die Projektionsebene. Der Modellschnittpunkt des *Primärkontakte*s würde durch Translation von der Bildschirmebene entfernt werden. Dies führt zur Aufhebung des durch *Depth-Levelling* erreichten Interaktionsziels. Stattdessen wird eine Rotation um den Bildschirmpunkt des *Primärkontakte*s vorgenommen. Des Weiteren muss eine Skalierung in Richtung des selben Referenzpunktes erfolgen. Mit diesem Schritt soll gewährleistet werden, dass der Richtungsvektor zwischen Geometrie- und Bildschirmpunkt des *Sekundärkontakte*s weiterhin senkrecht auf der Bildebene steht.

Die Parametrisierung der Rotation bestimmt sich wie folgt. Sei G_2 der Geometrieschnittpunkt des *Sekundärkontakte*s und S_2 der zugehörige Punkt der Eingabe auf dem Bildschirm, so bestimmt sich $V_{G_2S_2}$ als Richtungsvektor zwischen G_2 und S_2 . Wie beim *Depth-Levelling* (siehe Abschnitt 7.2) wird im ersten Schritt der Distanzfaktor (t_L) für Annäherung an die Bildebene ermittelt. Danach wird ein Punkt G_2' berechnet, der die Position von G_2 nach Verschiebung um t_L entlang $V_{G_2S_2}$ beschreibt. Als nächstes werden zwei Vektoren $V_{S_1G_2}$ und $V_{S_1G_2'}$ bestimmt. $V_{S_1G_2}$ ist hierbei der Richtungsvektor zwischen dem zugehörigen Bildschirmpunkt des *Primärkontakte*s S_1 und G_2 . $V_{S_1G_2'}$ ist der Richtungsvektor zwischen S_1 und G_2' . Der Rotationswinkel (α_L) und die Rotationsachse ($R_L = (x, y, z)$) ergeben sich aus:

$$\begin{aligned}\alpha_L &= \arccos(\|V_{S_1G_2}\| * \|V_{S_1G_2'}\|) \\ R_L &= \|V_{S_1G_2}\| \times \|V_{S_1G_2'}\|\end{aligned}$$

Wie bereits erwähnt dient eine Skalierung in Richtung des Geometriepunkts des *Primärkontakte*s zum Erhalt der Orthogonalität zwischen Geometrie- und Bildschirmpunkt des *Sekundärkontakte*s. Der Skalierungsfaktor (S_L) ist gegeben durch:

$$S_L = \frac{|V_{S_1G_2}|}{|V_{S_1G_2'}|}$$

Befindet sich der Geometrieschnittpunkt des *Sekundärkontakte*s nach Anwendung aller Transformationen auf der Bildebene, so ist das Interaktionsziel erreicht und das *Rotation-Levelling* beendet.

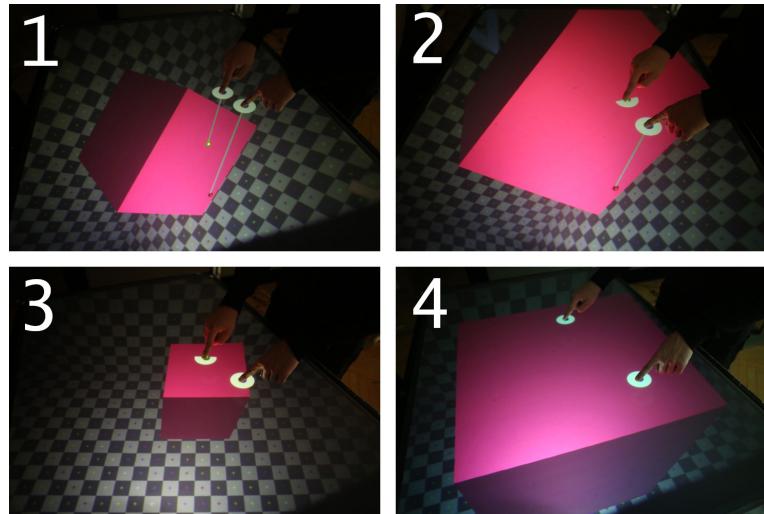


Abbildung 16: Ablauf einer *Levelling* Interaktion. Bild 1 zeigt die ermittelten Orthogonalschnittpunkte mit der Szenengeometrie. In Bild 2 hat der Geometrieschnittpunkt des *Primärkontakts* durch *Depth-Levelling* die Bildebene erreicht. Bild 3 zeigt die Ebnung des *Sekundärschnittpunkts* durch *Rotation-Levelling*. Danach sind lediglich RST Manipulationen möglich, wie Bild 4 verdeutlicht.

7.4 Vorteile und Limitierungen

Das am Lehrstuhl entwickelte Point Based Rendering System ermöglicht die Analyse von komplexen 3D Oberflächen-Scans in einem großen Umfang verschiedener Auflösungsstufen. Nach den in Abschnitt 1.2 beschriebenen Konzepten, ist die Betrachtung und Interaktion mit dreidimensionalen Modellen vor allem in Bereichen nahe der Projektionsebenen effektiv möglich. Aus diesem Zusammenhang leitet sich das Interaktionsziel, nutzerdefinierte Inhalte in Bildschirmnähe zu bringen, ab. Durch Anwendung von RTS+L wird der Nutzer ohne Einführung neuer Gesten bei der Erreichung dieses Ziels unterstützt. Hierbei wird die Komplexität der zur Manipulation erforderlichen Transformation vor dem Anwender verborgen.

Levelling ist als implizite Navigationstechnik nach RTS+L nicht ohne den Einfluss von RTS bedienbar. Umgekehrt betrachtet können Rotation, Translation und Skalierung bei RTS+L nur getrennt von *Levelling* bedient werden, wenn sich die Geometrieschnittpunkte beider *Touch-Kontakte* auf der Bildebene befinden.

Die am schwersten wiegende Limitierung ist, dass die durch *Levelling* hervorgerufene Manipulation bei RTS+L nicht umkehrbar ist. Demnach schließt RTS+L zwar die z-Translation, sowie 3D Rotation ein, jedoch nur in Richtung der durch die *Touch-Kontakte* definierten Vektoren. Abbildung 16 zeigt ein einfaches Beispiel der Folge dieses Problems. Hier wird ein Würfel unter der Tischebene positioniert. Nach mehrmaligem Anwenden liegt eine der Flächen der Geometrie in der Bildebene. Eine weitere Manipulation durch *Levelling* ist nicht möglich, da sich die Seitenflächen der Würfel orthogonal zur Bildebene befinden. Die gegenüberliegende Fläche ist durch die in der Projektionsebene liegende verdeckt und wird daher vom Geometrieschnitttest bei der Initiierung der Geste nicht getroffen.

Kapitel 8

Wechsel zwischen Interaktionstechniken

Wie in den Kapiteln 6 und 7 erläutert wird, können verschiedene implizite und explizite Techniken verwendet werden um alle Freiheitsgrade der Navigationsmanipulation in einer virtuellen Szene zu steuern. Durch die Vielzahl der unterschiedlichen Modi für die Interaktion, ist es wichtig geeignete Kriterien zu definieren, nach denen zwischen den Techniken gewechselt werden kann.

In Abschnitt 8.1 wird eine Herangehensweise an diese Problematik vorgestellt, die anhand der Evaluierung von Kontaktpunkten auf dem Bildschirmtisch eine geeignete Interaktionsform ableitet. Abschnitt 8.2 zeigt wie Menüstrukturen zur Festlegung des Navigationsmodus beitragen können. Zuletzt werden in Abschnitt 8.3 beide Ansätze verglichen und auf ihre Vorteile und Limitierungen untersucht.

8.1 Evaluierung der Bildschirmkontakte

Die in Kapitel 3 beschriebene, zugrunde liegende Touch-Eingabeerkennung gibt Aufschluss über vom Nutzer platzierte Bildschirmkontaktpunkte. Des Weiteren ermöglicht sie die Zuordnung dieser Fingerpositionen zu einzelnen Händen. Obwohl hierbei eine genaue Hand-Nutzer Zuweisung nicht möglich ist, kann die Evaluierung der gegebenen Eingabeinformationen bereits effektiv als Kriterium zur Festlegung des Navigationszustands genutzt werden.

In kollaborativen Anwendungsszenarien ist gemeinsame Analyse der gebotenen virtuellen Inhalte stark an das Zeigen auf bestimmte Bereiche der Visualisierung gebunden. Wie in der Anforderungsanalyse (siehe Kapitel 5) vorgestellt, verlieren Nutzer durch Stereoparallaxe leicht die visuelle Wahrnehmung des

physischen Bildschirms. In Folge dessen kann es passieren das Nutzer versehentlich mit einem Finger die Bildschirmoberfläche berühren und damit ungewollt Manipulationen an der Applikation vornehmen. Um diesen Eingabekonflikt zu verhindern, wird die Eingabe einer Hand nur für die Interaktion angewendet, wenn der Anwender mehr als einen Finger auf die Bildschirmoberfläche legt.

Da für die Berechnung der Manipulationsparameter nur ein Punkt der jeweiligen Hand als Eingabe nötig ist, wird eine Handzentrumsposition ermittelt. Diese wird aus den Positionen aller enthaltenen Finger berechnet. Sie kann wahlweise durch den gemittelten Positions値 aller Kontaktpunkte, oder den Mittelpunkt des fingerumschließenden Rechtecks definiert sein. Für die Berechnung der Manipulationsparameter wird die relative Translation der gesamten Hand des Nutzers über den Bildschirm auf die jeweiligen Interaktionsmodi übertragen. Hierbei ist zu beachten, dass die berechnete Handzentrumsposition kein fester Punkt auf der Hand des Nutzers ist. Folglich führt das Aufsetzen oder Entfernen von Fingern an der interagierenden Hand ebenfalls zur Verschiebung des Handzentrums. Die Auswirkung dieser Translation auf die Manipulation sollte verhindert werden, da sie zu sprunghaften Veränderungen der Applikation führt.

Vergleicht man alle in den Kapiteln 6 und 7 vorgestellten Interaktionsmodi hinsichtlich ihrer zur Berechnung der Manipulation verwendeten Kontaktpunkte wird deutlich, dass keine der Techniken mehr als zwei aktuelle Inputpositionen nutzt. Eingabekonflikte entstehen häufig, wenn mehrere Anwender gleichzeitig mit dem Bildschirmtisch zu interagieren versuchen. Dieses Problem kann dadurch begrenzt werden, dass nicht mehr als zwei Hände zur Interaktion zugelassen werden. Demzufolge werden genau die beiden Hände zur Eingabe genutzt, welche den Projektionstisch früher berühren. Abbildung 17 illustriert die Auswirkung dieser Technik auf ein beispielhaftes Szenario.

Die beschriebenen Navigationstechniken lassen sich ebenfalls durch Auswertung der vom Nutzer aufgesetzten Kontaktpunkte auswählen. Nach der oben genannten Erklärung wird für die Eingabe eines einzelnen Fingers kein Navigationsmodus betreten.

2D Translation ist einhändig bedienbar. Gleiches gilt für die in Abschnitt 6.3 beschriebene 3D Rotation, falls die interagierende Hand mehr als einen Finger beinhaltet. Diese Technik wäre auch mit den Zentrumspositionen zweier Hände berechenbar. Zugunsten der Verständlichkeit der Interaktion und um eine klare Definition für die Wahl des Modus zu gewährleisten, wird der 3D Rotationsmodus jedoch ausschließlich bei der Eingabe mit zwei Fingern, bei

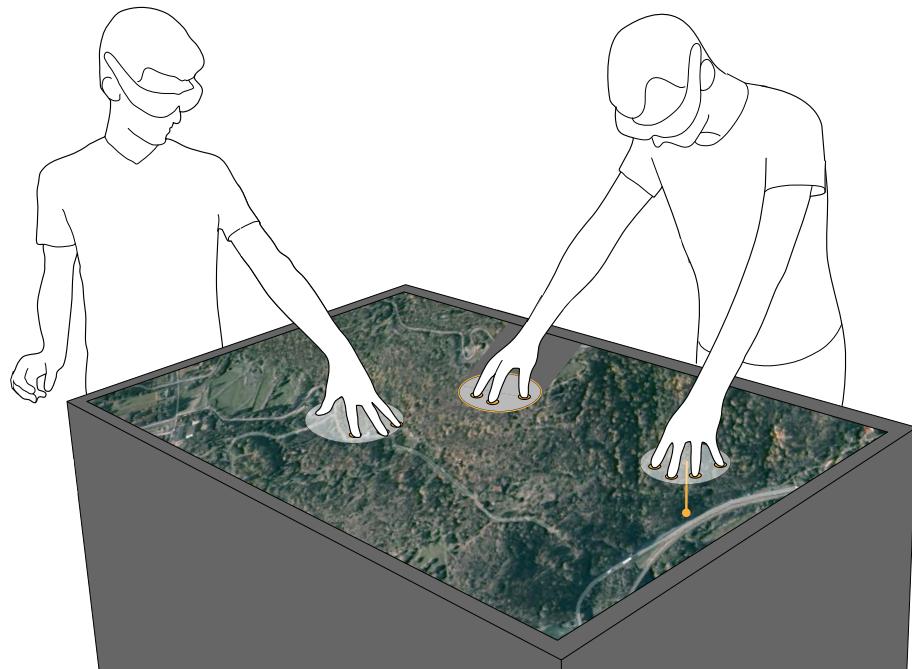


Abbildung 17: Zwei Nutzer interagieren gleichzeitig mit der Bildschirmfläche. Dies kann Konflikte bei der Eingabe hervorrufen. Aus diesem Grund werden nur maximal zwei Kontakte zur Manipulation der Applikation zugelassen. Alle bei Überschreitung dieser Limitierung gelieferten Eingaben werden vom System ignoriert.

genau einer aufgelegten Hand gesteuert. Somit bleiben Dreifinger-, Vierfinger- und Fünffinger-Einhand-Interaktion für die Kontrolle der 2D Translation.

Für den RTS+L Modus (siehe Abschnitt 7.1) werden zur Berechnung aller Manipulationsparameter zwei Kontaktpunkte benötigt. Selbiges gilt für die in Abschnitt 6.2 vorgestellte 3D Translation. Da letztere Interaktionstechnik jedoch keine Auswirkungen auf Skalierung und Rotation hat, wird ein allgemeiner Translationsmodus definiert. Dieser schließt die Verwendung von einhändiger 2D Translation ein. Setzt ein Nutzer nach einem bestimmten Zeitintervall der Einhand-Interaktion eine zweite Hand auf, wird die 3D-Translation angewandt. Demzufolge wird RTS+L bedient wenn zwei Hände mit zeitlicher Differenz kleiner des bestimmten Zeitintervalls aufgesetzt werden. Ein Zustandsdiagramm (siehe Abbildung XX) verdeutlicht den Umgang mit den Navigationsmodi durch Evaluierung der Bildschirmkontakte.

8.2 Auswahl durch Menüelemente

Die Nutzung von Betriebssystemen und sonstigen Programmen mit Grafischer Benutzeroberfläche macht den Umgang mit Menüstrukturen zur Konfiguration von Arbeitsabläufen alltäglich. Aus diesem Grund sind schaltbare, geometrische Flächen in Null-Parallaxe eine naheliegende Lösung zur Filterung von Navigationsmodi. Nach diesem Ansatz wird jeder Interaktionstechnik eine in der Bildeckene liegende Geometrie zugeordnet. Selektiert der Nutzer eine dieser und ist der referenzierte Modus aktiv, so wird dieser deaktiviert. Ist er deaktiviert, so erfolgt eine Aktivierung. Sind mehrere Modi gleichzeitig eingeschaltet, wird die Wahl der Technik nach den in Abschnitt 8.1 ermittelt.

Herrlich et al. stellen mit ihrer Technik *PieRotate* einen Ansatz zur Separierung der Freiheitsgrade der dreidimensionalen Rotation mit Touch-Eingaben vor [Herrlich et al., 2011]. *PieRotate* teilt den Interaktionsraum zur Manipulation eines Objekts in drei Zonen auf. Durch eine Rotationsgeste in der jeweiligen Zone, kann das Objekt um die zugewiesene Achse gedreht werden. Diese Lösung lässt sich auf die Umschaltung verschiedener Navigationsmodi übertragen. Demnach werden Interaktionsterritorien definiert, in denen bestimmte Techniken verwendet werden, welche durch den Nutzer erzeugbar und konfigurierbar sind. Bier et al. beschreiben die Verwendung von *Magic Lenses* zur Erzeugung von Filtereffekten in bestimmten Arealen des Bildes. *Magic Lenses* sind nach dieser Entwicklung durchsichtige, fensterartige Interfaces, welche das Rendering der unter ihnen liegenden Applikationsinhalte beeinflussen.

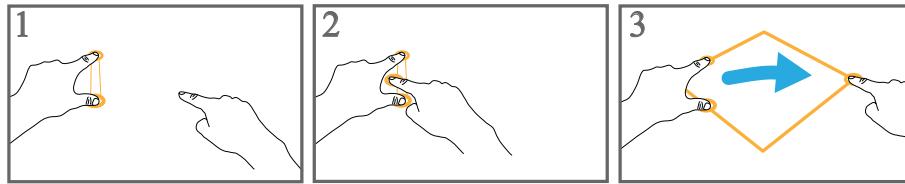


Abbildung 18: Die Erzeugung von Menuelementen durch eine Touch gesteuerte Crossing Geste nach Accot und Zhai [Accot and Zhai, 2002]. In Bild 1 ist die visuelle Verbindungsleitung zwischen zweie Fingern zu sehen. Darstellung 2 zeigt die Bestrebung des Anwenders diese zu durchkreuzen um ein neues Fenster zu erstellen. Zuletzt kann dieses durch fortwährende Streichbewegung skaliert werden, wie Abbildung 3 zeigt.

Ein Anfasser in der Ecke der *Magic Lense* macht das Werkzeug durch Dragging beweglich.

Die Erzeugung von Menüelementen und Fenstern lässt sich durch die Einführung einer zugewiesene Touch-Geste ergänzen. Demnach führt das Aufsetzen zweier Finger einer Hand, zur Konstruktion einer visuellen Verbindungsleitung zwischen den Fingern. Inspiriert vom *Crossing*-Verfahren nach Accot und Zhai [Accot and Zhai, 2002], führt das Durchstreichen dieser Linie zum Öffnen eines Fensters. Dieses dehnt sich durch die fortwährende Bewegung des durchstreichenden Fingers aus, bis er die Eingabefläche verlässt. Eine Implementierung dieses Konzepts ist im Zuge der Arbeit nicht entstanden. In Abbildung 18 wird dieses Konzept visualisiert.

8.3 Diskussion

Die Evaluierung der Bildschirmkontakte nach Abschnitt 8.1 gibt dem Anwender die Möglichkeit allein durch verschiedene Bewegungsabfolgen zwischen einer Reihe von Navigationsmodi zu wechseln. Aufgrund dessen sind flüssige Interaktionsabläufe möglich. Der Nutzer muss sich hierzu jedoch eine Reihe von Regeln zur Verwendung der jeweiligen Techniken merken, was die intuitive Bedienung des Systems strapaziert. Folgendes Szenario hebt die Komplexität und den kognitiven Aufwand bei der Zustandsunterscheidung hervor:
 RTS+L und 3D Translation sind an einen zweiseitigen Input gebunden. Die Entscheidung zur Aktivierung einer dieser Modi wird lediglich durch zeitliche Auswertung getroffen. Bei der Analyse des in Abbildung 19 dargestellten Zustandsdiagramms wird deutlich, dass der RTS+L Modus durch kurzzeitiges Anheben einer Hand verlassen wird. Ab diesem Zeitpunkt wird durch erneutes Aufsetzen der 3D Translationsmodus aktiviert. RTS+L kann erst durch

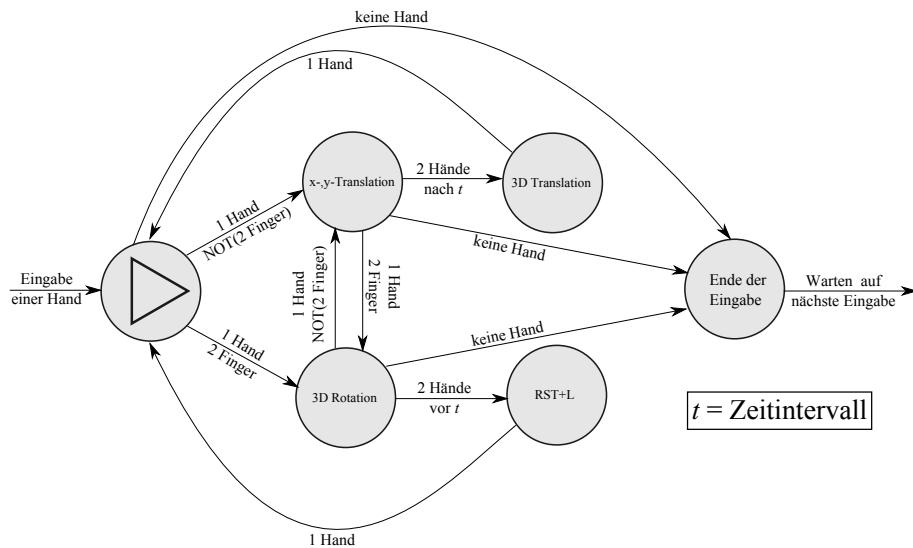


Abbildung 19: Ein Zustandsdiagramm welches die Übergänge zwischen den Interaktionsmodi visualisiert.

Entfernen der zweiten Hand vom Bildschirm und Aufsätzen beider Hände innerhalb des Zeitintervalls nochmals genutzt werden.

Die Einbindung einer Menüstruktur nach Abschnitt 8.2 gibt dem Nutzer durch visuelle Rückmeldung einen Überblick über alle verfügbaren Modi. Nach den Ansprüchen des jeweiligen Nutzers kann so individuell die Zustandsunterscheidung erleichtert werden. Eine Verwendung aller Navigationsmodi nach Abschnitt 8.1 ist noch immer möglich. Durch Visualisierung von Menüstrukturen auf der Bildebene werden hingegen Teile der Applikation verdeckt. Außerdem schmälert der durch Menüelemente eingenommene Raum die Interaktionsfläche für die Navigationsmanipulation. Zusätzlich verdecken Modellinhalte in Negativ-Parallaxe möglicherweise die dargestellten funktionalen Bestandteile der Benutzeroberfläche, was den Umgang mit ihnen erschwert.

Keiner der in diesem Kapitel genannten Ansätze vermeidet das Auftreten von Eingabekonflikte zwischen den Nutzern vollständig. Durch die fehlende Hand-Nutzer Zuordnung kann bei der zweihändigen Interaktion nicht verhindert werden, dass zwei Nutzer mit je einer Hand Veränderungen am Applikationszustand vornehmen.

Kapitel 9

Virtuelles Freischneiden der Hand während Berührungseingaben

Negativ-parallaxe Bildausschnitte können wie in Kapitel 5 beschrieben Wahrnehmungskonflikte hervorrufen. In diesem Kapitel wird die im Rahmen dieser Arbeit entstandene *See-Through* Technik basierend auf den Erkenntnissen von Argelaguet et al. [Argelaguet et al., 2010] vorgestellt. Die Strategie soll helfen beschriebenen Problemen der Tiefenwahrnehmung entgegenzuwirken.

Hierzu definiert Abschnitt 9.1 den konzeptuellen Ansatz. Abschnitt 9.2 erläutert die Integration der *See-Through* Technik in die Applikationsstruktur. Zuletzt werden in Abschnitt 9.3 die Vorteile und Limitierungen der Strategie diskutiert.

9.1 Ansatz

Wie in Abbildung 20 veranschaulicht wird nutzen Argelaguet et al. den von ihnen als *Show-Through* bezeichneten Ansatz zur Freilegung verdeckter virtueller Objekte [Argelaguet et al., 2010]. Bei perspektivischem Rendering kann der Blick auf verdeckte Objekte in manchen Betrachtungswinkeln auf der Projektionsfläche sichtbar sein. Die Auswahl dieser Geometrie schafft bei durch *Show-Through* nach Argelaguet et al. einen Sichtkanal auf diese Inhalte für jeden Betrachtungswinkel. Im Kontext dieser Arbeit soll durch Anwendung der Technik die von der Hand des Nutzers durchstoßene Geometrie oberhalb der Projektionsfläche entfernt werden.

Wird die Tischoberfläche berührt, so werden negativ parallaxe Modellbereiche zwischen der Eingabeposition und dem Kopf des Nutzers in einem be-



Abbildung 20: Im linken Bild ist die freie Sicht auf die ausgewählte Geometrie möglich. Diese ist in der mittleren Darstellung verdeckt. In der rechten Abbildung schafft der durch *Show-Through* nach Argelaguet et al. erzeugte Sichtkanal freien Blick auf die verdeckte Geometrie. Diese Abbildung entstammt vollständig einer Veröffentlichung von Argelaguet et al. [Argelaguet et al., 2010]

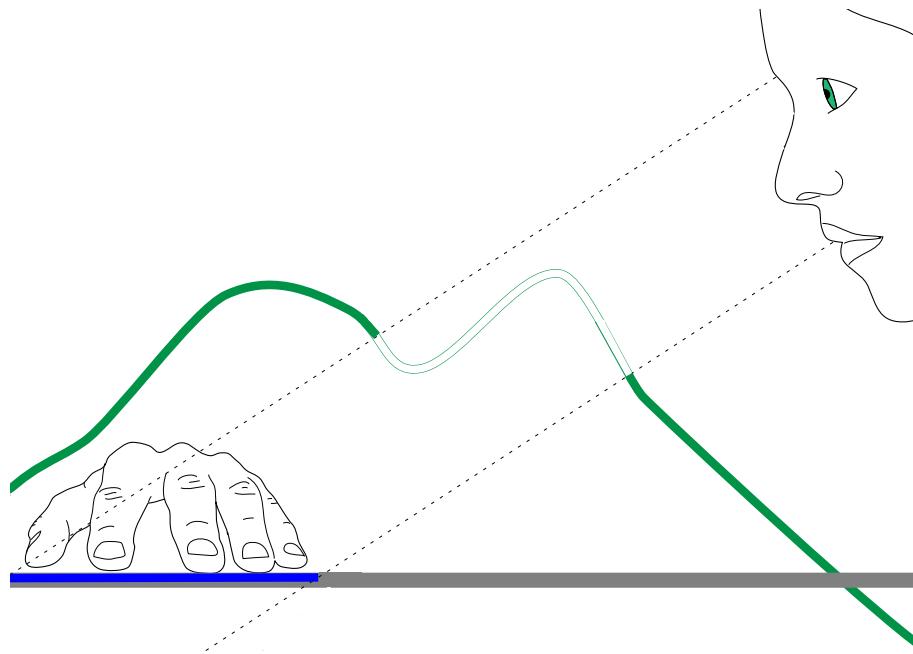


Abbildung 21: Visualisierung des *See-Through* Konzepts. Die blaue Fläche referenziert das Areal des fingerumschließenden Kreises

stimmten Durchmesser ausgeschnitten. Dies erzeugt einen zylinderförmigen Blicktunnel, welcher dem Nutzer freie Sicht bis zu seiner Hand auf der Tischebene bietet. Der Durchmesser des Ausschnittzylinders wird hierbei durch den fingerumschließenden Kreis bestimmt. Abbildung 21 illustriert die Anwendung der Technik.

In einem Mehrbenutzerszenario wird für jeden Nutzer eine eigene Projektion der virtuellen Szene auf dem Bildschirmtisch erstellt. Dies beeinflusst auch die Perspektive auf von Händen verdeckte, negativ parallaxe, Bereiche der Darstellung. Wird für alle um den Tisch versammelten Personen der gleiche Ausschnittzylinder verwendet, kann ausschließlich für einen der Nutzer eine konfliktfreie Wahrnehmung sichergestellt werden.

Um Wahrnehmungskonflikte für alle Anwender vermeiden zu können, muss die Orientierung des Ausschnittzylinders nutzerspezifisch angepasst werden. Infolgedessen wird eine eigene Repräsentation des Zylinders für jeden Nutzer bestimmt und auf die jeweilige Projektion angewandt.

9.2 Implementierung

Die Anwendung der *See-Through* Technik kann leicht beim Rendern der Geometriefragmente vorgenommen werden. Hierzu werden alle Eingabepositionen der Hände auf der Tischfläche, sowie der Radius des fingerumschließenden Kreises, in den Fragments shader der Applikation gegeben. Die Position der Kamera entspricht der Kopfposition des Nutzers und kann somit zur nutzerspezifischen Berechnung der Orientierung des Ausschnittzylinders genutzt werden. Zuletzt werden alle Fragmente, welche innerhalb des Blicktunnels liegen, verworfen.

9.3 Vorteile und Limitierungen

Durch die Anwendung der *See-Through* Technik können blickabhängig, von der Hand verdeckte, negativ parallaxe Modellausschnitte bei der Projektion der virtuellen Szene ausgeschlossen werden. Hierdurch bleibt die geometrische Relation zwischen realen und virtuellen Inhalte erhalten. Dies führt zu einer wahrnehmungsunterstützenden Visualisierung, welche maßgeblich zur Benutzbarkeit der Toucheingaben beiträgt.

Die *See-Through* Technik beeinflusst lediglich die Darstellung der virtuellen Szene. Eine Manipulation der Geometrie, oder des Blickpunktes, wird nicht vorgenommen. Dadurch kann mit der Szene in jeder geometrischen Lage inter-

KAPITEL 9. VIRTUELLES FREISCHNEIDEN DER HAND WÄHREND BERÜHUNGSEINGABEN

agiert werden.

Der Zylinder als Ausschnittgeometrie hat sich in der prototypischen Anwendung als gute Approximation des Nutzerarmes erwiesen. Es ist damit möglich den Großteil verdeckter, negativ parallaxer Modellausschnitte, von der Darstellung auszuschließen.

Die Ausschnittgeometrie umschließt jedoch ein Volumen, welches nicht exakt die Gliedmaßen des Nutzers repräsentiert. Infolgedessen können Bereiche der Visualisierung beschnitten werden, welche nicht vom Nutzer verdeckt werden und daher noch sichtbar sein sollten. Des Weiteren entspricht die Form und Orientierung der Ausschnittgeometrie nicht der komplexen Form des Arms eines Nutzers. Bei bestimmten Eingaben können somit nicht alle durch Negativparallaxe hervorgerufenen Wahrnehmungskonflikte vermieden werden.

Die *See-Through* Technik wird durch das Berühren der Bildschirmoberfläche initiiert. Aus diesem Grund kann das Hineingreifen in Inhalte oberhalb dieser Ebene, ohne Interaktion mit der Tischfläche, nicht verhindert werden.

Kapitel 10

Abbildung von Multi-Touch Techniken in der Applikationsstruktur

In den Kapiteln 6 und 7 werden einige Techniken vorgestellt, welche im Rahmen der Arbeit in einem neuartigen System zur Multi-Touch 3D Navigation vereinigt wurden. Durch die in Kapitel 8 eingeführten Ansätze können diese modular ausgewählt und bedient werden. Zusätzlich erfolgt die Verwendung der *See-Through* Technik (sieh Kapitel 9). Hierdurch werden Wahrnehmungskonflikte abgeschwächt, welche durch Eindringen der Hand des Nutzers in Negativ-Parallaxe hervorgerufen werden.

In diesem Kapitel wird die Einbindung dieser Konzepte in eine generische Applikationsstruktur präsentiert. Hierzu wird in Abschnitt 10.1 die Verarbeitung der vom TUIO Protokoll (siehe Unterabschnitt 3.3.1) gelieferten Eingabedaten vorgestellt. Abschnitt 10.2 erklärt den Aufbau und die Arbeitsweise der Touch Navigation als Applikationsmanipulator. In Abschnitt 10.3 wird die Abbildung der visuellen Repräsentation von Touch Eingaben im Szenegraph beschrieben. Zuletzt werden die in diesem Kapitel dargebotenen Zusammenhänge in Abschnitt 10.4 auf ihre Vorteile und Limitierungen untersucht.

10.1 Multi-Touch Input Pipeline

Die Verarbeitung von Touch Eingaben erfolgt in drei wesentlichen Schritten. Schritt 1 wird einmalig zu Applikationsstart ausgeführt. Hierbei wird für die Maximalanzahl an gleichzeitig erkennbaren Händen jeweils ein Touchwerkzeug angelegt. Während der MSER-Algorithmus (siehe Kapitel 3) zur Erkennung

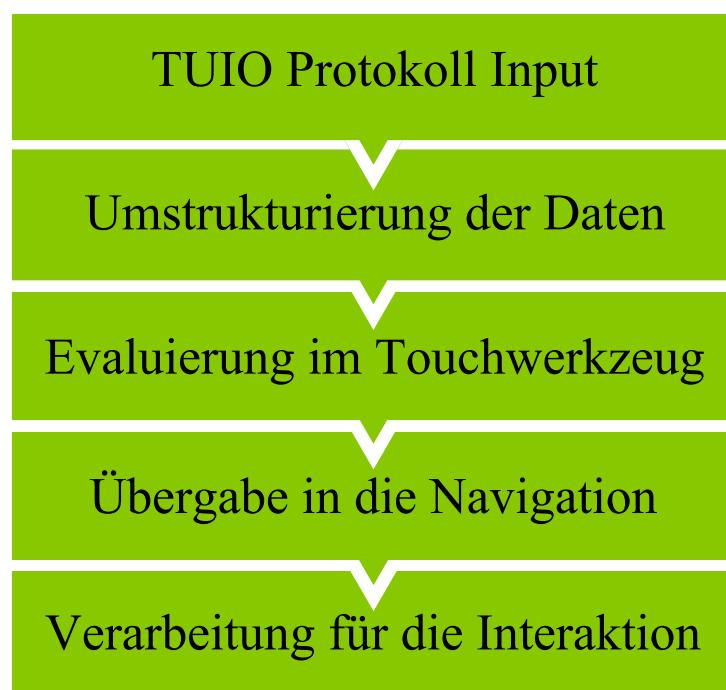


Abbildung 22: Verarbeitung des Touch Inputs.

von Eingaben keine Begrenzung aufweist, sind die in der Applikation nutzbaren Inputpositionen durch die Anzahl verfügbarer *Stations* im *Avango Daemon* begrenzt (siehe Unterabschnitt 3.3.1). Touchwerkzeuge sind vom Anwender konfigurierbar und sollen die Verbindung zwischen der taktilen Eingabe und der jeweiligen Auswirkung dieses Inputs in der Applikation herstellen. Jedes Touchwerkzeug verarbeitet den Input von genau einer Hand. Des Weiteren verfügt jedes dieser Werkzeuge über verschiedene Geometrien zur Visualisierung der Eingabe. Der Input einer Hand wird als Teil des Touchwerkzeugs in einem Datencontainer gehalten, welcher bei Bedarf in Schritt 2 gefüllt wird.

Sobald durch den in Abschnitt 3 vorgestellten MSER-Algorithmus Fingerkontaktepunkte ermittelt wurden, folgt Schritt 2. An dieser Stelle werden die Daten aus dem TUO Protokoll Transfer ausgewertet und umstrukturiert. Hierzu werden die angesprochenen Datencontainer gefüllt. Diese identifizieren jede erkannte Hand und enthalten außerdem eine Liste mit Positionen der einzelnen Finger, sowie eine errechnete Handzentrumsposition. Die Handzentrumsposition entspricht dem Mittelpunkt der um alle enthaltenen Finger gespannten Bounding Box. Durch die in Abschnitt 3.4 vorgestellten Komplikationen bei der Handerkennung, werden Finger nur beim ersten Aufsetzen auf die Tischplatte einer Hand zugewiesen. Bei der Eingabeverfolgung bleibt diese Zuordnung erhalten.

Da jeder der in Schritt 2 beschriebenen Datencontainer an ein Touchwerkzeug gebunden ist, erfolgt in Schritt 3 zuletzt die Verknüpfung der Eingabe mit der jeweiligen Auswirkung in der Applikation. Hierzu verfügen Touchwerkzeuge über einen Selektionsmechanismus, welcher basierend auf den vorliegenden Eingabedaten und dem aktuellen Applikationsstatus Kontakt zu einem Applikationsmanipulator herstellt. Dieser wird über Strahlenschnitttests realisiert. Abbildung 22 visualisiert einen Überblick über die einzelnen Schritte der Verarbeitung von Toucheingaben.

10.2 Touch Navigation

Touchwerkzeuge verfügen nicht über die Macht direkte Veränderungen an der Applikation vorzunehmen, ausgenommen der Darstellung ihrer eigenen geometrischen Repräsentation. Sie stellen lediglich durch die gegebenen Input Daten einen Kontakt zu Applikationsmanipulatoren her. Wir definieren diese Manipulatoren als Objekte zur funktionsspezifischen Änderung bestimmter Eigenschaften der Anwendung.

Die Touch Navigation ist ein Applikationsmanipulator zur Umsetzung der vorgestellten Multi-Touch Techniken auf die Transformation des virtuellen Bildschirms. Die Klasse benötigt zur Bestimmung der Interaktion Eingabedaten, welche wiederum von den Touchwerkzeugen geliefert werden. Jedes dieser verfügt über einen Selektionsmechanismus, welcher genutzt wird um Kontakt zu Applikationsmanipulatoren wie auch der Touch Navigation aufzubauen. Diese verfügt über eine durchsichtige Geometrie, welche an den physikalischen Bildschirm angehängt wird. Sie referenziert dadurch den für die Eingabe nutzbaren Bereich. Beim Aufsetzen einer Hand wird das zugewiesene Touchwerkzeug aktiviert und führt zwei Strahlenschnittteste aus. Der erste bestimmt den Auftreffpunkt auf der sichtbaren Szenengeometrie. Durch den zweiten Schnitt wird funktionale Geometrie getestet. Dies schließt die Fläche, welche durch die Touch Navigation definiert wurde, sowie etwaige Menüelemente ein. Neben einer Auftreffposition bietet der Schnitt mit funktionaler Geometrie auch die Referenz auf eine bestimmte Klasse. Auf diese Weise kann das Touchwerkzeug durch erfolgreichen Schnitttest der durchsichtigen Fläche auf dem Bildschirm Zugriff auf die Touch Navigation erhalten.

Jeder Applikationsmanipulator ist mit den Funktionen *addContact* und *removeContact* ausgestattet. Nachdem das Touchwerkzeug eine funktionale Geometrie getroffen hat, wird auf der Referenzierten Klasse *addContact* aufgerufen. Diese Funktion wird mit den Eingabedaten des Touchwerkzeugs, einer eindeutigen Identifikationsnummer (ID) für die Bindung, sowie dem Geometrieschnitt des Szeneninhalts parametrisiert. Wird der Kontakt von der Touch Navigation entgegengenommen, speichert das Werkzeug die Bindung und leitet etwaige Veränderungen des eigenen Inputs über *addContact* an den Manipulator weiter. Andernfalls gilt der gegebene Input als blockiert und das Touchwerkzeug hat keine Auswirkung auf die Navigationsinteraktion. Um an den Manipulator gebunden werden muss sich ein gebundener Kontakt lösen und das Touchwerkzeug erneut anmelden. Letzteres ist nach Lösen eines gebundenen Kontakts durch erneutes Aufsetzen der Hand möglich. Da die Touchwerkzeuge keinem Nutzer zugeordnet werden können, wird auf diese Weise die gleichzeitige Interaktion mehrerer Nutzer eingeschränkt. Verliert ein Werkzeug durch Anheben der jeweiligen Hand die Eingabe, so wird der Kontakt durch *removeContact* parametrisiert durch seine zugehörigen ID, von der Navigation abgemeldet.

Die Touch Navigation verwaltet eine Liste von Touch-Kontakten welche, durch die in Abschnitt 8.1 beschriebenen Zusammenhänge, auf eine Länge von zwei Elementen begrenzt ist. Erhält der Manipulator eine Bindungsanfrage über *addContact*, urteilt er anhand der Elemente in dieser Liste wie mit der Anfrage umzugehen ist. Existiert bereits ein Kontakt mit gleicher ID, wird eine

Aktualisierung dieses basierend auf den gelieferten Daten durchgeführt. Falls nicht und ist die Maximalanzahl der gleichzeitig verwertbaren Touch-Kontakte nicht erreicht, wird der Liste ein weiteres Element beigefügt. Die Anfrage wird in beiden Fällen als erfolgreich beantwortet. Ist die Kapazität der Listenbegrenzung erreicht und entspricht die ID nicht der eines in der Liste enthaltenen Elements, wird der Kontakt verworfen. Folglich wird die Anfrage als nicht erfolgreich beantwortet.

10.3 Repräsentation im Szenegraph

Im Gegensatz zu anderen physischen Werkzeugen ist der Multi-Touch Bildschirmstisch kein nutzerspezifisches Eingabegerät. Präzise ausgedrückt handelt es sich vielmehr um ein Mehrbenutzer Display mit Interaktionsschnittstelle. Demnach ist der Touch-Tisch gleichzeitig Darstellungs- sowie Eingabemedium, was für die Modellierung im Szenegraph berücksichtigt werden muss.

Die Konstruktion der Repräsentation des Tisches als virtuelles Display wird durch das verwendete Applikationsframework vorgenommen. In Abschnitt 10.1 wurden Touchwerkzeuge als Teil der Multi-Touch Input Pipeline vorgestellt. Ihre Geometrie visualisiert die von ihnen genutzten Inputdaten auf dem Bildschirm, sowie die Verbindungsgerade der von ihnen ermittelten Schnittposition mit der Szenengeometrie. Touchwerkzeuge können bislang nicht eindeutig einem bestimmten Nutzer zugeordnet werden, jedoch sind ihre Visualisierungen immer von der Transformation des Bildschirms abhängig. Des Weiteren sind die durch sie angestoßenen Interaktionsanfragen logischer Teil der Eingabe durch den Multi-Touch Tisch. Aus diesem Grund wird die Geometrie der Touchwerkzeuge dem Bildschirmknoten angehängt.

Auf eine Eltern-Kind Beziehung zwischen Hand und den Fingern wurde absichtlich verzichtet. Da die Fingerpositionen bei der Eingabe im Bildraum ausgedrückt sind, müssten sie hierfür in den Objektraum des Handknotens transformiert werden. Durch ständige Neupositionierung der Finger und folglich auch der Hand, würde diese Umwandlung die Performanz beeinträchtigen.

10.4 Diskussion

Da die meiste Interaktion ausgehend von einzelnen Händen, anstatt von Fingern, abgeleitet wird erscheint die Verwendung des Touchwerkzeugs als Datenstruktur naheliegend. Um die Eingabe der Werkzeuge zu steuern wird eine Umstrukturierung der Daten in der Applikation vorgenommen. Diese kostet

KAPITEL 10. ABBILDUNG VON MULTI-TOUCH TECHNIKEN IN DER APPLIKATIONSSTRUKTUR

Zeit und wirkt sich daher negativ auf die Performanz des Systems aus. Mit der einmaligen Handzuordnung wird die Toucherkennung merklich stabilisiert.

Das Konzept der Applikationsmanipulatoren macht die vorgestellte Struktur leicht erweiterbar. Bei der Programmierung können neue Manipulatoren eingeführt werden, welche durch den Selektionsmechanismus der Touchwerkzeuge bedient werden. Hierbei muss das Werkzeug die Klasse nicht kennen, solange sie die *addContact* und *removeContact* Funktionalität unterstützt. Hierbei scheint jedoch die Wahl des Schnitttests unsichtbarer Geometrie als Selektionsmechanismus verbesserungswürdig.

Kapitel 11

Diskussion

In diesem Kapitel soll eine abschließende Diskussion der im Rahmen dieser Arbeit entstandenen Konzepte zur Multi-Touch basierten Interaktion mit stereoskopischen Bildinhalten erfolgen. Hierzu soll die Bewertung dieser Ansätze in Abschnitt 11.1 anhand der in Kapitel 2 definierten Anforderungen bemessen werden. Abschließend werden in Abschnitt 11.2 einige Hypothesen definiert die aus Beobachtungen während dem Arbeitsprozess resultieren.

11.1 Anforderungsevaluierung

Der Grundstein für die Entwicklung der Touch Navigation ist durch das *MSER Hand und Finger Tracking* gegeben (siehe Kapitel 3). Dieses System bietet eine Reihe von Informationen über die Struktur der Eingabe auf dem Bildschirmtisch. Demnach konnte eine Finger-Hand Zuordnung sinnvoll in der Applikationsstruktur abgebildet werden (siehe Kapitel 10). Durch Einführen eines Zusatzkriteriums beim Umgang mit Interaktionstechniken, wurde die maximale Anzahl an für die Manipulation zugelassenen Hände auf zwei begrenzt. Eine Hand-Nutzer Zuordnung wird vom aktuellen System nicht unterstützt. In der Applikation kann die einhändige Eingabe zweier Nutzer somit nicht von der beidhändigen eines Nutzers unterschieden werden. Fehleingaben durch gleichzeitiges Interagieren der Nutzer können folglich nicht vollständig vermieden werden. Nach diesem Zusammenhang wird Anforderung 1 maßgeblich berücksichtigt und unterstützt, jedoch nicht vollständig erreicht.

Clipping von Modellbereichen kann Konflikte der Tiefenwahrnehmung durch Stereoparallaxe vermeiden [Ardouin et al., 2011]. Das in Kapitel 9 beschriebene Konzept zur Beschneidung der virtuellen Geometrie, durch einen perspektivabhängigen Ausschnittzyylinder nutzt diese Erkenntnis. Abschnitt 9.3 beschreibt, dass die Wahl und Orientierung des Volumens für diesen Prozess

nicht der realen Form des menschlichen Arms entspricht. Hierdurch kann nicht vollständig sichergestellt werden, dass der Anwender negativ-parallaxe Modellbereiche nicht berührt. Entstehende Wahrnehmungskonflikte wurden nach Anforderung 2 abgeschwächt, konnten jedoch nicht gänzlich vermieden werden.

Die in die Implementierung der Touch Navigation einbezogenen Techniken ermöglichen dem Nutzer flexiblen Umgang mit virtuellen Inhalten. Hierbei wird die Auswirkung tiefen-anpassender Gesten durch Distanz zwischen Bildschirm und Geometrie skaliert. Angewendete Transformationen sind sowohl im Nahfeld-Bereich als auch auf große Distanz präzise möglich, was das System im Kontext des 3D Pitoti Projekts effektiv nutzbar macht. Anforderung 3 wurde nach diesen Kriterien erreicht.

Eingebundene Manipulationsformen sind größtenteils am von Hancock et al. vorgestellten Konzept der orthogonalen Verbindung zwischen Geometrie und Bildschirm orientiert [Hancock et al., 2007, Hancock et al., 2009].

Dies unterstützt das Gefühl Modelle direkt zu Berühren. Nutzer können natürliche Bewegungsabfolgen demnach leicht auf angebotene Navigationsstrategien anwenden. In Kapitel 8 wird der Wechsel zwischen den Navigationsmodi vorgestellt. Während die einzelnen Gesten auch für unerfahrene Nutzer leicht steuerbar scheinen, ist das Einprägen der Routine zur Auswahl einer Navigationstechnik nach den in Abschnitt 8 Zusammenhängen fraglich. Anforderung 4 wurde im Arbeitsprozess maßgeblich berücksichtigt. Im Hinblick auf zukünftige Entwicklungen sollte eine Studie zur Bewertung der Touch Navigation hinsichtlich dieser Anforderung durchgeführt werden.

Die entwickelten Navigationstechniken ermöglichen die Manipulation aller sechs Freiheitsgrade der Transformation im dreidimensionalen Raum. RST+L bietet die gleichzeitige Kontrolle über x- und y-Translation, z-Rotation und uniforme Skalierung, erweitert durch die implizite Tiefenebnung der *Levelling* Strategie. Mithilfe der 3D Rotation sind die Drehungen um x-, y- und z-Achse getrennt zu bedienen. Der 3D Translations-Modus dient zur expliziten Kontrolle der x-, y- und z-Komponente bei der Bewegungstransformation. Die alleinige Skalierungsanpassung ist nach den in Kapitel 7 beschriebenen Voraussetzungen bei RST+L nur nach Erreichung der Interaktionsziels von *Levelling* möglich. Jedoch kann dies durch menüseitige Filterung der Interaktionsmodi von der Anwendung bei Rotation, Skalierung und Translation im Bildraum ausgeschlossen werden. Somit ist der Skalierungsfaktor auch getrennt von anderen Transformationen konfigurierbar. Anforderung 5 gilt demnach als erfüllt.

RST+L gibt dem Anwender die Möglichkeit bekannte Strategien der 2D Touch

Interaktion für die Applikationssteuerung einzusetzen. Währenddessen sorgt die Errechnung von Manipulationsparametern bei der *Levelling* Strategie für Ebnung von visuell relevanten Inhalten. Nach Bruder et al. ist die Analyse und Interaktion mit stereoskopischen Geometrien vor allem durch nah an der Bildebene liegende Objekte gegeben. Die Heranführung der vom Nutzer berührten Modelle an die Projektionsfläche ist folglich als Interaktionsziel zu sehen. Nach diesem Zusammenhang ist von der Erfüllung von Anforderung 6 auszugehen.

Zur Visualisierung der Interaktionsprozesse und der taktilen Eingabe wurden verschiedene virtuelle Darstellungen eingeführt. Diese verdeutlichen die Auswirkungen der Interaktion mit der Bildfläche. Sie verstärken die Verständlichkeit der einzelnen Navigationstechniken und machen fehlerhafte Verarbeitungsergebnisse des MSER Hand und Finger Trackings sichtbar. Somit konnte Anforderung 7 durch das System erreicht werden.

11.2 Hypothesen

Hypothese 1 *Fehleingaben der Nutzer können nur durch eine Verbesserung der Eingabeanalyse behoben werden.*

Das MSER Finger und Hand Tracking ist auf die Finger-Hand Zuordnung begrenzt. Dieses Kriterium genügt nicht um Aussagen über die Zugehörigkeit von Eingabepositionen zu einem bestimmten Nutzer zu tätigen. Dadurch können Fehleingaben durch den gleichzeitigen Input mehrerer Anwender von der Applikation nicht vollständig ausgeschlossen werden. Genauer Aufschluss über die Hand-Nutzer Relation ist die Voraussetzung für die Entwicklung von Mechanismen zur Vermeidung beschriebener Mehrbenutzerkonflikte. Die Verbesserung der Eingabeanalyse könnte die Extrahierung dieser Daten ermöglichen.

Hypothese 2 *Konflikte der Tiefenwahrnehmung können durch Volume Clipping behoben werden.*

Auch wenn die für die See-Through Technik verwendete Ausschnittgeometrie nicht dem menschlichen Arm entspricht, hat sich die Strategie dennoch als große Unterstützung für die kollisionsfreie Tiefenwahrnehmung gezeigt. Eine präzisere Abbildung physischer Gegebenheiten über dem Tisch könnte zu einer geeigneten Konstruktion des Clipping Volumens beitragen.

Hypothese 3 *Das entwickelte System ist auch für unerfahrene Nutzer leicht zu bedienen.*

Abschnitt 3.4 spricht einige Probleme an, welche durch die Touch Erkennung entstehen. Die Auswirkung dieser beeinflusst drastisch den Umgang mit der Touch Navigation. Aus diesem Grund wurde im Rahmen dieser Arbeit keine Nutzerstudie durchgeführt. Es ist von einer Verzerrung der Ergebnisse einer potentiellen Studie auszugehen. Testpersonen hätten die offensichtlichen Fehlfunktionen bei der Eingabeauswertung möglicherweise in die Bewertung des Navigationssystems eingebracht. Dieser Zusammenhang stellt die Nützlichkeit der Studienergebnisse zur Evaluierung der Touch Gesten in Frage. Für valide Ergebnisse sollte demzufolge eine Verbesserung der Input Verarbeitung erfolgen.

Hypothese 4 *Der Umgang mit Levelling ist leicht erlernbar und effektiv bei der 3D Multi-Touch Interaktion.*

Die Berechnung des *Rotation-Levelling* ist gegenüber dem Mapping expliziter Navigationstechniken komplex. Durch den visuellen Effekt der Manipulation wird diese Komplexität jedoch vor dem Nutzer verborgen. Der Zusammenhang die Geometrie durch *Levelling* auf der Bildfläche zu ebnen erscheint schlüssig und leicht zu verinnerlichen. Vor allem im Umgang mit komplexen Oberflächenstrukturen zeigt sich eine automatische Ausrichtungskorrektur anhand nutzerdefinierter Punkte als effektiv.

Kapitel 12

Fazit und zukünftige Arbeiten

Im Rahmen dieser Arbeit wurde ein System zur Multi-Touch basierten 3D Navigation für einen immersiven Bildschirmtisch implementiert. Hierzu werden verschiedene Techniken zur kollaborativen Exploration virtueller Welten angeboten. Die zur Steuerung der Applikation benötigten Parameter können in spezifischen Gruppen und getrennt voneinander bedient werden. In zukünftigen Arbeiten sollten die eingebundenen Ansätze in einer umfangreichen Nutzerstudie untersucht werden.

Entwickelte Modi sind durch die Evaluierung der Bildschirmkontakte vom System unterscheidbar und durch vereinfachte Menüstrukturen vom Anwender zu filtern. Die Umsetzung der Erzeugung neuer Menüelemente zur Laufzeit der Applikation konnte im Rahmen der Arbeit lediglich konzeptuell erläutert werden. Es sollte zukünftig eine Implementierung dieser Idee erfolgen.

Levelling erweitert die expliziten Manipulationsstrategien, um den Nutzer beim Umgang mit den dargebotenen Inhalten zu unterstützen. Unterschiedliche Visualisierungen referenzieren sowohl die Struktur der Eingabe auf dem Tisch, als auch die Interaktionsvorgänge. Hierbei wäre eine Nutzerstudie zur Effizienzbestimmung des Systems und im Hinblick auf weitere Verbesserungen angebracht. Des Weiteren sollte die *Levelling* Technik einer vergleichenden Betrachtung expliziter Gesten unterzogen werden.

Auftretenden Eingabekonflikten durch Einsetzen des Tisches in einem Mehrbenutzerszenario wird durch Analyse der Struktur des Inputs entgegengewirkt. Durch Einbindung der *See-Through* Technik bei Berührung der Projektionsfläche werden negativ-parallaxe Modellbereiche zur Vermeidung von Konflikten der Tiefenwahrnehmung freigeschnitten. Betrachtet man die Bilder die zur Analyse des MSER-Algorithmus genutzt werden, stellt sich die Frage ob weite-

KAPITEL 12. FAZIT UND ZUKÜNTIGE ARBEITEN

re Informationen über die Objekte über dem Tisch extrahiert werden können. Aktuell wird lediglich eine zweidimensionale Orientierung des Armes anhand einer die Wurzel-ER umschließenden Ellipse übermittelt (siehe Kapitel 3). Es bleibt zu untersuchen ob sich eine dreidimensionale Relation aus den Disparitäten ableiten ließe. Dies würde zur Untersuchung der Hand-Nutzer Zuweisung beitragen. Des Weiteren würde hierdurch eine Basis für die Bestimmung eines präziseren Clipping Volumens bei der *See-Through* Technik geschaffen werden.

Die Arbeitsweise der *Touchwerkzeuge* macht die Applikationsstruktur leicht erweiterbar für die Entwicklung zusätzlicher Interaktionsvorgänge welche durch taktile Eingaben gewährleistet werden. Außerdem sind integrierte Navigationstechniken mit einem anderen Input austauschbar. Es wäre demnach interessant zu sehen ob die Techniken auch mit anderen Eingabegeräten sinnvoll in Zusammenhang gebracht werden können.

Literaturverzeichnis

- [Accot and Zhai, 2002] Accot, J. and Zhai, S. (2002). More than dotting the i's—foundations for crossing-based interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 73–80. ACM.
- [Ardouin et al., 2011] Ardouin, J., Lécuyer, A., Marchal, M., and Marchand, E. (2011). Design and evaluation of methods to prevent frame cancellation in real-time stereoscopic rendering. In *3D User Interfaces (3DUI), 2011 IEEE Symposium on*, pages 95–98. IEEE.
- [Argelaguet et al., 2010] Argelaguet, F., Kunert, A., Kulik, A., and Froehlich, B. (2010). Improving co-located collaboration with show-through techniques. In *3D User Interfaces (3DUI), 2010 IEEE Symposium on*, pages 55–62. IEEE.
- [Autodesk, 2008] Autodesk (2008). Stereoscopic filmmaking whitepaper, the business and technology of stereoscopic filmmaking.
- [Avango,] Avango.
- [Benko and Feiner, 2007] Benko, H. and Feiner, S. (2007). Balloon selection: A multi-finger technique for accurate low-fatigue 3d selection. In *3D User Interfaces, 2007. 3DUI'07. IEEE Symposium on*. IEEE.
- [Bruder et al., 2013] Bruder, G., Steinicke, F., and Sturzlinger, W. (2013). To touch or not to touch?: comparing 2d touch and 3d mid-air interaction on stereoscopic tabletop surfaces. In *Proceedings of the 1st symposium on Spatial user interaction*, pages 9–16. ACM.
- [Casiez et al., 2012] Casiez, G., Roussel, N., and Vogel, D. (2012). 1€ filter: a simple speed-based low-pass filter for noisy input in interactive systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2527–2530. ACM.
- [de la Rivière et al., 2010] de la Rivière, J.-B., Kervégant, C., Dittlo, N., Courtois, M., and Orvain, E. (2010). 3d multitouch: when tactile tables meet

- immersive visualization technologies. In *ACM SIGGRAPH 2010 Talks*, page 56. ACM.
- [Ewerling et al., 2012] Ewerling, P., Kulik, A., and Froehlich, B. (2012). Finger and hand detection for multi-touch interfaces based on maximally stable extremal regions. In *Proceedings of the 2012 ACM international conference on Interactive tabletops and surfaces*, pages 173–182. ACM.
- [Giesler et al., 2014] Giesler, A., Valkov, D., and Hinrichs, K. (2014). Void shadows: multi-touch interaction with stereoscopic objects on the tabletop. In *Proceedings of the 2nd ACM symposium on Spatial user interaction*, pages 104–112. ACM.
- [Hancock, 2011] Hancock, M. (2011). Sticky tools: Full 6DOF force-based interaction for multi-touch tables. <https://www.youtube.com/watch?v=DW8vmbhxK04>. [Online; accessed 06-May-2015].
- [Hancock et al., 2007] Hancock, M., Carpendale, S., and Cockburn, A. (2007). Shallow-depth 3d interaction: design and evaluation of one-, two-and three-touch techniques. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1147–1156. ACM.
- [Hancock et al., 2009] Hancock, M., Ten Cate, T., and Carpendale, S. (2009). Sticky tools: full 6dof force-based interaction for multi-touch tables. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, pages 133–140. ACM.
- [Herrlich et al., 2011] Herrlich, M., Walther-Franks, B., and Malaka, R. (2011). Integrated rotation and translation for 3d manipulation on multi-touch interactive surfaces. In *Smart Graphics*, pages 146–154. Springer.
- [Kaltenbrunner et al., 2005] Kaltenbrunner, M., Bovermann, T., Bencina, R., and Costanza, E. (2005). Tuio: A protocol for table-top tangible user interfaces. In *Proc. of the The 6th Int'l Workshop on Gesture in Human-Computer Interaction and Simulation*, pages 1–5.
- [Lipton, 2007] Lipton, L. (2007). Vertical surround parallax correction. In *Electronic Imaging 2007*, pages 64900G–64900G. International Society for Optics and Photonics.
- [Martinet and Grisoni, 2012] Martinet, A., C. G. and Grisoni, L. (2012). Integrality and separability of multitouch interaction techniques in 3d manipulation tasks. *Visualization and Computer Graphics, IEEE Transactions on*, 18(3):369–380.

LITERATURVERZEICHNIS

- [Matas et al., 2004] Matas, J., Chum, O., Urban, M., and Pajdla, T. (2004). Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 22(10):761–767.
- [Moeller and Kerne, 2012] Moeller, J. and Kerne, A. (2012). Zerotouch: an optical multi-touch and free-air interaction architecture. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2165–2174. ACM.
- [Nistér and Stewénius, 2008] Nistér, D. and Stewénius, H. (2008). Linear time maximally stable extremal regions. In *Computer Vision–ECCV 2008*, pages 183–196. Springer.
- [Rousset et al., 2014] Rousset, É., Bérard, F., and Ortega, M. (2014). Two-finger 3d rotations for novice users: surjective and integral interactions. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces*, pages 217–224. ACM.
- [Scheurich and Stuerzlinger, 2013] Scheurich, D. and Stuerzlinger, W. (2013). A one-handed multi-touch mating method for 3d rotations. In *CHI’13 Extended Abstracts on Human Factors in Computing Systems*, pages 1623–1628. ACM.
- [Seyed et al., 2013] Seyed, T., Burns, C., Costa Sousa, M., and Maurer, F. (2013). From small screens to big displays: understanding interaction in multi-display environments. In *Proceedings of the companion publication of the 2013 international conference on Intelligent user interfaces companion*, pages 33–36. ACM.
- [TUIO,] TUIO.
- [Valkov et al., 2011] Valkov, D., Steinicke, F., Bruder, G., and Hinrichs, K. (2011). 2d touching of 3d stereoscopic objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1353–1362. ACM.
- [Valyus and Asher, 1966] Valyus, N. and Asher, H. (1966). *Stereoscopy*.
- [Wartell, 2001] Wartell, Z. J. (2001). *Stereoscopic head-tracked displays: Analysis and development of display algorithms*. PhD thesis, Georgia Institute of Technology.