

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
Betreuer: Alexander Kulik, André Kunert

Datum der Abgabe: 4. Mai 2015

Erklärung

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

Weimar, den 4. Mai 2015

.....
Sebastian Stickert

Zusammenfassung

Dieses Dokument dient als Muster für die Ausarbeitung einer Diplomarbeit am Lehrstuhl für Webtechnologie und Informationssysteme. 1 - 2 - 3 Ümläut-test.

Inhaltsverzeichnis

1	Einführung	4
1.1	Mehrbenutzer VR-Systeme	4
1.2	Multi-Touch Interaktion mit dreidimensionalen Szenen	4
1.3	Ziel der Arbeit	5
2	Finger und Hand Tracking auf Grundlage von Maximally Stable Extremal Regions	6
2.1	Maximally Stable Extremal Regions	7
2.2	Technische Voraussetzungen	8
2.3	Implementierung	8
2.3.1	TUIO Touch Protokoll	9
2.3.2	Umgang mit Jittering	9
3	Anforderungsanalyse	11
3.1	Mehrbenutzer Touch-Eingaben	11
3.2	Tiefenwahrnehmung durch Stereo- und Bewegungsparallaxe	12
3.3	Interaktionsziele	12
3.4	Explizite und implizite Kontrolle der involvierten Freiheitsgrade	12
3.5	Visuelle Rückmeldung	12
4	3D Interaktion mit Multitoucheingaben	13
4.1	Sticky Tools	13
4.2	Balloon Selection	14
4.3	Two Axis Valuator	15
4.4	Diskussion	15
5	Wahrnehmungskonflikte und Lösungsansätze	17
5.1	Frame Cancellation	17
5.2	Touch Interaktion mit stereoskopischen Visualisierungen	18
5.3	Diskussion	19

6	Explizite Multi-Touch 3D Navigation	21
6.1	Rotation, Translation und Skalierung im Bildraum	21
6.2	3D Translation	22
6.3	3D Rotation	24
6.4	Vorteile und Limitierungen	24
7	Levelling: Implizite Multi-Touch 3D Navigation	26
7.1	Definition und Interaktionsziele	26
7.2	Depth-Levelling	27
7.3	Rotation-Levelling	28
7.4	Vorteile und Limitierungen	29
8	Wechsel zwischen Interaktionstechniken	31
8.1	Evaluierung der Bildschirmkontakte	31
8.2	Auswahl durch Menüelemente	33
8.3	Diskussion	34
9	Abbildung von Touch Eingaben in der Applikationsstruktur	36
9.1	Multi-Touch Input Pipeline	36
9.2	Touch Navigation	37
9.3	Representation im Szenegraph	37
9.4	Diskussion	37
10	Virtuelles Freischneiden der Hand während Berührungseingaben	38
10.1	Ansatz	38
10.2	Implementierung	39
10.3	Vorteile und Limitierungen	39
11	Diskussion	41
11.1	Erfüllung der Vorgaben	41
11.2	Beobachtungen	41
11.3	Hypothesen	41
12	Fazit	42
12.1	Beiträge dieser Arbeit	42
12.2	Vorschläge zur Weiterentwicklung	42
A	Anhang 1	43
	Abbildungsverzeichnis	44
	Tabellenverzeichnis	45

Literaturverzeichnis	46
-----------------------------	-----------

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 Angebotenen Darstellungen interagieren. Des Weiteren können Anwender frei zwischen verschiedenen Bildschirmen wechseln.

1.2 Multi-Touch Interaktion mit dreidimensionalen Szenen

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 zweidimen-

sionalen Inhalten bereits etabliert. Durch diese Erfahrungen erwarten Nutzer intuitiv Eingaben an der Oberfläche eines Bildschirmtisches durchführen zu können. Auch stereoskopische Bildinhalte sind in der Öffentlichkeit, beispielsweise durch 3D Kinofilme, längst etabliert. Für die Verbindung dieser beiden Techniken gibt es bislang jedoch keine vollständig akzeptierte Lösung.

Stereoskopisches Rendering führt zu positiver, negativer, oder null Parallaxe bei der Darstellung von Bildinhalten. Das heißt virtuelle Modelle erscheinen für die Wahrnehmung unterhalb (positive Parallaxe), oberhalb (negative 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 ([?]).

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 2D Raum wird hierfür lediglich eine Achse benötigt. Da die Eingabeparameter eines Berührungspunktes der Bildschirmfläche zweidimensional sind, ergibt sich eine Herausforderung für die Entwicklung einer geeigneten 3D Manipulationsschnittstelle ([?]).

1.3 Ziel der Arbeit

Die Verwendung von Projektionstischen zur Darstellung stereoskopischer Bildinhalte begünstigt die Verwendung von taktilen Eingaben. Durch blickabhängige Renderingverfahren werden kollaborative Anwendungen leicht realisierbar. Ziel dieser Arbeit ist es, eine Mehrbenutzer-Schnittstelle zur touch-basierten Applikationsinteraktion mit dreidimensionalen Inhalten zu entwickeln.

Kapitel 2

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 2.1 vorgestellt. Abschnitt 2.2 beleuchtet die technischen Voraussetzungen zur Umsetzung des Multi-Touch Tisches. Letztlich wird in Abschnitt 2.3 ein Überblick über die softwareseitige Integration des MSER Finger und Hand Trackings in das genutzte Applikationsframework Avango gegeben.

2.1 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 XX.).

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 RE (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 Stewénus, 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 grundlegenden 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 Farbinsintensität aufweisen

Wie in Abbildung XX. zu erkennen, 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.

2.2 Technische Voraussetzungen

Für die technische Umsetzung des in Abschnitt 2.1 beschriebenen MSER Systems im Kontext der Multi-Touch Erkennung wird in unserem Labor ein rückseitiges, diffuses Infrarotbeleuchtungssetup genutzt. Hierbei wird eine diffuse Infrarot emittierende Lichtquelle unterhalb des Bildschirmtisches angebracht. Ein Spiegel lenkt das Licht senkrecht auf eine matte und lichtdurchlässige Projektionsfläche. Das von dieser Ebene reflektierte Licht wird durch den Spiegel gelenkt und von einer Infrarotkamera zur weiteren Verarbeitung aufgenommen. Abbildung XX 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.

2.3 Implementierung

Es existiert eine Anbindung des MSER Algorithmus zur Touch Erkennung, an das am Lehrstuhl entstandene Applikationsframework Avango¹. Die ana-

¹<http://www.avango.org/>

lysierten Eingabedaten werden basierend auf dem TUIO Protokoll² an die Anwendung gesendet.

In Abschnitt 2.3.1 wird eine Übersicht über die mit dem TUIO Protokoll bereitgestellten Datenstrukturen gegeben. Abschnitt 2.3.2 beschreibt das Auftreten von Jittering bei der Eingabeerkennung und wie mit diesem Problem umgegangen werden kann.

2.3.1 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*, die zugewiesene *SessionID*, LIST ALL PARAMETERS. Ein weiterer Stationstyp ist für die Verwaltung handspezifischer Details verantwortlich. Hierzu gehören die *SessionID* der Hand, eine Liste von *FingerSessionIDs* der zugewiesenen Finger, LIST ALL PARAMETERS.

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*.

2.3.2 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

²<http://www.tuio.org/>

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. Dieser Ansatz überzeugt vor allem durch seine niedrige Rechenintensität, sowie die geringe Latenzanfälligkeit bei schnellen Bewegungen.

Kapitel 3

Anforderungsanalyse

Hier wird das Kapitel Anforderungsanalyse beschrieben. Das Kapitel besteht aus den Abschnitten 3.1, 3.2, 3.3, 3.4 und 3.5.

3.1 Mehrbenutzer Touch-Eingaben

Anforderung: Fehleingaben durch gleichzeitiges Interagieren verschiedener Nutzer mit dem Bildschirmtisch sollten vermieden werden.

Die meisten 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 kann, besteht folglich die Möglichkeit, dass der Tisch zur gleichen Zeit von mehreren Personen Eingaben erhält. Diese Eingabekombinationen können leicht zu Manipulationen der Applikation führen, welche vom Nutzer nicht beabsichtigt waren und somit vermieden werden sollten.

Abbildung XX zeigt ein konkretes Anwendungsszenario. Hier versammeln sich eine bestimmte Anzahl Nutzer um gemeinsam Eigenschaften einer virtuellen Welt zu untersuchen. Während ein Nutzer beidhändig eine Interaktionsgeste ausführt, setzt ein anderer Nutzer seine Hand auf einen markanten Inhalt der Szene. Dabei berührt er ebenfalls die Tischfläche. In diesem Fall sollte sichergestellt sein, dass das System die Semantik der Touchpunkte erkennt und eine sinnvolle Entscheidung bezogen auf die abgeleitete Interaktion trifft.

3.2 Tiefenwahrnehmung durch Stereo- und Bewegungsparalaxe

Anforderung: Konflikten der Tiefenwahrnehmung durch Berührung negativ paralaxer Bildareale sollte entgegen gewirkt werden.

Anforderung: Die Interaktion mit positiv und negativ paralaxen Objekten sollte nicht durch Bewegungsparalaxe erschwert werden.

3.3 Interaktionsziele

Anforderung: Manipulationen der Applikation, sollten basierend auf dem Interaktionsziel des Nutzers bestimmt werden.

3.4 Explizite und implizite Kontrolle der involvierten Freiheitsgrade

Anforderung: Das System sollte auch für unerfahrene Nutzer leicht benutzbar sein.

Anforderung: Es sollte dem Nutzer möglich sein, alle für das jeweilige Interaktionsziel erforderlichen Freiheitsgrade getrennt voneinander zu bedienen.

Anforderung: implizite kontrolle????

3.5 Visuelle Rückmeldung

Anforderung: Der Umgang mit der Schnittstelle sollte mit visuellem Output unterstützt werden

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. (!?!?Des Weiteren wurden verschiedene Kriterien entwickelt, welche zur Bewertung von Multi-Touch Techniken dienen!?!?) 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 Anwendung 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 simulieren 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 (siehe Abb XX).

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 XX).

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 einem 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 XX 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 Tischebene nach oben. Der Ankerpunkt bestimmt durch Positionsveränderung die x- und y- Position des Ballons über der Interaktionsfläche (siehe Abb XX.).

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.

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 (siehe Abbildung XX.1). 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 (siehe Abbildung XX.2). Abbildung XX illustriert das 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 Balloon 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 zeigen sich durch ihre physikalisch motivierte Entwicklung als intuitiv und nutzerfreundlich für die 3D Objektmanipulation.

??CITE??

Aus diesem Grund fließen einige der von Hancock et al. entwickelten Konzepte maßgeblich in die entstandene 3D Multi-Touch Navigation ein. Sticky Tools ersetzen die Funktionalität der herkömmlichen 2D Skalierungsgeste mit einer expliziten Höhenanpassung. Durch die perspektivische 2D 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 Entfernungen zur Bildebene 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 Distanzen 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 fester Position 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. entwickelt abgeändert wurde, um einen 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 zusätzlichen 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.

Wartell beschreibt eine durch *Frame Cancellation* entstehende schwache Tiefenwahrnehmung 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 XX. zeigt, dass ohne Anwendung der Technik im Bild des rechten Auges ein kleinerer Ausschnitt der Geometrie zu sehen ist als im linken. 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]. Abbildung XX. zeigt ein einfaches Stereo-Viewing-Setup. Der dargestellte 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.

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 Tischebene zu erreichen. Valkov et al. beschreiben eine Wahrnehmungsdiskrepanz die sich daraus ableitet [Valkov et al., 2011]. Demnach ist die Wahl des Fokuspunktes für die Beobachtung des Interaktionsvorgangs doppeldeutig. Ein Objekt in Negativ- Parallaxe liegt oberhalb der Tischebene, 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 Wahrnehmung des parallaxen Inhalts bei Fokussierung der Hand verschwimmen. Abbildung XX 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 *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 [?]. 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

Stereoskopisch Projektionssysteme erzeugen durch physikalisch motiviertes Rendering die Illusion einer tatsächlichen Dreidimensionalität der dargebotenen Inhalte. Die in Abschnitte 5.1 und 5.2 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. beschriebenen 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 Burder 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 Tischebene erweitert werden.

Die im System von de la Rivière et al. integrierte Funktionalität beseitigt das Auftreten von Wahrnehmungskonflikten durch Eingreifen in virtuelle Modelle [?]. 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 wurden 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 CITE STICKY-TOOLS. Nach dieser Strategie soll zu jedem Zeitpunkt der Interaktion eine orthogonale Verbindung, zwischen der Eingabeposition auf dem Bildschirm und der darunter liegenden Geometrie, bestehen.

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.

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. Abbildung XX veranschaulicht diesen Zusammenhang.

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_2 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. Abbildung XX zeigt wie die Skalierung der Projektionsfläche einer virtuellen Kamera, die Größenrelation der dargestellten Geometrie beeinflusst. 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 einer von den Erkenntnissen der Balloon Selection CITE BALLOON 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 einer Seilwinde vergleichen. Abbildung XX illustriert diesen Zusammenhang.

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 der Geometrie.

TODO FORMEL FERTIG

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 REF ARCBALL RELATED vorgestellte Ansatz von CITE ARCBALL implementiert.

Für die Bestimmung der Manipulation wird die Hand eines Nutzers mit genau zwei aufgesetzten Fingern verfolgt. Initiiert der Nutzer die Geste durch Aufsetzen der Hand, so wird ein direkter Angriffspunkt auf der Geometrie mit orthogonaler Verbindungsgeraden zur Bildschirmfläche berechnet. Bis zur Beendigung 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 XX 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, 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 Translation und 3D Rotation jeweils nur eine Form der Transformation steuerbar. Im Kon-

text ihrer Anwendung zeigen sich 3D Translation, sowie 3D Rotation, ebenfalls als nutzerfreundliche Strategien zur Manipulationen der von ihnen bestimmten Freiheitsgrade.

Die Seilwindenstrategie 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 Seilwindenstrategie, 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 Freiheitsgrade explizit umzugehen. Die Bedienbarkeit eines solchen Systems könnte jedoch leicht komplex werden.

Wie Anforderung XX vorgibt soll das System für Nutzer, ungeachtet ihres Vorwissens, leicht handhabbar sein. Um diese Anforderung zu unterstützen beschreibt und evaluiert dieses Kapitel 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. Es wird dabei sowohl die Distanz der 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 XX. zeigt die Auswirkung von *Levelling* auf eine beispielhafte Visualisierung.

Die Punkte auf der Geometrie werden durch einen Schnittpunkt 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 XX 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) beigelegt werden. Wir nennen diesen Modus Rotation, Translation und Skalierung im Bildraum, erweitert durch *Levelling* (RTS+L).

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 *Primärkontakts* liegt näher an der Tischebene oder ist weiter über der dieser als der des *Sekundärkontakts*. Ziel des *Depth-Levellings* ist es, den Geometriereferenzpunkt des *Primärkontakts* durch Translation auf die Projektionsebene zu bringen. Um dies zu erreichen wird das Viewing Setup schrittweise entlang der Bildschirmnormalen verschoben.

¹im Screenspace

Die Schrittweite der Translation wird bei RTS+L durch einen Faktor festgelegt welcher aus der Relation zwischen den *Touch-Kontakte* 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-Kontakten 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.

CALCULATION

Abbildung XX zeigt die Auswirkung der verschiedenen Berechnung auf den *Levelling* Vorgang. Sollte die nach einem der genannten Ansätze ermittelte Levelling-Distanz größer oder gleich der Länge des Richtungsvektors zwischen dem Geometrie- und dem Bildschirmpunkt des *Primärkontakts* sein, so wird stattdessen eine Translation um diesen Vektor angewandt. Nach dieser Bewegung befindet sich der Geometriereferenzpunkt des *Primärkontakts* auf Tischebene 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 *Primärkontakts* würde durch Translation von der Bildschirmenebene entfernt werden. Dies führt zur Aufhebung des durch *Depth-Levelling* erreichten Interaktionsziels. Stattdessen wird eine Rotation um den Bildschirmpunkt des *Primärkontakts* vorgenommen. Desweiteren muss eine Skalierung in Richtung des selbigen Referenzpunkt erfolgen. Mit diesem Schritt soll gewährleistet werden, dass der Richtungsvektor zwischen Geometrie- und Bildschirmpunkt des *Sekundärkontakts* weiterhin senkrecht auf der Bildebene steht.

Die Parametrisierung der Rotation bestimmt sich wie folgt. Sei G_2 der Geometrieschnittpunkt des *Sekundärkontakts* 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 *Primärkontakts* 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:

$$\alpha_L = \arccos(|V_{S_1G_2}| * |V_{S_1G_2'}|)$$

$$R_L = |V_{S_1G_2}| \times |V_{S_1G_2'}|$$

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

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

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

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 Projektionsebene 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 XX zeigt ein einfaches Beispiel der Folge dieses Problems. Hier wird eine zweidimensionale Fläche windschief zur Tischebene unter dieser positioniert. Nach mehrmaligem Anwenden liegt die Geometrie in der Bildebene. Eine Umkehrung dieses Zustands ist nicht möglich.

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 Bildschirm 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 2 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 Eingabeinformation 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 Positionswert 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 das 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. Abbildung XX verdeutlicht diesen Zusammenhang.

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 XX 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 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 Bildebene 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. Ein

Anfasser in der Ecke der *Magic Linse* 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 Verbindungslinie 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. In Abbildung XX 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 zweihändigen Input gebunden. Die Entscheidung zur Aktivierung einer dieser Modi wird lediglich durch zeitliche Auswertung getroffen. Bei der Analyse des Zustandsdiagramms (siehe Abbildung XX) 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 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

Abbildung von Touch Eingaben in der Applikationsstruktur

Hier wird das Kapitel Abbildung von Touch Eingaben in der Applikationsstruktur beschrieben. Das Kapitel besteht aus den Abschnitten 9.1, 9.3, 9.2 und 9.4.

9.1 Multi-Touch Input Pipeline

Die Verarbeitung von Touch Eingaben erfolgt in drei wesentlichen Schritten, welche in Abbildung XX illustriert werden. Schritt 1 wird einmalig zu Applikationsstart ausgeführt. Hierbei wird für die Maximalanzahl an gleichzeitig erkennbaren Händen jeweils ein Touchwerkzeug angelegt. 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 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 2 beschriebenen MSER Algorithmus Fingerkontaktpunkte ermittelt werden, folgt Schritt 2. An dieser Stelle werden die Daten aus dem TUIO 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.

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.

9.2 Touch Navigation

Hier steht Inhalt zu Touchwerkzeugen und Applikationsmanipulatoren.

9.3 Representation im Szenegraph

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

TODO SCENEGRAPH PLUS ERKLÄRUNG

9.4 Diskussion

Hier steht Inhalt zur Diskussion.

Kapitel 10

Virtuelles Freischneiden der Hand während Berührungseingaben

Negativ parallaxe Bildausschnitte können wie in Kapitel 5 beschrieben Wahrnehmungskonflikte hervorrufen, wenn der Nutzer mit der Tischoberfläche interagiert. In den Abschnitten ?? und ?? wurden verschiedene Konzepte vorgestellt, welche zeigen wie mit diesem Problem umzugehen ist.

Diese Techniken umgehen jedoch lediglich den direkten Umgang mit negativ parallaxen Modellen, oder wurden im Kontext der 3D Multi-Touch Interaktion noch nicht umgesetzt.

In diesem Kapitel wird eine im Rahmen dieser Arbeit entstandene Implementierung der See Through Technik, für die Eliminierung von Wahrnehmungskonflikten, vorgestellt. Durch Anwendung der See Through Technik werden die von der Hand des Nutzers verdeckten Areale der Visualisierung, oberhalb der Tischoberfläche, freigeschnitten.

In Abschnitt 10.1 wird der theoretische Ansatz vorgestellt. Abschnitt 10.2 stellt die Integration der See Through Technik in die Applikationsstruktur vor. Zuletzt werden in Abschnitt 10.3 die Vorteile und Limitierungen des Ansatzes aufgeführt.

10.1 Ansatz

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

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 XX 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. Abbildung XX verdeutlicht diesen Zusammenhang.

Um Wahrnehmungskonflikte für alle Nutzer 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 (siehe Abbildung XX).

10.2 Implementierung

Die Anwendung der Show 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 Fragmentshader 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.

10.3 Vorteile und Limitierungen

Durch die Anwendung der Show 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 Show 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 interagiert 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 (siehe Abbildung XX).

Die Initiierung der Show 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 11

Diskussion

Hier wird das Kapitel Diskussion beschrieben. Das Kapitel besteht aus den Abschnitten 11.1, 11.2 und 11.3.

11.1 Erfüllung der Vorgaben

Hier steht Inhalt zur Erfüllung der Vorgaben.

11.2 Beobachtungen

Hier steht Inhalt zu Beobachtungen.

11.3 Hypothesen

Hier steht Inhalt zu Hypothesen.

Kapitel 12

Fazit

Hier wird das Kapitel Fazit beschrieben. Das Kapitel besteht aus den Abschnitten 12.1 und 12.2.

12.1 Beiträge dieser Arbeit

Hier steht Inhalt zu Beiträgen dieser Arbeit.

12.2 Vorschläge zur Weiterentwicklung

Hier steht Inhalt zu Vorschlägen zur Weiterentwicklung.

Anhang A

Anhang 1

Im Anhang kann auf Implementierungsaspekte wie Datenbankschemata oder Programmcode eingegangen werden.

Abbildungsverzeichnis

Tabellenverzeichnis

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.
- [Autodesk, 2008] Autodesk (2008). Stereoscopic filmmaking whitepaper, the business and technology of stereoscopic filmmaking.
- [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.
- [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 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.
- [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 Stewénus, 2008] Nistér, D. and Stewénus, 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.

[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.