Fachhochschule Bonn-Rhein-Sieg
University of Applied Sciences
Fachbereich Informatik

Masterarbeit

Verschiedene Ansichten für mehrere Anwender in gemeinsam genutzten virtuellen Umgebungen

Martin Friedrich

14. Oktober 2008

Betreuer:
Prof. Dr. Wolfgang Heiden
Externer Betreuer:
Dipl.-Inform. Roland Kuck

Eidesstattliche Erklärung:

Hiermit versichere ich, die vorliegende Arbeit selbständig und unter ausschließlicher Verwendung der angegebenen Literatur und Hilfsmittel erstellt zu haben.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Köln, 14. Oktober 2008

Martin Friedrich

Inhaltsverzeichnis

1 Einleitung **7**

1.1 Motivation . . . 7
1.2 Ziele der Arbeit . . . 8
1.3 Gliederung . . . 10

2 Grundlagen **11**

2.1 Virtuelle Umgebungen . . . 11
2.1.1 Stereoskopische Displaytechniken . . . 11
2.1.2 Tracking . . . 14
2.2 Multi-User VR-Displays . . . 15
2.2.1 Spatial Barrier Displays . . . 15
2.2.2 Multi-Viewer Displays . . . 17
2.2.3 Multi-Viewpoint Images . . . 20

3 Vorangegangene Arbeiten **21**

3.1 AVANGO™ . . . 21
3.1.1 Szenegraph . . . 22
3.1.2 Field Container . . . 22
3.1.3 Evaluation Order . . . 23
3.1.4 Scripting-Interface . . . 24
3.1.5 Container . . . 24
3.2 Virtualisierung von Skulptur . . . 24
3.3 Kollaborative Interaktion in virtuellen Umgebungen im kunsthistorischen Kontext . . . 27
3.3.1 Aufbau der Applikation . . . 27

4 Vorbereitende Arbeiten **31**

4.1 Portierung der Applikation nach AVANGO NG . . . 31
4.1.1 Programmstruktur . . . 31

4.1.2 Viewing Setup . . . 34
4.2 Generierung mehrerer Ansichten . . . 35
4.3 Preview . . . 37

5 Darstellung mehrerer Ansichten 39
5.1 Motivation . . . 39
5.2 Anforderungen . . . 40
5.3 Switchen . . . 42
5.4 Picture in Picture . . . 42
5.5 Splitscreen . . . 44
5.5.1 Rotation des Sichtkegels . . . 48
5.5.2 Translation des Sichtkegels . . . 52
5.6 Einbindung in die bestehende Applikationsstruktur . . . 56
5.7 Evaluation der Ergebnisse . . . 58
5.7.1 Vorgehensweise . . . 58
5.7.2 Aufgabenstellung . . . 59
5.7.3 Auswertung . . . 60

6 Pfadkorrektur 62
6.1 Motivation . . . 62
6.2 Anforderungen . . . 63
6.3 Methoden zur Pfadkorrektur . . . 64
6.3.1 Translation der virtuellen Kamera . . . 64
6.3.2 Anhalten der Tour . . . 66
6.3.3 Kombination beider Verfahren . . . 67
6.4 Erkennen der problematischen Situationen . . . 68
6.4.1 Pfadbasierter Ansatz . . . 68
6.4.2 Statebasierter Ansatz . . . 69
6.4.3 Verbesserter statebasierter Ansatz . . . 71
6.5 Einbindung in die bestehende Applikationsstruktur . . . 74
6.6 Ergebnisse . . . 75

7 Empirischer Test zum gemeinsamen Raum 78
7.1 Aufbau . . . 78
7.2 Ergebnisse . . . 81

8 Fazit 84

Abkürzungsverzeichnis **87**

A Auswertung des Fragebogens **88**

Abbildungsverzeichnis **91**

Literaturverzeichnis **93**

1 Einleitung

1.1 Motivation

Für eine detaillierte Analyse von Objekten ist es hilfreich, sie aus verschiedenen Blickwinkeln zu untersuchen. Ein direkter Vergleich eines Objekts aus unterschiedlichen Perspektiven kann dazu verwendet werden, bekannte Theorien zu bestätigen oder zu widerlegen. Kunstwissenschaftler weltweit setzen für diesen Zweck zwei Diaprojektoren ein, die nebeneinander unterschiedliche Aufnahmen eines Objekts projizieren. Diese Bilder können direkt miteinander verglichen werden, wodurch exaktere Ergebnisse als bei der Untersuchung eines einzelnen Bildes erreicht werden können. Die Übertragung dieser Vorgehensweise in die virtuelle Realität bietet für das Projekt *Virtualisierung von Skulptur* neue Möglichkeiten, um die Anordnung der Skulpturen von Gian Lorenzo Bernini in der Galleria Borghese zu analysieren. Das Projekt *Virtualisierung von Skulptur* [26] ist ein von der Deutschen Forschungsgemeinschaft (DFG) gefördertes interdisziplinäres Projekt des Fraunhofer-Instituts für Intelligente Analyse- und Informationssysteme (IAIS) und dem kulturwissenschaftlichen Forschungskolleg „Medienumbrüche“ (FK 615).

Die Galleria Borghese beinhaltet vier barocke Skulpturen, die von Gian Lorenzo Bernini (1598-1680) erstellt wurden. Die Aufstellung der Skulpturen wurde von Bernini so konzipiert, dass die Figuren miteinander interagieren. Da die Figuren in der Vergangenheit mehrfach an unterschiedlichen Orten aufgestellt wurden und die ursprüngliche Aufstellung nicht dokumentiert wurde, ist sie mit der Zeit verloren gegangen. Kunsthistoriker sind deshalb auf der Suche nach der ursprünglichen Aufstellung der Skulpturen, wie sie von Bernini vorgesehen war. Da ein manuelles Umstellen der Skulpturen zu aufwändig und im normalen Museumsbetrieb unmöglich wäre, wurden diese Skulpturen im Rahmen des Projekts *Bernini Digital* [9] digital repliziert und in einer virtuellen Umgebung zugänglich gemacht.

In der Diplomarbeit von Tim Jansen über die „*Kollaborative Interaktion in virtuellen Umgebungen im kunsthistorischen Kontext*“ [15], wurde ein Interaktionsinterface für diese virtuelle Umgebung entwickelt, mit der ein Benutzer die Begehung der Galleria Borghese steuern kann. Damit mehrere Kunsthistoriker die Galleria Borghese gemeinsam untersuchen können, erfolgt die Visualisierung der virtuellen Umgebung in einem Virtual-Reality-Display (VR-Display) namens TwoView. Das TwoView [11] wurde entwickelt, um kollaboratives Arbeiten in virtuellen Umgebungen für zwei Personen zu ermöglichen.

Die Begehung der Galleria Borghese im TwoView-Display erfolgt durch Wege. Diese Wege können sowohl vorgegebene Touren als auch selbst definierte Pfade sein. Befinden sich zwei Betrachter im TwoView, stehen sie immer an derselben virtuellen Position desselben Weges. Die Betrachter können sich im TwoView auf einer bestimmten Fläche frei bewegen, wodurch sie vom eigentlichen Wegverlauf abweichen können. Falls sich ein Benutzer zu weit von einem Weg entfernt, können beim Durchqueren von schmalen Passagen Probleme auftreten. In diesem Fall wird der Benutzer durch eine Wand geführt. Durch diese Kollision mit der Wand wird die Immersion des Benutzers gestört, weshalb Lösungen gefunden werden sollen, um dieses Kollisionsproblem beheben. Die Immersion beschreibt den Grad des Eintauchens in eine virtuelle Umgebung. Je länger ein Benutzer in der virtuellen Welt ungestört agieren kann, desto intensiver fühlt er sich in diese ein [8].

1.2 Ziele der Arbeit

In dieser Arbeit werden zwei unterschiedliche Aspekte zum kollaborativen Arbeiten in gemeinsam genutzten virtuellen Umgebungen behandelt. Einerseits sollen einem Betrachter mehrere Ansichten präsentiert, andererseits sollen Konzepte und Lösungen erarbeitet werden, die den Bruch der Immersion beim Passieren von Durchgängen verhindern. Für beide Aspekte werden verschiedene Lösungsmöglichkeiten vorgestellt, die die Interaktion der Benutzer in der virtuellen Umgebung verbessern sollen.

Darstellung mehrerer Ansichten:
Um verschiedene Ansichten gleichzeitig darstellen zu können, muss die Projektionsfläche aufgeteilt werden. Da im kunsthistorischen Bereich für diesen Zweck zwei Diaprojektoren eingesetzt werden, die nebeneinander zwei unterschiedliche Bilder präsentieren, könnte eine Art Splitscreen zur gleichzeitigen Visualisierung zweier Ansichten umgesetzt werden. In dieser Arbeit soll untersucht werden, inwieweit eine solche Aufteilung der Projektionsfläche das analytische sowie das kollaborative Arbeiten beeinflusst.

Pfadkorrektur:
Die Applikation aus der Diplomarbeit „*Kollaborative Interaktion in virtuellen Umgebungen im kunsthistorischen Kontext*“ [15] ermöglicht es, Wege in der virtuellen Galleria Borghese zu erstellen und abzulaufen. Beide Benutzer befinden sich immer an derselben virtuellen Position desselben Weges.

Durch die Bewegungsfreiheit, die den Benutzern innerhalb des TwoView gegeben wird, können sie ihre virtuelle Position beeinflussen. Ihre virtuelle Position bestimmt sich aus dem Wegpunkt und ihrer realen Position im TwoView. Dadurch können sie sich vom eigentlichen Wegverlauf entfernen. Verläuft ein solcher Weg durch eine Tür, und ein Betrachter hat sich zu weit vom Weg entfernt, wird dieser durch eine Wand geführt. Um dieses Kollisionsproblem zu beheben könnte eine Pfadkorrektur vorgenommen werden, die den Betrachter durch den Durchgang leitet.

Durch die Pfadkorrektur wird die virtuelle Position des Betrachters verändert. Da seine reale Position im TwoView jedoch gleich bleibt, entsteht eine Diskrepanz von der virtuellen zu der realen Position des Benutzers. Somit bekommt er aufgrund dieser Abweichung einen falschen Bildausschnitt dargestellt, der nicht zu seiner eigentlichen Position passt. Durch diesen Eingriff wird die exakte Interaktion beider Personen gestört und das kollaborative Arbeiten beeinträchtigt. Diese Problematik soll untersucht und eine Lösungsmöglichkeit präsentiert werden.

1.3 Gliederung

Diese Arbeit gliedert sich in acht Kapitel.

In Kapitel zwei werden die wichtigsten Komponenten virtueller Umgebungen erläutert, die für diese Arbeit relevant sind. Zudem werden einige repräsentative VR-Displays vorgestellt, die kollaboratives Arbeiten für mehrere Benutzer ermöglichen.

Anschließend werden im dritten Kapitel vorangegangene Arbeiten präsentiert, die die Grundlagen für diese Masterarbeit darstellen.

Alle vorbereitenden Arbeiten, die für die Anfertigung dieser Masterarbeit nötig waren, werden in Kapitel vier beschrieben.

Das fünfte Kapitel behandelt die Visualisierung mehrerer Ansichten. Hierzu werden verschiedene Verfahren vorgestellt, die abschließend in einem Usertest evaluiert werden.

In Kapitel sechs werden unterschiedliche Ansätze für eine Pfadkorrektur vorgestellt, um das Kollisionsproblem bei schmalen Durchgängen zu lösen.

Anschließend wird in Kapitel sieben ein Test beschrieben, um die Beeinträchtigung des kollaborativen Arbeitens durch unterschiedliche Kameratransformationen beider Benutzer zu bewerten.

Kapitel acht fasst die Ergebnisse der Arbeit kurz zusammen und gibt einen Ausblick für weitere Entwicklungen, die im Anschluss dieser Arbeit erfolgen können.

Die Ergebnisse der Fragebögen des Benutzertests werden im Anhang beigelegt.

2 Grundlagen

2.1 Virtuelle Umgebungen

In dem Projekt *Virtualisierung von Skulptur* [26] wurden die Skulpturen von Gian Lorenzo Bernini digital repliziert und in einer virtuellen Umgebung zugänglich gemacht, weshalb in diesem Abschnitt die grundlegenden Techniken virtueller Umgebungen erläutert werden.

Virtuelle Umgebungen sind computergenerierte Welten, die den Betrachter in eine virtuelle Realität eintauchen lassen. Um einen hohen Grad der Immersion zu erreichen, werden verschiedene Sinne des Benutzers angesprochen. Zudem wird die Möglichkeit bereitgestellt, mit den virtuellen Objekten zu interagieren. Das Hauptaugenmerk wird in der Regel auf den visuellen Reiz gelegt. Weiterhin können auch die auditive, olfaktorische und haptische Wahrnehmung simuliert werden. Da für diese Arbeit lediglich die visuelle Wahrnehmung relevant ist, wird auf die Simulation der anderen Reize nicht eingegangen.

2.1.1 Stereoskopische Displaytechniken

Stereoskopische Displaysysteme werden in vielen Bereichen eingesetzt, um dreidimensionale Umgebungen visuell darzustellen. Aufgrund der Vielzahl möglicher Techniken werden hier nur die gängigsten Verfahren vorgestellt. Alle hier beschriebenen Verfahren erzeugen ihre dreidimensionale Wirkung durch die Darstellung von stereoskopischen Bildpaaren, die auch als stereoskopische Halbbilder bezeichnet werden. Dabei werden für das linke und rechte Auge unterschiedliche zweidimensionale Bilder dargestellt. Bei allen hier vorgestellten Verfahren wird eine spezielle Brille vorausgesetzt, die die Bilder für das linke und das rechte Auge separieren kann [21], [25], [14]. Es existieren sowohl aktive als auch passive Stereo-Techniken, die

eine unterschiedliche Darstellung der stereoskopischen Halbbilder vornehmen. Passive Verfahren stellen die Bilder für beide Augen gleichzeitig dar, während aktive die beiden Bilder zeitlich versetzt darstellen, wodurch die Brille die Separation der Bilder aktiv vornehmen muss.

Anaglyphenverfahren

Das Anaglyphenverfahren ist eine passive Stereo-Technik, die das Bild für das linke und für das rechte Auge in Komplementärfarben darstellt. Hierzu können beispielsweise die Farben rot und grün verwendet werden. Die beiden unterschiedlich gefärbten Bilder werden gleichzeitig auf derselben Projektionsfläche dargestellt, wodurch sich die Einzelbilder überlagern. Für die Brillengläser werden Farbfilter in den gleichen Farben verwendet, um die Bilder für das linke und das rechte Auge zu separieren. Das grüne Brillenglas transmittiert dabei das grüne und reflektiert das rote Bild. Analog verhält es sich mit dem roten Brillenglas (siehe Abbildung 2.1(a)). Da für die Separation der Bilder für beide Augen Farbfilter verwendet werden, ist die Farbtreue der Szene durch dieses Verfahren nicht mehr gewährleistet. Die initialen sowie die Erweiterungskosten eines solchen Systems sind sehr gering.

Polarisationsverfahren

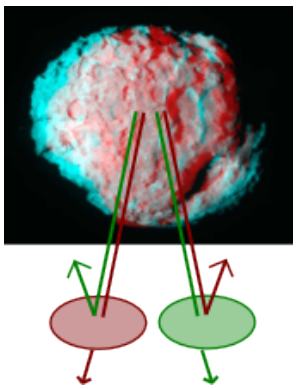
Bei der Polarisation erfolgt die Trennung der stereoskopischen Halbbilder durch unterschiedlich polarisiertes Licht. Dazu werden in der Regel zwei Projektoren eingesetzt, deren Polarisationsebenen orthogonal zueinander angeordnet sind. Die Brillengläser des Betrachters sind unterschiedlich beschichtet, so dass das linke Brillenglas beispielsweise nur vertikale und das rechte Brillenglas nur horizontale Lichtwellen transmittiert (siehe Abbildung 2.1(b)). Damit die Polarisation der Lichtwellen bei der Reflexion von der Projektionsfläche beibehalten wird, wird eine speziell beschichtete Leinwand benötigt. Da die stereoskopischen Halbbilder bei diesem Verfahren gleichzeitig dargestellt werden, handelt es sich um ein passives Verfahren.

Aufgrund dieser speziellen Leinwand und dem Einsatz von zwei Projektoren sind die initialen Kosten eines solchen Systems vergleichsweise hoch. Da die Brillen allerdings preiswert herzustellen sind, ergeben sich für eine Erweiterung des Systems auf mehrere Benutzer nur geringe Kosten.

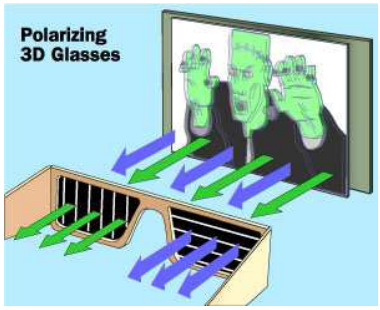
Shuttertechnik

Die Shuttertechnik ist im Gegensatz zu den bisher genannten Methoden ein aktives Verfahren. Das bedeutet, dass hierbei die Brille die Separation der dargestellten Halbbilder aktiv vornehmen muss. Die Bilder für das linke und das rechte Auge werden im schnellen Wechsel nacheinander angezeigt. Die Brille muss nun in der gleichen Frequenz ein Brillenglas auf durchlässig und das andere auf undurchlässig schalten (siehe Abbildung 2.1(c)). Die Brillengläser einer Shutterbrille bestehen deshalb aus LCD-Gläsern, deren Flüssigkristalle sich in der Frequenz der Bildwiederholungsrate des Ausgabegeräts auf transmittierend oder reflektierend schalten lassen. Um zu gewährleisten, dass ein angezeigtes Bild von nur einem Auge wahrgenommen wird, muss die Brille mit dem Ausgabegerät synchronisiert werden.

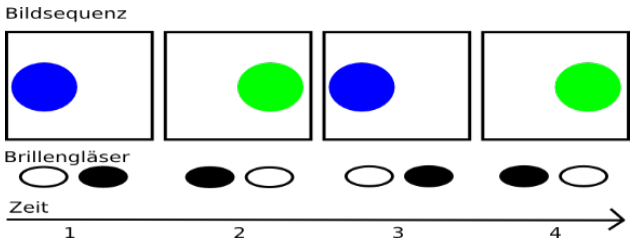
Die initialen Systemkosten sind im Vergleich zur Polarisation recht günstig, die Kosten für die Erweiterung allerdings aufgrund der teuren Shutterbrillen recht hoch.



(a) Anaglyphenverfahren (Bild nach [4])



(b) Polarisationsverfahren [12]



(c) Shuttertechnik

Abbildung 2.1: Beispiele für stereoskopische Displaytechniken

2.1.2 Tracking

Ein wichtiger Aspekt bei virtuellen Umgebungen ist die Anpassung des dargestellten Bildes an die reale Position und Blickrichtung eines Betrachters. Mit Hilfe von Tracking-Systemen können die Bewegungen der Benutzer in der realen Welt erfasst und in die virtuelle Welt übertragen werden. Wichtige Eigenschaften für Trackingverfahren sind Reichweite, Genauigkeit, Latenz und die Möglichkeit, die Position mehrerer Objekte zu bestimmen. Es gibt verschiedene Arten von Trackingverfahren, die je nach Anwendungszweck Vor- und Nachteile besitzen [24], [6]. Das Tracking in VR-Systemen erfolgt meist mit Hilfe von optischen oder elektromagnetischen Verfahren. Andere Trackingverfahren, wie beispielsweise mechanisches oder akustisches Tracking, spielen in diesem Kontext eine eher untergeordnete Rolle.

Optisches Tracking
Beim optischen Tracking werden spezielle Markierungen an Objekten angebracht, die mit Hilfe von Kameras und einer entsprechenden Software identifiziert werden können. Die Umsetzung des optischen Trackings kann sowohl aktiv als auch passiv erfolgen.

Beim passiven Tracking werden spezielle reflektierende Markierungen am Objekt angebracht, die durch Infrarotkameras erfasst werden. Um die korrekte Position bestimmen zu können, müssen mindestens zwei Kameras gleichzeitig das Objekt aufnehmen. Anhand der Position der Kameras und der aufgenommenen Bilder kann die Position des Ziels trianguliert werden.

Beim aktiven optischen Tracking können beispielsweise Leuchtdioden als Markierungen verwendet werden, die als Targets aktiviert oder deaktiviert werden können. Optische Trackingsysteme sind echtzeitfähig, einfach erweiterbar und liefern exakte Ergebnisse. Zudem können sie mehrere Objekte gleichzeitig erfassen.

Elektromagnetisches Tracking
Beim elektromagnetischen Tracking wird ein Transmitter im Raum platziert, der abwechselnd drei orthogonal zueinander stehende magnetische Felder erzeugt. Die Sensoren an den Zielobjekten können die unterschiedlichen magnetischen Feldervektoren erfassen und ihre Position und Ausrichtung im Raum ermitteln. Solche Verfahren sind kostengünstig umzusetzen, die Positionsbestimmung kann aber durch in der Nähe befindliche Metalle, Monitore oder Stromquellen gestört werden.

2.2 Multi-User VR-Displays

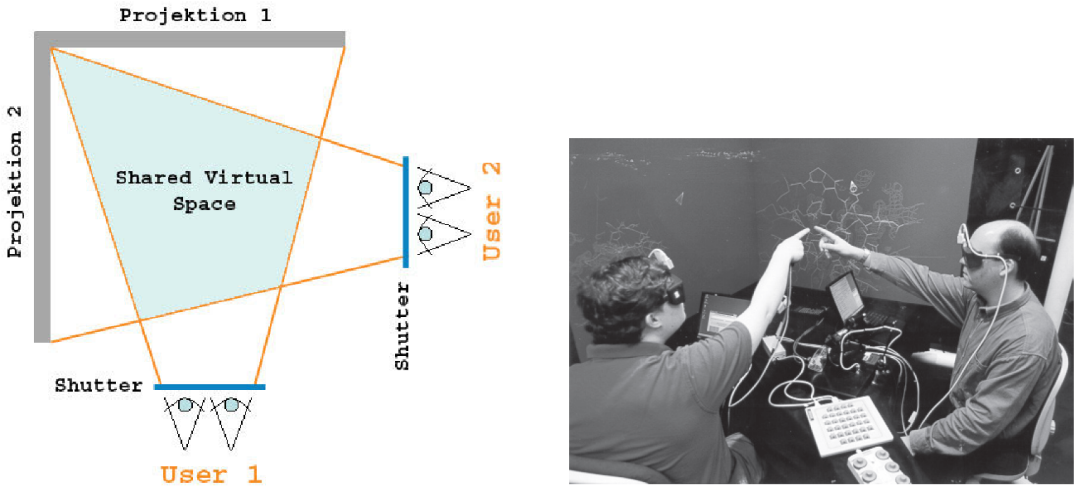
In dieser Arbeit werden verschiedene Aspekte zum kollaborativen Arbeiten in gemeinsam genutzten virtuellen Umgebungen behandelt. Hierfür ist es wichtig, dass mehrere Anwender exakt in einer solchen Umgebung miteinander interagieren können. Aus diesem Grund muss die korrekte Perspektive für alle Anwender in der virtuellen Umgebung gegeben sein. Herkömmliche VR-Displays gewährleisten die perspektivische Korrektheit für höchstens eine Person und stellen die Szene nur für eine bestimmte Position vor der Leinwand korrekt dar. Damit ein Benutzer die Szene perspektivisch korrekt sehen kann, muss er deshalb an genau dieser Stelle vor der Projektionsleinwand stehen. Andere Systeme passen die Perspektive mittels Tracking an die Position eines einzelnen Benutzers an. Befinden sich mehrere Benutzer in einer solchen Umgebung, teilen sie sich die gleiche Perspektive aus unterschiedlichen Positionen, was zu einer falschen Wahrnehmung der Objekte führen kann.

Multi-User VR-Displays bieten deshalb die Möglichkeit, eine virtuelle Umgebung für mehrere Personen perspektivisch korrekt darzustellen. Erst durch die korrekte Perspektive für alle Betrachter wird kollaboratives Arbeiten ermöglicht, wodurch sie gemeinsam und exakt miteinander in einer virtuellen Umgebung interagieren können. Multi-User VR-Displays können in drei Arten klassifiziert werden, auf die im folgenden eingegangen wird [22].

2.2.1 Spatial Barrier Displays

Bei Spatial Barrier Displays werden die Bilder für verschiedene Personen räumlich getrennt voneinander dargestellt. Die Einzelbilder überlagern sich somit nicht an einer Stelle der Projektionsleinwand, sondern werden an verschiedenen Positionen oder auf verschiedenen Leinwänden dargestellt. Folgende Systeme können hierzu eingesetzt werden:

Two-User Protein Interactive Theater
Das Two-User Protein Interactive Theater (PIT) wurde von der *University of North-Carolina* entwickelt und soll unter anderem unterstützend bei der Vorhersage von



(a) Aufbau des PIT [22]

(b) Zwei Anwender benutzen das PIT [29]

Abbildung 2.2: Two-User Protein Interactive Theater

Proteinfaltung eingesetzt werden [29]. Das PIT besteht aus zwei orthogonal zueinander angeordneten Projektionsflächen, die an einem Tisch montiert sind (siehe Abbildung 2.2). Vor jeder Projektionsfläche kann eine Person sitzen und die dargestellte Szene betrachten. Da sich durch die orthogonale Anordnung der Betrachter die Sichtkegel überlappen, entsteht ein gemeinsamer Raum, in dem beide das gleiche Bild aus ihrer eigenen Perspektive sehen. Dadurch können beide Benutzer exakt miteinander interagieren.

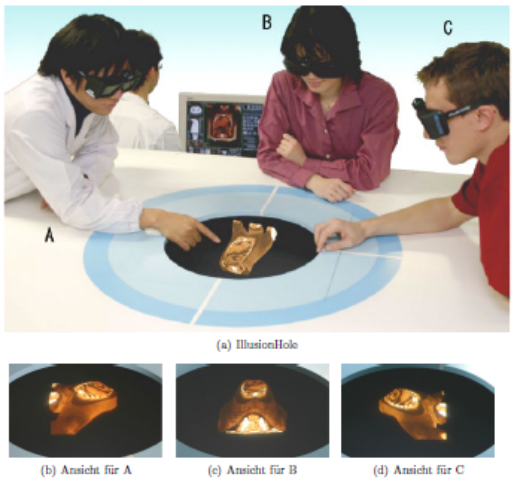
Studierstube

Die Studierstube [23] ist unter anderem ein Augmented-Reality-System, welches die reale Wahrnehmung des Betrachters um zusätzliche computergenerierte Informationen erweitert. Die Betrachter werden mit einem semi-immersiven Head Mounted Display (HMD) ausgestattet, wodurch sie sich frei in der gemeinsamen Umgebung bewegen können. Durch die Verwendung von HMDs können mehrere Betrachter sowohl dieselbe als auch unterschiedliche Applikationen ausführen.

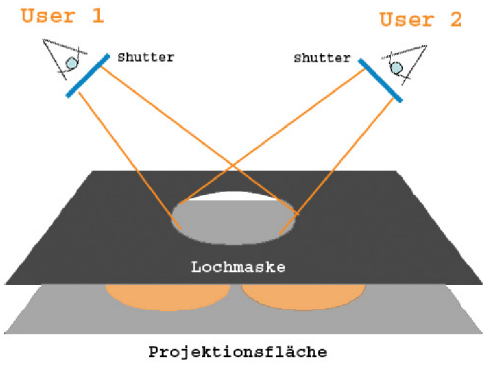
IllusionHole

Beim IllusionHole [17] ist die Projektionsfläche wie ein Tisch aufgebaut (siehe Abbildung 2.3). Die Benutzer positionieren sich um das IllusionHole und bekommen anhand ihrer getrackten Position ein perspektivisch korrektes Bild angezeigt. Die

verschiedenen Bilder der Betrachter werden an unterschiedlichen Stellen der Projektionsfläche wiedergegeben. Damit ein Betrachter nur das für ihn bestimmte Bild sehen kann, wird eine Maske mit einem Loch in der Mitte vor der Projektionsleinwand angebracht. Durch die Darstellung der Bilder an unterschiedlichen Positionen auf der Leinwand kann ein Betrachter, wenn er durch das Loch sieht, nur die für ihn bestimmten Bilder sehen. Alle anderen Ansichten werden von der Maske verdeckt.



(a) Drei Anwender benutzen das IllusionHole [18]



(b) Aufbau des IllusionHole für zwei Betrachter [22]

Abbildung 2.3: IllusionHole

VisualEyes

Das von General Motors entwickelte VisualEyes dient der Analyse von CAD-Modellen in einer CAVE [30]. Da VisualEyes an mehreren Standorten von General Motors eingesetzt wird, wurden die CAVEs untereinander vernetzt. Dadurch können die Entwickler, die sich an verschiedenen Standorten befinden, gemeinsam am Design eines Fahrzeugs arbeiten.

2.2.2 Multi-Viewer Displays

Eine zweite Klasse stellen die Multi-Viewer Displays dar. Hierbei werden die Bilder übereinander auf derselben Projektionsfläche dargestellt, um lokales kollaboratives Arbeiten zu ermöglichen. Im Folgenden werden unterschiedliche Ansätze vorgestellt:

Two-User Responsive Workbench

Die Two-User Responsive Workbench [7] ist eine Erweiterung der für einen Betrachter konzipierten Responsive Workbench [19]. Die Responsive Workbench simuliert das Arbeiten an einem Tisch. Dazu wird das Bild mit einem Projektor über ein Spiegelsystem auf eine Tischoberfläche projiziert. Bei der Two-User Responsive Workbench werden die Bilder für beide Personen mit der Shuttertechnik zeitlich versetzt dargestellt. Die Abfolge der Bilder kann *Viewer Sequential* oder *Viewer Interleaved* erfolgen.

Viewer Sequential
Die Bilder der Betrachter werden hintereinander angezeigt.
... L1 R1 L2 R2 ...

Viewer Interleaved
Die Bilder der Betrachter werden überkreuzt dargestellt.
... L1 L2 R1 R2 ...

L1: linkes Auge von Betrachter 1, R1: rechtes Auge von Betrachter 1
L2: linkes Auge von Betrachter 2, R2: rechtes Auge von Betrachter 2

Die Shutterbrillen müssen bei diesen Verfahren einen zusätzlichen Zustand besitzen, der beide Brillengläser schließt und somit das Bild für den anderen Betrachter vollständig ausblendet. Um eine flüssige Darstellung zu erreichen, muss der Stereoprojektor eine hohe Wiederholungsfrequenz besitzen.

TwoView

Das TwoView [11] wurde 2005 am Fraunhofer-Institut für Medienkommunikation (IMK) entwickelt. Um für zwei Personen ein separates stereoskopisches Bild zu erzeugen, wird sowohl eine Aktiv- als auch eine Passiv-Stereo-Projektion verwendet (siehe Abbildung 2.4). Die Trennung beider Benutzer erfolgt passiv mithilfe von Polarisationsfiltern. Das System verwendet zwei Projektoren, die mit unterschiedlichen Polarisationsfiltern ausgestattet sind. Somit können die Projektoren für jeden Benutzer ein eigenes aktives Stereobildpaar darstellen, die durch die Verwendung von Shutterbrillen wieder separiert werden können. Vor der Projektionsleinwand befindet sich eine $3 \times 2,4$ Meter große Fläche, auf der sich die Benutzer frei bewegen können. Die Position des Betrachters innerhalb dieser Fläche wird mittels optischen Trackings erfasst und für die Berechnung der richtigen Perspektive verwendet.

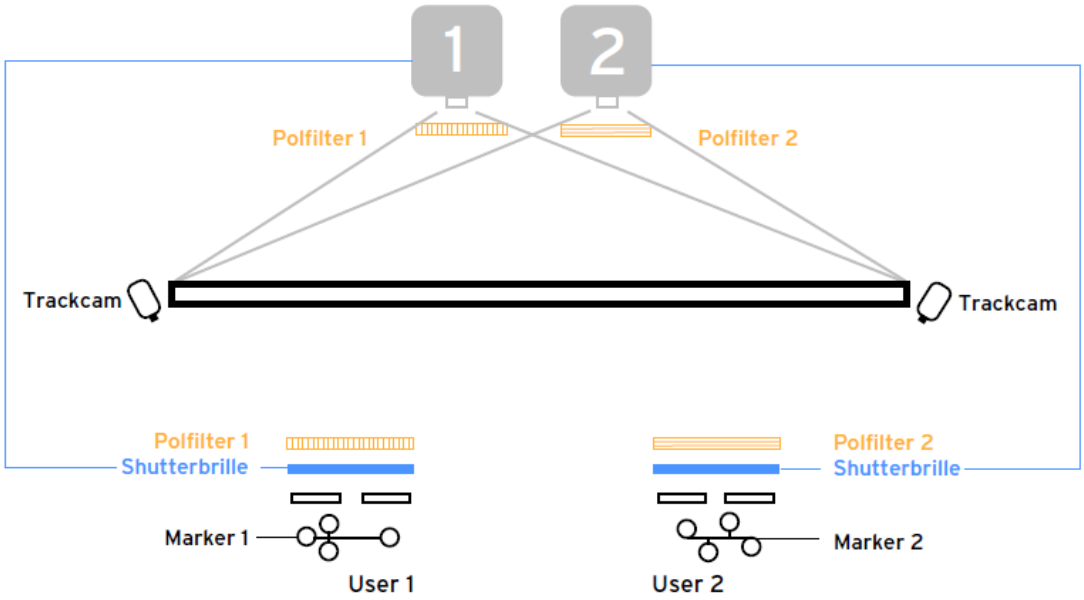


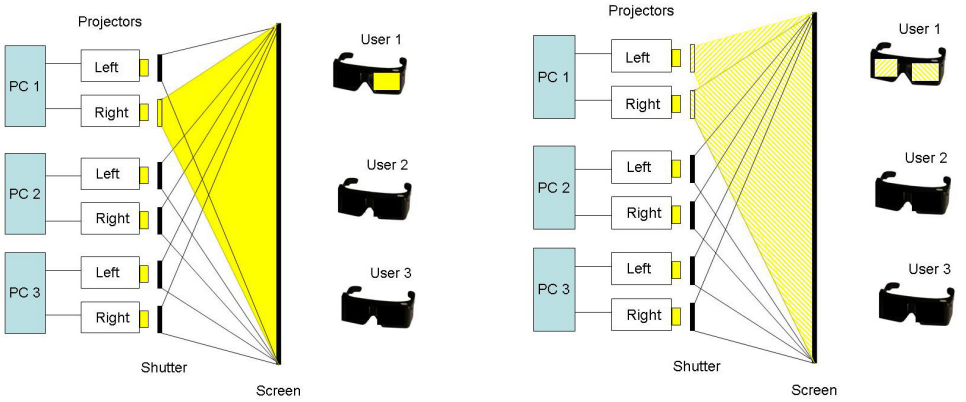
Abbildung 2.4: Aufbau des TwoView [15]

Shuttered Projectors

Bei Shuttered Projectors [13] wird das Prinzip der Aktiv-Stereoprojektion verwendet und auf mehrere Personen erweitert. Mit dem herkömmlichen Prinzip der Shuttertechnik kann nur eine Ansicht visualisiert werden. Um mehrere Ansichten unterstützen zu können, müssen die Wiedergabefrequenz der Projektoren und der Shuttermechanismus der Brillen modifiziert werden. Dadurch können mit zeitlicher Trennung unterschiedliche Bilder für mehrere Personen dargestellt werden. Ähnlich wie bei der Two-User Responsive Workbench gibt es zwei verschiedene Projektorkonfigurationen, die in Abbildung 2.5 dargestellt werden. Das Shuttered Projectors System besteht aus mehreren Aktiv-Stereo Projektoren, denen ein zusätzlicher Shutter zur Trennung der Ansichten vorgeschaltet ist. Dieser Shutter kann mechanisch als rotierende Scheibe oder durch die Verwendung einer Flüssigkristallscheibe umgesetzt werden. Anders als die bisher genannten Multi-Viewer Displays können hierbei mehr als zwei Ansichten dargestellt werden.

Das shutterbasierte Setup (Abbildung 2.5(a)) stellt die Bilder für das linke und das rechte Auge eines Benutzers zeitlich versetzt dar. Die Separation der stereoskopischen Halbbilder für die Betrachter erfolgt mit der Shuttertechnik. Bei der Kombination aus Polarisation und Shutter (Abbildung 2.5(b)) werden die stereoskopischen

Halbbilder gleichzeitig dargestellt. Die Separation der Bilder für das linke und das rechte Auge erfolgt hierbei durch das Polarisationsverfahren.



(a) Shutterbasiertes Setup (b) Kombination aus Polarisation und Shutter

Abbildung 2.5: Projektor-Konfigurationen für Shuttered Projectors [13]

2.2.3 Multi-Viewpoint Images

Die dritte Klasse von Multi-User VR-Displays stellen Multi-Viewpoint Images dar [27]. Diese sind eine softwaretechnische Erweiterung eines Single-User Displays. Die virtuelle Umgebung wird hierbei für einen festen Punkt perspektivisch korrekt berechnet und dargestellt. Befinden sich mehrere Betrachter in der virtuellen Umgebung, teilen sie sich diese Ansicht. Um dennoch exakt miteinander arbeiten zu können, besitzt jeder Betrachter ein spezielles Eingabegerät, um mit den Objekten interagieren zu können. Die virtuellen Cursor der Eingabegeräte werden für jede Person perspektivisch korrekt dargestellt. Dadurch wird die exakte Interaktion mit den virtuellen Objekten in einer perspektivisch unkorrekten Umgebung gewährleistet. Dieses Verfahren stellt eine kostengünstige Möglichkeit zum kollaborativen Arbeiten zur Verfügung. Durch die Verwendung eines Single-User Displays wird die Szene jedoch nur für höchstens eine Person perspektivisch korrekt dargestellt.

3 Vorangegangene Arbeiten

3.1 AVANGO™

AVANGO ist ein objektorientiertes Virtual-Reality-Framework [28] welches vom Fraunhofer-Institut IAIS entwickelt wurde. Es wird in vielen Projekten eingesetzt, und wurde auch für die Implementierung der Applikation aus der Diplomarbeit *„Kollaborative Interaktion in virtuellen Umgebungen im kunsthistorischen Kontext“* [15] verwendet. Da die *Darstellung mehrerer Ansichten* und die *Pfadkorrektur* (siehe Abschnitt *Ziele der Arbeit*) in die bestehende Anwendung integriert werden sollen, werden im Folgenden die zugrundeliegenden Konzepte von AVANGO näher erläutert.

AVANGO stellt eine leicht zu bedienende Schnittstelle bereit, um verteilte interaktive VR-Applikationen zu implementieren. Weiterhin werden durch das Framework verschiedenste Ein- und Ausgabegeräte abstrahiert. Die grundlegenden Komponenten von AVANGO sind ein Szenegraph-Modell, ein Scripting-Interface und das *Field-Container*-Konzept. Durch den modularen Aufbau und der Trennung der einzelnen Komponenten wird eine hohe Wiederverwendbarkeit und Flexibilität für die Entwicklung von VR-Applikationen gewährleistet.

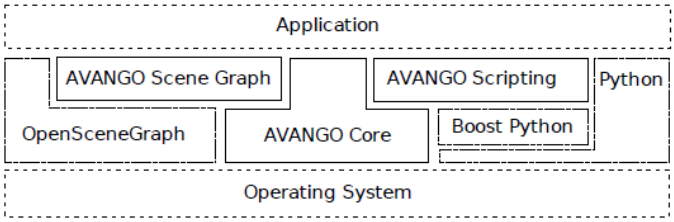


Abbildung 3.1: Aufbau von AVANGO NG

Im Jahr 2008 wurde AVANGO™ um zahlreiche Funktionalitäten erweitert und die nächste Version AVANGO NG (AVANGO Next Generation) entwickelt [20]. Der schematische Aufbau von AVANGO NG wird in Abbildung 3.1 veranschaulicht. Um eigene Applikationen zu schreiben, wird *Python* [3] als Scriptsprache eingesetzt. Für die Repräsentation einer Szene wird *OpenSceneGraph* [2] verwendet, dessen Objekte von AVANGO NG als *Field Container* in der eigenen Applikation zugänglich gemacht werden. Da die Implementierung der Darstellung mehrerer Ansichten und der Pfadkorrektur auf AVANGO NG erfolgt, werden die grundlegenden Komponenten dieses VR-Frameworks im Folgenden näher erläutert.

3.1.1 Szenegraph

Der Szenegraph ist eine Datenstruktur zum Verwalten und Beschreiben der darzustellenden Elemente einer Szene, wie zum Beispiel der Geometrie von Objekten, Lichtquellen oder Materialeigenschaften. Die Objekte des Szenegraphen sind hierarchisch angeordnet und können über Kanten verbunden werden. Diese Kanten definieren eine logische Beziehung zweier Objekte, durch die Eigenschaften vererbt werden können. Durch die hierarchische Anordnung können bestimmte Eigenschaften von Objekten an die darunterliegende Hierarchiestufe weitergegeben werden. Dies können zum Beispiel Transformationen sein, die auf ein oder mehrere Objekte angewendet werden soll.

In AVANGO™ wurde *OpenGL Performer* [1] als Szenegraph für die Repräsentation der virtuellen Umgebung verwendet. Aufgrund mangelnder Weiterentwicklung und fehlender Unterstützung der Windows Plattform wurde für AVANGO NG *OpenSceneGraph* [2] gewählt.

3.1.2 Field Container

Alle Objekte in AVANGO sind *Field Container*, die verschiedene Zustandsinformationen aufweisen können. *Field Container* können über *Field Connections* miteinander verbunden werden, um Ereignisse zu propagieren. Ändert sich der Zustand eines *Field Containers*, wird diese Änderung an alle verbundenen Objekte weitergegeben, die daraufhin aktualisiert werden müssen. Dies kann zum Beispiel durch die Manipulation eines *Field Containers* innerhalb der eigenen Applikation angestoßen werden.

3.1.3 Evaluation Order

Die Evaluation Order beschreibt die Aktualisierungsreihenfolge der *Field Container* in AVANGO. VR-Applikationen stellen Animationen dar, die sich aus vielen Einzelbildern (Frames) zusammensetzen. Die Aktualisierung der *Field Container* findet somit bei jeder Berechnung eines Frames statt, oder sobald sich der Zustand eines *Field Containers* verändert. Anschließend müssen alle verbundenen Objekte aktualisiert werden.

Die Evaluationreihenfolge ist hierarchisch strukturiert. Zuerst werden alle Sensorfelder ausgewertet, wie zum Beispiel Eingabegeräte oder Zeitzähler. Anschließend folgen die AVANGO-Objekte, die in der eigenen Applikation verwendet werden. Diese reagieren auf die geänderten Eingabewerte und berechnen ihren neuen Zustand. Die Ergebnisse werden anschließend an sogenannte Aktuatoren, wie beispielsweise die virtuelle Kamera, weitergeleitet und ausgewertet.

In manchen Fällen sind bestimmte Berechnungen abhängig von dem Ergebnis des vorangegangenen Frames. So kann die Berechnung einer neuen Position zum Beispiel von der zuvor berechneten Position abbhängig sein. Solche Verbindungen werden als Zyklen bezeichnet und sind in AVANGO prinzipiell nicht definiert. In manchen Situationen ist es nötig, einen solchen Zyklus zu definieren, dessen Auswertung aber erst im darauffolgenden Frame erfolgen soll. Solche Verbindungen zweier *Field Container* werden als *weak-connections* bezeichnet. Das Beispiel mit der Positionsberechnung kann deshalb mit Hilfe einer solchen *weak-connection* umgesetzt werden (siehe Abbildung 3.2). Die berechnete Position wird deshalb nicht direkt, sondern erst im darauf folgenden Frame ausgewertet.

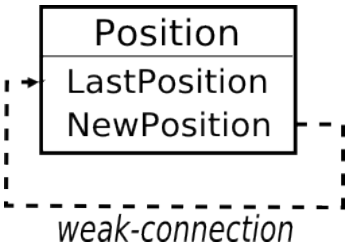


Abbildung 3.2: Zyklus mit einer *weak-connection*

3.1.4 Scripting-Interface

Das Scripting-Interface dient dazu, eigene VR-Applikationen zu implementieren. In AVANGO™ wurden Applikationen mit *Scheme* erstellt. *Scheme* [16] ist eine funktionale Programmiersprache, die nicht sehr weit verbreitet ist. Damit sich ein Benutzer beim Erlernen von AVANGO nicht zusätzlich in das Konzept funktionaler Programmierung einarbeiten muss, wird in AVANGO NG *Python* eingesetzt. *Python* bietet die Möglichkeit, objektorientiert zu arbeiten und erlaubt durch ihr leichtes Erlernen schnell, eigene Applikationen zu schreiben *(rapid prototyping)*. Die einzelnen Komponenten von AVANGO sind in *C++* geschrieben. *Python* bietet durch die *Boost Python* library eine direkte Anbindung an *C++*. Dadurch ist es nach der Umstellung auf *Python* nun möglich, eigene *Field Container* in der VR-Applikation zu definieren und dem Szenegraphen zuzufügen. In der alten Version von AVANGO mussten eigene *C++* Klassen implementiert werden, um selbst definierte *Field Container* in einem *Scheme*-Programm verwenden zu können.

3.1.5 Container

Applikationen können sich aus einer Vielzahl kleiner Komponenten zusammensetzen. Wenn mehrere Komponenten miteinander verknüpft werden, können lange Ketten entstehen, deren Verwaltung schnell unübersichtlich wird. Um dieses Problem zu reduzieren, wurde AVANGO NG um das Konzept des *Container* erweitert. Hierbei werden einzelne, unabhängige Subgraphen in einem solchen *Container* gekapselt. Die Applikation muss dadurch nur noch eine kleine Menge von *Containern* anstatt einer großen Menge von *Field Containern* verwalten.

3.2 Virtualisierung von Skulptur

Das Projekt „*B7 Virtualisierung von Skulptur: Rekonstruktion, Präsentation, Installation*“ [26] stellt den inhaltlichen Rahmen für die Implementierung der beiden Aspekte dieser Arbeit (*Darstellung mehrerer Ansichten* und *Pfadkorrektur*) dar. Hierbei handelt es sich um ein Teilprojekt des von der Deutschen Forschungsgemeinschaft (DFG) geförderten Forschungsprojekts „615“. Im Rahmen dieses Projekts untersucht das kulturwissenschaftliche Forschungskolleg „Medienumbrüche“

der Universität Siegen die beiden strukturell prägenden Medienumbrüche des 20. Jahrhunderts. Diese lassen sich als Umbruch zu den analogen Medien, zu Beginn des 20. Jahrhunderts, und als Umbruch zu den digitalen Medien, im Übergang zum 21. Jahrhundert, bestimmen.

Das Projekt „*Virtualisierung von Skulptur*“ ist ein interdisziplinäres Projekt zwischen den Bereichen Kunsthistorik, Informatik und Medienwissenschaft zur Untersuchung des Umbruchs zu den digitalen Medien. Analoge Medien, wie beispielsweise der Film, zeichnen sich durch Strukturmerkmale wie *Geschwindigkeit*, *Dynamik* und *Simultanität* aus. Hypermedien können Themen auf unterschiedliche Arten, wie zum Beispiel durch Texte, Grafiken, Videos und Tonaufzeichnungen, darstellen. Die Art und die Reihenfolge der angezeigten Informationen der Themen können frei vom Benutzer gewählt werden. Digitale Medien hingegen verwenden den Computer zur Darstellung verschiedener Hypermedien und zeichnen sich durch *Interaktivität*, *Multidimensionalität*, *Non-Direktionalität* und *Non-Sequentialität* aus [26].

Die Änderung der Strukturmerkmale bringt eine Änderung der Wahrnehmung der Medien mit sich, was bei der Betrachtung virtueller Szenen zum Ausdruck kommt. Hier steht die dreidimensionale Wahrnehmung im Vordergrund, was zu der Annahme führte, dass für die Phänomene der Virtualisierung zweidimensionale Modelle nicht mehr ausreichen. Anstelle von Malerei, Fotografie oder Film werden hierzu komplexe Kunstwerke, wie beispielsweise barocke Kunstinstallationen, zur Analyse von Figuration und Räumlichkeit benötigt. Bei solchen Kunstinstallationen spielen nicht nur die Objekte selbst, sondern auch ihre Wechselwirkung mit Licht und Schatten oder mit anderen Objekten eine wichtige Rolle. Solche Kunstinstallationen bieten deshalb neben der medienästhetischen Untersuchung von Raummodellen zusätzlich die Möglichkeit zur Untersuchung kunstwissenschaftlicher Fragen bezüglich Rekonstruktion, Präsentation, Installation und Interaktion der Modelle [26], [15].

Als Basis für das Forschungsprojekt B7 wurde deshalb die barocke Skulpturengruppe von Gian Lorenzo Bernini aus der Villa Borghese in Rom verwendet. Die Villa Borghese ist eine Parkanlage mit einem Kunstmuseum und umfasst die Galleria Borghese, die zu den berühmtesten und wertvollsten privaten Kunstsammlungen der Welt zählt. Gian Lorenzo Bernini (1598-1680) war ein bedeutender Bildhauer des 17. Jahrhunderts [9] und erstellte für die Galleria Borghese von 1618-1625 die vier Skulpturen David, Apollo und Daphne, Pluto und Proserpina sowie Äneas und Anchises (siehe Abbildung 3.3).

Abbildung 3.3: Skulpturengruppe aus der Galleria Borghese: David, Pluto und Proserpina, Äneas und Anchises, Apollo und Daphne [15]

Bezeichnend für den Barock war es, einzelne Elemente zu einem Gesamtkunstwerk zusammenzufügen. Durch das Zusammenspiel mehrerer Objekte sowie den Einfluss von Licht und Schatten erhalten die Skulpturen ihre besondere Ausdrucksstärke. So verhält es sich auch bei den vier Skulpturen von Bernini. Sie besitzen ein hohes Maß an Eigendynamik und sind als Gesamtkunstwerk zu sehen, deren einzelne Elemente sich gegenseitig beeinflussen.

Die Figuren Berninis wurden dreidimensional rekonstruiert, modelliert und in einer virtuellen Umgebung dargestellt. Da die Villa Borghese in den letzten Jahrhunderten mehrfach umgestaltet wurde, ist die ursprüngliche Aufstellung der Skulpturen mit der Zeit verloren gegangen. Nur wenige literarische Quellen enthalten Vorschläge für die ursprüngliche Gestaltung der Villa.

Da die Wechselwirkung der Skulpturen untereinander für die Wahrnehmung des Kunstwerks aber eine entscheidende Rolle spielt, ist die Aufstellung der vier Skulpturen in der Galleria Borghese für die Kunstwissenschaft von großer Bedeutung. Deshalb sind Kunsthistoriker auf der Suche nach der ursprünglichen Aufstellung der Figuren, wie sie von Bernini vorgesehen war. Da ein manuelles Umstellen der Skulpturen auf Dauer zu aufwändig wäre, wurde nach einem Weg gesucht, den Kunsthistorikern ihre Arbeit zu erleichtern. Im Rahmen des Projekts „Bernini Digital“ im Jahr 2002 wurden diese Skulpturen digital repliziert und in einer virtuellen Umgebung zugänglich gemacht. Durch die einfachen Positionierungs- und Umgestaltungsmöglichkeiten, die durch die Virtualisierung zur Verfügung stehen, können die Kunsthistoriker die Aufstellung und Wirkung der Skulpturen leichter erforschen [9].

3.3 Kollaborative Interaktion in virtuellen Umgebungen im kunsthistorischen Kontext

In der Diplomarbeit *„Kollaborative Interaktion in virtuellen Umgebungen im kunsthistorischen Kontext“* von Tim Jansen [15] wurde eine Applikation entwickelt, mit der eine exakte Analyse der Skulpturengruppe von Gian Lorenzo Bernini ermöglicht wird. Die in Abschnitt *Ziele der Arbeit* beschriebene *Darstellung mehrerer Ansichten* und *Pfadkorrektur* sollen diese Applikation erweitern, um die Interaktion der Benutzer in der virtuellen Umgebung zu verbessern, und neue Möglichkeiten zur Analyse der Sulpturengruppe bereit zu stellen.

3.3.1 Aufbau der Applikation

Die in der Diplomarbeit *„Kollaborative Interaktion in virtuellen Umgebungen im kunsthistorischen Kontext“* erstellte Anwendung besteht aus zwei Komponenten, die in diesem Abschnitt ausführlich beschrieben werden. Die Applikation selbst beinhaltet die Programmlogik und kann mit Hilfe eines PDA-Clients gesteuert werden. Die Kommunikation der beiden Komponenten erfolgt über ein spezielles Interface, mit dem die Benutzereingaben an die Anwendung übertragen werden können.

Um die Galleria Borghese zu begehen, können über den PDA (Personal Digital Assistant) selbst erstellte Pfade oder vorgegebene Touren ausgewählt werden, die im folgenden als Wege zusammengefasst werden. Nach der Auswahl eines solchen Weges geht der PDA-Client in den Play-Modus über. In diesem Modus steht dem Benutzer eine eigens für diesen Modus angepasste Bedienoberfläche zur Verfügung, mit der er die Animation steuern kann. Im Vordergrund steht hier der Play/Pause Button zum Fortführen oder Anhalten der Tour. Für eine realistischere Navigation kann der Benutzer die Geschwindigkeit einer Tour mit Hilfe des PDAs steuern. Je weiter der Benutzer den PDA vom Körper entfernt hält, desto schneller wird der Weg abgelaufen. Hält er den PDA ganz nah am Körper, wird die Animation angehalten.

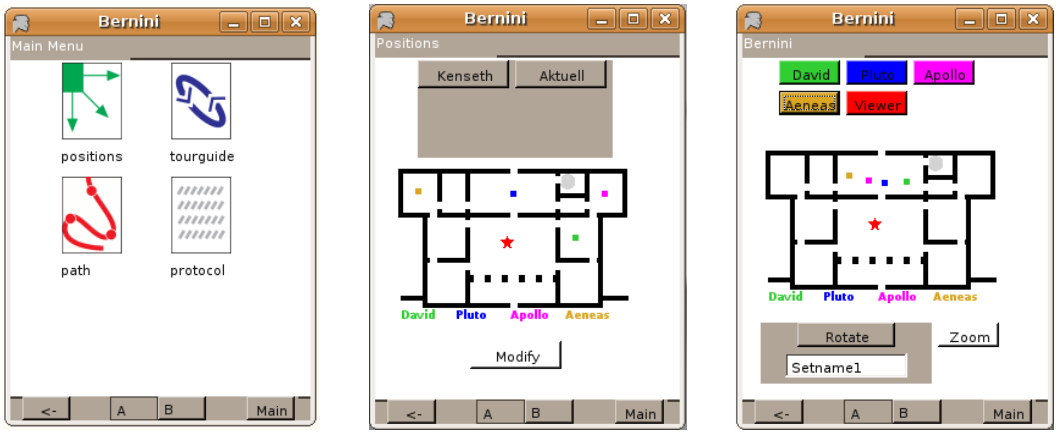
Um kollaboratives Arbeiten zu ermöglichen, wird die Anwendung im TwoView visualisiert. Die Steuerung der Applikation wird hierbei vom Anwender, der den PDA in der Hand hält, übernommen, während die zweite Person nicht in das Geschehen

eingreifen kann. Bei der Begehung der Galleria Borghese beschreiten beide Benutzer immer denselben Weg und befinden sich an derselben Position dieses Weges.

Die Steuerung der Applikation lässt sich in drei Kategorien einteilen, die auch als eigenständige Menüpunkte im Hauptmenü des PDA-Clients auftauchen: Positions, Tourguide und Path (siehe Abbildung 3.4(a)). Sowohl in dem Menüpunkt Tourguide als auch Path kann ein Weg zum Erkunden der virtuellen Galleria Borghese gestartet werden.

Menüpunkt Positions

Mit dem Menüpunkt *Positions* kann eine Aufstellung (Setting) der Skulpturen innerhalb der Galleria Borghese gewählt oder erstellt werden. Dazu werden unter diesem Menüpunkt vorgegebene Aufstellungsvarianten angeboten sowie eine Möglichkeit bereit gestellt, eine eigene Aufstellung der Skulpturen zu definieren. Um ein benutzerdefiniertes Setting zu erzeugen wird auf dem PDA-Client ein Grundriss der Galleria Borghese angezeigt, auf dem die farblich unterschiedlich gekennzeichneten Skulpturen platziert werden können (siehe Abbildung 3.4(b) und 3.4(c)).



(a) Hauptmenü

(b) *Positions:* Auswahl einer vorgegebenen Aufstellung

(c) *Positions:* Erstellung einer eigenen Aufstellung

Abbildung 3.4: *Hauptmenü* und *Positions* des PDA-Clients

Menüpunkt Tourguide
Der Tourguide bietet eine Auswahl vorgegebener Touren an, die in drei Kategorien eingeteilt sind:

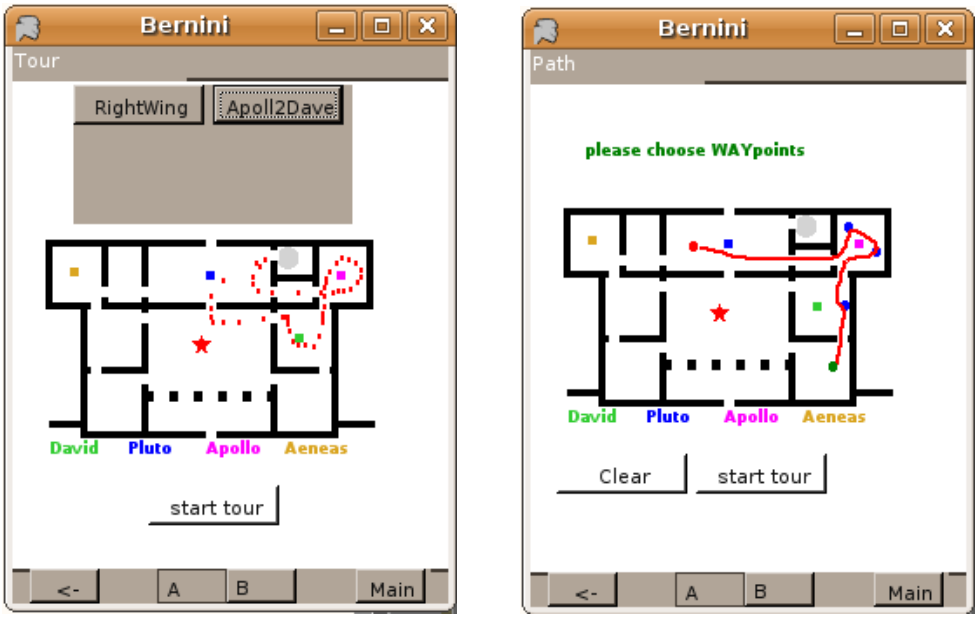
Sculptures
Hier werden Rundkurse um einzelne Skulpturen angeboten, die sich in der Galleria befinden.

Rooms
Die Kategorie *Rooms* bietet Touren durch einzelne Räume an, wobei die Untersuchung bestimmter Räume im Vordergrund steht.

Building
Die Kategorie *Building* beinhaltet Touren, die durch mehrere Räume der Galleria führen.

Für jede Kategorie werden mehrere Touren angeboten, die nach der Auswahl im Grundriss der Galleria auf dem PDA zur Vorschau angezeigt werden (siehe Abbildung 3.5(a)). Nach der Wahl einer Tour muss diese noch explizit gestartet werden, wodurch der PDA-Client in den Play-Modus übergeht.

Menüpunkt Path
Pfade bezeichnen Wege, die von dem Benutzer mit dem PDA-Client erzeugt wurden. Im diesem Menüpunkt kann der Anwender eigene Wege durch die Galleria über ein „2D-World in Miniature-Verfahren“ planen (siehe Abbildung 3.5(b)). Der PDA stellt hierfür einen Grundriss der Galleria Borghese zur Verfügung, in der die farblich gekennzeichneten Skulpturen eingezeichnet sind. Der Anwender kann nun einen Weg erstellen, indem er zunächst einen Start- und einen Endpunkt des Weges festlegt. Diese beiden Punkte werden über die kürzeste Entfernung miteinander verbunden. Anschließend können weitere Wegpunkte hinzugefügt werden, die ebenfalls über die kürzeste Entfernung miteinander verbunden werden. Nach dem Starten des Pfades geht die Applikation in den Play-Modus über. Beim Ablaufen des Pfades wird die Blickrichtung des Betrachters abhängig von den Positionen der Skulpturen berechnet. Sobald ein Objekt einen bestimmten Abstand zum Betrachter unterschritten hat, dreht sich die Kamera auf das Objekt zu.



(a) *Tourguide* (b) *Path* mit selbsterstelltem Pfad

Abbildung 3.5: Menü: *Tourguide* und *Path*

4 Vorbereitende Arbeiten

4.1 Portierung der Applikation nach AVANGO NG

Die Ergebnisse dieser Arbeit bauen auf der Applikation der Diplomarbeit *„Kollaborative Interaktion in virtuellen Umgebungen im kunsthistorischen Kontext“* auf. Die Umsetzung der *Darstellung mehrerer Ansichten* sowie der *Pfadkorrektur* erfolgt mit Python, basierend auf AVANGO NG. Da die Ergebnisse in die bereits bestehende Anwendung integriert werden sollten, musste diese Applikation zunächst auf die neue Version von AVANGO umgestellt werden. Vor allem die Verwendung des neuen Szenegraphen *OpenSceneGraph* und die Umstellung des Scripting-Interface von *Scheme* auf *Python* machten diese Portierung notwendig. Durch diese Veränderungen konnten zudem die neuen Konzepte von AVANGO NG, wie die Verwendung von *Containern* und ein stärkeres komponentenbasiertes Design, in der Applikation eingesetzt werden. Die Struktur des PDA-Clients hingegen ist weitestgehend unverändert geblieben.

4.1.1 Programmstruktur

Die Applikation erlaubt die Begehung der Galleria Borghese anhand von Wegen. Für die visuelle Darstellung wird deshalb in jedem Frame der aktuelle Status (*state*) einer Ansicht (*View*) berechnet. Die dargestellte Szene (*Scene*) besteht aus der Galleria Borghese sowie den dort positionierten Skulpturen. Um den Status einer Ansicht zu berechnen, werden anhand eines begangenen Weges, der Szene und dem Fortschritt der Tour die Positionen der Benutzer in der virtuellen Umgebung ermittelt. Anschließend kann die Szene für die Benutzer aus diesen Positionen dargestellt werden. Der Weg und die Aufstellung der Skulpturen können hierbei vom Anwender über den PDA-Client gewählt werden.

Ein Weg wird durch eine Menge von Wegpunkten (Viewerpositionen) definiert, die in einer Liste verwaltet werden. Ein Element dieser Liste repräsentiert einen bestimmten Wegpunkt und gibt die Referenzposition und Blickrichtung beider Benutzer zu einem bestimmten Zeitpunkt an. Beide Benutzer befinden sich an derselben Stelle desselben Weges. Die virtuelle Position der Benutzer berechnet sich aus dem Wegpunkt, an dem sie sich befinden, und ihrer realen Position im TwoView, die auf diese Referenzposition übertragen wird.

Die vorgegebenen Touren liegen als „Mov-Files“ vor und werden bei Bedarf vom PDA-Client eingelesen. Bei der Erstellung eines benutzerdefinierten Pfades müssen die Wegpunkte vom PDA selbst berechnet werden. Die Liste der Wegpunkte wird anschließend zusammen mit der Aufstellung der Skulpturen an die Applikation übertragen. Hierzu dient der *InputNode* als Schnittstelle für den Datenaustausch zwischen beiden Programmen. Der *InputNode* beschreibt somit alle Eingabewerte an die Applikation.

Anhand eines Weges, einer Szene und des Fortschritts der Tour kann die Viewerposition ermittelt werden. Hierzu wird zunächst die Geschwindigkeit der Tour durch die Klasse *SpeedControl* ermittelt. Die Geschwindigkeit wird anhand des Abstands vom PDA zum Betrachter berechnet. Mit dem aktuellen Wegpunkt und der Geschwindigkeit wird die nächste Viewerposition durch die Klasse *AnimationControl* ermittelt. Da ein Wegpunkt aber nur die Position für einen bestimmten Zeitpunkt beschreibt, kann der gesuchte Wert zwischen zwei Wegpunkten liegen. Deshalb muss die wirkliche Viewerposition durch die Klasse *MatrixInterpolator* aus den umgebenden Wegpunkten interpoliert werden. Die berechneten Positionen besitzen häufig noch ungewollte Transformationen, die zuletzt noch von den Klassen *FixRotation* und *GroundFollowing* korrigiert werden. Für die Bestimmung der Kameraposition ist die Höhe eines Wegpunktes irrelevant, und die Blickrichtung wird allein durch die Rotation um die vertikale Achse bestimmt.

Die beschriebenen Komponenten zum Ermitteln der Viewerposition bilden einen abgeschlossenen Subgraphen und können deshalb in einem *Container* zusammengefasst werden. Beginnend mit der Auswertung der Geschwindigkeit wird eine Evaluationskette ausgelöst, deren Komponenten unabhängig von äußeren Einflüssen sind. Zur besseren Handhabung wurden sie deshalb als *Mover* zusammengefasst.

Die Programmstruktur, die sich aus der Portierung auf AVANGO NG ergeben hat, wird durch das Diagramm 4.1 dargestellt.

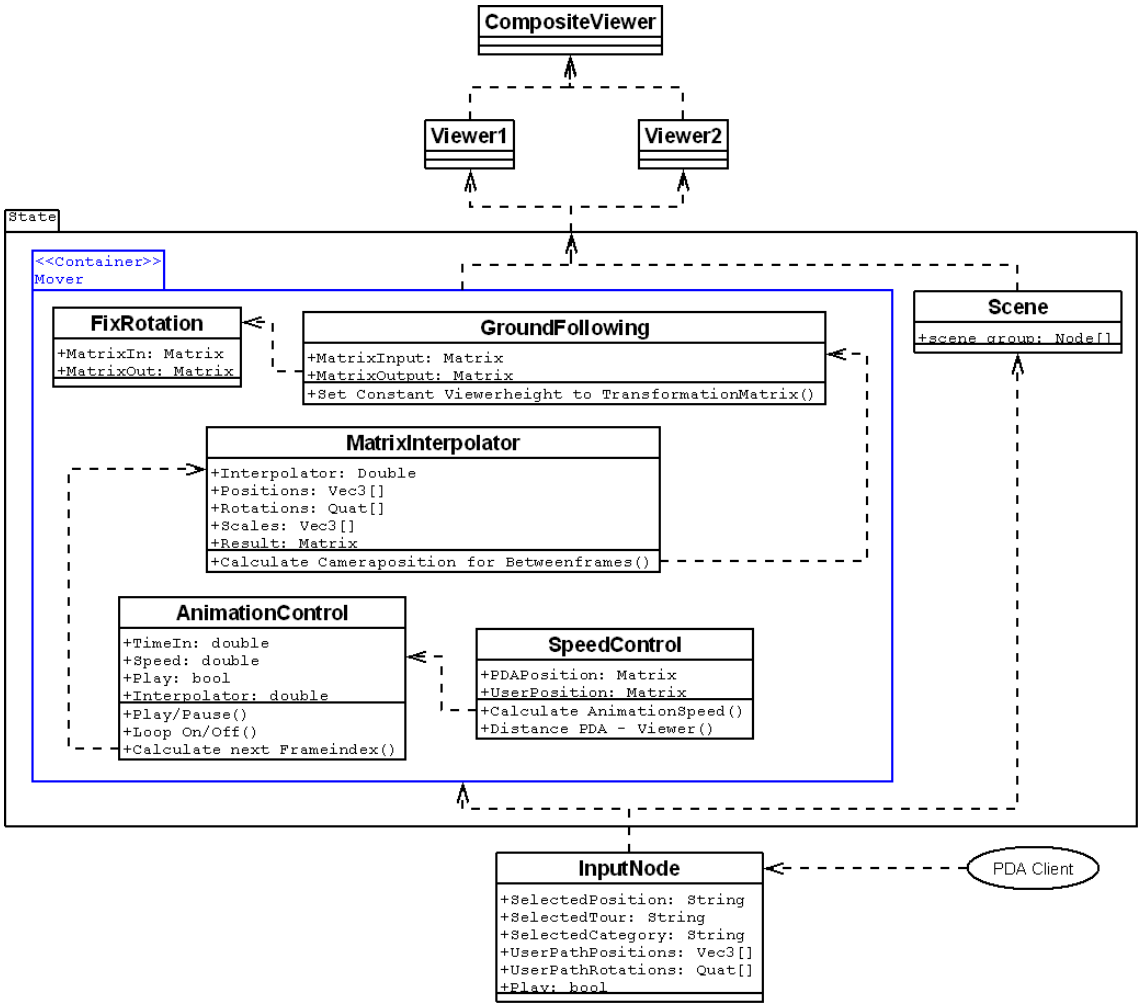


Abbildung 4.1: Applikationsstruktur nach der Portierung auf AVANGO NG. Der *Mover* ist blau markiert

4.1.2 Viewing Setup

Das *Viewing Setup* beschreibt die Struktur der Daten für die Darstellung einer Ansicht (*View*), die in Abbildung 4.2 dargestellt ist. Der *Composite Viewer* vereinigt mehrere *Views*, die wiederum aus zwei Unterkomponenten bestehen, einer Szene und einer Kamera.

Die Szene wird durch eine hierarchische Anordnung von Geometrie- und Transformationsknoten des Szenegraphen beschrieben. Sie beinhaltet die Räume der Galleria Borghese sowie die Skulpturen und deren Positionen. Ein Geometrieknoten beschreibt das Aussehen eines Objekts, welches mit einem Transformationsknoten in der virtuellen Umgebung positioniert werden kann.

Die Kamera repräsentiert den Betrachter in der virtuellen Umgebung. Die wichtigsten Attribute der Kamera sind *ViewerTransform*, *EyeTransform* und *ScreenTransform*. Das Attribut *ViewerTransform* beschreibt die interpolierte Position, die vom *Mover* berechnet wurde. Sie repräsentiert somit einen Punkt auf einem Weg, der für beide Anwender identisch ist. Die Benutzer können durch die frei begehbare Fläche des TwoView (*Moving Area*) ihre virtuelle Position beeinflussen. Die getrackte Position der Betrachter wird in der Matrix *EyeTransform* festgehalten. Um die virtuelle Position eines Betrachters (Kameraposition) zu berechnen, wird die Matrix *ViewerTransform* mit der Matrix *EyeTransform* verknüpft. Das Attribut *ScreenTransform* stellt die Position und Ausrichtung der Projektionsfläche bezogen auf die Viewerposition dar.

Die Projektionsleinwand (*Screen*) kann als Fenster zur virtuellen Umgebung angesehen werden, durch das die Kamera die virtuelle Szene aufnimmt. Dieses Fenster legt somit den Sichtkegel (*View frustum*) des Betrachters fest. Anhand der Maße der Leinwand (Höhe und Breite) und dem Abstand zum Betrachter lässt sich der Sichtkegel der virtuellen Kamera berechnen. Durch verschiedene Positionen eines Betrachters im TwoView ergeben sich unterschiedliche Sichtkegel. Je näher eine Person vor der Leinwand steht, desto größer wird der Öffnungswinkel seines Sichtkegels.

Das Bild, welches dem Betrachter angezeigt wird, wird in einem Ausgabefenster (*Window*) ausgegeben. Der Viewport beschreibt hierbei den Bereich des Ausgabefensters, in dem das Bild dargestellt werden soll.

Eine *View* definiert eine virtuelle Betrachterposition und einen Sichtkegel. Für eine perspektivisch korrekte Darstellung der Umgebung für zwei Betrachter werden somit zwei *Views* benötigt, die durch den *Composite Viewer* dargestellt werden können.

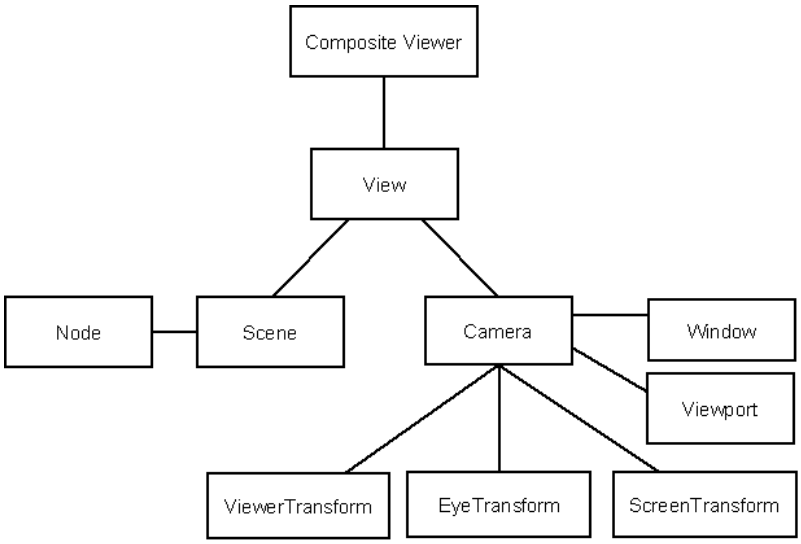


Abbildung 4.2: Viewing Setup einer Ansicht

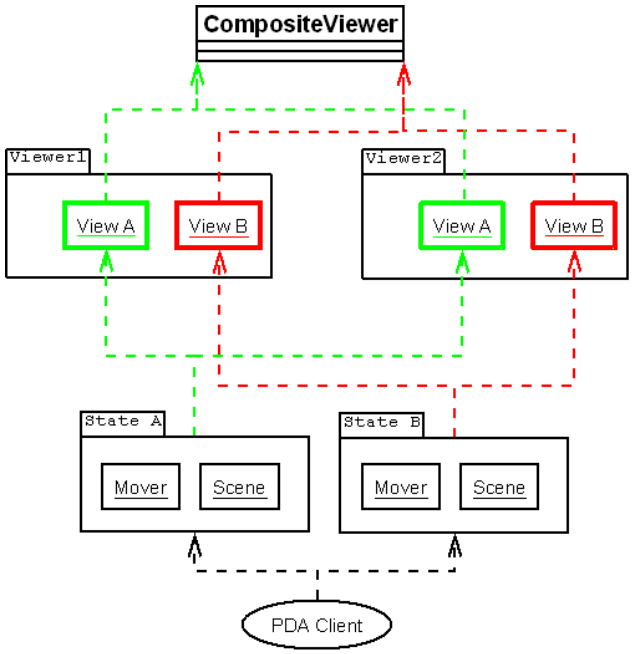
4.2 Generierung mehrerer Ansichten

In der bisherigen Applikation wurde eine Ansicht pro Person dargestellt. Damit jedoch eine gleichzeitige Darstellung verschiedener Ansichten erfolgen kann, wurde die Applikation zunächst um einen zweiten *View* pro Person erweitert (siehe Abbildung 4.3(a)).

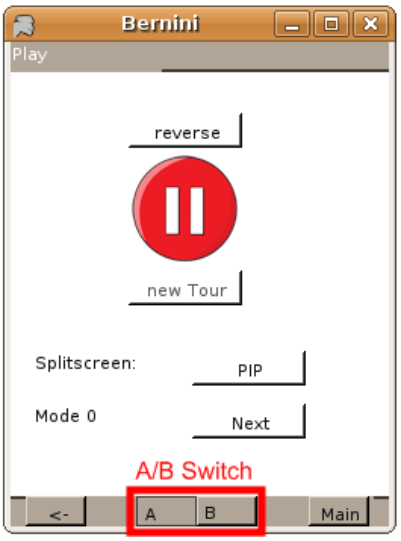
Eine *View* setzt sich im Wesentlichen aus einem *Mover* und einer Szene zusammen, woraus der aktuelle Status (*state*) der Ansicht berechnet wird. Da die Ansichten getrennt voneinander existieren sollen, werden zwei *Views* benötigt. Da eine *View* über den PDA-Client manipuliert werden kann, wurden dort alle Informationen, die an die Applikation übertragen werden, dupliziert. Dies sind zum Beispiel der gewählte Weg zur Begehung der Galleria Borghese oder die Aufstellung der Skulpturen. Diese doppelten Werte wurden auf dem PDA-Client in zwei Ansichten (*ViewA* und *ViewB*) eingeteilt, um eine eindeutige Trennung der Attribute zu erreichen.

Um zwischen verschiedenen Ansichten umschalten zu können, wurde eine Schaltfläche in den PDA integriert, die die aktuell ausgewählte Ansicht kennzeichnet (siehe Abbildung 4.3(b)). Von der Applikation werden nur die Attribute der aktiven Ansicht ausgewertet und dargestellt. Durch die Dopplung der Daten auf Seiten des PDA-Clients findet eine klare Trennung der beiden Ansichten statt. Der Status wird für beide Ansichten separat festgehalten und kann deshalb nach einem Wechsel wieder korrekt rekonstruiert werden.

Beide Ansichten werden beim Start der Anwendung instantiiert, von denen aber nur eine im Vollbildmodus angezeigt wird. Bei einem Wechsel der Ansicht wird die aktivierte Ansicht in den Vordergrund geholt, während die andere im Hintergrund erhalten bleibt. Um den Wechsel der dargestellten Ansicht möglichst performant zu gestalten, wird hierzu lediglich der Viewport der Ansichten angepasst.



(a) Erweiterung der Applikationsstruktur um eine zweite Ansicht pro Person



(b) PDA-Client mit A/B Switch zum Wechseln der aktiven Ansicht

Abbildung 4.3: Einbindung mehrerer Ansichten in die vorhandene Applikationsstruktur

4.3 Preview

Für die Umsetzung der *Pfadkorrektur* und die *Darstellung mehrerer Ansichten* müssen die Kameras beider Benutzer manipuliert werden. Da die resultierenden Bilder für die Kontrolle der Ergebnisse nur in geringem Maße hilfreich sind, wurde eine Vorschau (*Preview*) in die Anwendung integriert. Diese kann bei Bedarf aktiviert werden und soll während der Implementierung als Hilfmittel eingesetzt werden, um die Ergebnisse dieser Arbeit visuell zu verifizieren. Zusätzlich können den Betrachtern anhand der *Preview* bestimmte Sachverhalte, bezogen auf die Manipulation der Kameras, verständlicher dargestellt werden.

Die *Preview* (siehe Abbildung 4.4) zeigt die Position der virtuellen Kamera in der Galleria Borghese, den Sichtkegel der Kamera sowie den Bereich, in dem sich der Betrachter frei bewegen kann (*Moving Area*). Die *Moving Area* ist an die Viewerposition einer Ansicht gekoppelt. Die Viewerposition dient als Referenzpunkt für beide Benutzer, durch den sich deren virtuelle Position bestimmen lässt. Die getrackte Position beider Benutzer im TwoView wird auf die Viewerposition übertragen, um so die Kameraposition der Betrachter zu ermitteln.

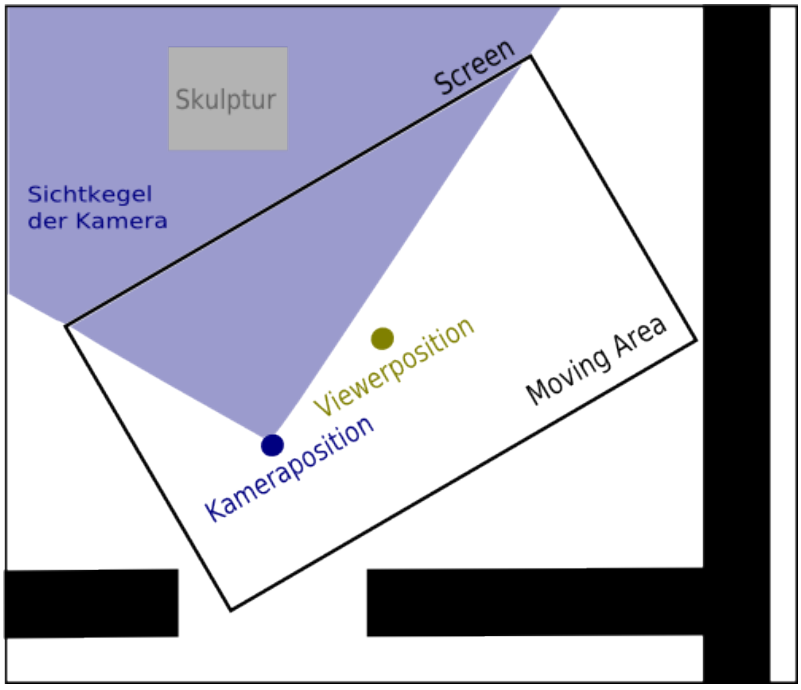


Abbildung 4.4: Skizze der *Preview* mit den dargestellten Komponenten

Der Sichtkegel stellt den sichtbaren Bereich der Kamera dar. Die Größe des *View frustums* richtet sich nach der Betrachterposition und der Größe der Projektionsleinwand, die sich an der vorderen Kante der *Moving Area* befindet. Für die bessere Orientierung und zur Kontrolle der virtuellen Position werden die Skulpturen abstrakt in der *Preview* dargestellt.

Damit der Status einer Ansicht fehlerfrei von der *Preview* wiedergegeben wird, werden lediglich die von der Applikation berechneten Ergebnisse dargestellt. Durch die Visualisierung der zusätzlichen Informationen, wie dem Sichtkegel und der Kameraposition in der virtuellen Umgebung, kann das Ergebnis der Berechnungen besser kontrolliert und verifiziert werden. Die angezeigte Kameraposition wird zudem immer an die getrackte Position des Betrachters angepasst.

Die *Preview* hat keinen Einfluss auf die eigentliche Anwendung und kann als separates Modul angesehen werden. Um möglichst wenig von dem eigentlichen Bild einer Ansicht zu verdecken, wird die Vorschau in einem kleinen Fenster im unteren Teil der Projektionsfläche angezeigt (siehe Abbildung 4.5).

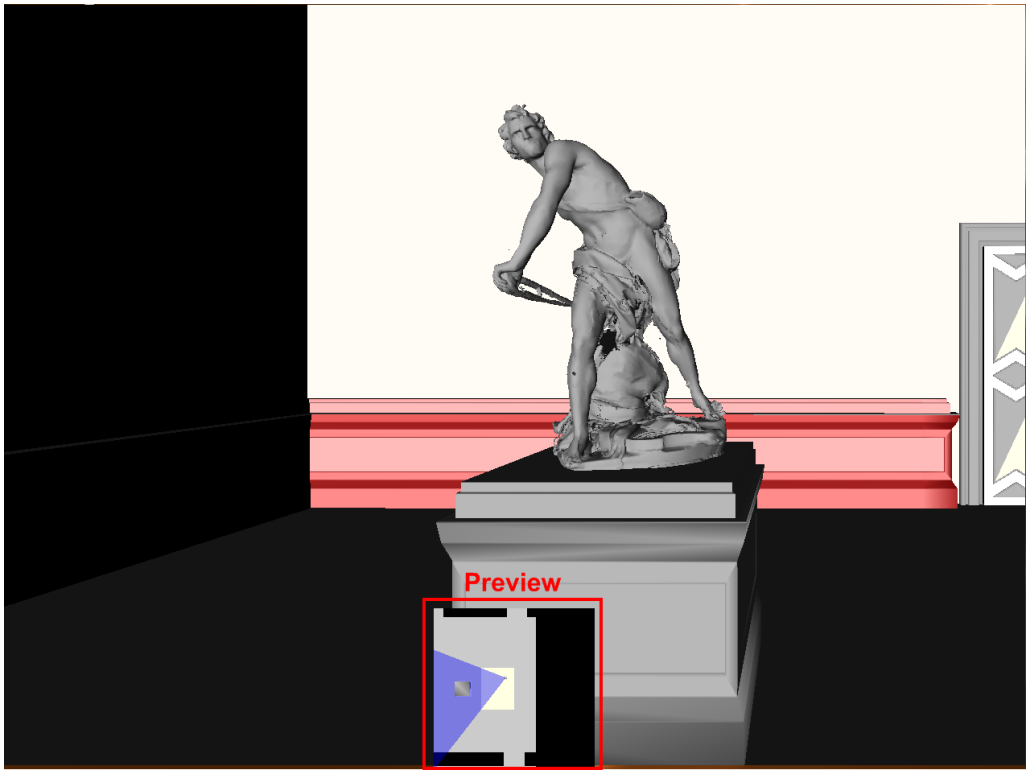


Abbildung 4.5: Screenshot mit aktivierter *Preview*

5 Darstellung mehrerer Ansichten

5.1 Motivation

Um exaktere Ergebnisse bei der Analyse von Objekten zu erhalten, ist es häufig notwendig, sie aus verschiedenen Blickwinkeln zu untersuchen. Kunstwissenschaftler weltweit setzen aus diesem Grund zwei Diaprojektoren ein, die unterschiedliche Bilder von Objekten nebeneinander projizieren. Für das Projekt *Virtualisierung von Skulptur* ist deshalb eine Übertragung dieser Vorgehensweise in eine virtuelle Umgebung wünschenswert. Durch die Einbindung dieser Funktionalität in die vorhandene Applikation stehen neue Wege für die Untersuchung der Galleria Borghese zur Verfügung. Bezogen auf das Projekt *Virtualisierung von Skulptur* des Forschungskollegs „Medienumbrüche“ (FK 615), werden durch die gleichzeitige Darstellung mehrerer Ansichten die Möglichkeiten bereitgestellt, ein Objekt aus verschiedenen Blickwinkeln zu betrachten, die Wirkung einer Skulptur an unterschiedlichen Standorten zu untersuchen oder den Standort einer Skulptur zu verifizieren, indem verschiedene Skulpturen an derselben Stelle positioniert werden.

In diesem Kapitel sollen deshalb verschiedene Methoden für die Visualisierung zweier Ansichten vorgestellt werden. Nach der Umsetzung werden die Methoden durch einen Usertest evaluiert, um ihre Praxistauglichkeit zu überprüfen. Durch die gleichzeitige Darstellung zweier Ansichten wird der Ansatz aus dem kunsthistorischen Bereich erweitert. Denn neben dem Vergleich zweier statischer Bilder können beispielsweise auch unterschiedliche Touren und Pfade miteinander verglichen werden.

5.2 Anforderungen

Auf Grundlage des Projekts *Virtualisierung von Skulptur* ließen sich die folgenden Anforderungen herleiten, um die korrekte und vollständige Umsetzung der *Darstellung mehrerer Ansichten* zu verifizieren.

Wiederherstellen einer Ansicht
Durch das Wechseln der Ansicht gerät die zuvor angezeigte Ansicht in einen passiven Status. Um sie reaktivieren zu können, soll der Status dieser Ansicht wieder hergestellt werden können, damit nach dem Wechsel der ursprüngliche Zustand wiederhergestelllt ist.

Trennung der Ansichten
Alle verfügbaren Ansichten sollen eigenständig sein und den vollen Funktionsumfang einer einzelnen Ansicht bieten. Durch diese Unabhängigkeit soll die größtmögliche Flexibilität und Anwendungsvielfalt erreicht werden. Die Trennung der Ansichten gewährleistet, dass jede Ansicht einen unterschiedlichen Weg durch die Galleria und eine eigene Aufstellung der Skulpturen aufweisen kann. Weiterhin wird durch diese Anforderung garantiert, dass alle Ansichten gleichwertig sind und identische Funktionalitäten besitzen.

Beibehalten der Funktionalität
Die bisher verfügbaren Funktionen für die Steuerung einer Ansicht sollen beibehalten werden. Dies beinhaltet die Möglichkeiten zum Wählen einer Tour, eines Pfades und einer Aufstellung der Skulpturen. Diese Funktionen sollen weder verändert noch neue Funktionalitäten hinzugefügt werden.

Verzerrungsfreie Darstellung
Bei der gleichzeitigen Darstellung mehrerer Ansichten muss die Projektionsfläche aufgeteilt werden. Für jede Ansicht steht somit nur noch ein Teil der ursprünglichen Darstellungsfläche zur Verfügung. Durch die Abbildung eines Bildes auf eine kleinere Projektionsfläche können Verzerrungen, wie beispielsweise Interpolationsartefakte, innerhalb des Bildes auftreten, die die Analyse der dargestellten Objekte verfälschen können.

Einheitliche Steuerung
Die Steuerung der einzelnen Ansichten soll separat und einheitlich erfolgen. Da die Steuerung der Applikation bereits vorgegeben ist, soll diese auch für alle Ansichten verwendet werden. Die Bedienelemente für die Verwaltung, wie zum Beispiel das Wählen der Methode zur Darstellung mehrerer Ansichten, sollen nahtlos in das bestehende Interface integriert werden.

Gleichzeitige Betrachtung
Es soll möglich sein, die verschiedenen Ansichten direkt miteinander vergleichen zu können. Dazu müssen sie gleichzeitig auf derselben Projektionsfläche dargestellt werden können.

Anzahl der Ansichten
Um den Benutzer nicht zu verwirren, wird die Anzahl der Ansichten auf zwei begrenzt. Werden mehr als zwei Ansichten angeboten, wäre es schwer, sich als Betrachter jeden Zustand zu merken und gezielt eine bestimmte Ansicht zu reaktivieren. Für die gleichzeitige Darstellung mehrerer Ansichten bekäme jede einen exklusiven Teil der Projektionsfläche zugewiesen. Durch die Aufteilung der Projektionsfläche in viele kleine Bereiche stünde jeder Ansicht anschließend nur noch ein kleiner Teil der Projektionsfläche zur Verfügung. Dadurch könnte der Betrachter die aktive Ansicht schnell aus den Augen verlieren.

Kollaboratives Arbeiten
Durch die Ausführung der Applikation im TwoView wird kollaboratives Arbeiten ermöglicht. Die Darstellung beider Ansichten soll hierauf möglichst keinen Einfluss nehmen.

5.3 Switchen

Das *Switchen* ist im Wesentlichen nur ein Resultat der *Generierung mehrerer Ansichten* aus Abschnitt 4.2. Bei diesem Vorgehen werden zwei unterschiedliche Ansichten definiert, die im schnellen Wechsel in den Vordergrund geholt werden können. Es werden keine zusätzlichen Zwischenschritte benötigt, um die Kamera vor einem Objekt neu zu positionieren. Eine gleichzeitige Darstellung zweier Ansichten ist nicht vorgesehen, weshalb dieses Verfahren für einen direkten Vergleich zweier Bilder ungeeignet ist. Geht es aber darum, zwei Ansichten unabhängig voneinander zu untersuchen, kann dieses Vorgehen den Anwender unterstützen, da keine zusätzlichen Zwischenschritte beim Wechsel der Ansichten erforderlich sind. Durch die getrennte Darstellung beider Ansichten treten keine Verzerrungen oder Verdeckungen auf.

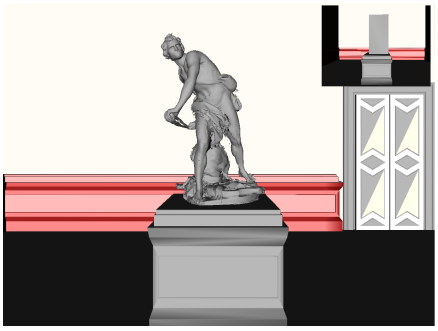
5.4 Picture in Picture

Die *Picture in Picture* Methode ist ein Verfahren, bei dem zwei Ansichten gleichzeitig dargestellt werden können. Hierbei wird die zweite Ansicht vollständig in einem kleinen Fenster innerhalb der Hauptansicht dargestellt. Dieses Fenster ist fest an der Projektionsfläche verankert und verdeckt dadurch einen Teil der Hauptansicht. Da die Hauptansicht möglichst wenig beeinflusst werden soll, wurde ein Kompromiss für die Größe dieses Fensters gefunden. Je größer das Bild für die zweite Ansicht dargestellt wird, desto mehr Details können in diesem Bild identifiziert werden, und desto stärker fällt die Verdeckung der Hauptansicht aus. Das kleinere Fenster wird am rechten oberen Rand der Projektionsfläche platziert. Bei einem Wechsel der Ansicht wird das Bild aus dem kleineren Fenster in das Hauptfenster verschoben, und die andere Ansicht wird auf der kleineren Fläche dargestellt (siehe Abbildung 5.1(a) und 5.1(b)).

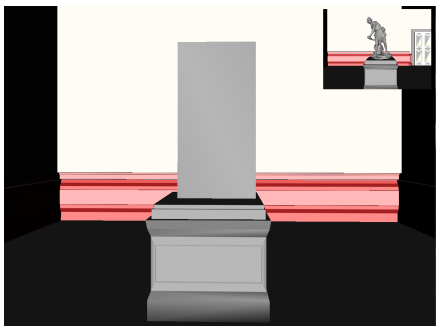
Um das gesamte Bild einer Ansicht auf einer kleineren Bildfläche unterzubringen, muss der *Screen* dieser Ansicht manipuliert werden. Der *Screen* repräsentiert ein virtuelles Fenster, durch das die Kamera die virtuelle Umgebung aufnimmt. Eine Manipulation des *Screens* verändert die Brennweite der virtuellen Kamera, wodurch bei der stereoskopischen Darstellung leichte perspektivische Verzerrungen auftreten können.

Wie in Abbildung 5.1(c) dargestellt, kann durch die Änderung des Viewports der Sichtkegel auf einen kleineren Bereich des *Screens* zugeschnitten werden. Dadurch sieht der Betrachter nur noch einen Teil des eigentlichen Bildes (blaue Fläche). Um nun das gesamte Bild einer Ansicht auf der kleineren Bildfläche darstellen zu können, muss der *Screen* näher zum Betrachter bewegt werden (rote Fläche). Durch den geringeren Abstand der Kamera zum *Screen* wird der Öffnungswinkel des Sichtkegels vergrößert und der ursprüngliche Sichtkegel wieder hergestellt.

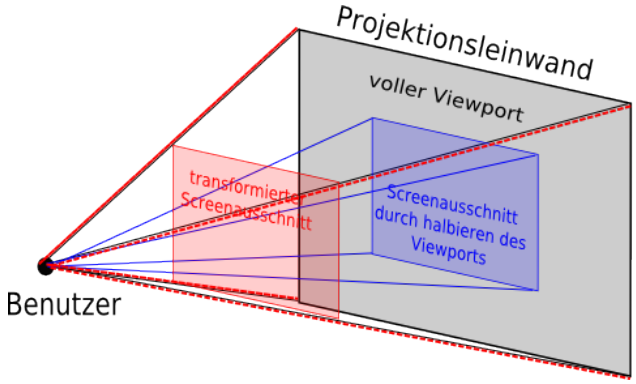
Zwar bietet der *Picture in Picture*-Modus die Möglichkeit, zwei Ansichten gleichzeitig darzustellen, aufgrund der geringen Größe der zweiten Ansicht ist sie für einen direkten Vergleich aber kaum geeignet.



(a) Ansicht A im Hauptfenster, Ansicht B im Nebenfenster



(b) Ansicht B im Hauptfenster, Ansicht A im Nebenfenster



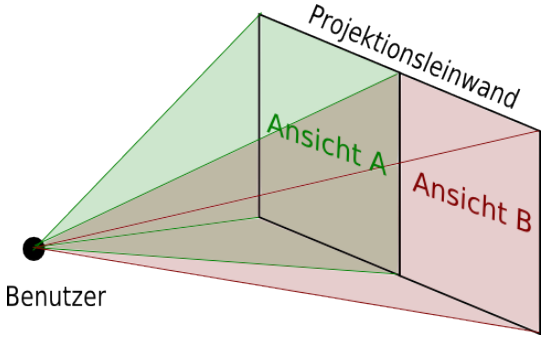
(c) Screentransformation für den *Picture in Picture*-Modus

Abbildung 5.1: *Picture in Picture*-Modus

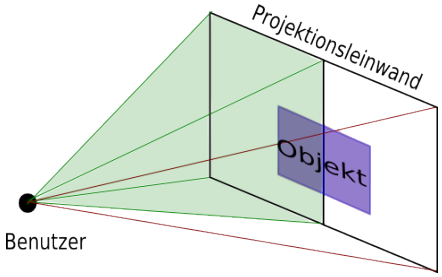
5.5 Splitscreen

Um eine gleichwertige Darstellung zweier Ansichten zu erreichen, wurde die *Splitscreen*-Methode umgesetzt. Durch das Aktivieren des *Splitscreens* wird das Hauptfenster vertikal in der Mitte geteilt (siehe Abbildung 5.2(a)). Beide Ansichten werden dabei in einer eigenen Hälfte der Projektionsleinwand dargestellt, Ansicht A in der linken und Ansicht B in der rechten.

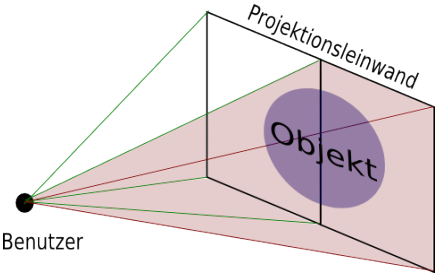
Für die Darstellung der Bilder auf einer kleineren Bildfläche wird auf eine Manipulation des *Screens* wie bei der *Picture in Picture*-Modus verzichtet. Da durch die Teilung aber nur noch ein Teil der ursprünglichen Darstellungsfläche pro Ansicht zur Verfügung steht, werden die Sichtkegel beider Ansichten (die Gesamtfrusta) auf die entsprechenden Bildhälften verkleinert. Dadurch können betrachtete Objekte, wie in Abbildung 5.2(b) und 5.2(c) dargestellt, abgeschnitten werden.



(a) Vertikale Teilung des Sichtkegels

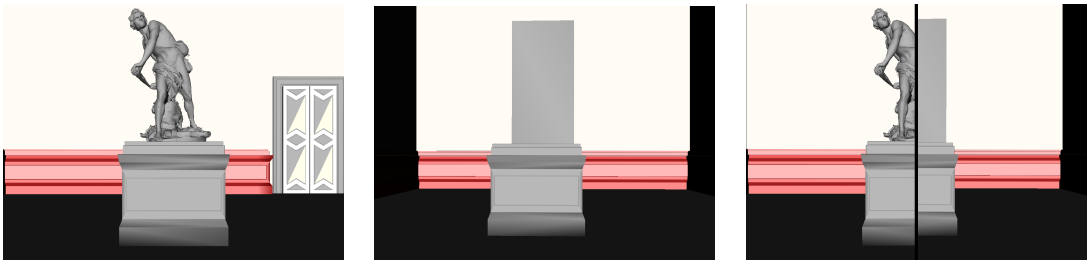


(b) Ansicht A wurde auf die linke Bildhälfte zugeschnitten, das betrachtete Objekt wird dabei abgeschnitten



(c) Ansicht B wurde auf die rechte Bildhälfte zugeschnitten, das betrachtete Objekt wird dabei abgeschnitten

Abbildung 5.2: Teilung des Sichtkegels in eine linke Hälfte (grün) und eine rechte Hälfte (rot)



(a) Ansicht A im Vollbildmodus

(b) Ansicht B im Vollbildmodus

(c) Resultat nach Aktivieren des *Splitscreens*

Abbildung 5.3: Screenshots des *Splitscreens* ohne Transformation der Sichtkegel. Die dargestellten Objekte können durch das Verkleinern des Sichtkegels abgeschnitten werden

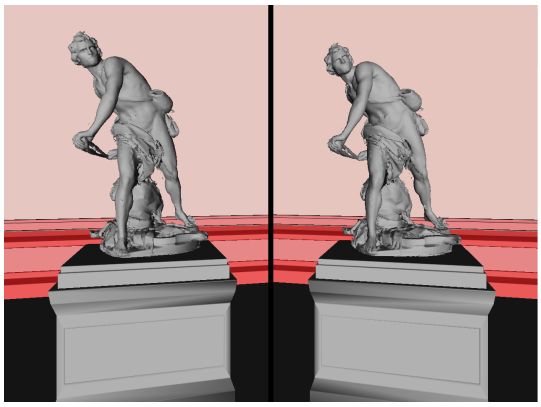
Da bei der Teilung des Gesamtfrustums die Sichtkegel auf einen kleineren Bildbereich zurecht geschnitten werden, können zuvor sichtbare Objekte aus dem Sichtbereich verschwinden oder abgeschnitten werden. In der Regel werden unterschiedliche Ansichten betrachtet, wodurch die abgeschnittenen Objekte nicht mehr exakt analysiert werden können (siehe Abbildung 5.3).

Wie in Abbildung 5.2 dargestellt, kann durch das Zuschneiden der Sichtkegel auf eine Bildhälfte das zuvor betrachtete Objekt aus dem sichtbaren Bereich verschwinden oder abgeschnitten werden. Die resultierenden Bilder sind somit für eine exakte Analyse der Objekte unbrauchbar. Deshalb müssen die zuvor fokussierten Objekte in das Zentrum der linken und rechten Ansicht zurückgebracht werden. Um Verzerrungen der Bilder auszuschließen, wird deshalb lediglich die Kamera einer Ansicht transformiert. Zu diesem Zweck kann die Kamera und der damit verbundene halbierte Sichtkegel auf das betrachtete Objekt translatiert oder rotiert werden, wodurch das dargestellte Bild für beide Personen perspektivisch korrekt bleibt.

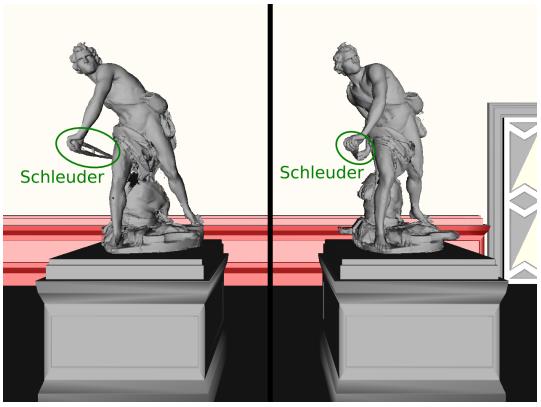
Die zwei verschiedenen Möglichkeiten, um den Sichtkegel auf das betrachtete Objekt auszurichten (Rotation und Translation), ergeben unterschiedliche Bilder (siehe Abbildung 5.4). Wird bei beiden Ansichten dasselbe Objekt aus derselben Perspektive betrachtet, kann man diese Unterschiede erkennen. Durch die Rotation erscheint das betrachtete Objekt auf beiden Seiten leicht eingedreht, während die Translation ein frontaleres, flacheres Bild liefert. Durch die Rotation des Sichtkegels ergeben sich auf der linken und rechten Bildhälfte sehr ähnliche Bilder, auf denen fast identische Details erkennbar sind. Die Translation hingegen liefert abweichende Bilder, wie man in Abbildung 5.4(c) an der Schleuder erkennen kann.

Durch die Teilung des Sichtkegels einer Ansicht (Gesamtfrustum) in eine linke und eine rechte Hälfte, besitzen sie eine andere Flucht als das Gesamtfrustum. Aus diesem Grund können die Bilder im *Splitscreen*-Modus nie exakt die gleiche Sicht auf das Objekt wie die Vollbildansicht liefern. Dennoch ist der *Splitscreen* für den direkten Vergleich zweier Bilder geeignet, da eine verzerrungsfreie und gleichwertige Darstellung beider Ansichten erfolgt.

(a) In Ansicht A und B wird dasselbe Objekt aus derselben Perspektive betrachtet



(b) *Splitscreen* mit Rotation der Kamera. Auf der linken und rechten Bildhälfte sind fast identische Objektmerkmale erkennbar



(c) *Splitscreen* mit Translation der Kamera. Auf der linken und rechten Bildhälfte sind unterschiedliche Objektmerkmale erkennbar

Abbildung 5.4: Screenshot einer Vollbildansicht und des *Splitscreens* mit unterschiedlichen Transformationen der Sichtkegel

Durch die Ausrichtung der Sichtkegel auf das fokussierte Objekt werden die Kameras beider Benutzer transformiert. Wird die Kamerabewegung in Abhängigkeit von der getrackten Position berechnet, ergeben sich unterschiedliche Bewegungen für beide Benutzer, da diese immer an unterschiedlichen Positionen im TwoView stehen.

Eine unterschiedliche Kameratransformation beider Benutzer kann den gemeinsamen Raum und das kollaborative Arbeiten beeinträchtigen. Einheitliche Transformationen hingegen haben keinen Einfluss auf den gemeinsamen Raum. Je weiter die Betrachter voneinander entfernt stehen, desto unterschiedlicher fallen diese Transformationen aus. Aus diesem Grund werden neben dynamischen Verfahren, die die getrackten Positionen der Betrachter berücksichtigen, auch statische Verfahren implementiert, die identische Kameratransformationen für beide Betrachter ausführen.

In Kapitel 7, *Empirischer Test zum gemeinsamen Raum*, wird ein Benutzertest zu diesem Effekt durchgeführt, der eine qualitative Bewertung der Beeinträchtigung des gemeinsamen Raums auf das kollaborative Arbeiten ermöglicht.

Nach dem Aktivieren des *Splitscreens* kann das betrachtete Objekt aus dem Sichtkegel gelangen oder abgeschnitten werden. Deshalb muss der halbierte Sichtkegel einer Ansicht auf dieses Objekt ausgerichtet werden. Hierzu kann der Sichtkegel rotiert oder translatiert werden. Um die Werte der Transformationen zu bestimmen, muss der Fokus der Gesamtansicht ermittelt werden und eine Annahme für die Blickrichtung der Betrachter getroffen werden. Mit der Kenntnis der Blickrichtung für einen Sichtkegel kann der ursprüngliche Fokus ermittelt, und der Sichtkegel auf diesen Bereich ausgerichtet werden. Für die Berechnung der Transformation wird davon ausgegangen, dass sich der Fokus des Betrachters auf dieser Blickgeraden befindet.

Es existieren zwei Annahmen, die die Blickrichtung (Sichtgerade) des Betrachters betreffen, die Bildmitte und die Winkelhalbierende, die in der Abbildung 5.5 dargestellt sind. Durch die Verwendung der Bildmitte wird angenommen, dass das betrachtete Objekt grundsätzlich in der Mitte der Projektionsfläche dargestellt wird. Deshalb kann das Zentrum der Projektionsleinwand als fokussierter Punkt angesehen werden. Eine alternative Annahme ist die Verwendung der Winkelhalbierenden eines Sichtkegels. Diese Annahme führt zu einer symmetrischen Teilung der Sichtkegel und passt sich an die Position des Betrachters vor der Projektionsfläche an. Beide Annahmen führen zu unterschiedlichen Sichtgeraden, auf Grund derer sich verschiedene Transformationen der Sichtkegel ergeben. Die resultierenden Bilder beider Annahmen können deshalb voneinander abweichen.

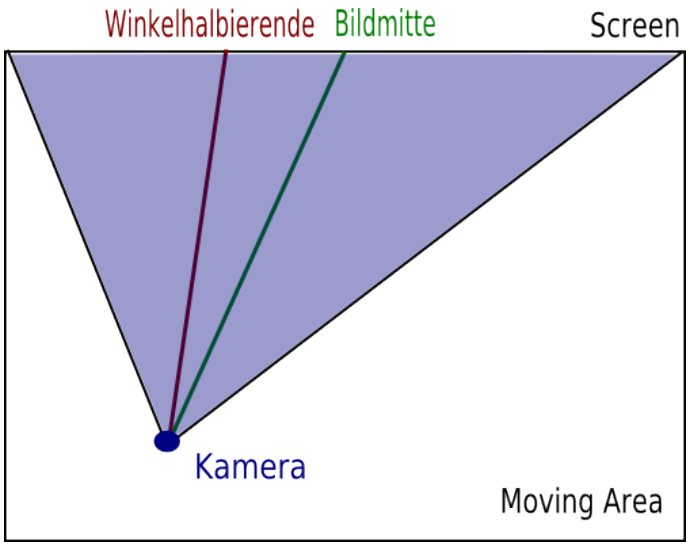


Abbildung 5.5: Unterschiedliche Annahmen für die Blickrichtung des Betrachters. Die Blaue Fläche repräsentiert den Sichtkegel

In den folgenden zwei Abschnitten werden die umgesetzten Verfahren den beiden Transformationen (Rotation und Translation) der Sichtkegel beschrieben. Bei den beschriebenen Verfahren wird angenommen, dass sich das betrachtete Objekt auf der Sichtgeraden des Gesamtfrustums befindet.

5.5.1 Rotation des Sichtkegels

Bei der Rotation des Sichtkegels wird die Kamera der Betrachter um die vertikale Achse gedreht, um den Sichtkegel auf das fokussierte Objekt einer Ansicht auszurichten. Mit der Annahme, dass sich das betrachtete Objekt in der Nähe der Blickrichtung des Gesamtfrustums befindet, werden die Sichtkegel der linken und der rechten Ansicht auf die Blickrichtung des Gesamtfrustums ausgerichtet, indem die Sichtgeraden übereinandergelegt werden. Dazu wird der Winkel zwischen den Sichtgeraden der halbierten Sichtkegel S und der Sichtgeraden des Gesamtfrustums berechnet und die Kamera anschließend um diesen Winkel gedreht.

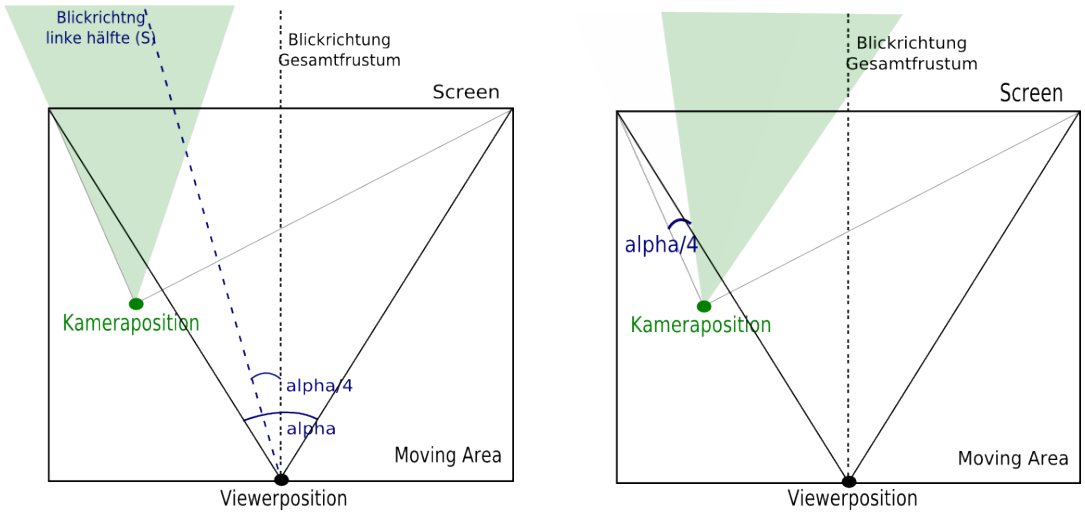
Für die Rotation wurden drei verschiedene Varianten umgesetzt. Es werden zwei dynamische Verfahren vorgestellt, die die getrackten Positionen der Benutzer für die Berechnung der Transformationen beider Personen berücksichtigen. Zusätzlich wird eine statische Variante beschrieben, die für beide Betrachter identische Rotationen, unabhängig von ihren Positionen, ausführt.

Offset Rotation

Die *Offset Rotation* stellt eine statische Variante für die Rotation dar (siehe Abbildung 5.6). Die Berechnung des Rotationswinkels ist hierbei unabhängig von der realen Position der Betrachter im TwoView, wodurch die Kameras beider Personen um den gleichen Winkel rotiert werden.

Um den Rotationswinkel zu bestimmen, wird zunächst der Öffnungswinkel α des Gesamtfrustums berechnet (siehe Abbildung 5.6(a)). Dieser ergibt sich aus der Viewerposition und dem *Screen*. Bei diesem Verfahren werden die Winkelhalbierenden als Blickrichtung verwendet. Die Winkel der Sichtgeraden der halbierten Sichtkegel und des Gesamtfrustums liegen in einem Winkel von $\alpha/4$ Grad auseinander. Um die Sichtgerade einer halbierten Ansicht auf die Blickrichtung des Gesamtfrustums zu legen, wird die Kamera, wie in Abbildung 5.6(b) zu sehen ist, um $\alpha/4$ Grad gedreht.

Da bei der *Offset Rotation* die reale Position des Benutzers nicht berücksichtigt wird, ergeben sich für beide Anwender identische Rotationen. Deshalb kann der Sichtkegel nur in wenigen Situationen optimal auf das fokussierte Objekt ausgerichtet werden. Durch die einheitliche Kameratransformation beider Benutzer wird der gemeinsame Raum bei diesem Verfahren nicht beeinträchtigt, weshalb sie weiterhin exakt miteinander interagieren können.



(a) *Offset Rotation:* vor der Aktivierung des *Splitscreens*

(b) *Offset Rotation:* nach der Aktivierung des *Splitscreens*

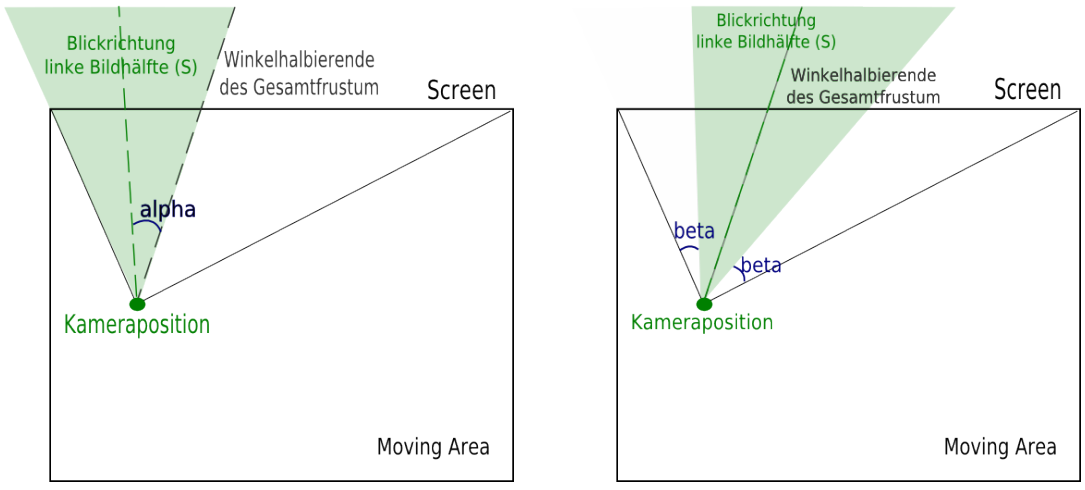
Abbildung 5.6: Offset Rotation

Übereinanderlegen der Winkelhalbierenden

Das *Übereinanderlegen der Winkelhalbierenden* stellt ein dynamisches Verfahren zur Rotation der Kamera dar, in dem die getrackten Positionen der Benutzer, im Gegensatz zu der vorher genannten Methode, berücksichtigt werden (siehe Abbildung 5.7). Durch die Kenntnis der virtuellen Kamerapositionen lässt sich der Sichtkegel genauer auf das fokussierte Objekt ausrichten.

Für verschiedene Positionen der Betrachter in der *Moving Area* ergeben sich unterschiedlich große Öffnungswinkel ihrer Sichtkegel. Steht ein Betrachter beispielsweise näher an der Projektionsleinwand, besitzt seine Kamera einen größeren Öffnungswinkel als bei einer weiteren Entfernung. Um die Rotationswinkel α für die Kameras beider Benutzer zu bestimmen, wird die Winkelhalbierende als Blickrichtung verwendet (siehe Abbildung 5.7(a)). Anschließend wird die Kamera vertikal um diesen Winkel gedreht (siehe Abbildung 5.7(b)). Aufgrund der unterschiedlichen Öffnungswinkel beider Benutzer ergeben sich unterschiedliche Rotationen. Durch die verschiedenen Kamerarotationen beider Benutzer wird der gemeinsame Raum zum kollaborativen Arbeiten beeinflusst.

Da die Winkelhalbierende das Gesamtfrustum symmetrisch aufteilt, ist der Rotationswinkel für Ansicht A und Ansicht B identisch. Durch die symmetrische Teilung des Gesamtfrustums liegt der rotierte Sichtkegel mittig auf der Sichtgeraden des Gesamtfrustums. Die Abweichung β zur linken und rechten Grenze des Gesamtfrustums ist identisch (siehe Abbildung 5.7(b)).



(a) *Übereinanderlegen der Winkelhalbierenden* vor der Aktivierung des *Splitscreens*

(b) *Übereinanderlegen der Winkelhalbierenden* nach der Aktivierung des *Splitscreens*

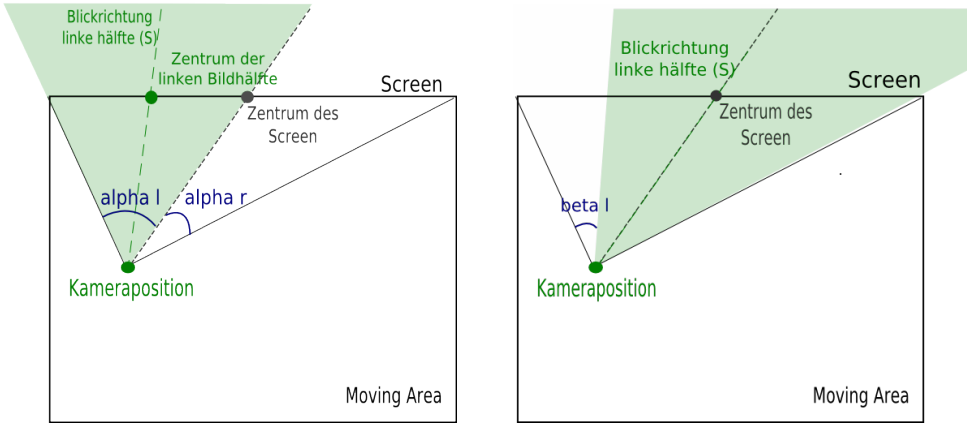
Abbildung 5.7: Übereinanderlegen der Winkelhalbierenden

Übereinanderlegen der Bildmitten

Das *Übereinanderlegen der Bildmitten* funktioniert ähnlich wie das *Übereinanderlegen der Winkelhalbierenden*. Auch hierbei handelt es sich um ein dynamisches Verfahren, welches die Rotationen der Kameras in Abhängigkeit der realen Betrachterpositionen berechnet.

Anders als bei den bisherigen Verfahren wird nun aber die jeweilige Bildmitte als Blickrichtung verwendet. Durch diese Annahme wird das Gesamtfrustum asymmetrisch aufgeteilt (siehe Abbildung 5.8). Die Sichtkegel der linken und der rechten Ansicht besitzen dadurch unterschiedliche Öffnungswinkel. Befindet sich der Benutzer beispielsweise vor der linken Hälfte der Projektionsleinwand, ist der Öffnungswinkel des Sichtkegels für die linke Bildhälfte (αl) größer als der Öffnungswinkel der rechten Bildhälfte (αr) (siehe Abbildung 5.8(a)). Aus diesem Grund ergeben sich andere Werte für die Rotationen der Kameras eines Benutzers in Ansicht A und Ansicht B.

Da die Bildmitte als Blickrichtung auch für die halbierten Ansichten angenommen wird, werden diese Sichtkegel ebenfalls asymmetrisch aufgeteilt. Nach dem Übereinanderlegen der Blickrichtungen ergibt sich deshalb eine unterschiedlich starke Abweichung β zu der linken und der rechten Grenze des Gesamtfrustums (siehe Abbildung 5.8(b)). In diesem Beispiel existiert keine Abweichung des rotierten Sichtkegels zur rechten Grenze des Gesamtfrustums. Wird diese Abweichung symmetrisch gestaltet, ergibt sich eine Übereinstimmung mit dem *Übereinanderlegen der Winkelhalbierenden*, da auch hier eine symmetrische Abweichung zu den Grenzen des Gesamtfrustums vorliegt.



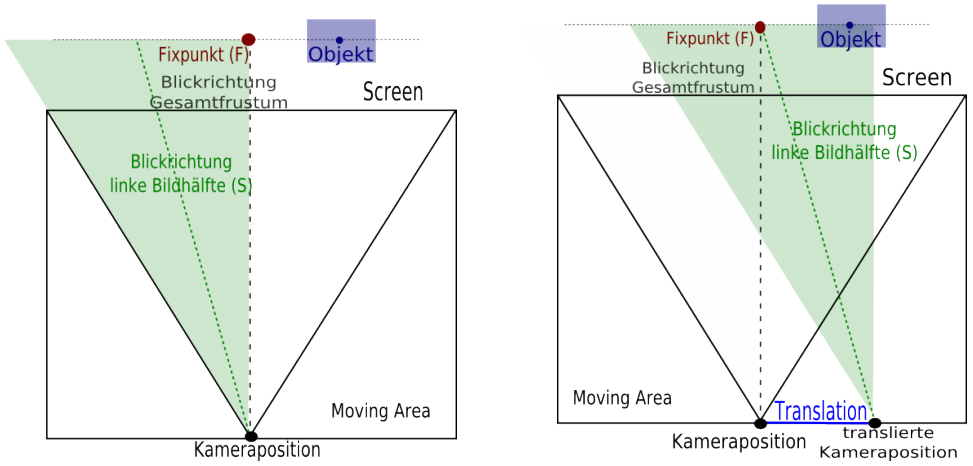
(a) *Übereinanderlegen der Bildmitten:* vor der Aktivierung des *Splitscreens*

(b) *Übereinanderlegen der Bildmitten:* nach der Aktivierung des *Splitscreens*

Abbildung 5.8: Übereinanderlegen der Bildmitten

5.5.2 Translation des Sichtkegels

Der zweite Ansatz, um den Sichtkegel auf einen fokussierten Punkt auszurichten, stellt die Translation des Sichtkegels dar. Hierbei werden die Kameras der Betrachter parallel zur Projektionsfläche verschoben. Um die Verschiebung der Kamera zu bestimmen, wird die Sichtgerade S der halbierten Sichtkegel auf den fixierten Punkt F gelegt. Anders als bei der Rotation bleibt hierbei die Ausrichtung des Sichtkegels erhalten, stattdessen wird die Position der Kamera verändert. Die Entfernung der Kamera zum fokussierten Objekt bestimmt hierbei maßgeblich den Wert der Translation. Je weiter das fokussierte Objekt vom Betrachter entfernt steht, desto weiter muss die virtuelle Kamera verschoben werden, um den Zielpunkt F mittig im halbierten Sichtkegel einer Ansicht zu positionieren. Wenn für die Berechnung des Werts der Translation die Entfernung nicht berücksichtigt wird, bewegen sich die Objekte mit zunehmender Entfernung an den Bildrand.



(a) Der rot markierte Fixpunkt F liegt auf der Sichtgeraden des Gesamtfrustums in der Höhe des Objekts

(b) Bestimmung der Translation, indem die Sichtgerade S auf den Fixpunkt F gelegt wird

Abbildung 5.9: Bestimmung der Translation

Für die Berechnung der Translation wird deshalb ein Fixationspunkt benötigt. Durch die Wahl einer Blickrichtung wird vorausgesetzt, dass der Fixpunkt auf der Sichtgeraden des Gesamtfrustums liegt. Befindet sich ein Objekt im Sichtbereich des Betrachters, wird der Punkt auf der Sichtgeraden in Höhe des Objekts als Fixpunkt F verwendet (siehe Abbildung 5.9(a)).

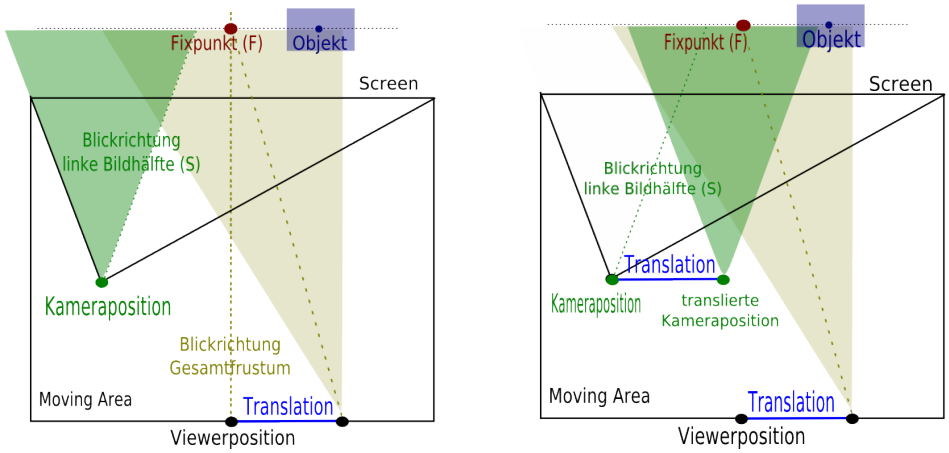
Zur Bestimmung der Translation einer Kamera wird die Sichtgerade S eines halbierten Sichtkegels auf den Fixpunkt gelegt und der Abstand zur vorherigen Kamerapo-

sition ermittelt (siehe Abbildung 5.9(b)). Je weiter also ein Objekt vom Betrachter entfernt steht, desto größer ist der Abstand von der Kamera zum Fixationspunkt. Dadurch ergibt sich eine betragsmäßig größere Translation der Kamera. Befindet sich kein Objekt im Sichtfeld des Betrachters, wird der Schnittpunkt der Sichtgeraden S mit dem *Screen* als Fixpunkt F verwendet. Auch für die Translation wurden drei Varianten umgesetzt: zwei dynamische Verfahren mit unterschiedlichen Blickrichtungen und ein statisches Verfahren.

Approximative Translation

Die *approximative Translation* ist wie die *Offset Rotation* eine statische Variante zur Transformation des Sichtkegels. Deshalb erfolgt die Berechnung der Translation unabhängig von der getrackten Position der Benutzer. Für die Berechnung der Translation wird die Viewerposition verwendet (siehe Abbildung 5.10(a)), die anschließend in Abbildung 5.10(b) auf die Kamera angewendet wird. Da sowohl für die Betrachterposition als auch für den Fixpunkt der Ansicht ein Näherungswert verwendet wird, stellt das Verfahren eine approximierende Berechnung der Translation dar.

Bei diesem Verfahren wird die Winkelhalbierende als Blickrichtung verwendet, wodurch die Sichtkegel symmetrisch geteilt werden. Da die reale Position des Benutzers außer Acht gelassen wird, ergibt sich für alle Positionen der Benutzer im TwoView der gleiche Abstand zum Objekt. Aus diesem Grund ist die Transformation der Kamera für beide Anwender identisch, und der gemeinsame Raum wird nicht beeinträchtigt.



(a) *Approximative Translation:* vor der Aktivierung des *Splitscreens*

(b) *Approximative Translation:* nach der Aktivierung des *Splitscreens*

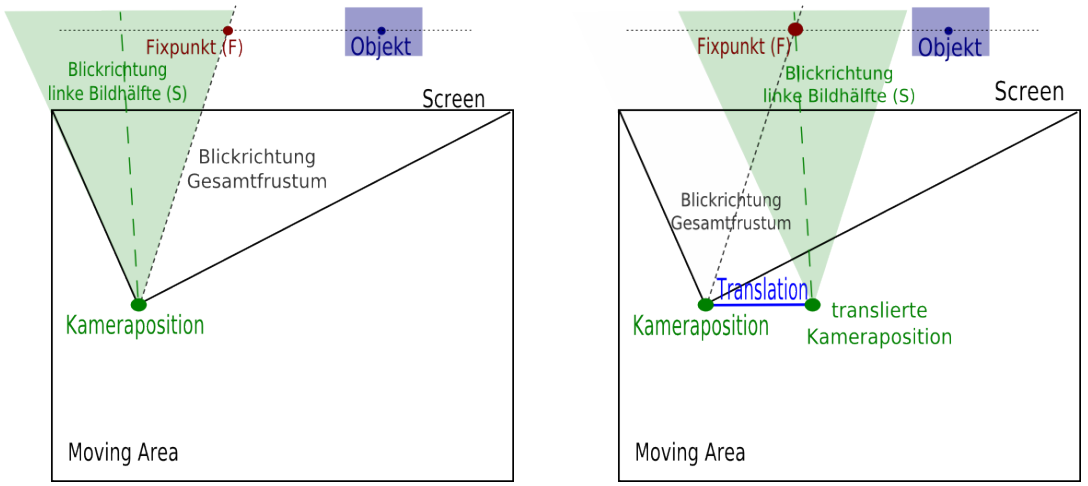
Abbildung 5.10: Bei der approximativen Translation wird die Translation für die Viewerposition ermittelt und auf die Kamera übertragen

Schneiden der Winkelhalbierenden

Das *Schneiden der Winkelhalbierenden* ist eine dynamische Variante zur Translation der Kamera. Durch das Verwenden der realen Positionen der Betrachter kann der Sichtkegel genauer auf den fokussierten Punkt ausgerichtet werden. Da aber durch diese Vorgehensweise die Kameras beider Betrachter unterschiedlich transformiert werden, kann das kollaborative Arbeiten beeinträchtigt werden.

Bei diesem Verfahren wird die Winkelhalbierende der Sichtkegel als Blickrichtung verwendet (siehe Abbildung 5.11). Durch die unterschiedlichen Positionen beider Benutzer ergeben sich unterschiedliche Öffnungswinkel der Sichtkegel. Deshalb werden für beide Betrachter unterschiedliche Translationen ausgeführt. Auch der Abstand zum fokussierten Objekt ändert sich je nach Position des Betrachters, was sich zusätzlich auf die Translation der Kamera auswirkt.

In Abbildung 5.11(b) wird die Sichtgerade S der linken Ansicht auf den Fixpunkt gelegt und anschließend die Kamera parallel zur Projektionsfläche verschoben.



(a) *Schneiden der Winkelhalbierenden:* vor der Aktivierung des *Splitscreens*

(b) *Schneiden der Winkelhalbierenden:* nach der Aktivierung des *Splitscreens*

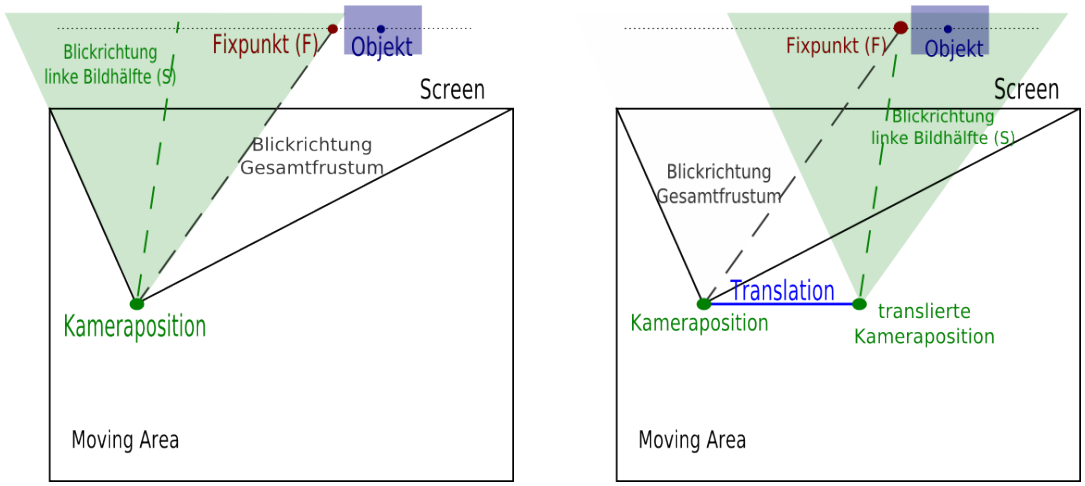
Abbildung 5.11: Schneiden der Winkelhalbierenden im Fixpunkt

Schneiden der Bildmitten

Das *Schneiden der Bildmitten* stellt ebenfalls ein dynamisches Verfahren zur Berechnung der Kameratranslation dar. Anders als beim *Schneiden der Winkelhalbierenden* wird bei dieser Variante die Gerade zwischen Betrachter und Bildmitte als Blickrichtung verwendet (siehe Abbildung 5.12). Durch diese Annahme werden die Sichtkegel asymmetrisch aufgeteilt, und es ergeben sich unterschiedlich große Sichtkegel für die linke und die rechte Bildhälfte.

Da hier ebenfalls die realen Positionen der Benutzer berücksichtigt werden, ergeben sich unterschiedliche Translationen für beide Anwender. Dadurch wird der gemeinsame Raum und das kollaborative Arbeiten beeinträchtigt.

In Abbildung 5.12(b) wird die Sichtgerade S der linken Ansicht auf den Fixpunkt gelegt und anschließend die Kamera parallel zur Projektionsfläche verschoben.



(a) *Schneiden der Bildmitten:* vor der Aktivierung des *Splitscreens*

(b) *Schneiden der Bildmitten:* nach der Aktivierung des *Splitscreens*

Abbildung 5.12: Schneiden der Bildmitten im Fixpunkt

5.6 Einbindung in die bestehende Applikationsstruktur

Die Realisierung der zuvor beschriebenen Verfahren zur simultanen Darstellung zweier Ansichten erfolgt in der Klasse *Splitter*. Um diese Klasse in die bestehende Applikationsstruktur zu integrieren, wird der *Splitter*, wie in Abbildung 5.13 dargestellt, zwischen den *Mover* und den *Viewer* geschaltet. Der *Mover* liefert die Viewerposition, mit der die virtuelle Position der Betrachter bestimmt werden kann. Da der *Splitter* die Kamerapositionen verändert, müssen für die Visualisierung die transformierten Kamerapositionen verwendet werden.

Damit der Benutzer das gewünschte Verfahren (*Switchen, Picture in Picture, Splitscreen*) mit dem PDA-Client wählen kann, wurde dort ein zusätzlicher Button zur Verfügung gestellt. Die gewählte Methode wird an den *Splitter* übertragen und dort ausgewertet. Je nach gewähltem Verfahren wird der Bildschirm unterschiedlich aufgeteilt. Auch die verschiedenen Varianten des *Splitscreens* können über den PDA-Client ausgewählt werden. Die Positionen der transformierten Kameras und die Aufteilung der Projektionsfläche werden anschließend an den *Viewer* für die Visualisierung weitergeleitet.

Anders als bei der Rotation der Sichtkegel erfolgt die Translation der Kameras in Abhängigkeit der Entfernung des fokussierten Objekts. Befindet sich kein Objekt im Sichtkegel des Gesamtfrustums, wird der Schnittpunkt der Sichtgeraden mit dem *Screen* als Fokus verwendet. Da die Galleria Borghese anhand von Wegen begangen wird, findet eine kontinuierliche Bewegung der Kamera statt. Dadurch können während der Begehung Objekte in den Sichtkegel des Betrachters gelangen und ihn auch wieder verlassen. In diesen Fällen findet eine sprunghafte Bewegung der Kamera statt, da die Entfernung zwischen den unterschiedlichen Fixationspunkten F recht groß sein kann. Um diese sprunghafte Bewegung zu beheben, wird im *Splitter* ein Grenzwert definiert, der die Transformation der Kamera einschränkt. Große Bewegungen werden somit auf mehrere Frames verteilt, wodurch eine kontinuierliche Bewegung der Kamera auf die Zielposition erfolgt.

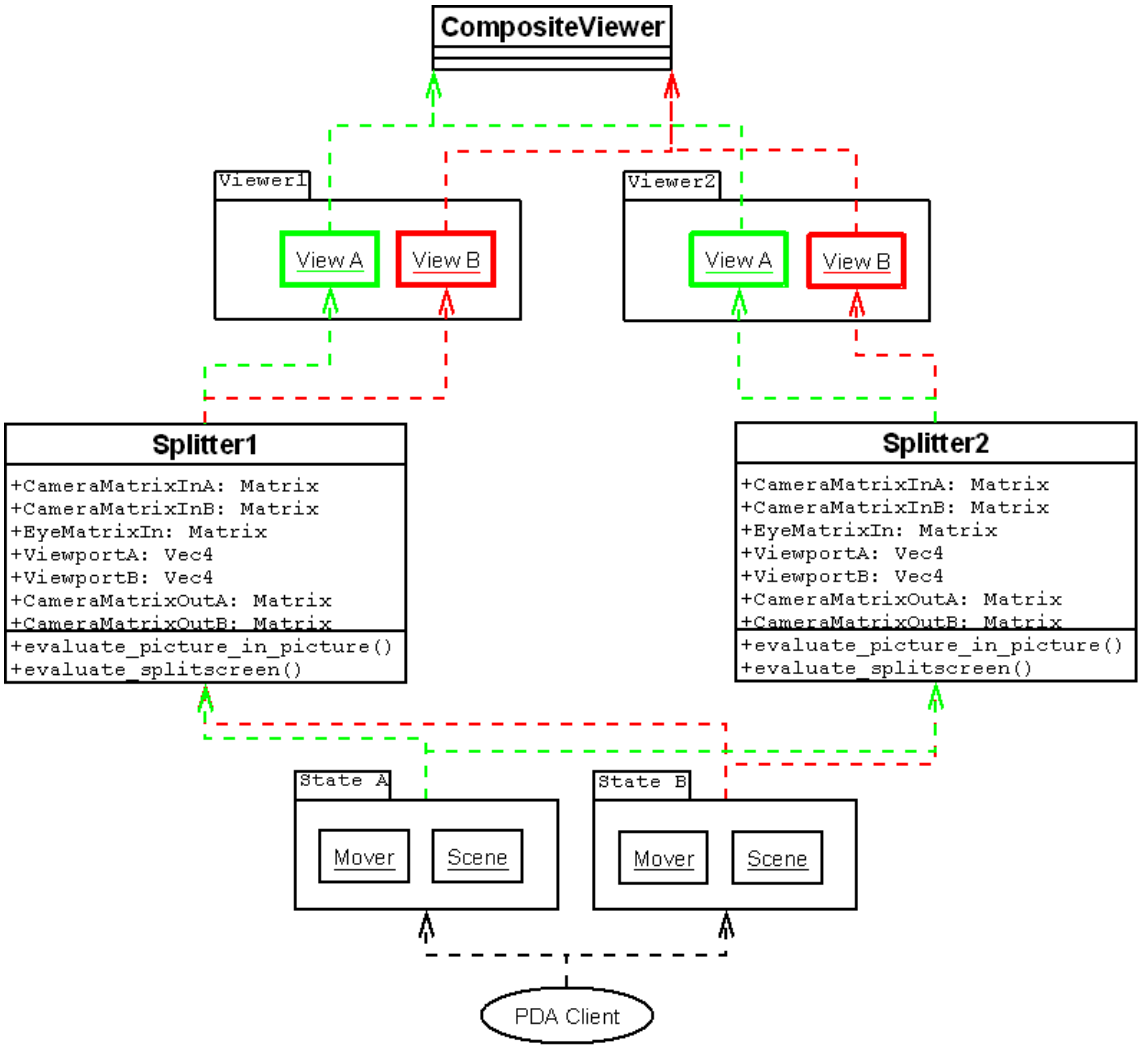


Abbildung 5.13: Applikationsstruktur mit integriertem *Splitter*

5.7 Evaluation der Ergebnisse

5.7.1 Vorgehensweise

Für die Bewertung der Ergebnisse wurde ein Usertest mit Hilfe von Medienwissenschaftlern der Universität Siegen durchgeführt. Der Test gliedert sich in ein Tutorial, einen Aufgabenteil, einen Fragebogen sowie eine abschließende Diskussion.

Beginnend mit dem Tutorial wurde den Probanden das vorhandene System erklärt. Zunächst wurden die grundlegenden Begriffe *Setting* (Aufstellung der Skulpturen), *Tour* (vorgegebene Wege) und *Path* (selbst erstellter Weg) erläutert, da diese die grundlegenden Funktionen der Applikation repräsentieren. Anschließend wurde zu jedem Menüpunkt eine Anleitung gegeben, die Schritt für Schritt zu befolgen war. Dadurch lernten die Probanden den grundlegenden Umgang mit der Applikation kennen und konnten sich einen Überblick über die zur Verfügung stehenden Funktionen verschaffen. Weiterhin konnten sie durch das Tutorial einen Eindruck vom Arbeiten im TwoView erhalten. Nachdem sich die Probanden mit der Steuerung der Applikation und ihren Möglichkeiten vertraut gemacht hatten, wurden die Aufgaben für den Usertest durchgeführt. Die gestellten Aufgaben waren von den Probanden selbstständig zu lösen. Um eine unvoreingenommene Bewertung der verschiedenen Verfahren zur *Darstellung mehrerer Ansichten* zu erreichen, wurden die genauen Vorgehensweisen der präsentierten Methoden vor dem Usertest nicht erläutert.

Um die Probanden nicht durch die Vielzahl der zu testenden Verfahren zu überfordern, wurde die Anzahl der verfügbaren *Splitscreen*-Varianten von sechs auf vier reduziert (die *Offset Rotation*, das *Übereinanderlegen der Winkelhalbierenden*, die *Approximative Translation* und das *Schneiden der Winkelhalbierenden*).

Die Resultate der verschiedenen *Splitscreen*-Verfahren weisen nur geringe Unterschiede auf. Viele Verfahren mit geringen Unterschieden würden auf die Tester verwirrend wirken, wodurch die Varianten des *Splitscreens* im Nachhinein nicht mehr differenziert und kein brauchbares Testergebnis erzielt werden könnte. Um eine Präferenz der Probanden, bezogen auf die Rotation oder Translation zu erhalten, wurden für den Test jeweils zwei Verfahren zur Rotation und Translation gewählt. Es wurde jeweils eine statische und eine dynamische Variante gewählt, um einen Eindruck der Beeinträchtigung des gemeinsamen Raums zu erhalten.

Für die Wahl der Blickrichtung konnte zwischen zwei Modi gewählt werden: die Winkelhalbierende und die Bildmitte. Bei der Bildmitte wird die Gerade zwischen Betrachter und Mitte der Projektionsleinwand als Blickrichtung verwendet. Die Winkelhalbierende wird durch die Gerade dargestellt, die den Öffnungswinkel des Sichtkegels symmetrisch teilt. Da im virtuellen Raum allerdings keine Projektionsfläche vorhanden ist, scheint Winkelhalbierende die sinnvollere Annahme zu sein. Durch die Verwendung der Bildmitte wird angenommen, dass die Benutzer immer in die Mitte der Projektionsleinwand sehen, auch wenn sie beispielsweise vor der rechten Bildhälfte stehen. Da der Betrachter in diesem Fall aber eher auf die rechte Bildhälfte fokussiert ist, spricht dies ebenfalls für die Winkelhalbierende. Zudem liefert die Winkelhalbierende einen realistischeren Blickwinkel auf die Projektionsleinwand. Durch den Blickwinkel zur Leinwand wird das Kontrastverhältnis und die Farbtreue des gesehenen Bildes beeinflusst. Je flacher der Blickwinkel auf die Leinwand fällt, desto eher treten hierdurch negative Abweichungen dieser beiden Attribute auf. Da die Winkelhalbierende einen stumpferen Winkel auf die Projektionsleinwand liefert als die Blickrichtung zur Bildmitte, stellt die Winkelhalbierende für diesen Aspekt ebenfalls die sinnvollere Annahme dar.

Aus diesem Grund wurden für den Test die *Offset Rotation*, das *Übereinanderlegen der Winkelhalbierenden*, die *Approximative Translation* und das *Schneiden der Winkelhalbierenden* ausgewählt.

5.7.2 Aufgabenstellung

Der Usertest konzentrierte sich auf die Wirkung der *Darstellung mehrerer Ansichten*. In den Aufgaben sollten die Probanden verschiedene Ansichten erzeugen und mit Hilfe der gegebenen Verfahren miteinander vergleichen. Um die Möglichkeiten zu erkennen, die ihnen durch die Neuerungen zur Verfügung stehen, sollten bei jeder Aufgabe unterschiedliche Aufstellungen und Wege für beide Ansichten gewählt werden. Dadurch sollte den Probanden aufgezeigt werden, dass die Ansichten vollkommen separat voneinander existieren und verschiedene Settings und Wege aufweisen können.

Mit den Aufgaben wurden alle drei verfügbaren Methoden (*Switchen*, *Picture in Picture* und *Splitscreen*) getestet. Zuletzt wurde eine freie Aufgabe gestellt, indem die Probanden ein eigenes Szenario erstellen und ihre bevorzugte Methode wählen

konnten. Um das kollaborative Arbeiten während der Aufgaben zu testen, sollte eine Person auf bestimmte Merkmale einer Skulptur deuten, die die andere Person identifizieren musste.

5.7.3 Auswertung

Die Methoden zur Darstellung mehrerer Ansichten wurden von den Probanden sehr gut aufgenommen. Das *Switchen* und der *Picture in Picture*-Modus wurden aber für einen direkten Vergleich zweier Bilder als wenig hilfreich angesehen. Beim *Picture in Picture*-Modus wurde die zweite Ansicht als Gedächtnisstütze empfunden, mit der man diese immer im Blickfeld hat und dadurch nicht in Vergessenheit gerät.

Der *Splitscreen* ist laut Testergebnis für den Vergleich zweier Ansichten am besten geeignet. Er spiegelt die klassische Arbeitsweise der Kunsthistoriker wider, bei der zwei Diaprojektoren zur Darstellung zweier Bilder verwendet werden. Durch die gleichwertige und verzerrungsfreie Darstellung beider Ansichten ist der *Splitscreen* gut für Detailvergleiche zweier Bilder geeignet. Die Unterschiede zwischen den verschiedenen Varianten des *Splitscreens* wurden als sehr gering empfunden. Aus diesem Grund konnte sich im Usertest kein Favorit herausstellen.

In der Nachbesprechung wurde die Funktionsweise der verschiedenen Varianten des *Splitscreens* aufgelöst. Mit Hilfe der *Preview* konnten die Unterschiede der Verfahren aufgezeigt und erklärt werden. Hierbei wurde auch die unterschiedliche Wirkung der Rotation und Translation angesprochen. Im Vergleich zur Rotation wurde die Translation als angenehmer empfunden, da sie ein frontaleres Bild liefert. Bei der Rotation hingegen erscheinen die Objekte leicht eingedreht zum Betrachter. Weiterhin wurde in der Nachbesprechung erwähnt, dass bei einem normalen Museumsbesuch das Museum als Ganzes im Vordergrund steht. Für einen solchen Zweck wäre die Translation beim *Splitscreen* völlig ausreichend, da mehrere Auswahlmöglichkeiten auf einen Besucher eher verwirrend wirken würden. Geht es aber darum, Objekte wissenschaftlich zu analysieren, können beide Verfahren hilfreich sein, da sie unterschiedliche Darstellungen der untersuchten Objekte liefern. Demnach kann für die jeweilige Aufgabe die geeignete Variante ausgewählt werden.

Nach der Aktivierung des *Splitscreens* haben sich die Probanden meist vor der Bildhälfte positioniert, die sie gerade betrachtet haben. Dadurch standen die Betrachter recht eng beieinander, wodurch die Transformationen der virtuellen Kameras für beide Benutzer recht ähnlich ausfielen. In diesem Fall fällt die Beeinträchtigung des gemeinsamen Raums bei dem *Splitscreen* gering aus. Da kein Vergleichswert zum kollaborativen Arbeiten im Vollbildmodus existiert, findet eine Auswertung dieses Aspekts in Kapitel 7, *Empirischer Test zum gemeinsamen Raum*, statt.

6 Pfadkorrektur

6.1 Motivation

Wie in Abschnitt 3.3 *Kollaborative Interaktion in virtuellen Umgebungen im kunsthistorischen Kontext* beschrieben, erfolgt das Erkunden der Galleria Borghese anhand von vorgegebenen oder selbst definierten Wegen. Durch die Visualisierung der Applikation im TwoView können sich zwei Benutzer in der virtuellen Galleria befinden und diese gemeinsam erkunden. Hierbei beschreiten beide Betrachter die gleiche Tour und befinden sich immer auf einer gemeinsamen Position des gleichen Weges.

Die Kameraposition eines Betrachters ermittelt sich aus der Viewerposition und seiner getrackten Position im TwoView. Durch den Bewegungsspielraum innerhalb der *Moving Area* können die Betrachter ihre virtuelle Position verändern und von dem eigentlichen Weg abweichen. Diese Abweichung kann beim Durchlaufen von schmalen Durchgängen, wie zum Beispiel von Türen, zu einem Kollisionsproblem führen. Hat sich beispielsweise ein Betrachter zu weit vom eigentlichen Weg entfernt, kann es vorkommen, dass er nicht mehr durch einen vor ihm liegenden Durchgang geleitet, sondern durch eine Wand geführt wird. In dem Beispiel aus Abbildung 6.1 führt der vorgegebene Weg (gelb markiert) durch eine Tür. Da die virtuelle Kameraposition nicht auf dem Wegpunkt liegt, läuft der Betrachter durch eine Wand. Dieses Phänomen wurde bei den bisherigen Usertests als sehr störend und unangenehm empfunden, weil dadurch auch die Immersion des Benutzers gestört wird. Denn je länger sich ein Benutzer ungestört in einer virtuellen Umgebung befindet, desto intensiver erlebt er das Gefühl, in diese Welt einzutauchen.

Die Galleria Borghese setzt sich aus vielen Räumen zusammen, die über Türen (Gates) miteinander verbunden sind. Deshalb tritt das beschriebene Problem bei der Begehung der Villa recht häufig auf. Eine Pfadkorrektur soll dieses Problem beheben, ohne den Benutzer zu stark in seiner eigentlichen Arbeit einzuschränken.

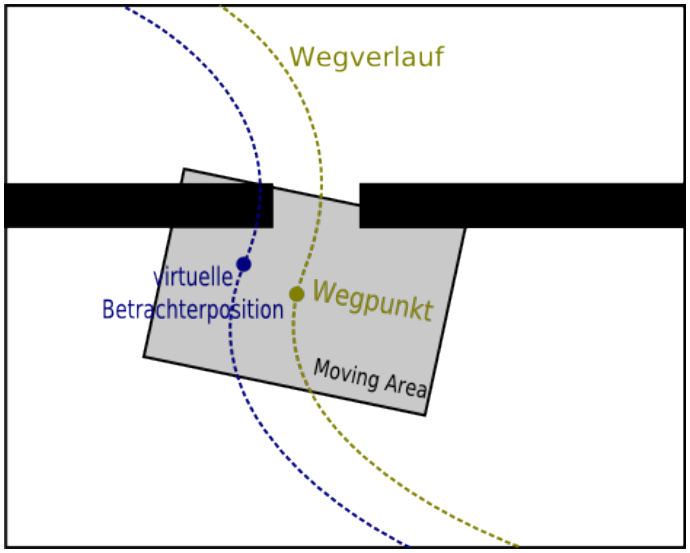


Abbildung 6.1: Steht der Betrachter (blau) zu weit vom eigentlichen Weg (gelb) entfernt, wird er durch eine Wand geführt

In diesem Kapitel sollen verschiedene Ansätze zur Lösung dieses Problems vorgestellt werden.

6.2 Anforderungen

Auf Grundlage des Projekts *Virtualisierung von Skulptur* ließen sich die folgenden Anforderungen herleiten, um die korrekte und vollständige Umsetzung der *Pfadkorrektur* zu verifizieren.

Passives Verhalten
Der Benutzer soll das Verhalten nicht explizit aktivieren oder deaktivieren müssen. Die kritischen Situationen sollen automatisch erkannt und eine entsprechende Problembehandlung ausgeführt werden. Dadurch müssen keine neuen Bedienungselemente auf dem PDA-Client eingefügt werden.

Temporäres Verhalten
Die Ausführung der Pfadkorrektur soll nur für einen kurzen Zeitraum erfolgen. Nachdem der Durchgang passiert wurde, soll der Ausgangszustand wieder hergestellt werden.

Keine Beeinträchtigung des Benutzers
Die Pfadkorrektur soll den Benutzer in seinem Verhalten nicht beeinflussen. Er soll seine Arbeit möglichst ungestört fortführen können.

Natürliches Verhalten
Die Pfadkorrektur soll ein natürliches Verhalten widerspiegeln. So sollte beispielsweise das Verändern der virtuellen Position nicht sprunghaft, sondern durch eine homogene Bewegung erfolgen.

Kollaboratives Arbeiten
Durch die Ausführung der Applikation im TwoView wird kollaboratives Arbeiten ermöglicht. Die Pfadkorrektur soll hierauf keinen negativen Einfluss nehmen.

6.3 Methoden zur Pfadkorrektur

Die Methoden der Pfadkorrektur geben verschiedene Vorgehensweisen an, mit dem das Kollisionsproblem behoben werden kann. Da zwei Personen im TwoView agieren können, wird zum besseren Verständnis die Person, die nicht durch den Durchgang passt, als Verursacher bezeichnet. In den folgenden Abschnitten werden drei Ansätze für eine Pfadkorrektur vorgestellt.

6.3.1 Translation der virtuellen Kamera

Das erste Verfahren für die Umsetzung der Pfadkorrektur ist die *Translation der virtuellen Kamera*. Falls ein Benutzer nicht durch einen Durchgang (Gate) passt, wird

die Kamera des Verursachers verschoben, um ihn durch das Gate zu leiten (siehe Abbildung 6.2). Um eine sanfte Translation der Kamera zu ermöglichen muss frühzeitig überprüft werden, ob der Benutzer durch den vor ihm liegenden Durchgang passt. Nachdem das Gate durchquert wurde, wird die Kamera des Verursachers wieder auf ihre eigentliche Position zurücktransformiert. Befinden sich zwei Betrachter im TwoView, können beide Personen ein Kollisionsproblem verursachen. In diesem Fall wird die *Translation der virtuellen Kamera* für beide Benutzer unabhängig voneinander vorgenommen.

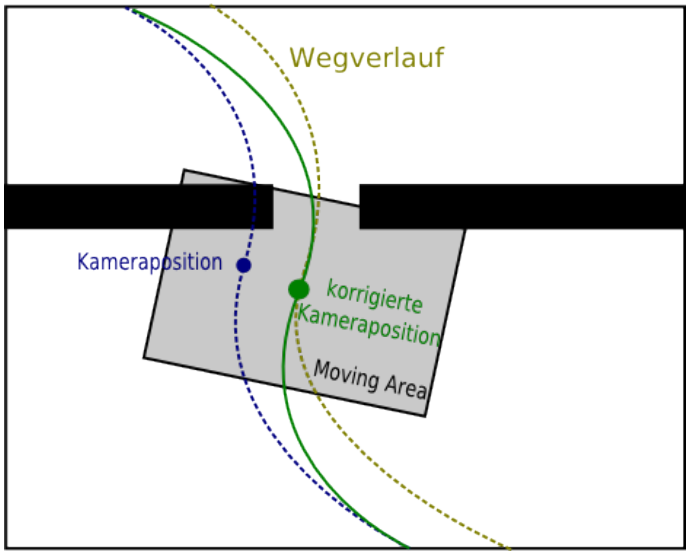


Abbildung 6.2: Da die Kameraposition (blau) mit einer Wand kollidiert, wird die Kamera auf dem grün markierten Weg durch das Gate geführt

Für die *Translation der virtuellen Kamera* können zwei verschiedene Vorgehensweisen ausgeführt werden. Zum einen kann eine Vereinigung der virtuellen Welten beider Anwender vorgenommen werden, zum anderen kann der Verursacher immer mittig durch ein Gate geführt werden. Eine Vereinigung der virtuellen Welten hätte zur Folge, dass die Kameraposition des Verursachers auf die Kameraposition seines Partners bewegt wird. Dadurch ist die virtuelle Position des Verursachers abhängig von der realen Position seines Partners im TwoView. Bewegt sich dieser und der Verursacher bleibt an seiner bisherigen Position stehen, wird die Kamera der Verursachers trotzdem bewegt. Dieses Vorgehen würde unruhig und störend auf den Betrachter wirken. Zudem müsste für den Fall, dass beide Benutzer nicht durch ein vor ihnen liegendes Gate passen, eine zusätzliche Behandlung erfolgen. Aus diesem Grund wird für die Pfadkorrektur immer ein mittiges Durchqueren eines Gates umgesetzt.

Wie bereits beim *Splitscreen* erkannt, wird durch eine ungleiche Transformation beider Kameras der gemeinsame Raum und das kollaborative Arbeiten beeinträchtigt. Aus diesem Aspekt ist es unerheblich, ob die virtuellen Welten vereint werden oder der Betrachter mittig durch einen Durchgang geleitet wird. Je weiter beide Benutzer in der *Moving Area* auseinander stehen, desto unterschiedlicher fallen die Translationen der Kameras für beide Anwender aus. Dadurch wird der gemeinsame Raum stärker beeinträchtigt als bei ähnlichen Kameratransformationen, und ein exaktes Interagieren beider Benutzer ist in diesen Situationen nicht mehr gewährleistet.

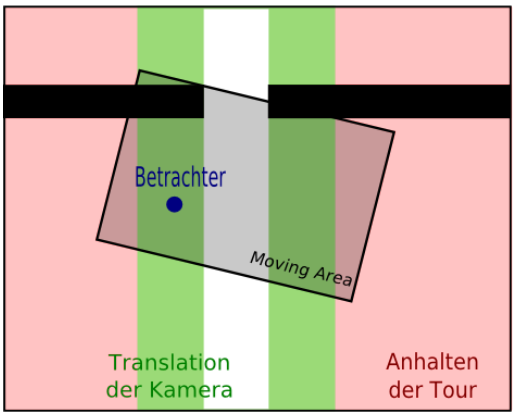
6.3.2 Anhalten der Tour

Ein anderer Ansatz, um auf das Problem zu reagieren, ist das *Anhalten der Tour*. Sobald ein Kollisionsproblem erkannt wird, beginnt die Applikation damit, langsam die Tour zu stoppen. Da ein Benutzer die Geschwindigkeit mit Hilfe des PDAs steuern kann, muss eine Überlagerung dieser Geschwindigkeit durch die Applikation vorgenommen werden. Nachdem die Tour angehalten wurde, müssen sich die Benutzer so positionieren, dass sie beim Fortsetzen der Tour problemlos durch das Gate geleitet werden. Daraufhin wird die Tour wieder auf ihre vorherige Geschwindigkeit beschleunigt. Die problematischen Situationen müssen somit frühzeitig erkannt werden, um das Abbremsen und Beschleunigen der Tour möglichst sanft zu gestalten.

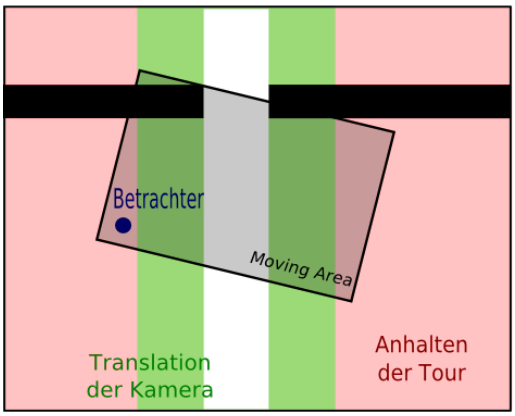
Da sich die Geschwindigkeit immer auf eine Tour und nicht auf einen Benutzer bezieht, kann keine separate Behandlung der Geschwindigkeitssteuerung für beide Anwender stattfinden. Wenn eine Person nicht durch das Gate geleitet werden kann, sind immer beide Anwender von dem Anhalten der Tour betroffen. Da die Durchgänge in der Galleria Borghese relativ schmal sind, müssen die Betrachter sehr eng zusammenrücken, um durch die Gates geführt zu werden. Deshalb kann das Stoppen der Tour relativ häufig auftreten, was auf die Benutzer schnell störend wirken kann. Ein Vorteil dieses Verfahrens ist jedoch, dass ‚anders als bei der *Translation der virtuellen Kamera*, der gemeinsame Raum nicht beeinträchtigt wird. Dadurch ist es während der Behandlung des Kollisionsproblems weiterhin möglich, kollaborativ zu arbeiten. Außerdem erhalten die Anwender ein direktes Feedback und müssen aktiv in das Geschehen eingreifen, was zu einer realeren Wahrnehmung der virtuellen Umgebung beitragen kann.

6.3.3 Kombination beider Verfahren

Beide bisher beschriebenen Verfahren weisen Nachteile auf. Aufgrund der oben genannten Probleme soll deshalb eine Kombination beider Ansätze die negativen Eigenschaften einschränken. Hierzu wurde ein Grenzwert eingeführt, der die Anwendung beider Verfahren ermöglicht. Durch die Einführung dieses Grenzwerts werden drei Zonen definiert, die eine unterschiedliche Behandlung der Pfadkorrektur mit sich bringen. In Abbildung 6.3 werden diese drei Zonen gezeigt. Ein Benutzer kann sich innerhalb dieser Zonen befinden. Befindet er sich im weißen Bereich, ist keine Pfadkorrektur erforderlich, befindet er sich außerhalb dieses Bereiches, verursacht er eine Kollision. Deshalb findet in der grünen Zone eine *Translation der virtuellen Kamera* (Abbildung 6.3(a)) statt und im roten Bereich das *Anhalten der Tour* (Abbildung 6.3(b)).



(a) Befindet sich ein Betrachter in der grünen Zone, wird er durch das Gate geführt



(b) Befindet sich ein Betrachter in der roten Zone, wird die Tour angehalten

Abbildung 6.3: Die Grenzwerte legen unterschiedliche Zonen fest:
Rot: Verringern der Geschwindigkeit
Grün: Translation der virtuellen Kamera
Weiß: keine Pfadkorrektur

Ein positiver Effekt der Kombination beider Verfahren ist, dass das *Anhalten der Tour* nicht mehr so häufig auftritt, und die Benutzer zudem nicht mehr so eng zusammenrücken müssen wie bei der alleinigen Anwendung dieses Verfahrens. Außerdem wird durch den Grenzwert eine maximale *Translation der virtuellen Kamera* vorgegeben, wodurch der gemeinsame Raum nicht mehr so stark beeinträchtigt wird

wie zuvor. Durch die Kombination beider Verfahren werden also die negativen Eigenschaften beider Vorgehensweisen abgeschwächt und ein angenehmeres Verhalten als bei der Anwendung eines einzelnen Verfahrens erzielt.

6.4 Erkennen der problematischen Situationen

Um ein angenehmes Verhalten der Pfadkorrektur zu ermöglichen, müssen die beschriebenen Methoden zur Pfadkorrektur früh genug angewendet werden können. Deshalb ist es wichtig, die problematischen Situationen rechtzeitig zu erkennen. Eine Pfadkorrektur ist prinzipiell immer beim Durchqueren eines Gates erforderlich. Deshalb muss vor dem Verlassen eines Raumes geprüft werden, ob die Benutzer durch das Gate passen oder nicht. Ist eine Pfadkorrektur erforderlich, wird eins der oben genannten Verfahren angewendet, damit die Benutzer beim Eintritt in das Gate auf einer sicheren Position stehen. Für das Erkennen des Kollisionsproblems werden im Folgenden drei verschiedene Vorgehensweisen erläutert.

6.4.1 Pfadbasierter Ansatz

Die Begehung der Galleria Borghese erfolgt anhand von Wegen (Touren und Pfaden). Um zu überprüfen, ob die Benutzer demnächst ein Gate durchqueren, kann der begangene Weg im Voraus untersucht werden. Deshalb wird vom aktuellen Wegpunkt aus getestet, ob der nachfolgende Wegabschnitt durch ein Gate verläuft. Ist dies der Fall, müssen die virtuellen Positionen beider Betrachter für diesen Zeitpunkt ermittelt werden, um anschließend überprüfen zu können, ob sie eine Kollision mit einer Wand verursachen.

Um festzustellen, ob eine Pfadkorrektur erforderlich ist, werden im nächsten Schritt die Eintrittspunkte der Betrachter in das vor ihnen liegende Gate ermittelt. Befinden sich diese Eintrittspunkte innerhalb der Gategrenzen, ist keine Behandlung der Situation erforderlich. Befinden sie sich neben der Türöffung, muss eine Pfadkorrektur erfolgen. Um die Kameras der Betrachter möglichst wenig zu bewegen, werden sie parallel zum *Screen* verschoben. Das *Anhalten der Tour* kann durch eine beliebige Funktion umgesetzt werden.

Um die Situationen frühzeitig zu erkennen, muss der Weg weit genug im Voraus untersucht werden. Da die Wegpunkte aber bei jeder Tour und jedem Pfad unterschiedliche Abstände besitzen, liefert die Anzahl der im Voraus untersuchten Wegpunkte keinen eindeutigen Hinweis auf die Entfernung zu einem Durchgang. Mit zunehmender Geschwindigkeit werden außerdem größere Schritte ausgeführt, wodurch einige Wegpunkte übersprungen werden. Da der Anwender die Geschwindigkeit einer Tour eigenhändig steuern kann, liefert das Wissen um den Wegpunkt beim Verlassen des Raumes keinen verlässlichen Hinweis auf den Zeitpunkt, wann die Pfadkorrektur beginnen soll.

Dieser Ansatz ist auf die Verwendung von Wegen angewiesen, weshalb die Pfadkorrektur bei einem freien Durchlaufen der Galleria Borghese mit diesem Verfahren nicht mehr anwendbar ist.

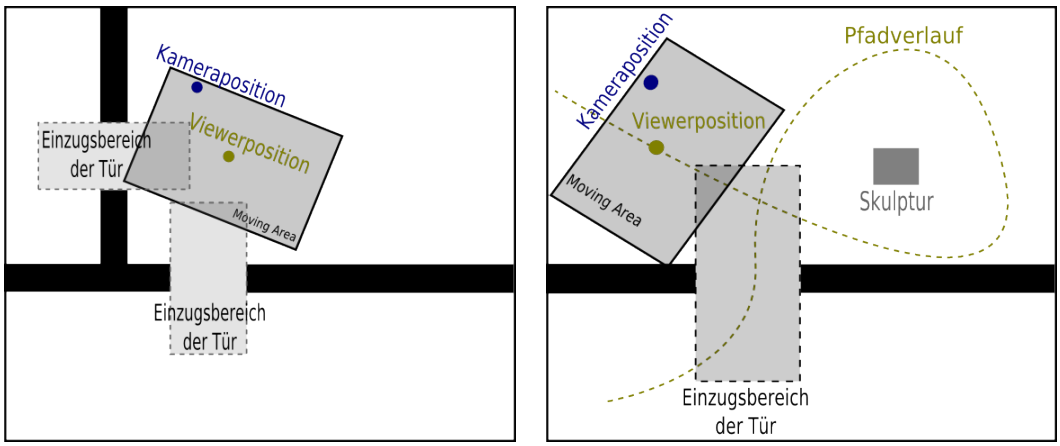
6.4.2 Statebasierter Ansatz

Der pfadbasierte Ansatz ermöglicht kein freies Begehen der virtuellen Umgebung und liefert keinen verlässlichen Startzeitpunkt für die Pfadkorrektur. Deshalb wird beim statebasierten Ansatz nur der Status der aktiven Ansicht ausgewertet und auf die Weginformation verzichtet.

Das Erkennen der Situationen, bei denen eine Pfadkorrektur erforderlich ist, soll hierbei ausschließlich anhand der virtuellen Positionen der Betrachter erfolgen. Um dies zu erreichen, wurde die *Moving Area* des TwoView in die virtuelle Umgebung übertragen und an der Viewerposition befestigt (siehe Abbildung 6.4). Zusätzlich wurden Einzugsbereiche der Gates definiert, um frühzeitig die problematischen Situationen zu erkennen (gestrichelt eingerahmte Fläche in Abbildung 6.4(a) und 6.4(b)). Überschneiden sich die *Moving Area* und der Einzugsbereich eines Gates wird davon ausgegangen, dass der Benutzer demnächst den aktuellen Raum verlässt.

Dieses Vorgehen kann in bestimmten Situationen zu einer fehlerhaften Erkennung des Kollisionsproblems führen. Liegen beispielsweise mehrere Türen nahe beieinander, kann das Gate, welches durchquert wird, nicht mehr eindeutig bestimmt werden. In dem Beispiel in Abbildung 6.4(a) befinden sich zwei Einzugsbereiche in der *Moving Area*, weshalb nicht identifiziert werden kann, welches Gate demnächst durchquert wird. In anderen Situationen kann es vorkommen, dass die Betrachter

wie in Abbildung 6.4(b)) nahe an einem Gate vorbeigehen, ohne es durchqueren zu wollen. In diesem Fall wird fälschlicherweise eine Pfadkorrektur durchgeführt. Die Überschneidung der *Moving Area* mit den Einzugsbereichen der Gates reicht deshalb als alleiniges Kriterium für die Erkennung problematischer Situationen nicht aus. Um die Kollisionen korrekt identifizieren zu können, wird deshalb zusätzlich noch die bisherige Laufrichtung des Anwenders sowie die Blickrichtung benötigt. Anhand dieser zusätzlichen Informationen kann ermittelt werden, ob sich die Benutzer auf das Gate zubewegen, und somit die Situationen korrekt identifiziert werden. Die Laufrichtung wird aus den letzten beiden Viewerpositionen ermittelt.



(a) Das durchquerte Gate kann nicht eindeutig bestimmt werden

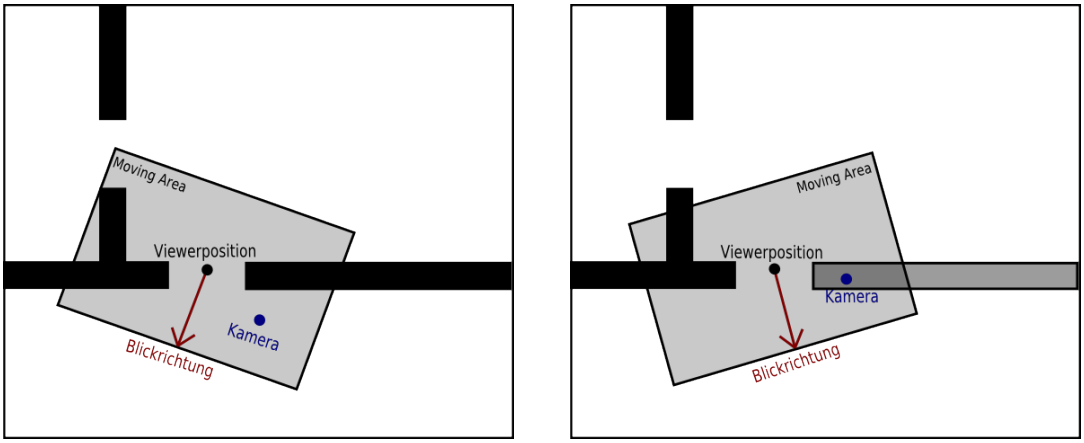
(b) Läuft der Benutzer nahe an einer Tür vorbei, soll keine Pfadkorrektur erfolgen

Abbildung 6.4: Einige Probleme beim statebasierten Ansatz

Für die Entscheidung, ob die Applikation eingreifen muss, wird, wie beim *pfadbasierten Ansatz*, der Eintrittspunkt des Betrachters in das Gate ermittelt. Dieser lässt sich aus der bisherigen Laufrichtung und der aktuellen virtuellen Position ermitteln. Liegt der Eintrittspunkt neben der Türöffnung, muss eine Pfadkorrektur durchgeführt werden. Die *Translation der virtuellen Kamera* erfolgt bei diesem Verfahren parallel zum *Screen*, um eine möglichst geringe Verschiebung der Kamera durchzuführen.

Ein Wegpunkt gibt neben der Position auch die Blickrichtung des Viewers an. Da die *Moving Area* an diesem Punkt befestigt ist, werden alle Positionen innerhalb dieses Bereiches kreisförmig um die Viewerposition bewegt. Befindet sich also die Kamera eines Benutzers nicht genau auf der Viewerposition, wird durch die Änderung der Blickrichtung auch die Kameraposition des Betrachters geändert. Treten diese Ro-

tationen vor oder innerhalb eines Gates auf, kann, wie in Abbildung 6.5 dargestellt, die Kamera eines Benutzers in eine Wand bewegt werden.



(a) Vor einer Rotation des Viewers stellt die Situation kein Problem dar

(b) Nach einer Rotation des Viewers befindet sich die Kamera des Betrachters in einer Wand

Abbildung 6.5: Bei einer Rotation des Viewers wird die Kamera des Betrachters um die Viewerposition gedreht

6.4.3 Verbesserter statebasierter Ansatz

Sowohl der pfadbasierte Ansatz als auch der statebasierte Ansatz versuchten, die Bewegung der Kamera möglichst gering zu halten. Deshalb wurden die Kameras der Betrachter immer parallel zum *Screen* verschoben. Da die translatierten Kameras aber nie auf der Viewerposition liegen, bewegen sie sich bei einer Änderung der Blickrichtung kreisförmig um die Viewerposition herum. Treten diese Rotationen vor oder innerhalb eines Gates auf, kann die Kamera in eine Wand geführt werden. Diese Bewegungen der Kameras fallen stärker aus, je weiter diese vom Rotationszentrum entfernt stehen. Da die Wege immer mittig durch ein Gate verlaufen und die Durchgänge eine bestimmte Breite aufweisen, kann ein kreisförmiger *sicherer Bereich* um die Viewerposition definiert werden (Abbildung 6.6), in dem die Bewegungen der Kameras unkritisch sind. Durch die Definition eines Kreises bleiben alle Punkte, die vor einer Rotation in diesem Bereich lagen, auch anschließend auf einer sicheren Position.

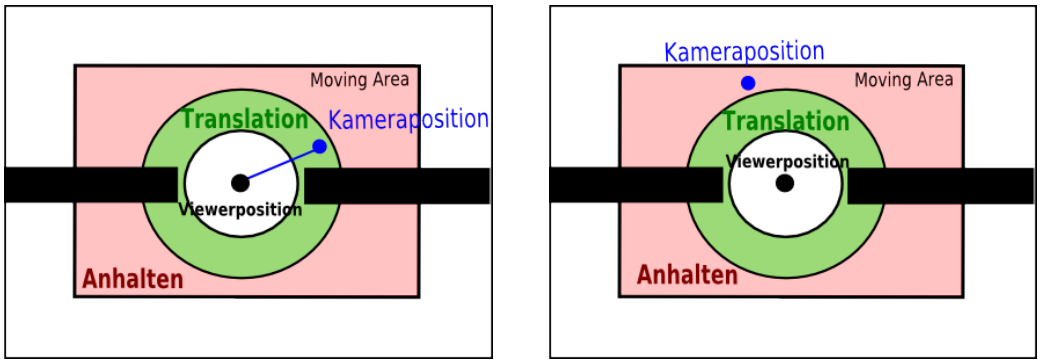
Die Größe des *sicheren Bereichs* wird so definiert, dass alle Punkte innerhalb dieser Zone durch das Gate passen. Somit werden für Kamerapositionen innerhalb dieses

Bereiches keine Pfadkorrekturen benötigt. Befindet sich ein Benutzer außerhalb dieses Kreises kann die *Translation der virtuellen Kamera* oder das *Anhalten der Tour* ausgeführt werden. Die *Translation der virtuellen Kamera* muss bei diesem Verfahren allerdings anders durchgeführt werden als bei den zuvor beschriebenen Ansätzen, da eine Bewegung der Kamera parallel zum *Screen* nicht mehr ausreichend ist, um in die sichere Zone zu gelangen. Stattdessen muss die Kamera immer auf die Viewerposition bewegt werden, wodurch Translationen in beliebige Richtungen innerhalb der *Moving Area* auftreten können (siehe Abbildung 6.6(a)). Alle anschließend auftretenden Änderungen der Blickrichtung haben eine Rotation der Kamera um ihre vertikale Achse zur Folge.

Für die *Kombination beider Verfahren* wurde ein zweiter kreisförmiger Bereich definiert, der die Anwendung der *Translation der virtuellen Kamera* und das *Anhalten der Tour* trennt. Wie in den Abbildungen 6.6(a) und 6.6(b) dargestellt, repräsentieren die Radien der Kreise Grenzwerte für den Abstand der Kamera zur Viewerposition. Der weiße Kreis repräsentiert die sichere Zone, in der keine Pfadkorrektur erforderlich ist. Um die Kamerabewegung einzugrenzen, wird die *Translation der virtuellen Kamera* angewendet, wenn die Kamera im grünen Bereich liegt. Befindet sich die Kamera im roten Bereich, wird die ausgeführte Tour angehalten.

Anders als bei dem statebasierten Ansatz wird nun nicht die Kameraposition, sondern die Viewerposition untersucht, um das Kollisionsproblem zu erkennen. Es werden Einzugsbereiche für die Gates definiert und überprüft, ob sich die Viewerposition in einem solchen Einzugsbereich befindet. Ist dies der Fall, wird davon ausgegangen, dass der Betrachter demnächst den Raum durch diese Tür verlässt. Damit die Reaktion früh genug erfolgen kann, müssen die Einzugsbereiche der Gates entsprechend groß definiert werden. Für die Entscheidung, ob eine Pfadkorrektur erforderlich ist, wird überprüft, ob die aktuelle Kameraposition im *sicheren Bereich* liegt oder nicht.

Wie bereits beschrieben, können bei diesem Verfahren Translationen in verschiedene Richtungen auftreten. Da die Durchgänge in der Galleria Borghese offenstehende Flügeltüren besitzen, ragen die Flügel der Türen in den Raum hinein. Wird nun die Kamera auf die Viewerposition bewegt, kann es passieren, dass diese Flügel durchlaufen werden. Da dies den gleichen Effekt auslöst wie das Laufen durch eine Wand, muss dieses Problem behoben werden. Deshalb wird die Translation der Kamera auf die Viewerposition überwacht, und es werden kreisförmige Bereiche um die Eckpunk-

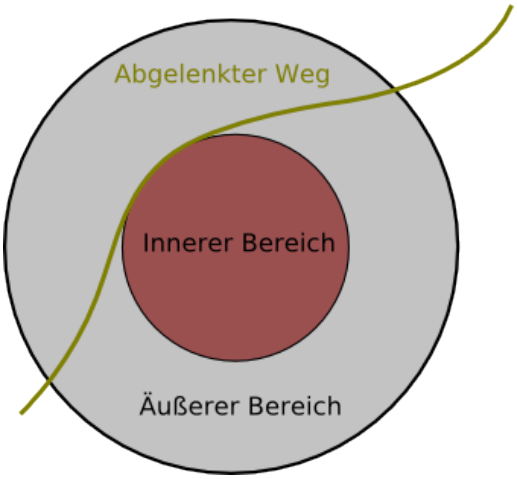


(a) Die blaue Linie repräsentiert die Translation der virtuellen Kamera auf die Viewerposition

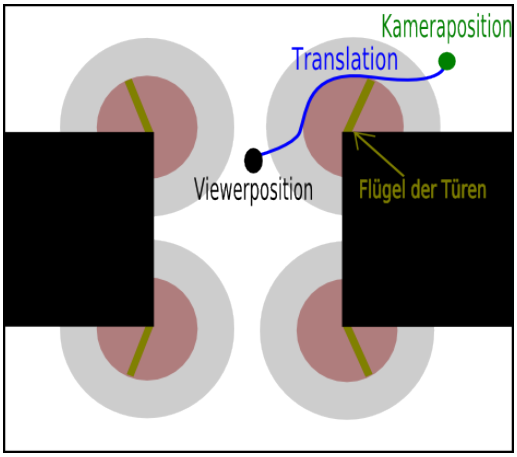
(b) Befindet sich die Kamera im roten Bereich, hält die Tour an

Abbildung 6.6: *Kombination der Verfahren* für den verbesserten statebasierten Ansatz. Die Radien der Kreise legen unterschiedliche Zonen fest:
Rot: Verringern der Geschwindigkeit
Grün: Translation der virtuellen Kamera
Weiß: sicherer Bereich

te eines Gates legt, wodurch der Benutzer in einer runden Bewegung (bumping) um die Flügel der Tür herumgeleitet wird. Diese *Bumping*-Methode (Bump = Beule) definiert Kreise, die in zwei Bereiche unterteilt sind, einen inneren Bereich (rot), und einen äußeren Bereich (grau) (siehe Abbildung 6.7(a)). Die roten Bereiche beschreiben den Bewegungsradius der Türflügel, und werden als nicht behgehbarer Bereich definiert (siehe Abbildung 6.7(b)). Somit muss die Kamera entlang der Grenzen dieses Bereichs geleitet werden. Der äußere Kreis wird benötigt, um eine angenehme Bewegung zu erzielen, da die Kamerapositionen in dieser Zone leicht nach außen gedrückt werden. Die äußeren Bereiche der *Bumping*-Kreise sollten möglichst nicht zu weit in das Innere des Gates hineinragen. Ansonsten würde die Position der Kamera innerhalb des Gates häufig durch das Bumping verändert, was zu einem unruhigen Verhalten führen würde.



(a) Der innere Bereich darf nicht betreten werden, der äußere Bereich drückt die Kamera leicht vom inneren Bereich weg, um eine sanfte Bewegung zu erhalten.



(b) Die Bumping-Kreise werden um die Ecken eines Gates gelegt. Die inneren Bereiche umschließen die Flügeltüren

Abbildung 6.7: Skizzen der Bumping-Kreise

6.5 Einbindung in die bestehende Applikationsstruktur

Die Implementierung der Pfadkorrektur erfolgt in der Klasse *Gatepassing*. Um sie in die bestehende Programmstruktur einzubinden, wurde sie zwischen den *Viewer* und den *Splitter* geschaltet (siehe Abbildung 6.8).

Durch die Aktivierung des *Splitscreens* können die Kameras durch die *Splitscreen*-Translation (siehe Abschnitt 5.5.2) parallel zum *Screen* verschoben werden. Lagen die Kameras zuvor in einem sicheren Bereich, können sie deshalb nun außerhalb dieser Zone liegen. Damit die Pfadkorrektur auch im *Splitscreen*-Modus verfügbar ist, muss die Klasse Gatepassing nach dem *Splitter* eingebunden werden.

Da der Benutzer, der den PDA kontrolliert, steuert die Geschwindigkeit einer Tour. Um dennoch das *Anhalten der Tour* zu erreichen, muss diese vom Benutzer definierte Geschwindigkeit von der Applikation überschrieben werden. Die Auswertung der Geschwindigkeit erfolgt im *Mover*, weshalb das Stoppen der Tour dem *Mover* mitgeteilt werden muss. Dadurch entsteht ein Zyklus, der als *weak connection* definiert

werden muss, damit die Berechnung der neuen Geschwindigkeit erst im nächsten Frame erfolgt.

Der Knoten für das *Bumping* wurde zwischen die Pfadkorrektur und den Viewer geschaltet (siehe Abbildung 6.8). Diese legt kreisförmige Bereiche um die Eckpunkte eines Gates und leitet den Betrachter auf einem sicheren Weg um die Flügeltüren herum. Da das Bumping für jede Kamera separat angewendet wird, werden insgesamt vier Instanzen dieses Knotens benötigt.

6.6 Ergebnisse

In diesem Kapitel wurden mehrere Strategien zur Pfadkorrektur vorgestellt: die *Translation der virtuellen Kamera*, das *Anhalten der Tour* sowie die *Kombination der Verfahren*. Bei der *Translation der virtuellen Kamera* wird der Betrachter auf einem sicheren Weg durch einen Durchgang geleitet. Da bei diesem Ansatz für beide Benutzer unterschiedliche Transformationen ausgeführt werden, wird der gemeinsame Raum und damit das kollaborative Arbeiten beeinträchtigt. Da sich dieser Effekt negativ auf die Zusammenarbeit beider Benutzer auswirken kann, wurde das *Anhalten der Tour* umgesetzt, welches den gemeinsamen Raum nicht beeinträchtigt. Da die Durchgänge in der Galleria Borghese relativ schmal sind, müssen die Betrachter sehr eng zusammenrücken, um die Tour fortzusetzen. Zudem tritt das *Anhalten der Tour* sehr häufig auf, weshalb sich die Benutzer viel bewegen müssen, um sich richtig zu positionieren, damit die Tour fortgesetz werden kann.

Eine Kombination beider Verfahren vorteilhaft, da sie die negativen Eigenschaften beider Ansätze abschwächen. Stehen die Anwender sehr weit auseinander, müssen deren Kameras stärker translatiert werden als bei einem geringeren Abstand. Aus diesem Grund wurde ein Grenzwert für das Ausmaß der Translation eingeführt. Wird dieser Grenzwert überschritten, wird die Tour angehalten. Bei kleineren Transformationen wird weiterhin die *Translation der virtuellen Kamera* vorgenommen, wodurch die daraus resultierende Beeinträchtigung des gemeinsamen Raums weniger ins Gewicht fällt. Zudem ist ein Anhalten der Tour nur noch selten erforderlich. Die negativen Eigenschaften beider Ansätze wurden somit eingeschränkt und ein guter Kompromiss erzielt.

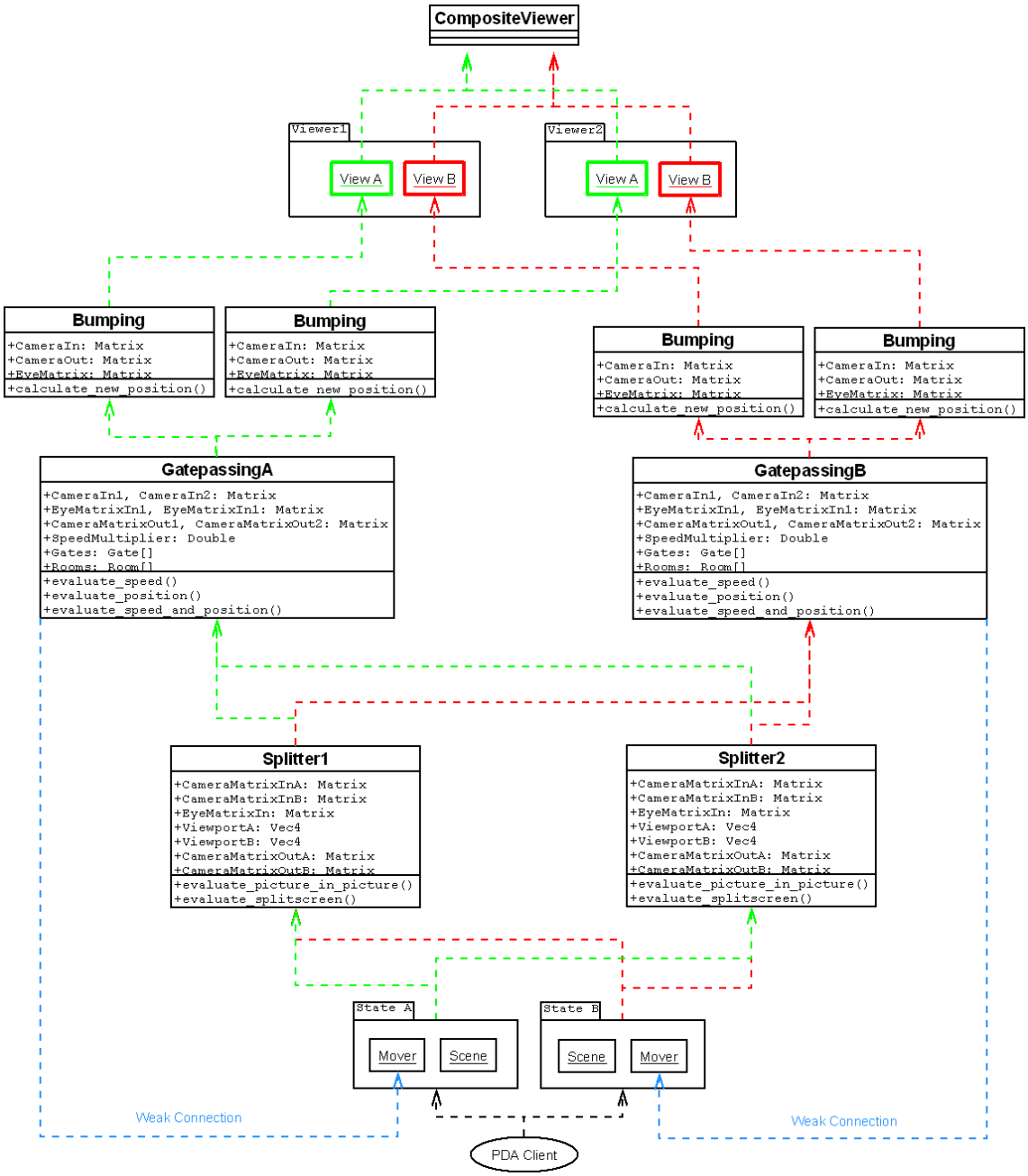


Abbildung 6.8: Applikationsstruktur mit Gatepassing und Bumping

Bei dem *Anhalten der Tour* kann es passieren, dass die Benutzer mit dem Rücken zu einem Gate stehen bleiben. In diesem Fall ist für die Benutzer nicht ersichtlich, wie sie sich positionieren müssen, um die Tour fortzusetzen. Da durch den *verbesserten statebasierten Ansatz* aber kreisförmige Bereiche um das Zentrum der *Moving Area* definiert werden, können sich die Anwender an diesem Punkt orientieren. Sollte die Tour anhalten, ohne dass ein Gate zu sehen ist, müssen sich die Benutzer in die Mitte der *Moving Area* begeben, um die Tour fortzusetzen.

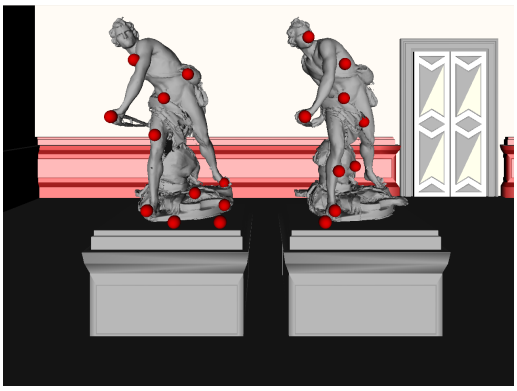
7 Empirischer Test zum gemeinsamen Raum

7.1 Aufbau

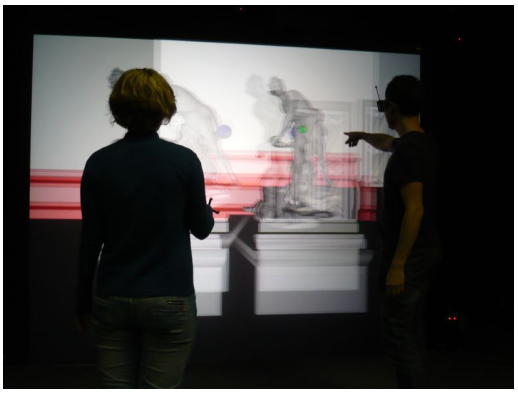
Die Visualisierung der Galleria Borghese im TwoView ermöglicht kollaboratives Arbeiten der Benutzer. Es wird ein gemeinsamer Raum erzeugt, in dem die Benutzer exakt miteinander interagieren können. Bei der *Darstellung mehrerer Ansichten* und der *Pfadkorrektur* können die Kamerapositionen beider Betrachter unterschiedlich zueinander manipuliert werden, wodurch der gemeinsame Raum und damit das kollaborative Arbeiten gestört werden kann. Einheitliche Kameratransformationen für beide Benutzer haben keinen Einfluss auf den gemeinsamen Raum und beeinträchtigen das kollaborative Arbeiten nicht. Um herauszufinden, wie stark die unterschiedlichen Transformationen den gemeinsamen Raum beeinflussen, wurde ein empirischer Test zur Bewertung dieses Effekts durchgeführt. Bei einem empirischen Test werden anhand von Beobachtungen oder Experimenten Daten gesammelt und anschließend ausgewertet.

Um die Beeinträchtigung des gemeinsamen Raums zu bewerten, wurde deshalb ein Benutzertest durchgeführt, bei dem zwei Probanden im TwoView miteinander interagieren und kollaborativ eine bestimmte Aufgabe lösten. Hierzu wurde eine Szene mit zwei Statuen aufgebaut, an denen bestimmte Positionen markiert wurden (siehe Abbildung 7.1(a)). Diese markierten Zielpositionen sollten mit einem Referenzobjekt (eine Kugel), die von einem Benutzer gesteuert werden kann, berührt werden. Dazu konnte das Referenzobjekt mit einer Wiimote®[5] in alle drei Freiheitsgrade in der virtuellen Umgebung bewegt werden.

Die Person, die das Referenzobjekt mit der Wiimote steuert (Akteur), kann die markierten Zielpositionen nicht sehen. Der Partner hingegen kann sowohl die Zielpositionen als auch die vom Akteur gesteuerte Kugel sehen. Somit kann er den Akteur durch Zeigen mit dem Finger und verbale Kommunikation anleiten, das Referenzobjekt auf die Zielpositionen zu bewegen (siehe Abbildung 7.1(b)).



(a) Testszene mit allen Zielmarkierungen. In einem Testdurchauf wurden zehn dieser Markierungen zufällig gewählt und nacheinander angezeigt



(b) Zwei Benutzer bei einem Testdurchlauf

Abbildung 7.1: Aufbau der Testszene; Benutzer in Aktion bei einem Testdurchlauf

In einer Testsitzung wurden verschiedene Testläufe durchgeführt. Um eine Bewertung der Beeinträchtigung des gemeinsamen Raums zu erhalten, wurden die Kameras der Benutzer um verschiedene Werte rotiert und translatiert. Dadurch ergaben sich mehrere Testdurchläufe mit unterschiedlichen Transformationsarten, die einen unterschiedlichen Einfluss auf das kollaborative Arbeiten hatten.

Um zu testen, inwieweit sich die Stärke einer Transformation auf den gemeinsamen Raum auswirkt, wurden für die Kameras sowohl große als auch kleine Bewegungen ausgeführt. Ein Testdurchlauf wurde ohne Manipulation des gemeinsamen Raums durchgeführt, damit ein Bezug der Ergebnisse auf das normale Arbeiten im TwoView hergestellt werden kann (Referenzdurchlauf). Eine Testsitzung besteht somit aus insgesamt fünf Durchgängen. Es wird das *normale* Arbeiten im TwoView getestet, sowie die Beeinträchtigung des gemeinsamen Raums durch eine *geringe Rotation*, eine *starke Rotation*, eine *geringe Translation* und eine *starke Translation*. Die Kameras wurden dabei wie in Tabelle 7.1 aufgelistet unterschiedlich transformiert.

Bezeichnung	**Transformation Benutzer 1 (Akteur)**	**Transformation Benutzer 2**
geringe Rotation	5 Grad	-5 Grad
starke Rotation	15 Grad	-15 Grad
geringe Translation	20 cm nach links	20 cm nach rechts
starke Translation	80 cm nach links	80 cm nach rechts

Tabelle 7.1: Auflistung der unterschiedlichen Kameratransformationen für beide Benutzer

Für jeden Testdurchlauf wurden die Zeiten für das Lösen der Aufgabe gestoppt, und später miteinander verglichen. Jeder Testdurchlauf wurde dabei nur einmal pro Person ausgeführt. Der Vergleich der Zeiten soll einen Hinweis darauf geben, bei welchem der fünf Durchläufe das kollaborative Arbeiten am Besten funktioniert hat.

Da die Probanden bei späteren Testdurchläufen mehr Erfahrung mit dem Umgang der Steuerung des Referenzobjektes hatten, und zudem nur eine begrenzte Anzahl von Zielmarkierungen vorhanden war, wurde die Reihenfolge der fünf Testdurchläufe für jede Testsitzung ausgelost.

Für die Testszene wurden zwei Skulpturen in einem Raum platziert, an denen die Markierungen befestigt wurden. Um das Arbeiten von Kunsthistorikern zu simulieren, wurde absichtlich auf die Verwendung abstrakter Objekte, wie zum Beispiel einem Würfel, verzichtet. Stattdessen wurden zwei Skulpturen verwendet, und die Zielpositionen an markanten Stellen befestigt. Dadurch wird das kollaborative Arbeiten nicht nur auf die Gestikulation reduziert, sondern auch die verbale Kommunikation berücksichtigt. Einige Zielpositionen wurden deshalb an weniger markanten Stellen positioniert. So wurden einige an einem Knie oder im Gesicht befestigt, andere hingegen auf dem Sockel oder einige Zentimeter vor dem Bauch der Skulptur. Es wurde immer nur eine Zielmarkierung auf einmal angezeigt. Nachdem die Referenzkugel erfolgreich auf die Zielposition bewegt wurde, erschien automatisch die nächste Markierung.

Da für den Test nur 20 Zielpositionen zur Verfügung standen, wurden pro Durchgang zehn zufällige Markierungen gewählt. Nach dem Test mussten die Probanden einen Fragebogen ausfüllen, der eine persönliche Einschätzung des kollaborativen Arbeitens, bezogen auf die unterschiedliche Beeinträchtigung des gemeinsamen Raums, abfragen sollte.

7.2 Ergebnisse

Insgesamt nahmen zehn Personen an dem Test teil. Um eine empirische Messbarkeit der Beinträchtigung des gemeinsamen Raums zu erhalten, wurde die Zeit für die Lösung der Aufgabe bei jedem Durchgang gestoppt. Die subjektive Meinung der Probanden wurde durch einen Fragebogen erfasst.

Auswertung der Zeitmessung

Die Ergebnisse der Zeitmessung sind in Abbildung 7.2 durch ein Boxplot-Diagramm dargestellt. Ein Boxplot-Diagramm (auch Box-Whisker-Plot) bietet eine Möglichkeit, numerische Daten, ohne eine Annahme der zugrundeliegenden statistischen Verteilung, darzustellen [10]. Dazu werden die Daten durch fünf Punkte zusammengefasst, der Median, zwei Quartile sowie zwei Whisker. Der Median wird durch einen schwarzen Balken dargestellt. Die blau markierte Box bezeichnet die beiden Quartile, die insgesamt 50% der Daten umfasst (jedes Quartil umfasst 25%). Die Whisker werden durch gestrichelte horizontale Linien markiert, deren Länge maximal das 1.5-fache der Quartile entspricht. Punkte, die außerhalb des Whiskers liegen, werden als Ausreißer behandelt und durch ein „o“ oder ein „+“ gekennzeichnet.

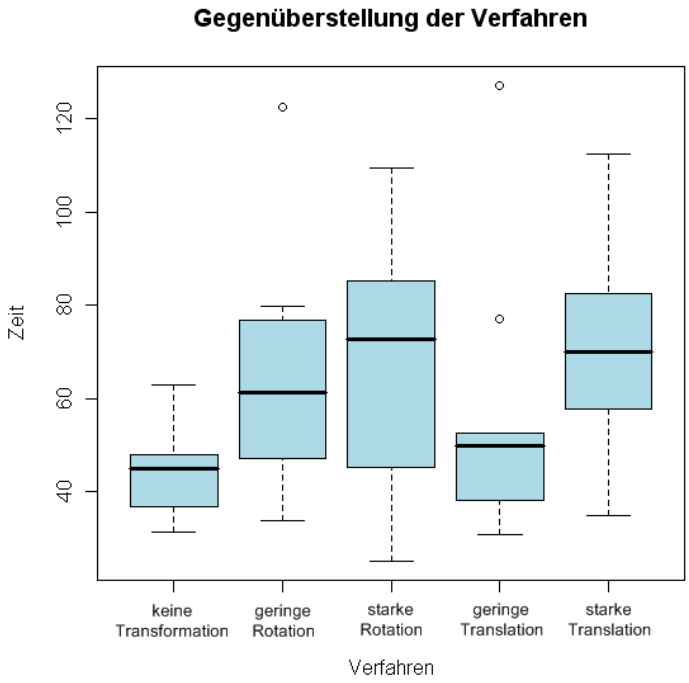


Abbildung 7.2: Dieser Boxplot stellt die gemessenen Zeiten der unterschiedlichen Testdurchläufe dar

Tendentiell hat das Arbeiten im normalen Modus, bei dem keine Transformation der Kameras durchgeführt wurde, am konstantesten abgeschnitten. Die gestellte Aufgabe konnte bei diesem Testlauf im Durchschnitt am schnellsten gelöst werden. Eine statistisch signifikante Unterscheidung der Testläufe durch eine Varianzanalyse konnte nicht nachgewiesen werden. Somit kann keine Bewertung erfolgen, welche Transformation der Kameras sich am stärksten bzw. am schwächsten auf das kollaborative Arbeiten ausgewirkt hat. Durch Übung bei den Probanden und das Wiedererkennen von Zielmarkierungen konnten auch bei einer starken Beeinträchtigung des gemeinsamen Raums gute Zeiten erzielt werden. Das lässt sich damit erklären, dass sich die Tester während der Durchläufe ein Vokabular für die Markierungen angeeignet haben, mit denen sie diese schneller identifizieren konnten. Somit konnten viele Zielpositionen auch ohne das Zeigen mit dem Finger schnell erkannt und abgearbeitet werden. Auf der anderen Seite kam es bei manchen Testdurchläufen zu Gesprächen der Probanden untereinander. Dadurch wurden langsamere Zeiten erzielt, die die Ausreißer bei der geringen Translation erklären. Da sich statistisch keine eindeutige Abgrenzung der Verfahren ermitteln ließ, sollte eine weiterführende Analyse zur Beeinträchtigung des gemeinsamen Raums durchgeführt werden.

Auswertung des Fragebogens

Durch den Fragebogen wurde die Meinung der Probanden zu den verschiedenen Testdurchläufen abgefragt. Am angenehmsten wurde der Testdurchlauf im normalen Modus empfunden. In diesen Durchgängen wurden die verbale Kommunikation und das Zeigen mit dem Finger als gleichwertige Hilfsmittel eingesetzt.

Auch in den Durchläufen mit geringer Beeinträchtigung des gemeinsamen Raums wurden das Reden und die Gestikulation als hilfreich empfunden. Die Transformationen der Kameras wurden tendentiell als wenig hinderlich empfunden, wobei sich keine Unterscheidung zwischen der Rotation und der Translation herausgestellt hat.

Bei einer starke ausgeprägten Transformation der Kameras wird die Zusammenarbeit der Probanden stärker beeinträchtigt als bei geringe oder garkeiner Transformation. Das Zeigen mit dem Finger wurde hierbei gar nicht mehr eingesetzt und von den meisten Probanden sogar als hinderlich empfunden. In diesen Durchgängen haben sich die Probanden überwiegend verbal verständigt, wodurch die Aufgabe dennoch sehr gut bewältigt werden konnte. Die Bewertung der Behinderung der

Kameratransformation bei der Lösung der Aufgabe fiel weitestgehend neutral aus. Die neutrale Einstellung setzt sich aus zwei Extremen zusammen. Die eine Hälfte der Befragten empfand die Transformation als sehr hinderlich, und die andere als überhaupt nicht hinderlich. Ein Unterschied zwischen der Rotation und Translation konnte nicht festgestellt werden.

Insgesamt funktionierte das Arbeiten im normalen Modus am besten, und wurde von den Benutzern als angenehm empfunden. Die Unterschiede zwischen einer geringen und einer starken Beeinträchtigung lassen sich statistisch nicht nachweisen. Aufgrund des Fragebogens stellten sich geringe Transformationen jedoch als weniger störend und angenehmer heraus als starke Transformationen.

8 Fazit

In dieser Arbeit wurden Lösungen zu zwei unterschiedlichen Aspekten der kollaborativen Arbeit in gemeinsam genutzten virtuellen Umgebungen behandelt. Ein Hauptaspekt dieser Arbeit ist die *Darstellung mehrerer Ansichten* in einem *Multi-Viewer Display*. Der zweite Hauptaspekt dieser Arbeit besteht darin, ein Kollisionsproblem zu beheben, welches auftreten kann, wenn sich mehrere Benutzer in einer gemeinsamen virtuellen Umgebung bewegen und zusammen einen schmalen Durchgang passieren. Da es hierbei vorkommen kann, dass ein Benutzer durch eine Wand geht, wurden verschiedene Lösungsansätze einer *Pfadkorrektur* vorgestellt, um dieses Problem zu beheben. Da aufgrund der Umsetzung beider oben genannten Aspekte die virtuellen Kameras der Benutzer unterschiedlich manipuliert werden, wird der gemeinsame Raum zum kollaborativen Arbeiten beeinflusst. Deshalb wurde zuletzt ein empirischer Test des gemeinsamen Raums durchgeführt, um eine Bewertung dieses Effekts zu erhalten.

Die Grundlage für diese Arbeit stellt die Diplomarbeit *„Kollaborative Interaktion in virtuellen Umgebungen im kunsthistorischen Kontext“* von Tim Jansen [15] dar. Das Ergebnis dieser Arbeit ist eine Applikation, die es ermöglicht, in einer virtuellen Umgebung exakte Analysen von Objekten durchzuführen. Die *Darstellung mehrerer Ansichten* und die *Pfadkorrektur*, wurden in die bestehende Applikation integriert werden.

Für die Visualisierung verschiedener Ansichten können unterschiedliche Ansätze verwendet werden (*Switchen*, *Picture in Picture* oder *Splitscreen*). Am sinnvollsten für den direkten Vergleich zweier Bilder hat sich der *Splitscreen* herausgestellt, da dieser die klassische Arbeitsweise von Kunsthistorikern widerspiegelt (siehe Abschnitt 5.5). Der *Splitscreen*-Modus liefert zwei gleichwertige Bilder unterschiedlicher Ansichten, die verzerrungsfrei dargestellt werden.

Da durch die Aktivierung des *Splitscreens* der Sichtkegel geteilt wird, kann das betrachtete Objekt aus dem Fokus des Benutzers gelangen. Deshalb müssen die virtuellen Kameras beider Betrachter neu auf das Objekt ausgerichtet werden. Hierzu kann die Kamera rotiert oder translatiert werden. Durch das Berücksichtigen der realen Betrachterpositionen im TwoView können sich dabei unterschiedliche Werte der Transformation für beide Benutzer ergeben. Dadurch kann der gemeinsame Raum und damit das kollaborative Arbeiten beeinträchtigt werden. Deshalb wurden neben dynamischen Verfahren, die die getrackte Position des Betrachters berücksichtigen, auch statische Varianten implementiert, die für die Kameras beider Benutzer identische Transformationen ausführen. Durch die statischen Methoden kann die Kamera allerdings nicht so genau auf das Objekt ausgerichtet werden wie bei den dynamischen Varianten. Eine Möglichkeit zur Erweiterung stellt die Anpassung der Kameratransformation an die getrackte Blickrichtung der Benutzer dar. Dadurch kann die Ausrichtung des Sichtkegels noch exakter erfolgen. Schauen beide Betrachter in unterschiedliche Richtungen, wird bei diesem Vorgehen aber ebenfalls der gemeinsame Raum beeinträchtigt.

Den zweiten Aspekt der Arbeit stellte das Kollisionsproblem bei schmalen Durchgängen dar. Um dieses Problem zu beheben, können zwei Arten der Pfadkorrektur angewendet werden. Entweder wird der Benutzer auf einem sicheren Weg durch diesen Durchgang geführt, oder die Tour wird angehalten. Beide Methoden besitzen Vor- als auch Nachteile. Eine Kombination beider Ansätze schwächt ihre negativen Eigenschaften ab und führt zu einem angenehmeren Verhalten. Aus diesem Grund stellt die Kombination beider Verfahren einen sinnvollen und anwenderfreundlichen Kompromiss dar.

Bei der Translation der Kamera auf die Viewerposition können die in den Raum ragenden Flügeltüren eines Gates durchlaufen werden. Deshalb wurde eine *Bumping*-Methode umgesetzt, die die Kamera um die Flügeltüren der Gates in der Galleria Borghese herumführt. Durch das Anhalten der Tour kann es vorkommen, dass die Benutzer mit dem Rücken zu einem Gate stehen bleiben. Da in dieser Situation nicht ersichtlich ist, wie sich die Benutzer positionieren müssen um die Tour fortzusetzen, wäre die Realisierung virtueller Pfeile denkbar, die die Benutzer unterstützen.

Für das Erkennen der problematischen Situationen wird lediglich der Status einer Ansicht ausgewertet. Dadurch ist die Pfadkorrektur auch bei einer freien Begehung der virtuellen Umgebung verfügbar. Da die *Bumping*-Methode bisher nur auf die

Gates beschränkt ist, könnte sie für diesen Zweck auf die Wände der Räume erweitert werden. Somit wäre das Kollisionsproblem bei einer freien Navigation in der virtuellen Umgebung behoben.

Um eine Bewertung der Beeinträchtigung des gemeinsamen Raums auf das kollaborative Arbeiten zu erhalten, wurde ein Benutzertest durchgeführt. Hierbei wurde der gemeinsame Raum gezielt auf verschiedene Arten beeinflusst und mit dem Arbeiten im normalen Modus, ohne Beeinträchtigung des gemeinsamen Raums, verglichen. Die Auswertung dieses Tests hat ergeben, dass sich die unterschiedlichen Kameratransformationen beider Benutzer negativ auf das kollaborative Arbeiten auswirken können. Da größere Rotationen und Translationen als störender empfunden werden, ist es sinnvoll, die Unterschiede der Kameratransformationen beider Benutzer zu minimieren. Da keine statistische Abgrenzung der Verfahren möglich war, könnte ein erweiterter Benutzertest mit einer größeren Anzahl an Zielmarkierungen durchgeführt werden. Dadurch kann ausgeschlossen werden, dass sich die Probanden an bestimmte Zielmarkierungen erinnern und diese schneller identifizieren können.

Für das kollaborative Arbeiten in einer virtuellen Umgebung hat sich die geringe Beeinträchtigung des gemeinsamen Raums als angenehmer herausgestellt als eine stärkere. Aus diesem Grund sollten die Unterschiede bei den Kameratransformationen beider Benutzer sowohl im *Splitscreen*-Modus, als auch bei der *Pfadkorrektur*, minimiert werden. Für den *Splitscreen*-Modus sollte deshalb eher eine statische Variante gewählt werden. Da sich die Benutzer meist vor der betrachteten Bildhälfte positionieren, stehen sie relativ eng beieinander, wodurch die Beeinträchtigung des gemeinsamen Raums bei einer dynamischen Variante eher gering ausfällt. Für die Pfadkorrektur stellt die Kombination der *Translation der virtuellen Kamera* und dem *Anhalten der Tour* die sinnvollste Lösung dar.

Abkürzungsverzeichnis

Moving Area	Die Moving Area ist eine 3×2.4 Meter große frei begehbare Fläche im TwoView.
Pfad	Ein Pfad stellt einen Benutzerdefinierter Weg dar, der mit Hilfe des PDA-Clients erzeugt wurde.
Screen	Der Screen stellt eine virtuelle Repräsentation der Projektionsleinwand dar.
Tour	Eine Tour ist ein vom PDA-Client Vorgegebener Weg.
View	Mit View wird eine Ansicht bezeichnet. Eine Ansicht besteht aus einer Szene und ein Weg der begangen wird.
View Frustum	Als View Frustum wird der Sichtkegel einer Kamera bezeichnet.
Viewer	Die Viewerposition stellt eine virtuelle Referenzposition für beide Betrachter im TwoView dar. Anhand dieser Position kann die virtuelle Position der Betrachter bestimmt werden. Die Viewerposition ist für beide Betrachter identisch. Zudem wird die Moving Area an dieser Position befestigt.
Weg	Der Begriff Weg wird als Überbegriff für Tour und Pfad verwendet.

A Auswertung des Fragebogens

Fragebogen zum Usertest

Normaler Modus:

1. **Wie gut hat allgemein das Zusammenarbeiten mit dem Partner in diesem Modus funktioniert?**

schlecht *gut*

2. **Wie hilfreich war das Zeigen des Partners?**

Wenig hilfreich *sehr hilfreich*

3. **Wie hilfreich war die Kommunikation mit dem Partner?**

Wenig hilfreich *sehr hilfreich*

4. **Wie wichtig waren markante Punkte, an dem sich die Zielpositionen befanden?**

Unwichtig *wichtig*

Geringe Transformation:

5. **Wie gut hat allgemein das Zusammenarbeiten mit dem Partner in diesem Modus funktioniert?**

schlecht *gut*

6. **Wie hilfreich war das Zeigen des Partners?**

Wenig hilfreich *sehr hilfreich*

7. **Wie hilfreich war die Kommunikation mit dem Partner?**

Wenig hilfreich *sehr hilfreich*

8. **Wie wichtig waren markante Punkte, an dem sich die Zielpositionen befanden?**

Unwichtig *wichtig*

9. **Wie hinderlich war die Rotation der Kamera?**

wenig hinderlich *sehr hinderlich*

10. **Wie hinderlich war die Translation der Kamera?**

wenig hinderlich *sehr hinderlich*

Starke Transformation:

11. **Wie gut hat allgemein das Zusammenarbeiten mit dem Partner in diesem Modus funktioniert?**

schlecht *gut*

12. Wie hilfreich war das Zeigen des Partners?

13. Wie hilfreich war die Kommunikation mit dem Partner?

14. Wie wichtig waren markante Punkte, an dem sich die Zielpositionen befanden?

15. Wie hinderlich war die Rotation der Kamera?

16. Wie hinderlich war die Translation der Kamera?

Abbildungsverzeichnis

2.1 Beispiele für stereoskopische Displaytechniken 13
2.2 Two-User Protein Interactive Theater 16
2.3 IllusionHole 17
2.4 Aufbau des TwoView [15] 19
2.5 Projektor-Konfigurationen für Shuttered Projectors [13] 20

3.1 Aufbau von AVANGO NG 21
3.2 Zyklus mit einer *weak-connection* 23
3.3 Skulpturengruppe aus der Galleria Borghese: David, Pluto und Proserpina, Äneas und Anchises, Apollo und Daphne [15] 26
3.4 *Hauptmenü* und *Positions* des PDA-Clients 28
3.5 Menü: *Tourguide* und *Path* 30

4.1 Applikationsstruktur nach der Portierung auf AVANGO NG. Der *Mover* ist blau markiert 33
4.2 Viewing Setup einer Ansicht 35
4.3 Einbindung mehrerer Ansichten in die vorhandene Applikationsstruktur 36
4.4 Skizze der *Preview* mit den dargestellten Komponenten 37
4.5 Screenshot mit aktivierter *Preview* 38

5.1 *Picture in Picture*-Modus 43
5.2 Teilung des Sichtkegels in eine linke Hälfte (grün) und eine rechte Hälfte (rot) 44
5.3 Screenshots des *Splitscreens* ohne Transformation der Sichtkegel. Die dargestellten Objekte können durch das Verkleinern des Sichtkegels abgeschnitten werden 45
5.4 Screenshot einer Vollbildansicht und des *Splitscreens* mit unterschiedlichen Transformationen der Sichtkegel 46

5.5 Unterschiedliche Annahmen für die Blickrichtung des Betrachters. Die Blaue Fläche repräsentiert den Sichtkegel 48
5.6 Offset Rotation 49
5.7 Übereinanderlegen der Winkelhalbierenden 50
5.8 Übereinanderlegen der Bildmitten 51
5.9 Bestimmung der Translation 52
5.10 Bei der approximativen Translation wird die Translation für die Viewerposition ermittelt und auf die Kamera übertragen 53
5.11 Schneiden der Winkelhalbierenden im Fixpunkt 54
5.12 Schneiden der Bildmitten im Fixpunkt 55
5.13 Applikationsstruktur mit integriertem *Splitter* 57

6.1 Steht der Betrachter (blau) zu weit vom eigentlichen Weg (gelb) entfernt, wird er durch eine Wand geführt 63
6.2 Da die Kameraposition (blau) mit einer Wand kollidiert, wird die Kamera auf dem grün markierten Weg durch das Gate geführt 65
6.3 Die Grenzwerte legen unterschiedliche Zonen fest:
Rot: Verringern der Geschwindigkeit
Grün: Translation der virtuellen Kamera
Weiß: keine Pfadkorrektur 67
6.4 Einige Probleme beim statebasierten Ansatz 70
6.5 Bei einer Rotation des Viewers wird die Kamera des Betrachters um die Viewerposition gedreht 71
6.6 *Kombination der Verfahren* für den verbesserten statebasierten Ansatz. Die Radien der Kreise legen unterschiedliche Zonen fest:
Rot: Verringern der Geschwindigkeit
Grün: Translation der virtuellen Kamera
Weiß: sicherer Bereich 73
6.7 Skizzen der Bumping-Kreise 74
6.8 Applikationsstruktur mit Gatepassing und Bumping 76

7.1 Aufbau der Testszene; Benutzer in Aktion bei einem Testdurchlauf . . 79
7.2 Dieser Boxplot stellt die gemessenen Zeiten der unterschiedlichen Testdurchläufe dar 81

Literaturverzeichnis

[1] OpenGL Performer. http://www.sgi.com/products/software/performer. Letzter Zugriff: 11.10.2008.

[2] OpenSceneGraph. http://www.openscenegraph.org. Letzter Zugriff: 11.10.2008.

[3] Python. http://www.python.org. Letzter Zugriff: 11.10.2008.

[4] Referenzbild für Anaglyphenverfahren. http://commons.wikimedia.org/wiki/Image:Comet_Wild2_Anaglyph.jpg. Letzter Zugriff: 11.10.2008.

[5] Wii. http://de.wii.com. Letzter Zugriff: 11.10.2008.

[6] EVE - The Encyclopedia of Virtual Environments. http://www.hitl.washington.edu/scivw/EVE/, 1993. Letzter Zugriff: 11.10.2008.

[7] Maneesh Agrawala, Andrew C. Beers, Ian McDowall, Bernd Fröhlich, Mark Bolas, and Pat Hanrahan. The two-user Responsive Workbench: support for collaboration through individual views of a shared space. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 327–332, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

[8] Richard Bartle. *Designing Virtual Worlds*. New Riders Games, 2003.

[9] Manfred Bogen and Roland Kuck. Reconstructing and Presenting Bernini's Borghese Sculptures. http://www.archimuse.com/mw2005/papers/bogen/bogen.html, 2005. Letzter Zugriff: 11.10.2008.

[10] David d'Angelo. Interaction Techniques for Assembly based Modeling in Virtual Environments. Master thesis, Fachhochschule Bonn-Rhein-Sieg, 2008.

[11] Abteilung Virtual Environments. TwoView: Neues Display-System für Arbeiten im virtuellen Raum. http://www.iais.fraunhofer.de/3916.html, 2005. Letzter Zugriff: 11.10.2008.

[12] Loris Fauster. Stereoscopic Techniques in Computer Graphics. 2007.

[13] Bernd Fröhlich, Roland Blach, Oliver Stefani, Jan Hochstrate, Jörg Hoffmann, Karsten Klüger, and Matthias Bues. Implementing Multi-Viewer Stereo Displays. In *WSCG (Full Papers)*, pages 139–146, 2005.

[14] L. Hill and A. Jacobs. 3-D Liquid Crystal Displays and Their Applications. *Proceedings of the IEEE*, 94(3):575–590, March 2006.

[15] Tim Jansen. Kollaborative Interaktion in virtuellen Umgebungen im kunsthistorischen Kontext. Diplomarbeit, Georg-Simon-Ohm-Fachhochschule Nürnberg, September 2007.

[16] R. Kelsey, W. Clinger, and J. Rees. The Revised5 Report on the Algorithmic Language Scheme. *Higher-Order and Symbolic Computation*, 1998.

[17] Yoshifumi Kitamura, Takashige Konishi, Sumihiko Yamamoto, and Fumio Kishino. Interactive stereoscopic display for three or more users. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 231–240, New York, NY, USA, 2001. ACM.

[18] Yoshifumi Kitamura, Takashi Nakashima, Keisuke Tanaka, and Takeshi Johkoh. The IllusionHole for Medical Applications. In William R. Sherman, Ming Lin, and Anthony Steed, editors, *VR*, pages 231–234. IEEE Computer Society, 2007.

[19] W. Krüger and B. Frhlich. The Responsive Workbench [virtual work environment]. *Computer Graphics and Applications, IEEE*, 14(3):12–15, May 1994.

[20] Roland Kuck, Jürgen Wind, Kai Riege, and Manfred Bogen. Improving the AVANGO VR/AR Framework - Lessons Learned. In *Virtuelle und Erweiterte Realität: 5. Workshop der GI-Fachgruppe VR/AR*, pages 209–220, 2008.

[21] David F. McAllister. *Stereo and 3D Display Technologies*. Wiley Encyclopedia of Imaging, Jan 2002.

[22] Kai Riege. Mehrbenutzerinteraktion in einem projektionsbasierten Two-Viewer System. Diplomarbeit, Bauhaus-Universität Weimar, März 2006.

[23] Dieter Schmalstieg, Anton Fuhrmann, Gerd Hesina, Zsolt Szalavári, L. Miguel Encarnaçäo, Michael Gervautz, and Werner Purgathofer. The studierstube augmented reality project. *Presence: Teleoper. Virtual Environ.*, 11(1):33–54, 2002.

[24] Kai Schrader. *Entwurf und Realisierung eines Ergonomie-Mock-Ups unter Verwendung von Methoden der virtuellen Realität*. PhD thesis, Technische Universität Braunschweig, 2003.

[25] I. Sexton and P. Surman. Stereoscopic and autostereoscopic display systems. *Signal Processing Magazine, IEEE*, 16(3):85–99, May 1999.

[26] Universität Siegen. Virtualisierung von Skulptur: Rekonstruktion, Präsentation, Installation. http://www.fk615.uni-siegen.de/de/teilprojekt.php?projekt=B7. Letzter Zugriff: 11.10.2008.

[27] Andreas Simon and Sascha Scholz. Multi-Viewpoint Images for Multi-User Interaction. In *VR '05: Proceedings of the 2005 IEEE Conference 2005 on Virtual Reality*, pages 107–113, Washington, DC, USA, 2005. IEEE Computer Society.

[28] Henrik Tramberend. Avocado: A Distributed Virtual Reality Framework. 1999.

[29] UNC. Computational Geometry in Molecular Biology. 2004.

[30] VisualEyes. http://www.evl.uic.edu/index2.php. Letzter Zugriff: 11.10.2008.