

Einsatz von Eye-Tracking zur Interaktion in Mixed Reality Umgebungen

Diplomarbeit

im Studiengang



mit dem Schwerpunkt
Informationstechnologie



Fachhochschule Offenburg
University of Applied Sciences

am **Fraunhofer**



Institut
Angewandte
Informationstechnik

Februar 2005

von

Markus Sareika

Betreuer

Prof. Dr. Tom Rüdibusch (FHO)

Dipl.-Inform. Michael Wittkämper (Fraunhofer FIT)

Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die vorliegende Arbeit wurde am Fraunhofer Institut für Angewandte Informationstechnik FIT unter wissenschaftlicher Betreuung angefertigt.

Bonn, den 16.02.2005

M.Sareika

Abstract

Mixed Reality (MR) technology synchronizes information between a digitally existing virtual space and the human cognition. It also enables interaction between cognition and virtual space. The transmission from this space to the human is mainly done by addressing the visual sense. Interaction is usually done by using tactile or auditory channels. This is not appropriate for all interaction situations. Another reasonable transmission channel from the human to the virtual space increases the grade of immersion and enables further possibilities for applications.

Integrating eye tracking into MR enables the bidirectional exchange of information. This also means interaction through the visual sense. To be able to reasonably use and evaluate this new form of interaction while taking account for intuitive natural and familiar human actions in connection with tactile and auditory forms of interaction in the given research environment, an appropriate eye tracking system was selected and acquired as well as integrated into a MR framework. For the implementation of the user interface an interaction model was created which served as a base for the „Eye Interact“ application. The application enables the stereoscopic calibration and recalibration of the eye tracking with respect to cognition, allows a generic connection with VRML scenes for interaction, has special debugging features and offers a base for evaluating multimodal interactions. Recalibrating is an often used feature to level out varying offsets which occur between the user and the HMD (head mounted device). The claim for a full automatic eye tracker calibration was a result of the evaluation. The evaluation also showed that gaze interaction is best used for selecting in progressive multimodal user interfaces.

Kurzfassung

Die Mixed Reality (MR) Technologie synchronisiert Information aus einem virtuellen, digital existierenden Raum mit der Realität für die Wahrnehmung des Menschen und ermöglicht die Interaktion. Die Übertragung vom virtuellen Raum zum Menschen erfolgt hauptsächlich über den visuellen Sinn. Die Interaktion geschieht gewöhnlich taktil oder auditiv und ist nicht für alle Anwendungssituationen geeignet. Ein weiterer sinnvoller Übertragungskanal in einer MR-Anwendung führt zu einem höheren Immersionsgrad und weiteren Einsatzmöglichkeiten.

Die Integration von Eye-Tracking in MR ermöglicht den bidirektionalen Informationsaustausch, die Interaktion über den visuellen Sinn. Um die neue Interaktionsform unter Berücksichtigung von intuitiven, natürlichen und gewohnten Handlungen des Menschen in Verbindung mit taktilen und auditiven Interaktionsformen im gegebenen Forschungsumfeld sinnvoll einsetzen und evaluieren zu können, wurde ein geeignetes Eye-Tracking System ausgewählt und beschafft sowie in ein MR Framework integriert. Für die Implementierung der Anwendung wurde ein Interaktionsmodell als Basis hergeleitet, aus der die „Eye Interact“ Anwendung entstand. Sie berücksichtigt die Wahrnehmung für die stereoskopische Kalibrierung und Rekalibrierung des Eye-Trackers, ermöglicht die generische Anbindung von VRML Szenen zur Interaktion, besitzt spezielle Möglichkeiten zur Fehlersuche, ist applikationsübergreifend einsetzbar und bietet eine Basis zur Evaluation multimodaler Interaktionsmöglichkeiten. Die Rekalibrierung ist häufig notwendig, um schwankende Offsets zwischen Anwender und HMD (Head Mounted Device) auszugleichen. Die Evaluation der Blickinteraktion ergab die Forderung nach einer vollautomatischen Eye-Tracker Kalibrierung und verdeutlichte, dass der Blick in progressiven, multimodalen Benutzerschnittstellen am Besten zur Selektion eingesetzt wird.

Inhaltsverzeichnis

Abbildungsverzeichnis.....	ix
Formelverzeichnis.....	xi
1 Einleitung.....	1
1.1 Mixed Reality.....	2
1.2 Wahrnehmung.....	6
1.3 Mixed Reality Anwendungen.....	8
1.4 Motivation und Zielsetzung.....	11
2 Eye-Tracking.....	15
2.1 Visuelles System des Menschen.....	16
2.1.1 Anatomie und Physiologie.....	16
2.1.2 Augenbewegungen.....	20
2.2 Entwicklung des Eye-Tracking.....	23
2.3 Eye-Tracking Technologien.....	23
3 MR Technologie und Blickinteraktion.....	27
3.1 Mixed Reality Technologie.....	27
3.2 Augenbasierte Interaktionstechniken.....	30
3.3 Mixed Reality Interaktion.....	35
4 Auswahl und Integration des Eye-Tracking.....	37
4.1 Mixed Reality geeignete Eye-Tracking Systeme.....	37
4.1.1 Anforderungen zur Mixed Reality Integration.....	37
4.1.2 Auswahl eines geeigneten Systems.....	38
4.2 Entwicklung der HMD Befestigung.....	41
4.3 Framework-Integration des Eye-Trackers.....	44
4.3.1 CORBA und das Morgan Framework.....	44
4.3.2 Zeitnähe und Synchronisation.....	48
4.3.3 Geometrie.....	50
4.3.4 Eye-Tracker.....	55
4.3.5 Tracker Combiner.....	56
4.3.6 Eye-Manager.....	58
4.3.7 Fadenkreuzanzeige im Viewer.....	59

5 Benutzerschnittstelle und Blickinteraktion.....	61
5.1 Benutzerschnittstellen Design.....	62
5.2 Implementierung.....	64
5.3 Eye-Tracker Kalibrierung.....	67
5.4 Applikationsübergreifender Einsatz.....	70
6 Evaluation.....	73
6.1 Usability Szenarien.....	73
6.2 Durchführung der Benutzertests.....	76
6.3 Selektion statischer Objekte.....	77
6.4 Selektion dynamischer Objekte.....	79
6.5 Interaktion durch Bewegung.....	80
6.6 Resümee der Evaluation.....	81
7 Zusammenfassung und Ausblick.....	83
7.1 Zusammenfassung.....	83
7.2 Ausblick.....	85

Abbildungsverzeichnis

Abbildung 1: Mixed Reality.....	1
Abbildung 2: Mixed Reality Kontinuum nach [Mil94].....	5
Abbildung 3: Das Wahrnehmungssystem [url22].....	6
Abbildung 4: Arthur Project [Bro04a].....	8
Abbildung 5: Produktionsplanung [Doi03].....	9
Abbildung 6: Augmentierte Daten der Computertomografie [Vog03].....	9
Abbildung 7: MR-Fußball.....	10
Abbildung 8: Augenmuskel des linken Auges [Roh85].....	16
Abbildung 9: Augenquerschnitt [Lev00].....	17
Abbildung 10: Retinale Abbildung bei Tageslicht [url5].....	18
Abbildung 11: Visuelles Feld des rechten Auges [Bof88].....	18
Abbildung 12: Informationsfluss zum visuellen Cortex [url4].....	19
Abbildung 13: Transport des visuellen Felds im Gehirn [url4].....	19
Abbildung 14: Fixationsdauer und Informationsverarbeitung [url5].....	21
Abbildung 15: Kopietest aus der Werbebranche [url6].....	22
Abbildung 16: EOG Messung [url6].....	24
Abbildung 17: Kontaktlinse [url5].....	24
Abbildung 18: Blickachsenmessung [Joo02].....	25
Abbildung 19: Optisches See-Through HMD.....	28
Abbildung 20: Interaktionsmodell.....	31
Abbildung 21: Selektionsschema.....	33
Abbildung 22: Mixed Reality Stage - Menü und Kopfzeiger [url19].....	35
Abbildung 23: Auswählen eines Menüpunkts [Lin03].....	36
Abbildung 24: Vergleichstabelle Eye-Tracking Systeme.....	39
Abbildung 25: HMD-Produkte unterschiedlicher Bauart.....	39
Abbildung 26: CMOS und CCD Kameras.....	40
Abbildung 27: Eye-Tracker.....	41
Abbildung 28: Eye-Tracker Befestigung I.....	41
Abbildung 29: Eye-Tracker Befestigung II.....	42
Abbildung 30: Eye-Tracker Befestigung II (vorne).....	43
Abbildung 31: Eye-Tracker Befestigung III.....	43
Abbildung 32: Anfrage (call) vom Client an das implementierte Objekt..	46
Abbildung 33: AR-API und das Morgan Framework.....	47
Abbildung 34: Informationsfluss zwischen Benutzer und Anwendung. .	47
Abbildung 35: Datenreihen zur Prädiktion mit Polynom 2. Grades.....	49

Abbildung 36: Quellcode Polynomextrapolation.....	49
Abbildung 37: Augengeometrie.....	53
Abbildung 38: Vererbungsschema Arrington-Eye-Tracker.....	55
Abbildung 39: Trackervererbungsschema.....	57
Abbildung 40: Screenshot Eye-Manager.....	58
Abbildung 41: Berechnung des Fadenkreuzoffsets.....	60
Abbildung 42: Modell multimodaler Interaktionsmöglichkeiten.....	62
Abbildung 43: Multimodales Interaktionsmodell.....	62
Abbildung 44: 3-fache Blickrichtungsvisualisierung.....	64
Abbildung 45: MR-Zeiger.....	66
Abbildung 46: Screenshot Eye-Tracking Software.....	67
Abbildung 47: Screenshot Eye-Tracking Software kalibriert.....	68
Abbildung 48: Kalibrierungswürfel.....	69
Abbildung 49: Fokussierungssituationen.....	70
Abbildung 50: Applikationsübergreifende Synchronisation.....	71
Abbildung 51: Handzeiger mit Tasten.....	74
Abbildung 52: Zeigerselektionszeiten statischer Objekte.....	77
Abbildung 53: Auslöserselektionszeiten.....	78
Abbildung 54: Selektion dynamischer Objekte.....	79
Abbildung 55: Bewegung.....	80
Abbildung 56: Bevorzugter Zeiger zum Bewegen.....	80

Formelverzeichnis

Formel 1: Affine Transformation.....	51
Formel 2: Projektion.....	51
Formel 3: Vektoraddition.....	52
Formel 4: Reihenfolge invertierter Matrizen.....	52
Formel 5: Modell-View-Matrix.....	53
Formel 6: Absolute und relative Transformtionen.....	54
Formel 7: Kombination von absoluten und relativen Koordinaten.....	54
Formel 8: Kombination von absoluten und relativen Koordinaten getrennt nach Position und Orientierung.....	54
Formel 9: Ergebnis der Kombination.....	54

1 Einleitung

Die Kombination von Mixed Reality (MR) und Eye-Tracking zur Interaktion ist neu und bietet deshalb eine besondere Herausforderung sowie Chancen und Möglichkeiten für Anwendungen in den unterschiedlichsten Branchen.

Mixed Reality, die gemischte Realität, ist eine Symbiose von Räumen die im Sinne der Wahrnehmung synchronisiert werden. Der eine Raum entspricht dabei der gewohnten Umgebung des Menschen, der andere ist virtuell und existiert digital. Er enthält Information, die meist in Form von Objekten visualisiert und für die Wahrnehmung des Rezipienten durch die Technologie der Mixed Reality aufbereitet wird. Die virtuelle Teekanne in Abbildung 1 ist tatsächlich nur durch das Display des Rezipienten zu sehen und wird hier durch Einsatz eines Handzeigers durch den Raum bewegt.



Abbildung 1: Mixed Reality

Die Information aus dem virtuellen Raum gelangt hauptsächlich über den visuellen Sinn zur Wahrnehmung. Der Informationsfluss in entgegengesetzter Richtung, vom visuellen Sinn zur digitalen Welt kann durch den Einsatz von Eye-Tracking ermöglicht werden. Die Bewegung des Auges wird erfasst und die Blickrichtung ermittelt, in der Information aus dem Kontext der aktuellen Wahrnehmungssituation enthalten ist. Die Integration von Eye-Tracking in Mixed Reality ermöglicht den bidirektionalen Informationsaustausch über den visuellen Sinn, dadurch wird über die Mensch-Maschine-Schnittstelle eine neue Form von Interaktion möglich. Um die neue Interaktionsform unter Berücksichtigung

EINLEITUNG

von intuitiven, natürlichen und gewohnten Handlungen des Menschen in Mixed Reality Umgebungen nutzen zu können, muss einerseits ein geeignetes Eye-Tracking System ausgewählt und integriert werden, andererseits ist die Evaluation geeigneter Interaktionen für den Einsatz notwendig. Die Voraussetzung für die Evaluation ist eine progressive Schnittstelle zum Benutzer, die neben dem bidirektionalen visuellen Kanal weitere Kanäle zur Wahrnehmung des Benutzers einsetzt und durch die vorhandene Technologie der Mixed Reality entsteht. Vergleich und Kombination von verschiedenen Interaktionstechniken liefern durch die Evaluation ein Ergebnis, das zur weiteren Entwicklung von modernen Benutzerschnittstellen beiträgt. Der Einsatz von Eye-Tracking zur Interaktion in Mixed Reality Umgebungen ist aber auch entscheidend von der Anwendungssituation abhängig, da sie mitbestimmt, welche Interaktionstechniken von Vorteil sind. Eine präzise Analyse für die Motivation und Zielsetzung dieser Arbeit erfolgt in den nächsten drei Abschnitten.

Im ersten Abschnitt wird die historische Entwicklung von künstlichen Welten in Verbindung mit gesellschaftlichen Strukturen betrachtet und anhand gegebener Definitionen von virtueller Realität der Begriff und das Ziel von Mixed Reality verdeutlicht. Der zweite Abschnitt geht auf entscheidende Erkenntnisse über die menschliche Wahrnehmung ein, in der die Informationen der verschiedenen Sinne verarbeitet wird und über die der Mensch mit seiner Umwelt kommuniziert. Anwenden von Mixed Reality ist das übergeordnete Ziel, deshalb folgen im dritten Abschnitt mögliche und bestehende Anwendungen und Einsatzbereiche. Darauf aufbauend folgt schließlich die Motivation und die Zielsetzung der Arbeit.

1.1 Mixed Reality

Mixed Reality (gemischte Realität), wie man die Erscheinungsform nennt, ist eine Mischung aus der Realität, der Wirklichkeit, die von jedem Menschen wahrgenommen wird, und dem, was als Virtual Reality bezeichnet wird.

Der Begriff Virtual Reality (VR) wurde durch den vielseitig begabten Künstler und Vordenker Jaron Lanier in den 80er Jahren eingeführt. Er gründete Anfang der Dekade die Firma VPL Research, die als Erste komplette VR-Systeme herstellte und dadurch einen Standard für Kultur und Industrie setzte [url1]. Der oft kontrovers diskutierte, charismatische Pionier des VR-Hypes wurde zu jener Zeit in einem Atemzug mit Thomas Edison genannt.

1.1 MIXED REALITY

Virtual Reality bedeutet übersetzt virtuelle Realität und adressiert eine besondere Art von Realität, in der künstliche Welten vom Menschen modelliert und über technische Hilfsmittel der Wahrnehmung zugänglich gemacht werden. Die Idee, künstliche Welten zu erschaffen, sie nach eigenen Vorstellungen zu gestalten, sich in ihnen zu bewegen und mit ihnen zu interagieren, ist genauso alt wie die Menschheit selbst. Jede Geschichte, sei es von Jagd, Kämpfen oder Geistern, entführte die Zuhörer nicht nur damals in die Welt des Erzählers. Die Erzählung ist immer geprägt von den subjektiven Erfahrungen des Homo sapiens. *„Mit der Herausbildung des Individuums Mensch, der Entwicklung der bildenden Künste und der Literatur wurden abertausende künstliche Welten geschaffen“* [Gam03].

Mit der ersten Fotografie im Jahre 1826 wurde die Ära der technisch reproduzierbaren Kunstwerke eingeleitet. Die Evolution der Technik brachte danach ruckelnde Bilder, die sich im Laufe der Zeit zum bunten Tonfilm weiterentwickelten und den visuellen Sinn in einer neuen Form ansprachen. Diese damals neuen Formen von virtueller Realität, die sich des Mediums Film bedienen, um die Wahrnehmung der Rezipienten zu erreichen, sind jedoch durch den Informationsfluss in nur einer Richtung beschränkt. Walter Benjamin lieferte schon in seinem Essay *„Das Kunstwerk im Zeitalter seiner technischen Reproduzierbarkeit“*, das 1936 erstmals veröffentlicht wurde, eine Fülle von präzisen Analysen der gegenwärtigen Produktions- und Rezeptionsbedingungen [Ben36]. Dabei nahm er erstens an, dass die technische Entwicklung für eine Veränderung des Wesens der Kunst verantwortlich ist, und zweitens, dass *„die Masse eine Matrix (ist), aus der gegenwärtig alles gewohnte Verhalten Kunstwerken gegenüber neugeboren hervorgeht“* [Ben36]. Die bahnbrechende medientheoretische Analyse war insbesondere für die Filmkritik und die Kunsttheorie in den siebziger Jahren von kaum zu überschätzendem Einfluss. Im Film verschmelzen reale und virtuelle Bildelemente zu einem neuen Bild, das den menschlichen Wahrnehmungsapparat veranlasst, in künstlich erschaffene Traumwelten einzutauchen. Die Imagination entsteht durch die Wahrnehmung. Die Phantasie- und Traumwelten, die in der Vorstellung des Rezipienten induziert werden, unterscheiden sich jedoch von den künstlichen Welten der Virtual Reality, die die Herausforderung darin sieht, die Illusion der synthetischen Welt so weiter zu entwickeln, dass sie dem Eindruck der Wirklichkeit möglichst nahe kommt. In wie weit man in die künstliche Welt eintaucht, wird am Grad der Immersion festgestellt. Er erhöht sich, wie dies die geschichtliche Entwicklung zeigt, mit jedem Sprung, der von der technischen Evolution ausgeht. Demzufolge ist mit einer weiteren Veränderung der von Benjamin angesprochenen Massenmatrix zu rechnen, da sich nun neben dem technisch reproduzierbaren Kunstwerk auch die Art der möglichen Kommunikation verändert. Der Rezipient ist nicht mehr passiv, wie beim Medium Film, sondern er kann interagieren, Einfluss auf die Illusion nehmen, die durch technische Sinnestäuschung die Wahrnehmung beeinflusst. Für die Interaktion ist der Austausch von Information charakteristisch, der die vollwertige Kommunikation ausmacht. Die Auswirkung der Interaktion auf die Wahrnehmung wird

EINLEITUNG

hiermit deutlich. Auch die Immersion wirkt auf die Wahrnehmung, aus der dann die Imagination, eine Form schöpferischen Denkens, Fantasie oder Vorstellung hervorgeht. Burdea und Coiffet [Bur03] fassen die drei Begriffe Immersion, Interaktion, Imagination, die als „I³“ bekannt sind, als das Ziel von Virtual Reality zusammen.

Ein entscheidender Punkt für das ungetrübte Gefühl der Immersion ist die Interaktion in Echtzeit, die den Menschen, wie das zum Beispiel bei Computerspielen oft zu beobachten ist, an die dargebotene Welt fesselt. Die Interaktion zwischen Mensch und Maschine bzw. zwischen Wahrnehmung und der simulierten Welt wird durch eine Schnittstelle ermöglicht. Sie legt durch die eingesetzte Technologie die Bandbreite der Interaktion fest. In dem Buch „Virtual Reality Technology“ [Bur03] wird Virtual Reality frei übersetzt definiert als:

„Die virtuelle Realität ist eine progressive Benutzerschnittstelle, die Simulation und Interaktion in Echtzeit durch multiple, sensorische Kanäle ermöglicht. Diese sensorischen Modalitäten sind visuell, auditiv, taktil, olfaktorisch (Geruch) und gustatorisch (Geschmack).“

Eine frühere Definition dieses Begriffes des deutschen VR-Autors Sven Bormann [Bor94] lautet:

„Die VR-Technik umfasst Ein- und Ausgabetechniken, die es dem Menschen erlauben, eine sinnliche Erfahrung zu machen, die einer physikalisch existierenden Wirklichkeit nicht entspricht oder eine physikalisch existierende Wirklichkeit um normalerweise so nicht wahrnehmbare Dimensionen erweitert, beispielsweise mit Durchsichtssystemen, bei denen sich reales Bild und Computergrafik überlagern. Zusätzlich kann der Mensch mit dieser Simulation interagieren. VR bezeichnet ebenso diesen künstlich geschaffenen Lebensraum.“

In Science Fiction Fernsehserien wird die virtuelle Realität im fortgeschrittenen Stadium als Holodeck „Holografische Suite“ (Raumschiff Enterprise) bezeichnet. In ihr ist die Illusion so perfektioniert, dass sie nicht von ihrem physischem Vorbild, der Realität, unterschieden werden kann. Diese Vision, einen so hohen Immersionsgrad zu erreichen, mag gegenwärtig, unter wissenschaftlichen Betrachtungen, als unseriös erscheinen, aber auch manche Einschätzung entwickelt sich im Laufe der Zeit.

Die Mischung aus Wirklichkeit und Synthetik integriert die unterschiedlichen Welten in eine neue Form der Realität: Mixed Reality. Innerhalb dieses Kontinuums zwischen real und virtuell [Mil94], das in Abbildung 2 dargestellt ist und der VR Definition von Bormann entspricht, unterscheidet man zwischen Augmented Reality (AR), einer realen Umgebung die um virtuelle Objekte erweitert wird, und Augmented Virtuality (AV), einer virtuellen Umgebung die um reale Objekte erweitert wird. Beide rücken immer mehr in die Gegenwart der Allgemeinheit. Mixed Reality erbt Definitionen und Ziele der Virtual Reality. Dazu zählt auch I³, das

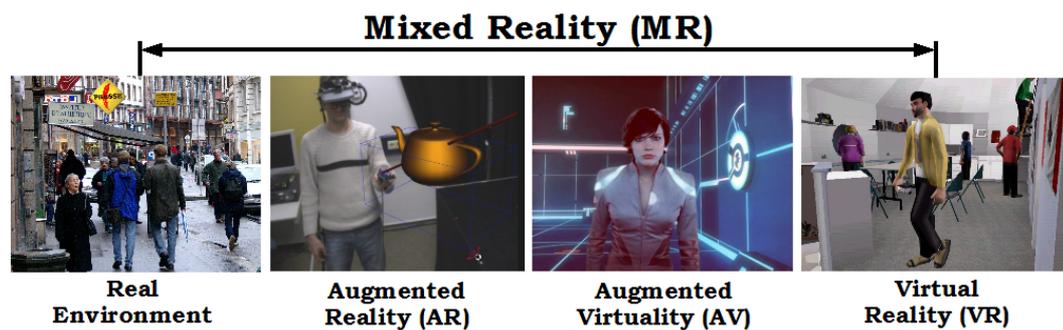


Abbildung 2: Mixed Reality Kontinuum nach [Mil94]

Immersion, Interaktion und Imagination als Ziel von MR zusammenfasst. Unter Immersionsgrad wird in diesem Zusammenhang der Grad verstanden, zu dem man in den virtuellen Teil der MR eintaucht. Als entscheidende Faktoren dafür zählen hier: die Bandbreite der Kommunikation und Interaktion, die auch vom Design der Benutzerschnittstelle abhängig sind, die Reproduktionsgenauigkeit die von Milgram und Kishino als RF (Reproduction Fidelity) [Mil94] definiert wurde und EPM (Extend of Presence Metaphor) [Mil94] ist ein Maß dafür, wie präsent man sich in der angezeigten Szene fühlt. Sie definierten einen weiteren Faktor, der indirekt auch zum Immersionsgrad beiträgt: EWK (Extend of World Knowledge) [Mil94], das ist das Maß für den vorhandenen Kenntnisstand über die reale Welt im Computer. Gerade weil AR eine Erweiterung des gewohnten alltäglichen Umfeldes ermöglicht, ist sie von besonderem Interesse. MR erhöht die Wahrnehmung von und die Interaktion mit der Welt [Azu97][Azu01]. Der virtuelle Anteil von MR kann Informationen anbieten, wie in Abbildung 2 zu sehen, die der Mensch mit seinen Sinnen nicht direkt erfassen kann. In vielen Arbeitsabläufen fallen Aufgaben an, bei denen man mit der realen Umgebung interagiert. Dabei könnte die Handlung des Menschen durch virtuelle Informationen unterstützt und durch neue Interaktionsmechanismen vereinfacht werden. Der Austausch von Information über die progressive Benutzerschnittstelle Mixed Reality auf der einen Seite und der Wahrnehmung des Menschen auf der anderen Seite erfolgt, wie in der MR Definition festgestellt, über multiple sensorische Kanäle. Die Wahrnehmung ist die Zentrale, in der die Informationen der unterschiedlichen Sinneskanäle zusammenkommen.

1.2 Wahrnehmung

Wahrnehmung steht im Zentrum des Informationsaustauschs des Menschen und am Anfang jeder Naturwissenschaft, deshalb ist sie elementar für unser Wissen über die Welt und die Kommunikation in ihr. In dieser Arbeit ist sie von besonderer Bedeutung, da der visuelle Sinn in etwa 90% von ihr einnimmt. Die klassische Psychologie unterscheidet zwischen Empfindung, bei der „einfache“ interne Erfahrungen des Menschen durch „einfache“ Reize hervorgerufen werden, und Wahrnehmung, einer „komplexen“ Erfahrung, die aus der Deutung eines Empfindungskomplexes resultiert. In der modernen Psychologie wird Wahrnehmung als Aufnahme von Information in das Bewusstsein eines Lebewesens betrachtet. Wie in Abbildung 3 zu sehen, ist die Wahrnehmung ein komplexer Prozess, in dem zahlreiche Komponenten zu einem einzigen Wahrnehmungsergebnis beitragen.

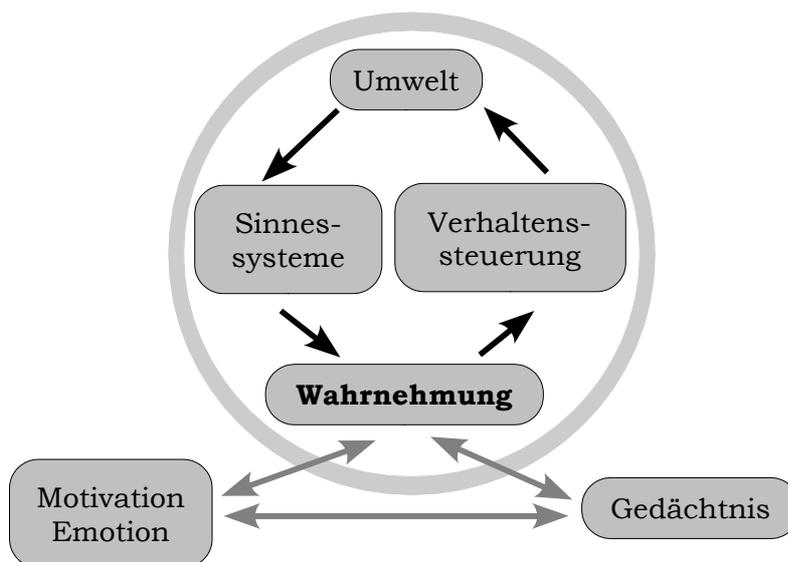


Abbildung 3: Das Wahrnehmungssystem [url22]

Die Psychologie und die Physiologie sind die unterschiedlichen Wissenschaften, die versuchen, ihre Erkenntnisse über die Wahrnehmung in Einklang zu bringen. Die Worte des Psychologen J.Müller aus dem 19. Jahrhundert, „one is not a psychologist who is not also a physiologist“ verdeutlichen dies. Auf der einen Seite der psychologische Verstand und auf der anderen das physiologische Gehirn. Beim Erforschen der Sinne betrachtet man zwei Gesichtspunkte: Gefühl und Wahrnehmung. Das Gefühl wird durch einen Reiz aus der Umgebung ausgelöst, für den Empfang der Information bzw. des Reizes sind die Rezeptoren von Augen, Ohren, Nase, Zunge, Haut etc. zuständig. Die Wahrnehmung ist das Ergebnis der empfangenen und interpretierten Information, die

1.2 WAHRNEHMUNG

durch die Sinne aufgenommen und verarbeitet wurde. Zusammengefasst bedeutet das, man fühlt den Reiz und nimmt wahr, was er ist [Lev00].

Der visuelle Sinn als eine Komponente des Wahrnehmungssystems setzt sich aus unterschiedlichen Bereichen zusammen, die in der Kombination Informationen für die Wahrnehmung von Licht, Bewegung und Räumlichkeit einfangen und aufbereiten. Interessant für diese Arbeit sind (im Hinblick auf die Wahrnehmung) vordergründig die Augen und ihre Eigenbewegung sowie das Entstehen des Raums, der visuellen Umwelt des Menschen. Die Augen können „frei“ im Raum bewegt werden, das geschieht indirekt über den Kopf, der durch die Motorik des Menschen gleichzeitig orientiert und positioniert werden kann. Zusätzlich führen die Augen Eigenbewegungen aus, die meist weder bewusst initiiert noch bemerkt werden. Auch die visuellen Informationen, die sie liefern, werden nicht in ihrer Reinform rezipiert, wie in Abbildung 3 dargestellt, sondern sind nur ein Teil des ganzen Wahrnehmungssystems, von dem man sagt, dass es mehr ist, als die Summe seiner Teile [url22].

Der Weg der Information im visuellen System sowie die unterschiedlichen Augenbewegungen sind in dem Abschnitt *2.1 Visuelles System des Menschen* zusammengefasst und werden zusammen mit der Anatomie und Physiologie des Auges betrachtet.

Das räumliche Sehen, das stereoskopische Erfassen der Umwelt, also die Wahrnehmung der dritten Dimension, kann, wie alle Komponenten der Wahrnehmung, auch nicht isoliert betrachtet werden, da ja nur das Zusammenspiel der Wahrnehmungskomponenten zu einem Resultat führt und unter anderem ein räumliches Bild liefert. Als Ausgangspunkt kann man sicherlich die Anatomie und Physiologie nennen, die den visuellen Input liefern, der im Auge auf den beiden zweidimensionalen Retinas entsteht und anschließend verarbeitet wird. Zur Wahrnehmung von Räumlichkeit, der dritten Dimension, der Distanz zwischen Objekten im Raum, dem Erfassen von Tiefeninformation, gibt es mehrere Signale. Die Signale kann man unterscheiden, indem sie den Kategorien monokular, kinetisch und binokular zugeordnet werden [Lev00].

Monokulare Signale werden oft von Künstlern für die räumliche Gestaltung von Gemälden eingesetzt. Die Bezeichnung monokular weist darauf hin, dass für ihre Wirkung nur ein Auge benötigt wird. Man bezeichnet sie auch als Tiefenkriterien. Monokulare Tiefenkriterien sind zum Beispiel: gegenseitiges Verdecken von Objekten, Schatten, gewohnte und relative Größen, lineare und atmosphärische Perspektiven (Geometrie, Schärfe und Farbton) sowie der Texturgradient (Oberflächenstruktur). Als weiteres monokulares Signal ist die Akkommodation des Auges, das Fokussieren der Augenlinse auf ein Objekt zu nennen.

Kinetische Signale setzen eine Bewegung voraus, die entweder vom Betrachter ausgeht oder im Blickfeld geschieht. Die Bewegungsparallaxe,

EINLEITUNG

die von Objekten in unterschiedlichen Entfernungen ausgeht, liefert Information über ihre relative Distanz zueinander.

Als binokulare Signale bezeichnet man solche, für die beide Augen erforderlich sind. Dazu zählen die Informationen, die aus den leicht unterschiedlichen Abbildungen auf den Retinas entstehen, die man binokulare Disparität oder auch binokulare Parallaxe nennt. Die auftretende Konvergenz bzw. Divergenz der beiden Augen bei der Beobachtung von Objekten, die sich in Richtung des Betrachters bzw. von ihm weg bewegen, gehört auch in diese Signalgruppe [Lev00].

Die Wahrnehmung eines Menschen ist nicht nur das Ziel der an ihn gerichteten Information, sondern sie ist auch Rezipient seiner Umwelt und spielt die zentrale Rolle bei der Aufnahme und Verarbeitung von Information sowie dem Resultat, das aus der Verarbeitung durch das Individuum hervorgeht. Das Individuum wird durch sie beeinflusst. Sein Handeln geht als Resultat aus der Verarbeitung von Informationen aus den verschiedenen Sinneskanälen hervor. Die Interaktion und der Austausch von Information geschieht auf der einen Seite durch die Wahrnehmung und auf der anderen Seite durch die MR Anwendung. Die Vermittlung der verschiedenen beteiligten Kanäle erfolgt dabei über die Mixed Reality Technologie. Für die heutige Technologie gibt es bereits einige Prototypen, Anwendungen und Produkte.

1.3 Mixed Reality Anwendungen

Virtual Reality und Mixed Reality Anwendungen beginnen die Industrie zu erobern und verteilte virtuelle Spielwelten gehören bereits zur Alltagskultur, für die sich besonders Kinder und Jugendliche schnell begeistern lassen. Die Anwendungen führen in der Wirtschaft zum effizienten Umgang mit Prozessen und zeigen Kunst, Kultur und Marketing neue Wege auf, um Menschen anzusprechen und Wissen zu vermitteln. In der Architektur beispielsweise kann die Technologie für Design und Integration von Gebäuden in die zukünftige Umgebung genutzt werden [Ish02], [Kat03], [Bro04a]. In der Abbildung 4 ist ein Szenario für die Integration von Gebäuden zu sehen. Die Previsualisation, die Visualisierung im Vorfeld einer Produktion, wird nicht nur in der Architektur angewandt, sondern auch bei der

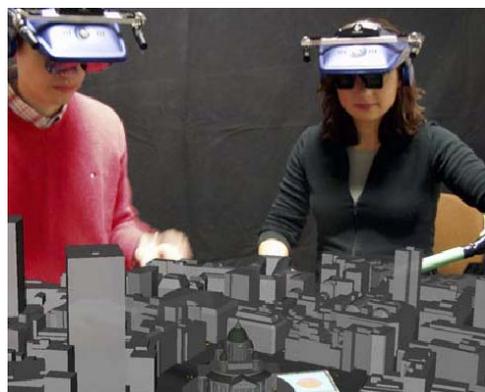


Abbildung 4: Arthur Project [Bro04a]

1.3 MIXED REALITY ANWENDUNGEN

Planung und Gestaltung von komplexen Bühnen- und Filmszenen. Die Kosten am Set der später folgenden Produktion können so minimiert werden, da sich zeit- und kostenintensive Situationen im Vorfeld mit geringerem Aufwand visualisieren und erproben lassen [Bro04b]. Produktdesign und Produktionsplanung der Industrie sind weitere Einsatzbereiche [Kli02], [Doi03].



Abbildung 5: Produktionsplanung [Doi03]

Abbildung 5 zeigt ein Szenario mit Industrierobotern in der Produktionsplanung. Der Einsatz von Mixed Reality im Produktionsprozess ist auch denkbar und bringt mit geeigneter MR-Technologie einen Mehrwert [Tan03]. Entwicklung, Produktion und Wartung von komplexen technischen Produkten und Systemen wird von ARVIKA [Woh00] unterstützt. Auch in der Medizin existieren bereits vielfältige Anwendungen [Vog03], [Sol04], [Mck04], die sowohl auszubildende wie praktizierende Ärzte bei der Diagnose und Behandlung effektiv unterstützen, als auch dem Patienten umfangreiche Einblicke in seine Krankheit bieten. In der Abbildung 6 sieht man Daten, die durch Computertomografie erfasst wurden und einem realen Modell überlagert sind. Dem Militär sind die Vorteile und Chancen dieser Technologie bekannt, die neuesten technischen Entwicklungen sind meist im Umfeld dieser finanzkräftigen Institution zu finden. Die Wissenschaft bedient sich der Technologie, um Daten, die zum Beispiel von Sonden auf weit entfernten Planeten ermittelt wurden, für den Menschen zur Analyse in einer möglichst leicht zu rezipierenden Form aufzubereiten. Ein weiteres Beispiel ist die Archäologie, für die es Applikationen zur Archivierung von Artefakten und Fundorten gibt [Ike03], [Ben04]. Die hochgradig komplexe Information, die mit erheblichem Aufwand eingefangen wurde, lässt sich durch den Einsatz einer Technologie, die anstrebt, künstlich erschaffene Realitäten zu optimieren und die Schnittstelle zum Menschen so effizient wie möglich zu gestalten, mit einem hohen Wirkungsgrad darstellen. Dadurch wird die Wahrnehmung der Wissenschaftler, die zum Erlangen neuer Erkenntnisse aus der aufbereiteten Information entscheidend ist, intensiver angesprochen.

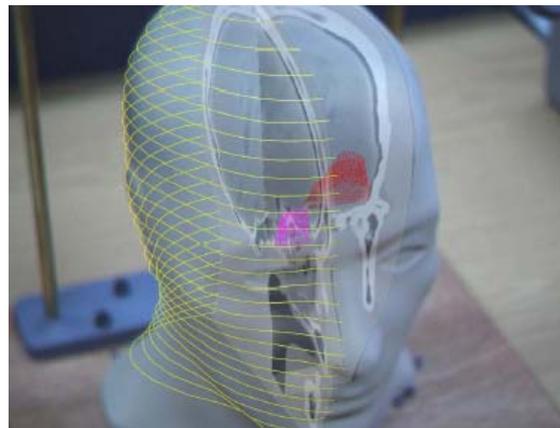


Abbildung 6: Augmentierte Daten der Computertomografie [Vog03]

EINLEITUNG

Die Anwendungsbereiche von VR und MR weiten sich immer mehr aus und sind nicht mehr nur in speziellen Branchen anzutreffen. Die Unterhaltungsbranche arbeitet von jeher mit künstlichen Welten. In Japan hat man ein System zur dreidimensionalen Liveübertragung von Fußballspielen entwickelt [Koy03]. Das augmentierte Blickfeld von einem MR-System, das ermöglicht ein Fußballspiel in Miniatur auf dem Wohn-

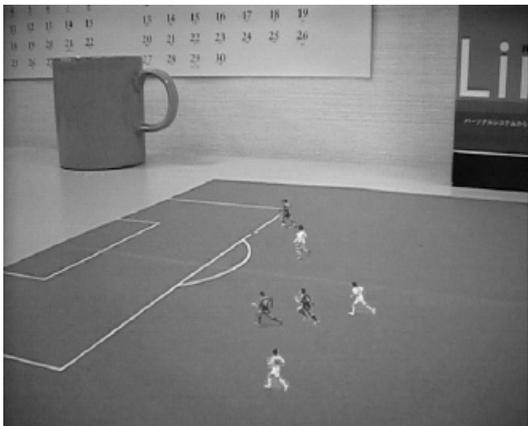


Abbildung 7: MR-Fußball

zimmertisch zu verfolgen, ist in Abbildung 7 zu sehen [Ina03]. Weltweit erfreuen sich verteilte Netzwerkspiele immer größerer Beliebtheit und sind bereits in vielen Haushalten vertreten [url23]. Auch neue Wege der Interaktion mit den Spielen etablieren sich. Aktuelles Beispiel ist das erfolgreiche „Eye-Toy“ der Firma Sony [url13]. Quake, ein Computerspiel, das man gewöhnlich auf Desktop Computern spielt, wurde in ein mobiles MR-System portiert und kann nun im

Freien gespielt werden [Tho03]. Ein weiterer Prototyp, das Spiel Net-Attack, wurde vollständig neu entwickelt und erfordert das Zusammenspiel innerhalb von Teams, die sowohl drinnen wie draußen gegeneinander antreten [url25]. Im Projekt iPerG werden Pervasive Games entwickelt. Das sind alle Räume und Medien durchdringende Spiele [url24], die crossmedial die verschiedensten Kommunikationskulturen miteinander verbinden. Für eine weitere Pervasive Anwendung wurde die Nutzung des Chemielaborbuches in eine „allesdurchdringende“ computergestützte Laborumgebung verwandelt [Sch04a]. Anwendungen und Geräte, die Sachverhalte überall unterstützen, visualisieren oder auf den Kontext zugeschnittene Informationen anbieten, sind noch nicht ganz alltäglich. „Ambient Wood“ ist ein System, das Wissen aus der Ökologie im Kontext der aktuellen realen Position des Anwenders vermittelt [Rog04].

Der Mehrwert und das Ziel der vorgestellten MR-Anwendungen ist, dem Menschen die Kommunikation und die Interaktion mit Geräten in seinem Umfeld zu erleichtern und den Informationsaustausch zu optimieren. Dazu zählt die Unterstützung der unterschiedlichsten Formen der Kommunikation und Interaktion, die nicht nur zwischen Mensch und Maschine, sondern auch zwischen Menschen über räumliche Distanz im Kontext der aktuellen Handlung und Aufmerksamkeit erfolgen kann.

1.4 Motivation und Zielsetzung

In Abschnitt 1.1 *Mixed Reality* wurde die Entwicklung künstlicher Welten und die Bedeutung von Mixed Reality betrachtet. Durch die informationsvermittelnde Mixed Reality Technologie wird der bidirektionale Informationsfluss zwischen der menschlichen Wahrnehmung und einer Anwendung möglich. Der Austausch von Information heißt Kommunikation. Die beiden Beteiligten der Kommunikation wurden in den Abschnitten 1.2 *Wahrnehmung* und 1.3 *Mixed Reality Anwendungen* vorgestellt. Wahrnehmung ist individuell und doch gibt es Gemeinsamkeiten, bestimmte Signale, die als Basis der Kommunikation dienen. Für die angestrebte Kommunikation ist eine gemeinsame Basis maßgebend, die auf der einen Seite durch die menschliche Wahrnehmung festgelegt ist und auf der anderen Seite von den Anforderungen der angestrebten Anwendung definiert wird.

Der Einsatz von Mixed Reality ermöglicht in einem sehr großen Spektrum von Einsatzbereichen neue Anwendungen (siehe 1.3 *Mixed Reality Anwendungen*). Mit dem Einsatz von Mixed Reality verändert sich die Beziehung zwischen Mensch und Computer, da der Computer vom Anwender immer weniger direkt wahrgenommen wird. Die Vision von Mark Weiser (1952-1999), einer neuen Ära der Informationstechnologie namens „Ubiquitous Computing“, hat bereits begonnen. Der ubiquitäre, allgegenwärtige, überall verbreitete Einsatz von Computern erfordert eine Infrastruktur, die das Zusammenwirken der einzelnen Computer ermöglicht. In der Studie von Endres und Butz [End05] sind einige vorhandene Systeme zusammengefasst. Das MR Framework Morgan [Ohl04] bietet eine ähnliche Infrastruktur an, die mehrere Computer vernetzt und eine Entwicklung dynamischer Mehrbenutzer Mixed Reality Umgebungen ermöglicht.

Mixed Reality, die progressive Benutzerschnittstelle, nutzt mehrere Kanäle für die Informationsübertragung. Durch die Vielfalt der Einsatzmöglichkeiten, die durch die Mixed Reality Technologie entstehen, ergeben sich unterschiedliche Anforderungen an die Anwendung. Die gemeinsame Basis, die maßgebend für die angestrebte Kommunikation ist, ist auf der einen Seite durch die Wahrnehmung festgelegt und lässt sich durch die Auswahl von Modalitäten und dem Design der Benutzerschnittstelle auf Seite der Anwendung schaffen.

*„Bei der Schnittstellengestaltung werden mit dem Begriff „Modalität“ die Interaktionsmöglichkeiten des Menschen mit der Maschine beschrieben. Sofern die Interaktion über mehrere Wahrnehmungskanäle erfolgen kann, wird von einer **multimodalen** Schnittstelle gesprochen.“ [Tim02]*

EINLEITUNG

Die Anwendungssituation ist für die optimale Beschaffenheit der multimodalen Benutzerschnittstelle entscheidend. Die Schnittstelle hängt aber auch von der verfügbaren MR-Technologie ab. Die vorgestellten MR-Anwendungen setzen Interaktionsparadigmen ein, die taktile (Hand) und auditive (Sprache) Modalitäten für die Interaktion nutzen. Der visuelle Kanal wird zur Übertragung der Information vom virtuellen Raum zur Wahrnehmung eingesetzt. In manchen Anwendungssituationen wäre es wünschenswert, die Hände frei zu haben, während man der Maschine Anweisungen erteilt. Das kann durch den Einsatz von Sprachschnittstellen erreicht werden, die aber in lauten Umgebungen nicht angemessen sind um nur ein Beispiel zu nennen. Daraus folgt, dass Hände und Sprache nicht für alle Interaktionssituationen geeignet oder ausreichend sind. Der Einsatz von Eye-Tracking in Mixed Reality Umgebungen stellt einen weiteren Kanal zur Verfügung und ermöglicht dadurch die Interaktion über den visuellen Kanal. Die Bewegung des Auges wird erfasst und die Blickrichtung ermittelt, in der Information im Kontext der aktuellen Wahrnehmungssituation enthalten ist. Die Integration von Eye-Tracking in Mixed Reality ermöglicht den bidirektionalen Informationsaustausch über den visuellen Kanal. Dadurch wird über die Mensch-Maschine-Schnittstelle eine neue Form von Interaktion möglich. Der Einsatzbereich von Eye-Tracking zur Interaktion hängt entscheidend von der Anwendungssituation ab, da sie bestimmt, welche Interaktionstechniken von Vorteil sind.

Eye-Tracking könnte in bestimmten Situationen, in denen die Eingabe über Sprache oder die Hände nicht in Frage kommt, sinnvoll für die exklusive Eingabe genutzt werden. Novak verwendet es zum Beispiel bei der Entwicklung von Benutzerschnittstellen im Auto, die über die Aufmerksamkeit des Fahrers bedient werden [Nov04]. Der Einsatz in Verbindung mit anderen Modalitäten könnte aber auch einen Mehrwert bringen. Um unter Berücksichtigung von intuitiven, natürlichen und gewohnten Handlungen des Menschen einen sinnvollen Einsatzbereich für die Augenbewegung in Verbindung mit weiteren Modalitäten zur Eingabe herauszufinden, ist eine Evaluation notwendig. Das Ziel der Mensch-Computer-Interaktion ist, die Schnittstelle zur Maschine so zu gestalten, dass sie nicht mehr bewusst wahrgenommen wird und intuitive, natürliche und gewohnte Handlungen im alltäglichen Leben fließend aus der gemischten Realität unterstützt werden. Genau wie Mixed Reality das Verschmelzen von Umgebung und digitaler Information schon heute der Wahrnehmung zugänglich macht. Ein möglichst hoher Immersionsgrad der MR-Anwendung ist dabei das übergeordnete Ziel (siehe auch 3.2 *Augenbasierte Interaktionstechniken*).

Um Eye-Tracking im gegebenen Forschungsfeld sinnvoll einsetzen zu können, ist zunächst eine Recherche und Marktanalyse erforderlich, die durch einen Vergleich existierender Eye-Tracking Systeme eine Entscheidungsbasis für die Anschaffung eines für MR geeigneten Eye-Tracking Produktes liefert. Anschließend ist die Herausforderung einerseits bei der Integration der Hardware, dem mechanischen Verbinden von Eye-Tracker und Display zu sehen und andererseits beim Einbinden

1.4 MOTIVATION UND ZIELSETZUNG

und Anpassen der Eye-Tracking Software in das Morgan Framework. Dabei sind als erstes Ziel die zeitnah aufbereiteten und visualisierten Eye-Tracking Daten zu nennen, die als Grundlage für die Interaktion mit dem Blick zur Verfügung stehen sollen. Die Datenaufbereitung in ein geeignetes Bezugssystem und die eingesetzten Synchronisierungsmechanismen für Raum und Zeit spielen dabei eine zentrale Rolle, da erst durch sie die näherungsweise präzise, simultane Interaktion in Mixed Reality möglich wird. Die stereoskopische Kalibrierung des Eye-Trackers im Sinne der Wahrnehmung ist eine weitere Voraussetzung. Mit den Augenbewegungsdaten, der synchronisierten Realitäten als Basis, lassen sich darauffolgend, unter Berücksichtigung der Eigenheiten der Augenbewegungen und unter Einbeziehen bestehender Erkenntnisse, mögliche Interaktionstechniken bzw. Mechanismen realisieren. Die anschließende Evaluation der wichtigsten Interaktionen, für die Eye-Tracking geeignet erscheint und die dabei gesammelten Erfahrungen sollen schließlich Auskunft und Ergebnisse liefern, die zur weiteren Verfahrensweise und zur möglichen zukünftigen Entwicklung beitragen können.

In **Kapitel 2** wird das visuelle System des Menschen und die bisherige Entwicklung von Eye-Tracking Technologien vorgestellt. Bestehende Erkenntnisse im Kontext der Mixed Reality Technologie und der Blickinteraktion sind in **Kapitel 3** dokumentiert. Außerdem wird dort das Interaktionsmodell hergeleitet, das als Basis für die Implementierung der MR-Benutzerschnittstelle dient. **Kapitel 4** enthält die Anforderungen und die Marktanalyse für die Anschaffung eines Mixed Reality tauglichen Eye-Trackers sowie die Integration des angeschafften Systems in ein Mixed Reality Display. Die Einbindung der Software in das bestehende Morgan Framework sowie die Erweiterung vorhandener Mechanismen im Kontext geometrischer kontinuierlicher echtzeitnaher Daten folgt darauf. In **Kapitel 5** wird das Design und die Implementierung einer multimodalen Benutzerschnittstelle besprochen, mit der sich das Eye-Tracking System applikationsübergreifend stereoskopisch kalibrieren lässt. Bei den Benutzertests zur Evaluation der Blickinteraktion in **Kapitel 6** werden spezielle Szenarien genutzt, um geeignete Einsatzbereiche der Blickinteraktion für die natürliche Interaktion zu ermitteln. **Kapitel 7** enthält eine Zusammenfassung der vorliegenden Arbeit und liefert einen Ausblick.

EINLEITUNG

2 Eye-Tracking

Eye-Tracking bedeutet das Verfolgen der Augenbewegung und ist mittlerweile ein feststehender Begriff für die verschiedensten Forschungsrichtungen geworden. Seit mehr als 100 Jahren werden Systeme zum Erfassen der Augenbewegung entwickelt. Dabei boten die anfangs ermittelten Daten zunächst die Grundlage, um mehr über die Bewegungen der Augen zu erfahren. Die Erkenntnisse aus der Analyse dieser Daten waren die Basis für die weitere technische Entwicklung, die es ermöglichte, immer präzisere Erhebungen durchzuführen. Es zeichnete sich bald ab, dass das Verständnis um den menschlichen Wahrnehmungsapparat dabei immer mehr in den Mittelpunkt der technischen Evolution des Eye-Tracking rücken sollte, da die Bewegung des Auges ein Teil des Wahrnehmungssystems ist. Die Forschungsgebiete sind mittlerweile weit gefächert und erstrecken sich von der Physiologie über die Psychologie als auch von der Mensch-Maschine-Kommunikation in der Informatik bis hin zur Technik, die immer kleinere und schnellere Geräte zur Datengewinnung und -verarbeitung anbietet. Die wahrnehmungspsychologischen Erkenntnisse fließen in die Entwicklung von Algorithmen ein, die relevante Daten aus aufgenommenen Bildern extrahieren und veredeln. Die sich ergänzenden Disziplinen machen den Entwicklungsprozess aus und liefern immer ergonomischere und effizientere Eye-Tracking Systeme. Zur kommerziellen Anwendung kommen die Eye-Tracking Systeme gegenwärtig zum Beispiel in der Marketing- und Werbebranche, die mit sogenannten Kopietests die Wirksamkeit von Anzeigen und Gestaltungselementen in den Print-Medien, aber auch in den elektronischen Medien evaluieren. Als Standardwerk ist „*Eye tracking in human-computer interaction and usability research: Ready to deliver the promises*“ [Jac93] zu nennen. Als Voraussetzung für die Arbeit mit Eye-Tracking Systemen ist das Wissen über das visuelle System des Menschen anzusehen, das ausschnittsweise im Folgenden dargestellt wird. Anschließend fasst ein Abschnitt die zeitliche Entwicklung des Eye-Tracking zusammen. Darauf folgt die Beschreibung von aktuellen Technologien, die das Erfassen von Augenbewegungen ermöglichen.

2.1 Visuelles System des Menschen

Der visuelle Sinn sammelt Eindrücke, die uns mit der Realität verbinden und nimmt ca. 90% der menschlichen Wahrnehmung ein. Dieser Sinn besteht aus vielen, sich ergänzenden Komponenten, die noch nicht alle vollständig erforscht und in Beziehung zueinander gesetzt wurden.

Damit die Bewegung des Auges technisch erfasst werden kann, sind nicht nur die anatomischen und physiologischen Eigenschaften des Auges von Bedeutung, da die Augenbewegung, wie schon angedeutet, in engem Zusammenhang mit anderen Komponenten des visuellen Systems steht und zum Wahrnehmungsergebnis beiträgt. Deshalb wird auf die fundierten Erkenntnisse der Anatomie und Physiologie des Auges etwas näher eingegangen. Danach steht die Beschreibung der verschiedenen Arten von Augenbewegungen im Vordergrund.

2.1.1 Anatomie und Physiologie

Die Augen befinden sich ungefähr auf der horizontalen Mittellinie des Kopfes, in der sogenannten Augenhöhle. Abbildung 8 vermittelt einen groben Überblick der Anatomie des Auges, das durch drei gegenseitig angeordnete Muskelpaare bewegt wird.

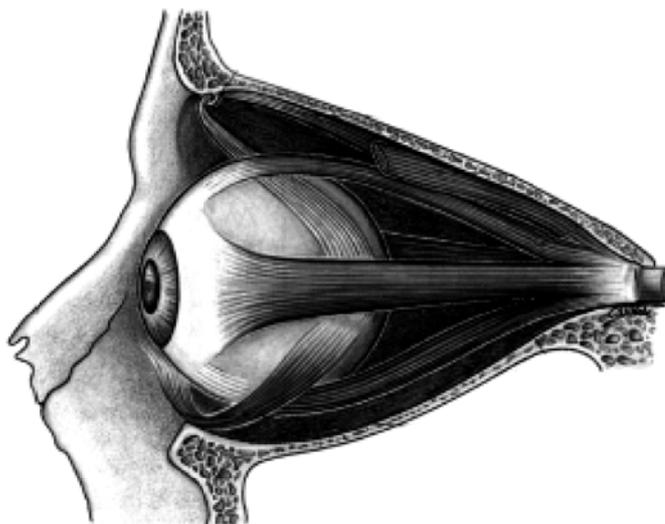


Abbildung 8: Augenmuskel des linken Auges [Roh85]

2.1 VISUELLES SYSTEM DES MENSCHEN

Für die horizontale und vertikale Bewegung ist je ein Muskelpaar zuständig. Das dritte Paar ist für die Torsion verantwortlich, die für die Anpassung des räumlichen Sehens in der Ferne und Nähe eine Rolle spielt. Nach einer Schätzung können die mit höchster Präzision arbeitenden Augenmuskeln die Augen so bewegen, dass sich der Blick auf mehr als 100.000 verschiedene Punkte des Gesichtsfeldes richten kann. Außerdem arbeiten die Muskeln, die Motorik beider Augen zusammen. Das ist wichtig, damit sich der Blick beider Augen auf den gleichen Punkt richtet und somit die entsprechenden Reize entstehen, die bei der Wahrnehmung für die Interpretation der dritten Dimension mit verantwortlich sind. Der Welle-Teilchen-Dualismus ist bekannt, jedoch ist hier wegen der abstrahierten Betrachtungsweise von Teilchen des Lichts, also den Lichtstrahlen, die Rede. Der Strahlengang in das Auge lässt sich in Näherung mit dem in einer Kamera vergleichen. Sicherlich ist die Optik professioneller Kameras, wie auch beim Auge, äußerst komplex, doch kann man sie für die einfache Betrachtung auf die Wirkung einer dünnen Linse reduzieren [Stö00]. Die Blende, mit der das einfallende Lichtbündel begrenzt wird, ist beim Menschen die Iris (Regenbogenhaut), deren Funktionsweise ebenfalls näherungsweise analog zu der in einer Kamera betrachtet werden kann.

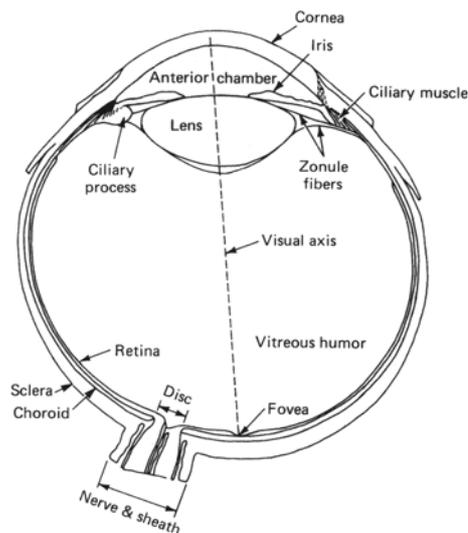


Abbildung 9: Augenquerschnitt [Lev00]

Die von der Realität emittierten und reflektierten, durch die Pupille auf die Linse des Auges einfallenden Lichtstrahlen, bilden auf der Netzhaut, der Retina, die Außenwelt verkleinert ab. Die gewölbte Augenvorderseite und die in der Dicke durch Muskelkraft veränderbare Linse verändern die Brennweite und stellen das Bild scharf (siehe Abbildung 9). Da sich der Brennpunkt des menschlichen optischen Systems zwischen Auglinse und Retina befindet, erscheint das Abbild vertikal und horizontal gespiegelt. Die Rezeptoren auf der Netzhaut empfangen die „Nachricht“, die über Synapsen codiert, als elektrische Impulse über den Sehnerv an das Gehirn weitergeleitet werden. Dieses sorgt unter anderem auch dafür, dass das Abbild nicht spiegelverkehrt wahrgenommen wird.

EYE-TRACKING

Subjektiv bemerkt man nicht, dass das Gesichtsfeld aus einem mittleren Bereich des scharfen Sehens und einem äußeren Bereich mit zunehmender Unschärfe besteht. Das liegt daran, dass die Augen ständig in Bewegung sind, sodass immer andere Teile des Gesichtsfeldes in den Bereich der Fovea gelangen, wenn sich die Aufmerksamkeit erst auf diesen und dann auf jenen Gegenstand richtet. Im mittleren Bereich der menschlichen Retina, nahe der optischen Achse, befindet sich die Macula, deren Zentrum ein kleines, etwas vertieftes Gebiet, die Zone des schärfsten Sehens darstellt, auch Fovea Centralis genannt. Hier ist die Dichte der lichtempfindlichen Rezeptoren, den für das Farbsehen zuständigen Zapfen, am größten. Die Einzelheiten eines betrachteten Objektes können dort deshalb besonders gut aufgelöst werden.

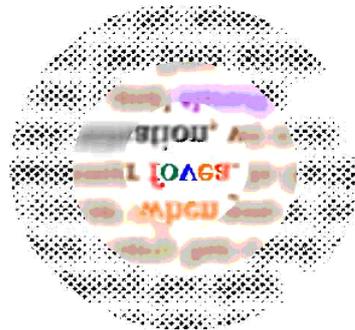


Abbildung 10: Retinale Abbildung bei Tageslicht [url5]

Schärfstes Sehen ist in einem Winkel von ca. 1° um den fixierten Blickort möglich. Das entspricht bei ausgestrecktem Arm ungefähr der Größe des Daumennagels. Die an das Gebiet der Fovea Centralis angrenzenden 3° werden als Parafovea bezeichnet und Gebiete außerhalb dieses Bereiches als Peripherie [Bal91]. Praktisch hat das zur Folge, dass vom Blickort weiter entfernt liegende Objekte mit zunehmend verringerter Auflösung und farbloser auf der Retina abgebildet werden. Dies wird in der Abbildung 10 illustriert. Der Grund dieser peripheren Unschärfe besteht in der Anbindung der Rezeptoren an den Sehnerv.

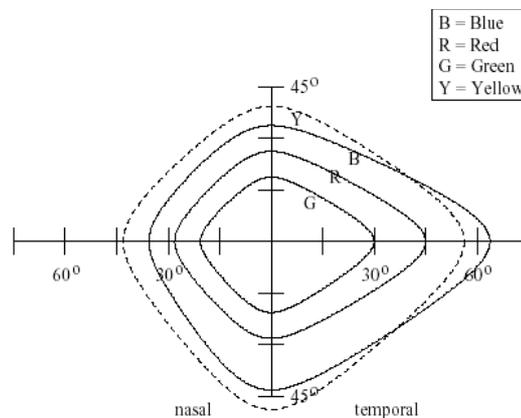


Abbildung 11: Visuelles Feld des rechten Auges [Bof88]

2.1 VISUELLES SYSTEM DES MENSCHEN

Bereits bei einer Abweichung von 3° vom Fixationsort vermindert sich die Sehschärfe um die Hälfte. Das visuelle Feld der Augen ist naturgemäß begrenzt und je nach wahrgenommener Wellenlänge unterschiedlich groß (siehe Abbildung 11) [Dur00].

Beim Sehen werden nicht nur Helligkeitsunterschiede und Farben erfasst, sondern es entsteht – über die Wahrnehmung unterschiedlicher Entfernungen und die Lagebezeichnungen von Objekten durch beide Augen – auch ein räumliches Bild.

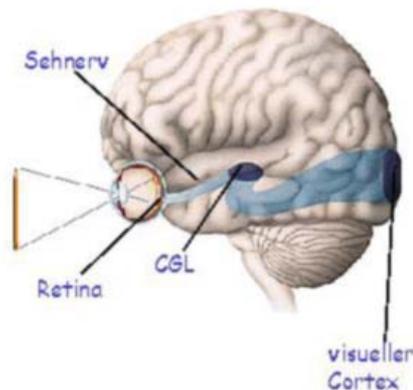


Abbildung 12: Informationsfluss zum visuellen Cortex [url4]

Das Auge ist im visuellen System für die Fokussierung der einfallenden „Lichtstrahlen“ zuständig, die von der Retina in elektrische Aktivitäten gewandelt werden. Das Nervensystem transportiert die generierten elektrischen Impulse zunächst an den CGL (Corpus Geniculatum Laterale), der den Informationsfluss zum visuellen Cortex reguliert. Dort angekommen, werden die visuellen Reize verarbeitet. Der Weg der Information vom visuellen Feld beider Augen ist in Abbildung 12 zu sehen und in Abbildung 13 farblich gekennzeichnet. Dadurch wird deutlich, wie sie auf dem Weg zum visuellen Cortex sortiert wird.

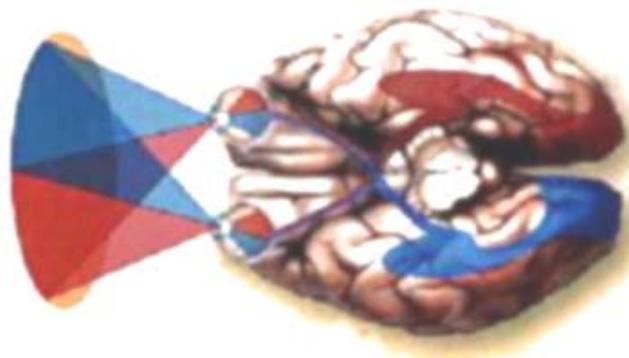


Abbildung 13: Transport des visuellen Feldes im Gehirn [url4]

EYE-TRACKING

Im CGL kommen die visuellen Informationen nach linkem und rechtem Sichtfeld getrennt an. Sie werden nicht vermischt, wie man denken könnte, sondern situationsbedingt gefiltert und an den visuellen Cortex weitergeleitet.

Der visuelle Cortex ist für den weiteren Verlauf bis hin zur Wahrnehmung zuständig. Da gegenwärtig noch keine sicheren Aussagen über die weitere Verfahrensweise mit der visuellen Information gemacht werden können, wird an dieser Stelle auf die Wahrnehmungspsychologie verwiesen zum Beispiel [Lev00].

2.1.2 Augenbewegungen

Das Auge ist nicht nur ein Sensor, der einen kleinen Wellenlängenbereich (380nm-720nm) dem System von Synapsen des Menschen zugänglich macht, sondern es wird auch zur aktiven Exploration der Umgebung eingesetzt. Es ist in der Lage, eine Vielzahl unterschiedlicher Bewegungen auszuführen, die neben dem Verschmelzen der Abbildungen auf der Retina von großer Bedeutung für das räumliche Sehen und das Abschätzen von Größen und Entfernungen sind. Die Motorik der Augen steht nicht nur im Zusammenhang mit den laufenden Handlungen der Person, sondern sie eilt ihnen um etwa eine halbe Sekunde voraus. Mit der Analyse der Augenbewegung beschäftigt sich die Psychologie. Unterschieden wird zwischen Vestibulärbewegungen, die zur Verfolgung von fixierten Objekten in der Umgebung sowie zum Ausgleich von Körperbewegungen während der Fixierung von Gegenständen erfolgen, und Sakkaden, den Bewegungen, die zur Ausrichtung des Auges auf neue Sehobjekte notwendig sind. Außerdem gibt es noch die Mikrobewegungen, die bei Fixierung eines statischen Objektes auftreten [Joo02]. Die reale Bewegung setzt sich aus diesen elementaren Augenbewegungen zusammen und ist von der momentanen Intention und Wahrnehmungssituation des Individuums entscheidend abhängig.

Vestibulärbewegung

Die Vestibulärbewegung, die auch als Folgebewegung (smooth pursuit) bezeichnet wird, ist eine langsame, gleitende Bewegung, die als Reaktion auf die Bewegungen des Körpers erfolgt und dadurch die Körperbewegung kompensiert, damit der visuell wahrgenommene Bereich auf der Retina näherungsweise konstant bleibt. Weiterhin tritt die Vestibulärbewegung auf, wenn man bewegte Objekte in der visuellen Umwelt verfolgt. Ein Beispiel ist, das Beobachten eines Hasen, der über ein Feld rennt,

2.1 VISUELLES SYSTEM DES MENSCHEN

während der Betrachter sich am gleichen Ort befindet und den Kopf still hält.

Sakkaden

Die Augen bewegen sich nicht gleichmässig, sondern springen von einem Ort zum nächsten. Die Bewegung ist sehr kurz und sehr schnell. Auf die Bewegung folgt eine kurze Ruhephase. Diesen Ablauf nennt man sakkadische Augenbewegung. Sakkaden sind ballistische Bewegungen, die während ihrer Dauer von 30-40ms die visuelle Wahrnehmung drastisch einschränken und die Informationsaufnahme verhindern. Dieser Effekt wird als *saccadic suppression* bezeichnet. Sakkaden sind für die Erkundung zuständig und treten sporadisch oder als Reflex auf. Der Mensch hat keine direkte Kontrolle über sie, obwohl bei der gezielten Erkundung oder Suche in der Umwelt die Häufigkeit der Sakkaden deutlich zunimmt.

Fixierung und Microbewegung

Fixiert man einen unbewegten Gegenstand, steht das Auge keinesfalls still. Es bewegt sich andauernd und springt gelegentlich ruckartig im

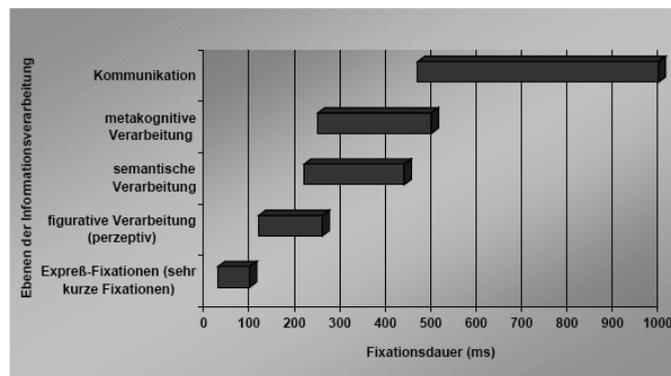


Abbildung 14: Fixationsdauer und Informationsverarbeitung [url5]

Blickfeld umher. Die geringe Bewegung wird vom Gehirn gefiltert und deshalb nicht wahrgenommen. Sie wird auch als Microsakkade bezeichnet und tritt im Bereich von einigen Winkelminuten mit einer Frequenz von 20-150 Hz auf. Diese Bewegung ist notwendig, damit die Rezeptoren im Auge kontinuierlich stimuliert bzw. gereizt werden. Ohne diese Bewegung würde das statische Bild langsam verschwinden. Sie kann bis zu 0,3 Grad betragen und etwa 20 bis 30 ms dauern.

Die Fixierung, also das scheinbare Stillstehen der Augenbewegung, hat grundsätzlich etwas mit Verarbeitung von Information zu tun, kann aber auch eine Aussage über das Interesse an dem fixierten Ort machen. Ca. 90% der Zeit, in der man etwas betrachtet, wird mit Fixierung verbracht.

2.2 Entwicklung des Eye-Tracking

In den Anfängen bediente man sich eher invasiver Methoden, die direkten mechanischen Kontakt mit der Hornhaut herstellten, um herauszufinden, was das Auge anschaut bzw. was es gerade fixiert hat. 1901 wurde die erste berührungsfreie Technik entwickelt, die Reflektionen von der Hornhaut auf fotografischen Platten festhielt. Mit der Entwicklung von statischen zu „laufenden Bildern“ – dem Film – konnte nun auch der zeitliche Verlauf der Augenbewegung festgehalten und analysiert werden. In den 50er Jahren entstanden die ersten Blickpfade von Piloten, die man während des Landeanflugs aufnahm. Die erste Anwendung war das Usability-Engineering – die systematische Studie von Benutzern, die mit Produkten interagieren. Ziel dieser Studien ist es, das Design der Produkte wahrnehmungseffizient unter Berücksichtigung von ergonomischen Aspekten zu optimieren. Im Laufe der 70er Jahre beschäftigte man sich erfolgreich damit, die Augenbewegungen mit der Wahrnehmung in Verbindung zu bringen. Dazu wurden die aufgezeichneten Bewegungspfade des Blicks mit psychologischen Theorien kombiniert. Der Fortschritt in der Technik erlaubte es mittlerweile, echtzeitfähige Eye-Tracking Systeme herzustellen. Dadurch profitierte die Datenanalyse, die zuvor nur in Retrospektive erfolgen konnte. Die Erkenntnis, dass verschiedene Reflektionen des Lichts auf der Hornhaut des Auges zur Berechnung der Blickrichtung herangezogen werden können, erhöhte die Genauigkeit der ermittelten Daten entscheidend und trug außerdem dazu bei, dass die Augenbewegung unabhängig von der Bewegung des Kopfs betrachtet werden konnte. Als sich in den 80er Jahren die Computer immer weiter verbreiteten, überlegte man erstmals, wie die Augenbewegung zur Interaktion zwischen Mensch und Maschine genutzt werden kann [Jac03]. Die Technologien, die heute für die Erfassung von Augenbewegungen zur Verfügung stehen, werden im Nachfolgenden betrachtet.

2.3 Eye-Tracking Technologien

Je nach Einsatzbereich bedient man sich unterschiedlicher Verfahren zum Erfassen der Bewegung des Auges. Um zwischen den Verfahren differenzieren zu können und um die Unterschiede zwischen ihnen aufzuzeigen, wird ihre Funktionsweise näher erläutert.

EYE-TRACKING

Das Elektro-Okulogramm (EOG) entsteht durch Aufzeichnen von Potenzialdifferenzen, die man mit Hilfe von Elektroden, die in der Augengegend angebracht werden, misst. Der entscheidende Nachteil bei diesem Verfahren ist der geringe Messbereich von ca. 30° und die Notwendigkeit, Elektroden im Gesicht befestigen zu müssen. Die Methode lässt sich in einer Laborumgebung realisieren. Außerhalb jedoch erscheint sie etwas umständlich in der Handhabung.



Abbildung 16: EOG Messung [url6]

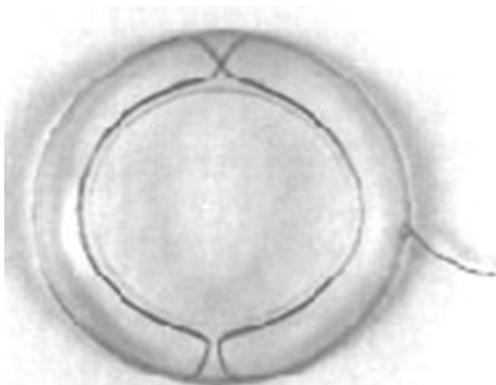


Abbildung 17: Kontaktlinse [url5]

Die genaueste, aber gleichzeitig auch aufwendigste Methode zur Registrierung der Augenbewegung, setzt Kontaktlinsen ein. Die Linsen werden entweder mit einem Spiegel präpariert, der eine Lichtquelle reflektiert, oder sie werden mit einer Spule, wie in der Abbildung 17 gezeigt, präpariert. Beide Fälle sind mit äußerster Vorsicht zu bewerkstelligen und erlauben in jedem Fall nur eine kurze Tragedauer. Deshalb sind sie nicht besonders praktikabel [Joo02].

Die Blickachsenmessung (auch Point of Regard Measurement), eignet sich gut zum Einsatz mit Desktopcomputern, da sie, wie in Abbildung 18 dargestellt, absolut berührungslos eingesetzt werden kann. Die Voraussetzung für ein solches Messsystem sind zwei Bezugspunkte im Auge. Mit ihnen kann die Eigenbewegung des Auges bestimmt werden, da die Bewegung, die mit dem Sensor (Kamera) erfasst wird, die Kopf- und Augenbewegung enthält. Als einen Punkt nimmt man meist die Pupille und als anderen einen Lichtreflex. Am häufigsten wird dafür der korneale Reflex genutzt, den man auch als erstes Purkinje-Bild kennt und der durch eine Reflektion, meist einer Infrarot-Lichtquelle, auf der Hornhaut des Auges entsteht. Mit dieser Information kann die Blickachse errechnet werden, da sich die beiden Punkte trotz der Kopfbewegung näherungsweise in einem festen Verhältnis zu kleinen Winkeln der Augenbewegung verhalten. Ein ähnliches Verfahren nutzt das Free-Gaze-System, das auf dem Desktop zur berührungslosen Erfassung eingesetzt werden kann [url8], [Ohn02].

Ebenfalls zum Einsatz auf dem Desktop eignet sich ein weiteres Verfahren, das die Kopfbewegung mit einer beweglichen Kamera auf dem Tisch

2.3 EYE-TRACKING TECHNOLOGIEN

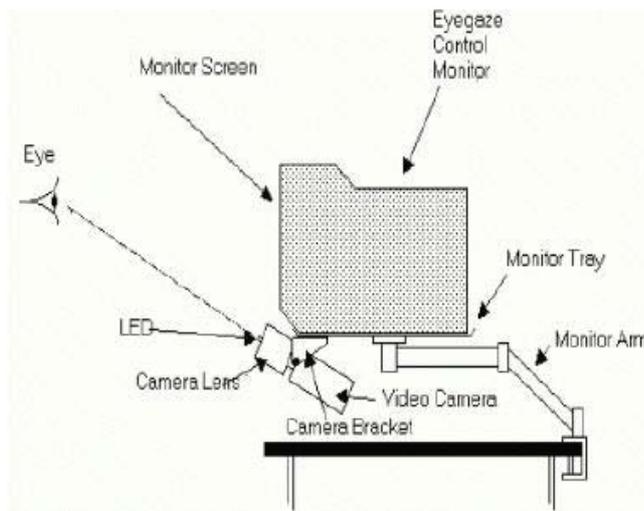


Abbildung 18: Blickachsenmessung [Joo02]

ausgleicht. Die Kamera liefert das Abbild des Auges, aus dem die Augenbewegung direkt berechnet werden kann. Auch hier wird eine Infrarot-Lichtquelle eingesetzt, um ihre vorteilhafte Reflektion im Auge bei der Bildverarbeitung zur Berechnung der Augenbewegung nutzen zu können.

Ist das Messgerät, meist eine CCD-Kamera, am Kopf befestigt, geht man davon aus, dass ein Bezugspunkt, der sich simultan mit den Augenbewegungen verändert, reicht, um die Bewegungen der Augen zu ermitteln. Auch hier setzt man das für den Menschen visuell nicht wahrnehmbare infrarote Licht ein und nutzt die reflektiven Eigenschaften des Auges in diesem Wellenlängenbereich. Algorithmen, die zur Berechnung der Augenbewegung eingesetzt werden, sind sehr unterschiedlich und bestehen in der ersten Stufe meist aus einem Netzwerk von Filtern, die aus dem eingefangenen Bild alle möglichen Störungen entfernen. Darauf folgend gibt es die unterschiedlichsten Ansätze zur weiteren Verfahrensweise. Dazu zählen beispielsweise auch neuronale Netze, die speziell für ihren Einsatz trainiert wurden. Camus und Wildes präsentieren einen vielversprechenden Algorithmus zum Extrahieren der Augenbewegung aus Nahaufnahmen des Auges [Cam02].

Die vorgestellten Technologien und deren verwendete Algorithmen können in unterschiedlichen Situationen effektiv zum Erfassen von Augenbewegungen eingesetzt werden. Die dadurch erlangten Daten lassen sich durch die Interpretation von einer Anwendung veredeln und für einen bestimmten Zweck verwenden. Das weitere Verfahren und der dadurch erlangte Mehrwert sind vom Kontext abhängig, in dem das Eye-Tracking eingesetzt wird. Es folgt nun die nähere Betrachtung des Einsatzbereichs und danach die Anforderungen und die Auswahl eines geeigneten Eye-Trackers.

EYE-TRACKING

3 MR Technologie und Blickinteraktion

Die Kombination von Mixed Reality und Eye-Tracking zur Interaktion ist neu und stellt deshalb eine besondere Herausforderung dar.

Die Ausgangssituation definiert sich durch den Stand von Technik und Forschung in den verschiedenen Disziplinen, die zur Realisierung der Interaktion mittels des menschlichen Blicks in einer Mixed Reality Umgebung beitragen. Dazu gehört unter anderem die Informatik, die Wahrnehmungspsychologie, das Eye-Tracking und die Mensch-Maschine-Kommunikation.

Der Stand der Forschung lässt sich aus den Fortschritten der Mixed Reality Technik sowie im Umfeld des Eye-Tracking ausmachen und wird durch Auftreten verschiedener Probleme verdeutlicht.

Zunächst soll eine kurze Zusammenfassung der Technik, die zur Realisierung einer Mixed Reality Umgebung notwendig ist, zeigen, in welchem technischen Umfeld die Blickinteraktion realisiert wird. Anschließend wird auf die bestehenden Erfahrungen mit der Interaktion mittels des Auges eingegangen.

3.1 Mixed Reality Technologie

Die Technik, die für die Visualisierung des virtuellen Bildanteils notwendig ist, erfordert Tracker. Die Aufgabe eines Trackers im abstrakten Sinn ist es, die realen Gegebenheiten, nämlich die Position und die Orientierung eines dynamischen Objektes in Daten umzusetzen und Veränderungen laufend zu aktualisieren, damit diese Information virtuell

MR TECHNOLOGIE UND BLICKINTERAKTION

vorhanden ist. Dadurch wird ein Übergang von der Realität zur Virtualität geschaffen.

Weiterhin ist eine Anzeige notwendig, über die die berechneten virtuellen Bilder dargestellt werden können. Dafür verwendet man in Mixed Reality Umgebungen meist Head Mounted Devices (HMD), die wie die Bezeichnung aussagt, auf dem Kopf getragen werden. HMDs, die sich für Mixed Reality Anwendungen eignen, sind entweder durchsichtige Displays (Optical-See-Through) oder undurchsichtige Displays (Video-See-Through). Letztere benötigen eine Kamera, die die Umwelt bzw. die Umgebung aufnimmt, damit sie zusammen mit den virtuellen Bildelementen angezeigt werden kann. Das Verfahren nennt man Video-Augmentierung. In Abbildung 19 ist ein klassisches Setup mit einem Optical-See-Through-Display skizziert.

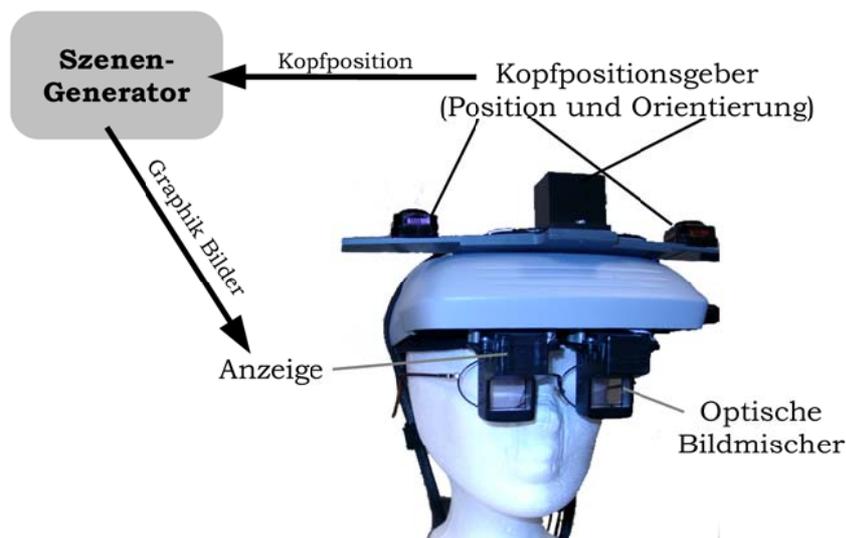


Abbildung 19: Optisches See-Through HMD

Der Kopfpositionsgeber, auch Headtracker genannt, überträgt 6 Freiheitsgrade, das bedeutet je 3 für die Position (x,y,z) und die Orientierung (α,β,γ) , vom Kopf des Betrachters im Raum an den Szenen-Generator, der als Teil einer MR-Applikation, basierend auf allen zur Verfügung stehenden Daten das aktuelle, virtuelle Bild aus der Sicht des Betrachters berechnet. Wie am Anfang dieses Abschnitts besprochen, ist ein Tracker für den Übergang geometrischer Daten in die Virtualität zuständig, die in diesem Fall, zur Berechnung des aktuellen Blickfelds des Anwenders beitragen und dadurch die Augmentierung ermöglichen. Der Rechenvorgang wird als Rendern bezeichnet. Damit die Render-Engine alle Objekte innerhalb des aktuellen Blickfelds berechnen kann, benötigt sie neben den geometrischen Informationen, die das Blickfeld definieren, Informationen über alle darzustellenden Objekte. Dazu gehören außer den geometrischen Definitionen der Objekte, die unterschiedlichen Eigenschaften, die z.B. das Aussehen durch Farbe und das Verhalten im Zusammenhang mit Licht festlegen. Die Beschreibung und die Zustände

3.1 MIXED REALITY TECHNOLOGIE

der virtuellen Objekte sind im Szenengraphen hierarchisch, ähnlich der Baumstruktur eines Dateisystems, angeordnet und definieren alle virtuellen Bildelemente zum Renderzeitpunkt. Für die Beschreibung der Szene und der darin enthaltenen Objekte gibt es viele proprietäre Sprachen, sowie die Standards VRML'97 [url15] und den Nachfolger X3D [url16]. Dynamische Objekte haben Parameter oder bildlicher gesprochen Attribute, die sich im Laufe der Zeit ändern können. Handelt es sich dabei um Objekte, die in beiden Realitäten eine Rolle spielen, so wird für Übertragen und Aktualisieren der geometrischen Lage ein Tracker verwendet. Aus technischer Sicht gibt es viele verschiedene Trackingverfahren, die alle mehr oder weniger die gleiche Aufgabe erfüllen, aber in ihrer Funktionsweise und technischen Umsetzung unterschiedlich sind. Deshalb ist die Anwendungssituation entscheidend für die Wahl eines geeigneten Trackingverfahrens.

Nachdem der Mensch gewohnt ist, nicht nur wegen des Abstands seiner zwei Augen voneinander, räumlich zu sehen und das Ziel, eine möglichst hohe Reproduktionsgenauigkeit RF (siehe *1.1 Mixed Reality*) zu erreichen angestrebt wird, sollte die Darstellung der virtuellen Realität im HMD möglichst viele Kriterien einsetzen, die eine räumliche Wahrnehmung der Szene ermöglichen. Für eine stereoskopische Anzeige ist es notwendig, das angezeigte Bild für jedes Auge separat zu berechnen, da die Augen des Betrachters aus unterschiedlichen Positionen im Raum auf die Szene blicken und die Wahrnehmung unter anderem den Unterschied der angezeigten Information nutzt, um den Eindruck von Räumlichkeit zu vermitteln. Weitere Kriterien, die zur Wahrnehmung von räumlicher Tiefe beitragen, sind im Kapitel *1.2 Wahrnehmung* aufgeführt.

Für die ungetrübte Wahrnehmung und in Hinblick auf EPM (siehe *1.1 Mixed Reality*) als Teil des Immersionsgrads ist auch die Synchronität der beiden Realitäten in Zeit und Raum wichtig und nicht immer unproblematisch [Azu97]. Die räumliche Synchronität für Mixed Reality erreicht man, indem die Parameter für die Berechnung der Anzeige richtig eingestellt [Sch04b] und die Tracker kalibriert werden. Eine präzise Kalibrierung sorgt dafür, dass kein wahrnehmbarer Unterschied zwischen den Realitäten bemerkt wird. Allerdings ist dies nur dann möglich, wenn auch eine zeitliche Synchronität gegeben ist, was bedeutet, dass die Übertragung der Daten von den Trackern in Echtzeit gegeben sein muss. Echtzeit heißt in diesem Zusammenhang, so schnell, dass die Wahrnehmung keine Latenzen erkennt, also nahe Echtzeit. Die Trägheit des visuellen Systems des Rezipienten wird auch bei Film und Fernsehen genutzt, wobei es möglich ist, der Wahrnehmung mit einer Bildrate ab etwa 20 Bildern pro Sekunde einen „flüssigen“ Eindruck zu vermitteln. Die Flimmergrenze liegt etwas höher: erst bei einer Bildfrequenz von 48 Hz wird im Kino das Flimmern des Films nicht mehr wahrgenommen. In der Regel sagt man, dass ab 60 Hz, besonders bei der Wiedergabe von sehr hellen Flächen, der Flimmereffekt beim Rezipienten ausbleibt und Bewegung sauber dargestellt wird. Bei diesem Wert wäre also die Toleranz im optimalen Fall 16 ms. Diese Zeit steht für Übermittlung der Daten und die Berechnung der Anzeige zur Verfügung. Man könnte

aber unter Umständen auch mit einem Wert bis zu 50 ms arbeiten. Entscheidend ist dabei unter anderem der subjektive Eindruck, der bei Benutzung des Systems entsteht, weil die individuelle Wahrnehmung des Rezipienten durch Gewohnheiten und Erfahrungen geprägt ist und geprägt wird. Deshalb existiert kein absolut einheitlicher Wert. Der gewonnene Eindruck trägt zum EPM (siehe 1.1 *Mixed Reality*) bei. Entspricht ein Mixed Reality System den genannten Anforderungen, so kann man von Echtzeit im Sinne der Wahrnehmung reden.

Ein weiterer Faktor, der den Immersionsgrad mitbestimmt, ist das Design der Benutzerschnittstelle, die die Interaktion zwischen Mensch und Maschine ermöglicht.

3.2 Augenbasierte Interaktionstechniken

Zur Zeit dominiert der Einsatz von taktilen Geräten zur Bedienung des Computers. Die Hand- und Fingerbewegung zur Interaktion über Tastatur und Maus ist im Büro- und Computeralltag Usus. Das Ziel, die Mensch-Computer-Interaktion, kurz HCI (human-computer-interaction), zu optimieren bzw. die Bandbreite für die Interaktion zu erhöhen und den Umgang mit dem Computer menschlicher zu gestalten, ist unbestritten. Das impliziert sogleich den Einsatz von immer multimodaleren und intuitiveren Benutzerschnittstellen, die ergonomisch abgestimmte Interaktionsmechanismen bereit stellen. Dabei gibt es Bestrebungen, Benutzerschnittstellen zu entwickeln, die nicht wie gewohnt auf die Eingabe von Kommandos angewiesen sind und eine Selektion des zu manipulierenden Objektes erfordern, sondern das Ziel haben, grundsätzlich intuitiver zu sein. Dazu zählt auch die Mixed Reality, die unter anderem von der Definition der virtuellen Realität den Anspruch nach multimodaler Interaktion in Echtzeit erbt.

Einen Ansatz dokumentiert die Arbeit „*Mutual Disambiguation of 3D Multimodal Interaction in Augmented and Virtual Reality*“ [Kai03]. Dort setzt man auf die Kombination von symbolischen und statistischen Informationen, die aus Körpersprache, Sprache und der Umgebung des Anwenders abgeleitet werden, um mit einer augmentierten oder virtuellen Umgebung interagieren zu können. Die Kombination der Informationen, die aus Symbolik und Statistik hervorgehen, soll die Ungewissheit ihrer Bedeutung minimieren, damit die Intention des Anwenders besser zur Interaktion genutzt werden kann. Der Prototyp des Systems interpretierte die Intention des Anwenders in 59 Prozent der Fälle richtig und erreichte eine mittlere Antwortzeit von 1,1 Sekunden (Standardabweichung = 0,68 Sekunden). Da für ein „Echtzeitsystem“ ein Wert von 16ms wünschenswert wäre und die Ansprüche des Menschen,

3.2 AUGENBASIERTE INTERAKTIONSTECHNIKEN

richtig verstanden zu werden, bestimmt nahe der 100% Marke liegen, wurde gezeigt, dass eine intuitivere Interaktion des Menschen mit der Maschine grundsätzlich möglich ist, aber auch noch Entwicklungszeit benötigt wird, um den beiden genannten Ansprüchen gerecht werden zu können. Die Ansprüche stellen sicher, dass ein Anwender durch die Benutzerschnittstelle nicht be-, sondern entlastet wird. Gegenwärtig bereitet ein Interface, das eine deutliche Tendenz zu einem höheren Grad an Intuition anstrebt, noch Probleme in der technischen Umsetzung. Darum hat auch die Evaluation in diesem Bereich einen hohen Stellenwert.

Interaktion impliziert eine Wechselbeziehung. Laut Duden [Dro96] ist Interaktivität auch der Dialog zwischen Mensch und Computer. Etwas detaillierter kann man den Dialog zwischen einem Objekt im Computer und dem Menschen betrachten, der aus Austausch von Informationen besteht. Interpretiert man das Wort Interaktion etwas näher, so stellt man fest, dass es auch etwas mit Aktion zu tun hat. Zusammenfassend kann man sagen: es finden Aktionen und der Austausch von Informationen statt. Der Austausch geschieht über die Schnittstelle, die nicht nur für den Fall einer Mixed Reality Anwendung das Ziel hat, eine möglichst hohe Bandbreite für die Informationsübertragung zum Anwender zur Verfügung zu stellen.

Hat der Mensch die Intention, eine Veränderung seiner Umgebung herbeizuführen, so ist eine Aktion notwendig. Das Ziel in einer Mixed Reality Umgebung ist meist eine Veränderung, eine **Modifikation** der Szene zu bewirken, etwas zu gestalten, egal, ob die angezeigten Objekte Geometrien aus der Architektur, Medizin oder Automobilindustrie visualisieren oder nutzen, um gezielt Informationen darzustellen und zu organisieren oder ob die Objekte irgendeine Form von Kommunikation mit etwas oder jemandem ermöglichen. Nachdem alles Mögliche an Information in einer Mixed Reality Umgebung in Form von Objekten abstrahiert und visualisiert werden kann, ist das Ziel ein Objekt. Da gewöhnlich mehrere Objekte dargestellt werden, ist die **Selektion** des Ziels der angestrebten Aktion notwendig. Die Art der Modifikation, die mit dem selektierten Objekt geschehen soll, geht aus der Intention des Benutzers hervor. Je nach Komplexität und Kontext der Intention muss sie meist in mehrere Schritte aufgeteilt und untergliedert werden, wie das auch bei der gängigen Betrachtung von Arbeitsabläufen passiert. Entscheidend dafür ist die angestrebte **Anwendung**, der Kontext und das Design der Benutzerschnittstelle. In Abbildung 20 sind die Zusammenhänge im Kern des besprochenen Interaktionsmodells dargestellt.



Abbildung 20: Interaktionsmodell

Leider ist die Funktionalität des Computers auf die Aktionen, die ihm vom Menschen beigebracht wurden beschränkt, d.h. er kann nur die Aktionen seriell ausführen, die von einem Menschen programmiert wurden. Der Mensch hingegen ist es gewohnt, seine Intention über unterschiedliche Kanäle simultan zu übermitteln. In ihrer über mehrere Kanäle kodierten Form (Sprache, Mimik, Gestik usw.) wird die Intention noch nicht einmal von jedem anderen Menschen richtig verstanden. Da der Mensch jedoch von der Beschränktheit der Maschine weiß, passt er sein Verhalten ihr gegenüber, also der Benutzerschnittstelle, über die kommuniziert wird, an und nutzt eine ihr verständliche Sprache oder **Kommunikationsform**, die aus dem Design der Schnittstelle hervorgeht, damit ein Austausch stattfinden kann. Durch diese Anpassung geht, wie bei einer Übersetzung, Information verloren. Außerdem erfordert die Anpassung Wissen über die Maschine und der ihr verständlichen „Sprachen“ sowie die Fähigkeit, ihre Funktion zumindest im Anwendungskontext nachvollziehen zu können. Über Umwege kann also ein Austausch stattfinden. Das geschieht jedoch auf Kosten der möglichen Kommunikationsbandbreite. Im Vergleich zu der gewohnten, natürlichen Kommunikation des Menschen, entsteht zusätzlich eine erhebliche **kognitive Last** durch die Übersetzung der Intention. Als Folgerung daraus könnte man feststellen: je natürlicher es dem Menschen möglich ist mit der Maschine zu kommunizieren, desto größer ist die Bandbreite der Kommunikation. Das folgende Zitat über das Ziel des Designs einer **optimalen Benutzerschnittstelle** geht auch davon aus, dass sie kaum wahrgenommen wird, also eine geringe kognitive Last während der Kommunikation erzeugt.

„When an interactive System is well designed, the Interface almost disappears, enabling users to concentrate on their work, exploration or pleasure.“ [Shn98]

Nachdem die Bandbreite der Kommunikation bzw. der Informationsübertragung auch den Immersionsgrad beeinflusst und das Design der Schnittstelle, sowie das Nutzen von natürlichen und gewohnten Kommunikationsformen entscheidende Faktoren für ein modernes Interface mit großer Bandbreite sind, stellt sich die Frage, wie das **natürliche Design** einer modernen Benutzerschnittstelle aussieht.

Als eine wesentliche Komponente des menschlichen Wahrnehmungssystems bietet sich die Integration des visuellen Sinns oder zumindest einige seiner Teilbereiche in einen solchen Interaktionsmechanismus an. Die Bewegung der Augen ist prädestiniert dafür, da sie als Quelle mit hoher Bandbreite Input vom Computeranwender liefert [Jac03].

Sicherlich ist der Umgang mit den aus dem Eye-Tracking ermittelten Daten und die Art der Integration für die genutzte Bandbreite entscheidend. Das Potential, das sich hinter der Augenbewegung verbirgt ist immens, aber wichtige Vorgänge in diesem Zusammenhang sind kaum entschlüsselt. Man stelle sich nur mal die Bewegungen der menschlichen Motorik zur Interaktion mit der Umwelt vor. Ihnen geht

3.2 AUGENBASIERTE INTERAKTIONSTECHNIKEN

meist ein **Blick** voraus, der unbestritten als kürzeste Reaktionszeit des Menschen gemessen wird. Der Blick lokalisiert außerdem **ohne bewusstes Zutun** des Menschen das favorisierte Objekt.

Es gibt einige Arbeiten, die den Einsatz von Eye-Tracking zur Interaktion in Benutzerschnittstellen als Musersatz für Desktop-Computer erforschten und feststellten, dass die Interaktionsgeschwindigkeit mit dem Auge schneller ist. Dazu gehören „*Evaluation of Eye Gaze Interaction*“ [Sib00] und „*Eye Tracking in Virtual Environments*“ [Wil02]. Der simple Ansatz erscheint auf Grund des gewohnten und weit verbreiteten „Zeigen und Klick Paradigma“ naheliegend. Jedoch gibt es auch gravierende Unterschiede, die langfristig gesehen sicherlich nicht unberücksichtigt bleiben können, weil das gesteckte Ziel von HCI und Mixed Reality, die höhere Interaktionsbandbreite, nicht aus den Augen verloren werden soll. Die Problematik bei der Blickinteraktion ist, dass der Mensch nicht gewohnt ist durch die Augenbewegung mit seinem Umfeld so abstrakt zu interagieren, wie es für den Musersatz notwendig wäre [Jac93].

Die Erkenntnisse beziehen sich auf **Blickinteraktion** mit dem Computerbildschirm, deshalb lassen sich nicht unbedingt alle Erkenntnisse auf die Interaktion im augmentierten dreidimensionalen Raum übertragen, da die Augenbewegung in Kombination mit der Kopfbewegung genutzt werden kann, so wie das der Mensch im Alltag gewohnt ist.

Auch in rein virtuellen Umgebungen wurde die Blickrichtung zur Selektion genutzt und Vorteile bei der intuitiven Nutzung der Blickrichtung zur Interaktion erkannt [Tan00]. In der Arbeit von Tanriverdi und Jacob wurde der Vergleich zwischen einem 3D-Zeiger der am Finger der Probanden befestigt war, einem *Fingerzeiger*, und einem Zeiger der durch die Blickrichtung des Menschen im Raum ausgerichtet wurde, ange stellt. In jedem Fall wird innerhalb einer virtuellen Welt mittels des Zeigers selektiert. In Folge dieser Arbeit kommt man zu dem Schluss, dass Augenbewegungen besonders in Benutzerschnittstellen, die nicht kommandobasierend sind, ihre Vorteile geltend machen können.

Die **Selektion** ist ein notwendiger Schritt für die Interaktion, da durch sie die Frage nach dem „Womit?“ geklärt wird. Ein Zeiger, der durch die Richtung des Blicks ausgerichtet wird, ist ein *Blickzeiger*. Er ist nicht nur wegen dem Geschwindigkeitsvorteil sehr vielversprechend, sondern auch, weil die Exploration und die Interessensbekundung durch längere Fixierungen Teil des natürlichen Blickverlaufes sind. Um zwischen bloßem Umherschweifen des Blicks, dem Explorationsverhalten und den Fällen differenzieren zu können, in denen eine Aktion auf das gerade betrachtete Objekt ausgelöst werden soll, ist ein **Auslösemechanismus** notwendig. Das Selektionsschema in der Abbildung 21 zeigt den Zusammenhang zwischen Selektion, Zeiger und Auslöser.



Abbildung 21: Selektionsschema

Problematisch stellt sich der Auslösemechanismus bei der Selektion mit dem Blick dar, da das Auslösen nicht zu dem natürlichen Blickverhalten gehört. Der wahrnehmungsökonomische Auslöser für die augenbasierte Interaktion ist noch nicht zweifelsfrei gefunden. Sicherlich gibt es schon einige Sensoren, die sich für diese Aufgabe eignen würden. Auch gibt es einige Arbeiten im Umfeld von Jakob, die sich mit der Thematik befassen oder ihn referenzieren. Dabei fällt die Arbeit von Slavko Milekic [Mil02] besonders auf, die sich mit dem Einsatz von Blickgesten in Kontext von „neuen“ Benutzerschnittstellen und Interaktionsmechanismen beschäftigt. Auch in der Arbeit „*Evaluation of Gaze Interaction*“ [Sib00] wird über eine Verweilzeit ausgelöst. Das bedeutet, die Blickdauer auf einen Bildschirmbereich löst nach ca. 150 ms die mit dieser Position verbundene Aktion aus. Beim Einsatz dieser Methode ist die Anpassung der Verzögerungszeit elementar, da sich dadurch die Benutzerfreundlichkeit entscheidend verändert und auch die Überselektion von Bildelementen, der sogenannte Midas Touch, vermieden wird. Ist für die Verweilzeit ein zu geringes Intervall festgelegt, würden durch das schlichte Umhersehen, dem natürlichen Explorationsverhalten des Menschen im aktuellen Interaktionsbereich, ungewollt alle möglichen Aktionen initiiert bzw. Objekte selektiert werden.

Eine weitere Problematik ist die Stabilität und die Genauigkeit der Eye-Tracker wie in der Arbeit von Jakob und Karn [Jac03] festgestellt wird. In der Arbeit „*Eye Tracking in Virtual Environments*“ [Wil02] wird für HMD-Eye-Tracker das Verrutschen der Apparatur auf dem Kopf als problematisch angesprochen und die Forderung nach einem Mechanismus zur schnellen *Rekalibrierung* (siehe Abschnitt *Eye-Tracker Kalibrierung*) laut. Auch der Wunsch nach einem kabellosen HMD wird in diesem Zusammenhang geäußert.

Die Frage nach dem natürlichen Design der MR Benutzerschnittstelle ist nach wie vor offen und lässt sich solange nicht zweifelsfrei klären, bis neue Erfahrungen aus der Praxis zu neuen Erkenntnissen beitragen. Die Voraussetzung dafür ist eine **Mixed Reality Benutzerschnittstelle**, welche die Interaktion mit dem Auge zulässt und die damit verbundenen Eigenheiten in sein Design integriert. Die Möglichkeit, verschiedene **multimodale Interaktionen** evaluieren zu können, sollte dabei gegeben sein (siehe *Benutzerschnittstelle und Blickinteraktion*). Bestehende Erkenntnisse aus der Interaktion mit der Mixed Reality sind der Ausgangspunkt und tragen zur weiteren Entwicklung bei.

3.3 Mixed Reality Interaktion

Die Mixed Reality Stage ist eine Applikation, die Planung sowie choreografische Gestaltung von Bühnen-Events auf einer 4:1 verkleinerten Bühne mit Unterstützung von Mixed Reality ermöglicht [Bro04b]. Zur Interaktion innerhalb der Mixed Reality Stage wählt man Befehle mittels eines Zeigers aus, der durch die Kopfbewegung ausgerichtet wird, im Folgenden wird der Zeiger deswegen als *Kopfzeiger* bezeichnet. Dabei wird zunächst das Fadenkreuz, das mittig auf der Anzeige zu sehen ist, mit der Kopfbewegung auf ein Objekt im Menü oder in der Szene ausgerichtet.



Abbildung 22: Mixed Reality Stage - Menü und Kopfzeiger [url19]

Durch die Bewegung des Kopfs, dem Verändern von Position und Orientierung des Benutzers im virtuellen Raum, können virtuelle Objekte und auch Menüeinträge durch Überlagerung mit dem Fadenkreuz ausgewählt werden. Zum Selektieren, der zuvor im Menü ausgewählten Befehle, benutzt man Sprachkommandos. Die Selektion bewirkt das Ausführen einer Aktion, die durch den entsprechenden Menüeintrag definiert wird.

„The initial test runs, however, showed that the View Pointer was quite intuitive to use. Because of the small field of view the user has to turn her head anyway to see the object of interest.“ [Lin03]

In dem Zitat wird die intuitive Art und Weise der Interaktion mit dem Kopfzeiger festgestellt, jedoch auch die Notwendigkeit der Kopfbewegung zur Lokalisierung von Objekten in der Szene, die aus dem eingeschränkten Blickfeld resultiert.

MR TECHNOLOGIE UND BLICKINTERAKTION

Während der Anwendung des geschilderten Szenarios konnte die Interaktionssituation nachvollzogen und im anschließenden Gespräch mit den Beteiligten des Projekts festgestellt werden, dass die anstrengenden aber notwendigen Kopfbewegungen einerseits aus dem Abstand zwischen Menü und dem entsprechenden Objekt resultieren, andererseits aber auch auf die Fixierung des Fadenkreuzes in der Mitte des Blickfelds zurückzuführen sind.

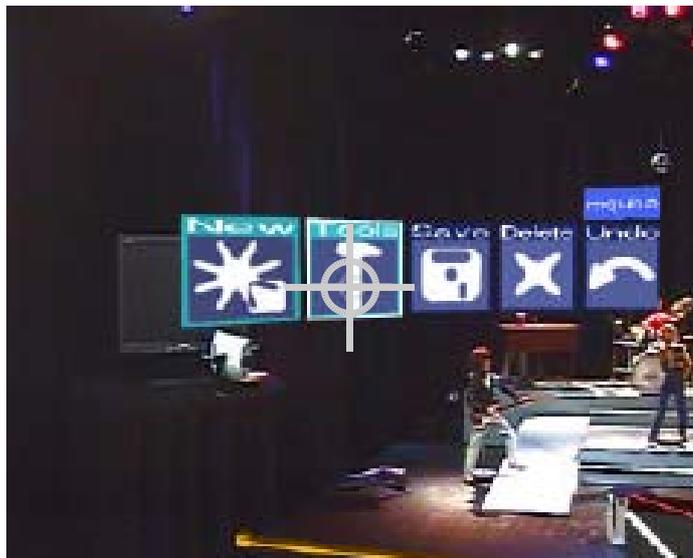


Abbildung 23: Auswählen eines Menüpunkts [Lin03]

Um den Kopf zu entlasten und um der intuitiven, natürlichen sowie gewohnten Interaktionsform näherzukommen, wurde die Überlegung angestellt den Eye-Tracker in die Benutzerschnittstelle zu integrieren. Dadurch wäre das Auswählen in der Szene durch die Kombination von Kopf und Blick möglich. Weiterhin wäre es dann auch denkbar, die Menüs so zu gestalten, dass sie sich mit dem Kopf bewegen und deshalb immer im Blickfeld zu sehen wären. Die Selektion der Menüpunkte könnte dann exklusiv mit den Augen erfolgen.

Zur Realisierung einer Interaktion mit dem Blick ist zunächst die Integration des Eye-Tracking Systems notwendig, die es ermöglicht den Blick des Anwenders zu erfassen. Es folgt nun ein Kapitel, das sich mit der Beschaffung und Integration der Eye-Tracking Hardware auseinandersetzt.

4 Auswahl und Integration des Eye-Tracking

Der nächste Abschnitt befasst sich mit Eye-Tracking Systemen, die für den Einsatz in MR geeignet sind. Danach folgt die Entwicklung einer Haltevorrichtung, die es ermöglicht die angeschaffte Hardware in Verbindung mit verschiedenen HMDs einzusetzen. Die Integration der Eye-Tracking Software in das Morgan Framework sowie die Erweiterung vorhandener Mechanismen im Kontext geometrischer, kontinuierlicher, echtzeitnaher Daten in unterschiedlichen Bezugssystemen schließt dieses Kapitel ab.

4.1 Mixed Reality geeignete Eye-Tracking Systeme

Die Anforderungen an Eye-Tracking Systeme, die den Einsatz in Mixed Reality Umgebungen gewährleisten, werden im Folgenden festgelegt. Danach steht der Vergleich existierender Eye-Tracking Systeme unter Berücksichtigung der zuvor definierten Anforderungen im Vordergrund.

4.1.1 Anforderungen zur Mixed Reality Integration

Die Anforderungen an ein Eye-Tracking System für die Integration in ein Augmented Reality Display ergeben sich zum großen Teil aus den Spezifikationen der verwendeten Displays. Als Wichtigstes ist bestimmt die

AUSWAHL UND INTEGRATION DES EYE-TRACKING

technische Integrierbarkeit zu nennen, da es möglich sein muss, die beiden Geräte miteinander zu kombinieren, während ihre Funktionalität erhalten bleiben sollte. Die Schwierigkeit dabei ist es, im Vorfeld abzuschätzen, wie die Lage der beiden optischen Systeme zueinander gestaltet werden kann, da die optische Achse jedes Systems auf das Auge gerichtet sein muss und das AR-Display selbstverständlich den wichtigsten Teil des natürlichen Blickfelds einnimmt. Außerdem darf das Blickfeld des Auges nicht zu sehr in Mitleidenschaft gezogen werden, da sich ansonsten der Immersionsgrad des gesamten Systems verringert. Weiterhin wäre eine modulare Lösung der Integration begrüßenswert, damit verschiedene Modelle von HMDs mit der Neuerung ausgestattet werden können oder das Eye-Tracking System ohne erheblichen Aufwand mit einem anderem HMD verbunden werden kann. Zusätzlich dazu wäre es wünschenswert, dem Benutzer weiterhin das Tragen einer Sehhilfe zu ermöglichen, da auch manche Displays dafür konstruiert wurden. Das Blickfeld auch „Field of View“ kurz FoV genannt, liegt bei den gängigen Displays zwischen 20° und 30° horizontal. Daraus folgt für den Eye-Tracker ein Minimum von 30°, damit er den kompletten augmentierten Bereich abdecken kann.

Aus Sicht der Softwareintegration sollten alle relevanten Parameter, die zur Interaktion beitragen können, zur Verfügung stehen bzw. über eine API (Application Programming Interface) am besten in C++ ansprechbar sein. Essentiell gehört die Blickrichtung dazu. Außerdem wäre ein Zugriff auf die Kalibrierung des Systems gut. Zusätzliche Daten, wie Torsion, Geschwindigkeit, Fixierungspunkte und Fixierungsdauer, sind optional zu sehen. Was die zeitliche Auflösung angeht, kann man als Untergrenze die Bildwiederholrate der Augmentierung festsetzen. Abschließend ist noch der Wunsch nach einer möglichst günstigen Lösung zu nennen, da nach einem ersten Überblick die Systeme nicht gerade preisgünstig zu erstehen sind.

Die wichtigsten Anforderungen zusammengefasst:

- **AR-HMD Integrierbarkeit**
- **Field of View min. 30°**
- **C++ API (Blickrichtung, Kalibrierung)**
- **preisgünstig**

4.1.2 Auswahl eines geeigneten Systems

Die Recherche ergab 10 Hersteller weltweit, die Eye-Tracking Systeme mit stark variierenden Preisen anbieten. Die technischen Daten wurden

4.1 MIXED REALITY GEEIGNETE EYE-TRACKING SYSTEME

in einer Tabelle zum Vergleich gegenübergestellt, die aus Platzgründen in Abbildung 24 in gekürzter Form zu sehen ist. In *Anhang B* ist die vollständige Version zu finden. Die Tabelle diente als Entscheidungsgrundlage für die Anschaffung eines geeigneten Systems.

Hersteller	FoV			Auflösung		Sampling	Gewicht	Preis	AR/VR tauglich	API	Kalibrierung	Realtime Parameter
	H	V	T	Track	Blickpos.							
Optom	+/-15°	---	---	<0,1°	---	1kHz	---	6.620 €	---	Seriell	---	---
Polhemus	---	---	---	---	---	---	---	12.246 €	---	Nein	---	---
Electronica	+/-30°	+/-20°	+/-45°	+/-0.016°	+/-0.1°	480/240/60Hz	---	---	---	---	---	---
El-mar	+/-40°	+/-30°	---	---	+/-0.1°	120Hz	300g	---	---	---	---	---
SMI	+/-30°	+/-20°	---	<0,1°	0,5-1°	50/60 Hz	450g	20.000 € +Int.ca. 9.000 €	Teilweise	UDP	über UDP initialisierbar	Blickposition, Pupillendurchmesser etc.
Ect	+/-15°	+/-10°	---	---	1°	?	12g	~6.600 €	spec. Integr.	ja	---	?
Eye Link	+/-20°	+/-18°	---	<0.022°	<0.5°	250Hz	420g	---	---	---	s.u.	s.u.
Eye Link	+/-30°	+/-20°	---	<0.01°	<0.5°	500Hz	420g	23.600 €	Teilweise	C/Python	3-13 Punkte	3-10ms: Blickrichtung, Pupillengröße 30ms: Saccaden, Fixationen, Blinzeln
ASL	50°	40°	---	0,1°	0,5°	50/60 Hz	241g	24.432 €	---	---	s.u.	s.u.
ASL	50°	40°	---	0,1°	0,5°	60 Hz	35g	27.500 €	AR/VR	C++/RS232	9 Punkte	Blickrichtung, Pupillendurchmesser
Arrington	+/-44°	+/-20°	---	0,15°	0,25-1°	30/60Hz	ähnlich asl 501vr	5.500 € +Int. 3.800 €	spec. Integr.	C++	User Profile, Auto-Calibrate, Slip-Correction	s.u.
Arrington	---	---	---	---	---	---	---	3.400 €	Ohne Optik	C++	User Profile, Auto-Calibrate, Slip-Correction	0,1-0,5ms: Pupillenposition, Geschwindigkeit, Fixationen, Pupillenabmaße, Torsion
nVision	---	---	---	---	---	---	1,6 kg	~130.000 €	---	---	---	---

Abbildung 24: Vergleichstabelle Eye-Tracking Systeme

In Abbildung 25 sind einige dieser Systeme unterschiedlicher Bauart zu sehen. Dabei wird deutlich, dass die Anforderung an die HMD-Integrierbarkeit nicht ohne weiteres zu erfüllen ist. Aufgrund des Eindrucks, der



Abbildung 25: HMD-Produkte unterschiedlicher Bauart

beim Anblick der Bilder entstand, musste eine Vorauswahl der betrachteten Systeme getroffen werden. Außerdem wurden Überlegungen und kleinere Recherchen angestellt, die auf Alternativen zur Anschaffung eines „fertigen“ Eye-Tracking Systems ausgerichtet waren. Das Ergebnis der Nachforschungen im Bereich der Bauteile sowie der Untersuchung bestehender Algorithmen, die in der Bildverarbeitung zur Gewinnung der Blickdaten erfolgreich eingesetzt werden könnten, zeigte, dass ein nicht

AUSWAHL UND INTEGRATION DES EYE-TRACKING

unerheblicher Aufwand betrieben werden müsste, um ein annähernd geeignetes System zu entwickeln.



Abbildung 26: CMOS und CCD Kameras

Es stellte sich die Frage: Kaufen oder selbst entwickeln? Da mit der Anschaffung von einem kompletten System unter anderem nicht unerhebliche Kosten verbunden sind.

Sicherlich hätte die angestrebte Interaktion in Hinblick auf den zeitlichen Aufwands für die Entwicklung eines Eye-Trackers nicht realisiert werden können und das Ziel dieser Arbeit hätte sich verlagert. Die Hardware aus dem Umfeld der Bildverarbeitung erschien tauglich, aber etwas zu voluminös und zu schwer für den angestrebten Anwendungsfall. Die Möglichkeit, den Algorithmus für die Datenextraktion selbst zu entwickeln und dadurch in der Lage zu sein, ihn für den Anwendungsfall optimieren zu können, wurde als Vorteil bewertet. In diesem Fall wären sicherlich auch die Software Bibliothek von Intel, die Image Processing Library und die Open Source Computer Vision Bibliothek zur Anwendung gekommen.

Die Entscheidung fiel zugunsten der amerikanischen Firma Arrington Research aus, da einerseits alle Anforderungen eingehalten bzw. übertroffen wurden und das Preis-Leistungs-Verhältnis überdurchschnittlich war. Das Gerät ist in der Abbildung 27 zu sehen. Die Hardware ist auf das Notwendigste reduziert, während die Software neben dem SDK, über das die wichtigsten Funktionen des grafischen Interfaces angesprochen werden können, drei Algorithmen mit unterschiedlichen Eigenschaften zur Datengewinnung anbietet. Weiterhin ist es möglich, durch den eingebauten Kalibrierungsmechanismus die zunächst inhomogenen Daten der Augenbewegung, unter Berücksichtigung der Bildschirmauflösung und dem Abstand des Betrachters zum Bildschirm, in eine homogene Form zu bringen, die für die weitere Verarbeitung notwendig ist.

4.2 Entwicklung der HMD Befestigung

Ein wesentlicher Vorteil des ausgewählten Eye-Trackers ist unter anderem, dass er, wie in der Abbildung zu sehen, sehr klein und dadurch auch besonders flexibel einzusetzen ist.



Abbildung 27: Eye-Tracker

Die Befestigung des Eye-Trackers am HMD, wurde im Rahmen dieser Arbeit konzipiert und konstruiert. Der erste Ansatz zur Befestigung der Miniatur CCD Kamera am Kopf eines Probanden bestand aus der Kombination von handelsüblichen Komponenten, die in jedem Baumarkt erhältlich sind. Der gebogene Edelstahl draht wurde gewählt, da das Verhalten zwischen plastischer und elastischer Verformung, also die Elastizitätsgrenze, zunächst als optimal betrachtet wurde, um durch Verbiegen des Drahtes eine geeignete geometrische Lage der CCD Kamera bestimmen zu können. Als Halterung diente eine Schweißbrille, der die Gläser entnommen wurden.



Abbildung 28: Eye-Tracker Befestigung I

Nachdem die Funktion des Eye-Trackers mit der mitgelieferten Software überprüft wurde und die Daten des Blicks in Augenschein der grafischen Anzeige plausibel erschienen, ging man zum nächsten Schritt über: der Befestigung des Eye-Trackers am HMD. Schnell stellte sich heraus, dass

AUSWAHL UND INTEGRATION DES EYE-TRACKING

eine solide Verankerung notwendig ist, da sich nach der Justierung der Kamera ihre Lage zum Display nicht mehr verändern sollte. Zusätzlich muss aber auch die Justierbarkeit erhalten bleiben, da auf Grund der Einzigartigkeit eines jeden Menschen die Kameraposition seiner Anatomie so angepasst werden muss, dass sie das ganze Auge im Bild hat.

Ein weiteres Kriterium beim Einrichten ist das Ausrichten der Infrarot-Lichtquelle auf das Auge. Dabei ist ein Mindestabstand einzuhalten, damit die Intensität der Strahlung, die in etwa quadratisch im Verhältnis zum Abstand abnimmt, das Auge nicht austrocknet. Die spektrale Strahlungsdichte ist gering. Folgeschäden bei zu geringem Abstand sind nicht bekannt. Jedoch merkt man einen zu geringen Abstand schnell, da sich die Frequenz des Lidschlags beim Probanden drastisch erhöht wodurch das Auge seiner Austrocknung entgegensteuert. Außerdem ist auch besonders auf eine möglichst homogene Ausleuchtung der Augenhöhle zu achten, damit dem Bildverarbeitungsalgorithmus der Eye-Tracking Software ein möglichst großes Kontrastspektrum zur Verfügung steht. Diese Anforderung ist essentiell und nicht immer leicht zu realisieren, da manchmal, anatomisch bedingt, von den Wimpern Schatten geworfen werden, die es zu vermeiden gilt, damit der Algorithmus in allen Blicksituationen ein optimales Bild von der CCD Kamera geliefert bekommt und unverfälschte Daten extrahieren kann. Sicherlich ist für die Qualität auch die Auswahl eines geeigneten Algorithmus ausschlaggebend.

Mit zunehmender Erfahrung im Justieren und Konfigurieren des Eye-Trackers kristallisierte sich die Forderung nach einer besseren Befestigungsmethode heraus. Der Bewegungsspielraum war gering und das Handhaben zum Justieren der Kamera- und Beleuchtungslage war zu umständlich.

Die nächste Idee zur Befestigung wurde mittels einer umgebauten Mikrofonhalterung, wie in der Abbildung 29 zu sehen, realisiert.

Die Kamera ist ausschließlich mit kleinen Kabelbindern an der Halterung befestigt worden.



Abbildung 29: Eye-Tracker Befestigung II

4.2 ENTWICKLUNG DER HMD BEFESTIGUNG

Auch das andere Ende der Halterung wurde mit Kabelbindern an soliden Teilen des HMD befestigt.



Abbildung 30: Eye-Tracker Befestigung II (vorne)

Die aktuelle und dritte Konstruktion ist noch solider und ermöglicht eine einfache Feinjustierung. Wie in der Abbildung 31 zu sehen ist, kommt als Halterung ein Kühlmittelschlauch zum Einsatz, der normalerweise in

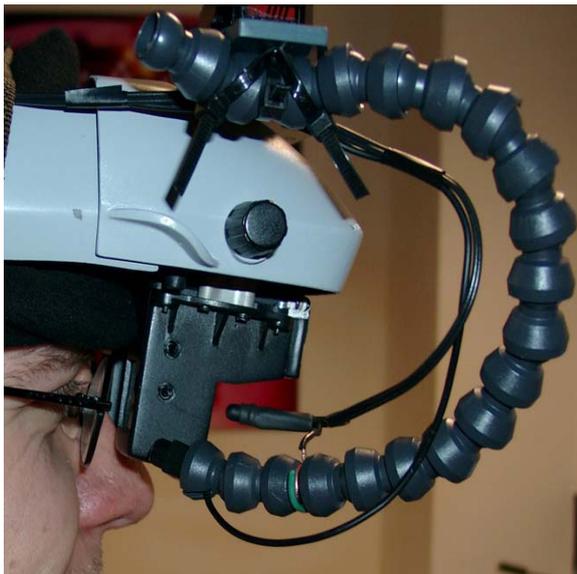


Abbildung 31: Eye-Tracker Befestigung III

Werkzeugmaschinen für die Kühlmittelzufuhr sorgt. Er ist starr und ermöglicht die Anpassung der Lage fast optimal. Für die Verbesserung der Feinjustierung wurde mit dem vorletzten Kugelgelenk des Schlauchs experimentiert. Versuche, die Reibung des Kugelgelenks von dem betroffenen Glied mit Fett oder Teflonspray zu minimieren, damit es sich leicht justieren lässt, waren erfolglos. Der Versuch, den Radius des Gelenks mit Sandpapier zu verkleinern und dadurch ein leichtgängigeres Gelenk zur Feineinstellung zu erhalten, gelang.

Nachdem die Integration der Hardware erfolgreich durchgeführt wurde und in mehreren Entwicklungsschritten eine Lösung gefunden werden konnte, stand als nächstes die Integration der Eye-Tracking Software in das Morgan Framework an.

4.3 Framework-Integration des Eye-Trackers

Als strukturelle Basis des Frameworks ist die CORBA-Spezifikation zu sehen, die von unterschiedlichen Herstellern interpretiert und implementiert wird. Der folgende Abschnitt *4.3.1 CORBA und das Morgan Framework* soll einen Überblick liefern und die bestehende Struktur des Frameworks näherbringen, da für die Integration des Eye-Trackers in das Morgan Framework Kenntnisse über die einzelnen Komponenten, die das Framework ausmachen, erforderlich sind. Auch das Zusammenspiel der Komponenten sowie die Mechanismen, die zwischen ihnen wirken, sind für die Integration von Bedeutung und werden im Abschnitt *4.3.4 Eye-Tracker* erwähnt. Die zeitkritische Betrachtung der dynamischen Daten und die Synchronisierung der Systemzeit erfolgt im Abschnitt *4.3.2 Zeitnähe und Synchronisation*. Darauf folgt dann der Abschnitt *4.3.3 Geometrie*, der die Grundlagen für die geometrische Transformation dokumentiert sowie die durch den Eye-Tracker ermittelte Blickrichtung in Zusammenhang mit der Mixed Reality Anzeige bringt und für die Anwendung in unterschiedlichen Bezugssystemen umformt. Anschließend stehen Entwurf und Implementieren des *4.3.4 Eye-Tracker* im Vordergrund, worauf die Implementierungen von *4.3.5 Tracker Combiner* und *4.3.6 Eye-Manager* folgen. Abschließend wird auf die Fadenkreuzanzeige im Viewer eingegangen.

4.3.1 CORBA und das Morgan Framework

CORBA steht für Common Object Request Broker Architecture, dabei handelt es sich um eine Spezifikation der OMG (Object Management Group) [url3]. Die herstellerunabhängige Architektur konzipiert eine Infrastruktur, die es erlaubt, über Netzwerke verteilte Anwendungen weitestgehend plattformunabhängig zu realisieren. Zusätzlich wird der Einsatz von verschiedenen Programmiersprachen zur Implementierung einer verteilten Anwendung von der Spezifikation unterstützt. Eine solche Anwendung kann unter Berücksichtigung wichtiger Eigenschaften, wie Skalierbarkeit und Fehlertoleranz, konzipiert werden. Die Implementierung von CORBA bleibt den verschiedenen Herstellern überlassen. Das Konzept der universellen Schnittstelle ermöglicht es auch, die verschiedenen Implementierungen der Hersteller für unterschiedliche Programmiersprachen zu kombinieren. Die Kommunikation von CORBA durch das Netzwerk erfolgt über das Standardprotokoll IIOP. Spezialisierte Definitionen von CORBA ergänzen die Architektur um Echtzeitfähigkeit und ermöglichen den Einsatz auf Embedded-Systemen.

4.3 FRAMEWORK-INTEGRATION DES EYE-TRACKERS

In der folgenden Tabelle sind andere Technologien sowie ihre Nachteile, im Vergleich zu CORBA aufgeführt.

	Technologie	Nachteile
RPC	Remote Procedure Call	nicht objektorientiert TCP/IP-Bindung
DCOM	Distributed Common Object Modell	Betriebssystem abhängig Microsoft Windows
RMI	Remote Methode Invocation	Software abhängig Java, TCP/IP-Bindung
CORBA	Common Object Request Broker Architecture	-

Aus programmiertechnischer Sicht bietet sich die Möglichkeit, Objekte auf verschiedene Systeme zu verteilen, die in der Summe eine Anwendung ausmachen. Ein nicht zu verachtender Vorteil dabei ist, dass sich die Performanz durch die geschickte Konzipierung der verteilten Anwendung auf unterschiedlich leistungsfähige Systeme enorm steigern lässt. Im Vergleich zu einem einzelnen Computersystem kann die Anwendung auf dem verteilten System skaliert werden und durch Load-Balancing ist es möglich, die Rechenlast zwischen den Systemen effizient zu verteilen. Die Verbindung zwischen den Systemen wird weitgehend von CORBA abstrahiert, dabei spielt die IDL (Interface Definition Language) für die Definition der Systemschnittstellen, den Interfaces der Computer eine zentrale Rolle.

Die IDL definiert, wie der Name schon sagt, eine Schnittstelle zwischen Objekten und liefert dadurch auch die Basis für den Aufruf von Objekten, die in einem entfernten System existieren. Weiterhin ermöglicht die Schnittstellensprache den Einsatz von unterschiedlichen Programmiersprachen in einer verteilten Anwendung. Das bedeutet, ein Java- oder Python-Objekt kann ein C++ Objekt aufrufen oder umgekehrt – natürlich ist dabei fast jede Kombination von unterschiedlichen Sprachen möglich, für die es eine CORBA-Anbindung gibt. Bei der Implementierung wird zuerst über IDL die Schnittstelle definiert und anschließend mit einem IDL-Kompiler das CORBA-Gerüst erstellt, das den Aufruf von verteilten Objekten ermöglicht. Der Kompiler wird für die Programmiersprache, in der das Objekt implementiert werden soll, ausgewählt. Das Ergebnis der Kompilierung sind mehrere Dateien, die die Schnittstelle abstrahieren und je nach Kompiler eine Basis des Codes für die Objektimplementierung liefern. Die Kompiler legen den Code so an, dass man von einem aus IDL kompilierten Objekt ableitet, um die Implementierung des Objektes zu realisieren. Das Resultat ist ein CORBA-Objekt, das über die in der IDL spezifizierten Schnittstelle angesprochen werden kann.

Die Anfragen an CORBA-Objekte innerhalb einer verteilten Anwendung, die auch über die lokalen Systemgrenzen hinweg erfolgen können und

AUSWAHL UND INTEGRATION DES EYE-TRACKING

grundsätzlich über die IDL-Schnittstelle laufen, gehen zunächst an den Broker. Die volle Bezeichnung lautet: Object Request Broker (ORB) und bedeutet sinngemäß übersetzt: „Händler für Objektanfragen“. Er regelt die Kommunikation zwischen den verteilten Objekten und hält Referenzen zu ihnen. In der Abbildung 32 wird das Funktionsprinzip dargestellt. Für den Anwendungsentwickler ist es dadurch möglich, eine objekt-orientierte verteilte Anwendung zu implementieren.

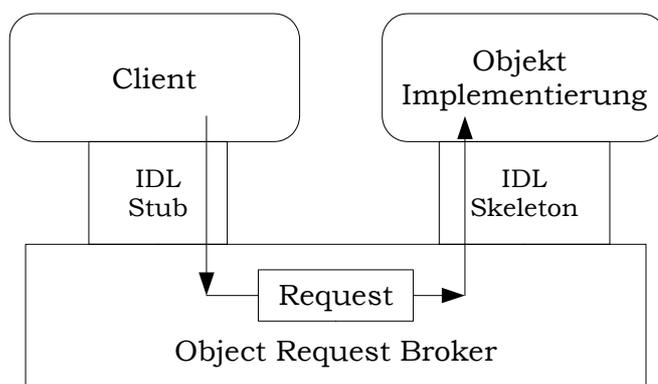


Abbildung 32: Anfrage (call) vom Client an das implementierte Objekt

Das Morgan Framework baut auf die CORBA-Implementierung von Orbacus auf und nutzt zur Implementierung gegenwärtig ausschließlich die Programmiersprache C++. Die „Zentrale“ des Morgan Frameworks ist der ORB, der dafür sorgt, dass zur Laufzeit des Programmes angeforderte Referenzen zu Dunits ermittelt werden. Eine Dunit (distributed unit) ist eine abstrakte Einheit, ein CORBA-Objekt, das eine seiner Implementierung entsprechende Funktionalität zur Verfügung stellt. Eine Dunit entsteht in einer sogenannten Factory, die vom Broker die Anweisung erhält, eine neue Dunit zu erstellen. Factories ermöglichen zur Laufzeit der Anwendung, das Erstellen von Objektinstanzen, die dank CORBA auf unterschiedlichen Systemen laufen können. Der eine Rechner könnte zum Beispiel für die Bildverarbeitung des Eye-Trackers genutzt werden, während auf einem anderen Rechner die Bilder für die Anzeige gerendert werden. Weiterhin könnte man sich zusätzliche Rechner vorstellen, die aufwendige Berechnungen für die angestrebte Anwendung durchführen. Alle Rechner arbeiten also mehr oder weniger für eine Anwendung zusammen. Auf den einzelnen Rechnern entstehen in Prozessen Instanzen von unterschiedlichen Dunits so z.B. Tracker, die über das Morgan Framework vernetzt werden.

Damit für die Entwicklung von MR-Anwendungen nicht unbedingt CORBA-Kenntnisse erforderlich sind und MR-Programmierer außerdem nicht mit den Mechanismen des Morgan Frameworks vertraut sein müssen, gibt es die AR-API. Über diese API können alle notwendigen Funktionen angesprochen werden, die zur Umsetzung einer MR-Anwendung notwendig sind. Die Abbildung 33 zeigt den Zusammenhang zwischen den Komponenten der API.

4.3 FRAMEWORK-INTEGRATION DES EYE-TRACKERS

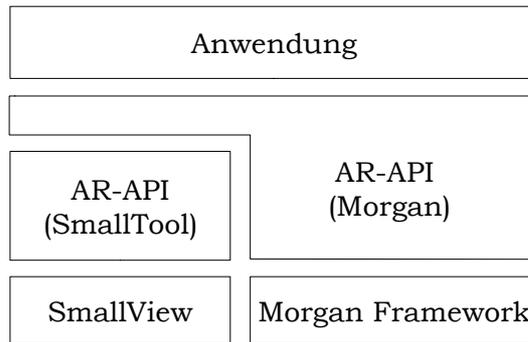


Abbildung 33: AR-API und das Morgan Framework

Die meist externen Geräte, die man auch als Devices bezeichnet, werden über das Morgan Framework eingebunden. Dazu zählen hauptsächlich Tracker, die kontinuierliche Daten liefern. Ihre Aufgabe wurde im Abschnitt 3.1 *Mixed Reality Technologie* besprochen. Es gibt aber auch einen Sonderfall: den Viewer SmallView. Historisch bedingt, kommuniziert die AR-API mit ihm über eine TCP/IP Verbindung. SmallView ist unter anderem für die Berechnung der virtuellen Bilder, die im HMD angezeigt werden, zuständig. Der Informationsfluss zwischen dem Benutzer und einer MR-Anwendung ist zusammenfassend in Abbildung 34 visualisiert. In diesem Überblick ist auch die Integration des Eye-Trackers in die Hardware und in das Framework dargestellt.

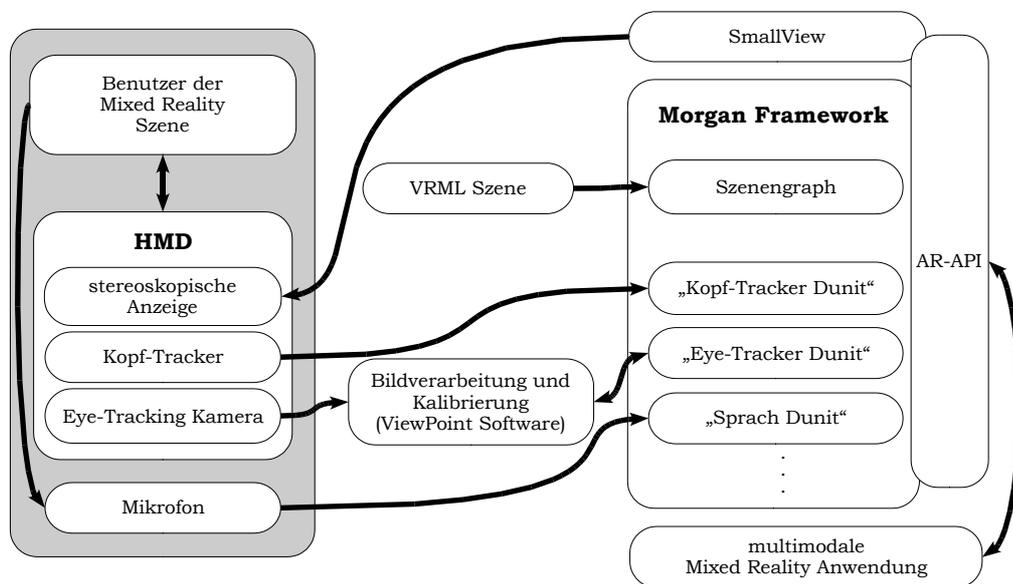


Abbildung 34: Informationsfluss zwischen Benutzer und Anwendung

Für das Zusammenspiel der einzelnen Framework-Komponenten ist eine zeitkritische Betrachtung der dynamischen Daten und die Synchronisierung der Zeit aller beteiligten Computersysteme erforderlich.

4.3.2 Zeitnähe und Synchronisation

In einem verteilten System, das in Echtzeit im Sinne der Wahrnehmung läuft, ist die Synchronität der beteiligten Computersysteme wichtig, damit zwischen den einzelnen Rechnern eine gemeinsame Basis existiert, mit der einheitlich gerechnet werden kann. Die Betrachtung der Toleranz, die für alle auftretenden Latenzen zur Verfügung steht und insgesamt innerhalb von ca. 16-50 ms zu sehen ist, wurde im Abschnitt 3.1 *Mixed Reality Technologie* durchgeführt.

Um die Zeit zwischen vernetzten Rechnern zu synchronisieren, gibt es das SNTP Protokoll, das im RFC 2030 spezifiziert ist [url18]. Da alle für das Framework genutzten Computer das Betriebssystem XP einsetzen, war es notwendig, eine Applikation zu finden, die in regelmäßigen Abständen die Rechneruhren so genau wie möglich aufeinander abstimmt. Die Software „Windows Time Synchronizer“ [url17] wurde nach einer kurzen Recherche als geeignet betrachtet, da sie in festlegbaren Intervallen die Zeit bis auf die Millisekunde genau zwischen den Rechnern synchronisiert. Eine Messung der Latenzen von übertragenen Daten im Morgan Framework wurde nicht durchgeführt, da die geeigneten Mittel für einen zuverlässigen Messaufbau nicht zur Verfügung standen.

Für die Integration des Eye-Trackers in eine Anwendung ist es erforderlich, die relativ zum Kopf des Benutzers ermittelten geometrischen Daten mit den absoluten Daten des Head-Trackers zu kombinieren (siehe 4.3.3.2 *Blickrichtungs-Transformation* und 4.3.5 *Tracker Combiner*). Um die Kombination der Daten in einer zeitnahen Umgebung zu realisieren, ist eine Betrachtung der Latenzzeiten und Frequenzen sowie deren Auswirkung auf die Daten notwendig.

Zunächst wurde davon ausgegangen, dass auf Grund von Latenzen, die durch die Netzwerkübertragungen oder durch Rechenzeiten entstehen, die Notwendigkeit besteht, einen Ausgleich für die gealterten geometrischen Daten zu schaffen. Weiterhin ging die Überlegung davon aus, dass Daten (Position und Orientierung) aus verteilten diskreten Quellen, selbst bei synchronisierten Systemzeiten und gleichen Frequenzen, nicht ohne weiteres miteinander kombiniert werden können, da kein Mechanismus im Framework vorhanden war, der die Frequenzen miteinander im Millisekundenbereich synchronisiert und sicherstellt, dass Abweichungen innerhalb eines gewissen Toleranzbereiches bleiben. Das Ergebnis der Überlegung war, einen Algorithmus zu implementieren, der in die Zukunft der Daten schaut, also eine Prediktion für einen bestimmten Zeitpunkt durchführt und das am besten gleichzeitig für mehrere Datenquellen. Gleichzeitig bedeutet in diesem Kontext: so schnell wie möglich, da Echtzeit auf Grund der benötigten Rechenzeit nicht möglich ist und Echtzeit nur in einem unendlich kleinen Moment, im Jetzt, existiert – vor ihm liegt die Zukunft und nach ihm die Vergangenheit.

4.3 FRAMEWORK-INTEGRATION DES EYE-TRACKERS

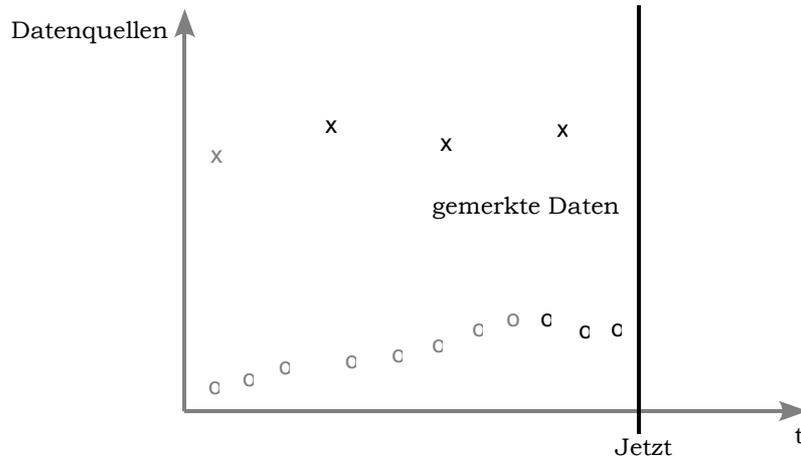


Abbildung 35: Datenreihen zur Prädiktion mit Polynom 2. Grades

Das Resultat war eine Implementierung des Newtonschen Polynoms, um von nicht synchronen, diskreten Datenreihen mit unterschiedlichen Frequenzen einen Wert zu einem für alle Datenreihen einheitlichen Zeitpunkt zu erhalten. Der implementierte Mechanismus hat ein Gedächtnis, einen Speicher, der sich entsprechend dem eingestellten Polynomgrad, die für die Berechnung notwendigen vergangenen Werte der Datenreihen merkt. Zum konfigurierbaren Zeitpunkt, der durch eine eingestellte Frequenz, mit der die Datenreihen kombiniert werden, aus der aktuellen Systemzeit ermittelt wird, erfolgt die Berechnung der gegenwärtigen Werte. Das Ergebnis der Extrapolation ist in Abbildung 35 mit *jetzt* gekennzeichnet. Um die Datenreihen mit keinem weiteren Fehler zu belasten, wurden sie in der ersten Implementierung des Tracker Com-

```
double Morgan::TrackerNewtonPrediction_impl::DataPump::predict(
    double* xi, int* n, double* x, double* y )
{
    // divided differences
    double u[MAX_POLYNOM_POWER][MAX_POLYNOM_POWER];
    // coefficients
    double c[MAX_POLYNOM_POWER];
    // predicted value
    double yi;
    // counter
    int k;
    // calculate coefficients
    for ( int j ( 0 ); j <= (*n); j++ )
    {
        for ( k = 0 ; k <= j; k++ )
        {
            if ( k == 0 )
                u[j][k] = y[j];
            else
            {
                if ( ( x[j] - x [j - k] ) == 0 )
                    u[j][k] = 1.7E308; // largest possible value instead of infinity !
                else
                    u[j][k] = ( u[j][k - 1] - u[j - 1][k - 1] ) / ( x[j] - x[j - k] );
            }
        }
        c[j] = u[j][j];
    }
    // predict
    yi = c[ (*n) ];
    k = ( *n ) - 1;
    for ( ; k >= 0; k-- ) yi = c[k] + ( ( *xi ) - x[k] ) * yi;
    return yi;
}
```

Abbildung 36: Quellcode Polynomextrapolation

AUSWAHL UND INTEGRATION DES EYE-TRACKING

biners direkt im Anschluss an ihre Extrapolation durch eine geometrische Transformation miteinander kombiniert. In Abbildung 36 berechnet der Algorithmus im ersten Teil die Koeffizienten des Polynoms, die dann in der letzten for-Schleife für die Extrapolation zum Zeitpunkt x_i genutzt werden.

Das Ergebnis der kombinierten Daten war leider nicht zufriedenstellend, da das ohnehin schon sprunghafte Signal von der Augenbewegung nach der Prädiktion noch unruhiger wirkte als es im Viewer visualisiert und in Augenschein genommen wurde.

Die Implementierung einer Splinextrapolation hätte zusätzlich zur Gewichtung der vorhandenen Daten geführt und bestimmt zu einer leichten Verbesserung der Prädiktion beigetragen. Sie wurde nicht getestet.

Anstatt mit einer Extrapolation in die Zukunft der Daten zu schauen, die durch das Verfahren einen nicht akzeptablen Fehler erhielten, wurde beschlossen mit den unverfälschten echten Werten zu arbeiten. Die verbleibenden zeitlichen Differenzen, zwischen Werten, deren Datenreihen aus verschiedenen Quellen hervorgehen, entstehen durch unterschiedliche Übertragungszeiten und unterschiedliche Frequenzen. Will man nun diese Datenreihen miteinander kombinieren führt das zu einer Ungenauigkeit des Ergebnisses, da zum Zeitpunkt der Kombination nur die letzten gemerkten diskreten Daten aus der Vergangenheit mit unterschiedlichen Zeitstempeln zur Verfügung stehen. Der entstandene Fehler, den das aus der Kombination resultierende Signal enthielt, war nicht wahrzunehmen, als es im Viewer visualisiert wurde. Deshalb fiel die Wahl auf dieses Verfahren zur Kombination von absoluten und relativen geometrischen Daten. Die Implementierung ist im Abschnitt 4.3.5 *Tracker Combiner* beschrieben. Am Ende des nun folgenden Abschnitts: 4.3.3 *Geometrie* ist die Abfolge von geometrischen Transformationen, die fürs Kombinieren der relativen mit den absoluten Daten notwendig sind, beschrieben.

4.3.3 Geometrie

Alle sichtbaren virtuellen Objekte werden geometrisch in einem Raum berechnet, der sich aus dem Blickfeld des Betrachters definiert und mit dem ihm entsprechenden realen Raum synchronisiert wird. Damit die Berechnung erfolgen kann, sind Position und Orientierung vom Kopf des Betrachters notwendig, die über den Head-Tracker ermittelt und digitalisiert werden. Die geometrischen Betrachtungen beziehen sich auf den dreidimensionalen Raum. Die beteiligte Zeit bezieht sich immer näherungsweise auf die Gegenwart, es kann deshalb von Echtzeit im Sinne

4.3 FRAMEWORK-INTEGRATION DES EYE-TRACKERS

der Wahrnehmung geredet werden (siehe 3.1 *Mixed Reality Technologie*). Zur Berechnung von Lagebeziehungen, die Position und Orientierung enthalten, sind geometrische Transformationen notwendig, die im nachfolgenden Abschnitt dokumentiert sind. Anschließend steht die geometrische Betrachtung und Umwandlung der Blickrichtungsdaten im Vordergrund.

4.3.3.1 Geometrische Transformation

Betrachtet man einen Raum mit drei Dimensionen (R^3) so kann jeder Punkt durch die Angabe der Koordinaten x , y und z in einem kartesischem Koordinatensystem identifiziert werden. Seine Orientierung ergibt sich aus den Winkeln α , β und γ . Jeder Punkt im Raum lässt sich durch einem Vektor \vec{v} , der vom Ursprung des Koordinatensystems ausgeht, repräsentieren.

$$\vec{v}' = A * \vec{v}$$
$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & t_x \\ a_{21} & a_{22} & a_{23} & t_y \\ a_{31} & a_{32} & a_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Formel 1: Affine Transformation

Eine geometrische Transformation legt im Allgemeinen die Beziehung zwischen zwei Punktmengen fest. So lässt sich zum Beispiel auch ein Punkt verschieben, wenn man eine Transformation auf ihn anwendet, das Ergebnis wäre dann \vec{v}' . Als affine Transformation bezeichnet man die linearen Transformationen \mathbf{a} Rotation, Skalierung, Dehnung und Scherung, außerdem auch die nichtlineare Translation \mathbf{t} [url9].

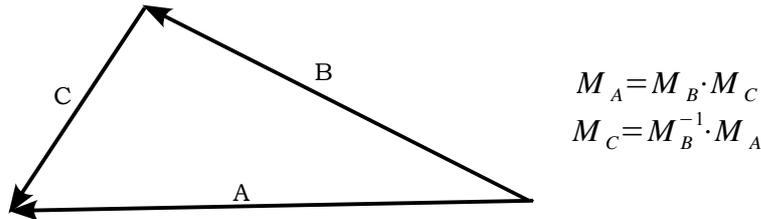
$$v' = M * v$$
$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & t_x \\ a_{21} & a_{22} & a_{23} & t_y \\ a_{31} & a_{32} & a_{33} & t_z \\ p_x & p_y & p_z & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Formel 2: Projektion

Die affine Transformation wird in Formel 1 durch eine homogene 4x4 Matrix \mathbf{A} dargestellt. In der Formel 2 legt der Parameter \mathbf{p} die Projektionseigenschaften der Matrix \mathbf{M} fest.

AUSWAHL UND INTEGRATION DES EYE-TRACKING

Das eben besprochene Format der 4x4 Matrix soll hier als Standard betrachtet werden, da sie sowohl Position wie auch Orientierung enthalten kann und in dieser Form in der vorhandenen Implementierung genutzt wird. Die Parameter zur Bestimmung der Lage sind die wichtigsten für die meisten geometrischen Berechnungen, die im Rahmen dieser Arbeit anfallen. Um Matrizen anwenden zu können, sind einige grundlegende Regeln zu beachten. Wie in Formel 3 zu sehen, ist eine Matrix zu invertieren, wenn sich die Richtung des Vektors umkehrt.



Formel 3: Vektoraddition

Wendet man Translation und Rotation in separaten Matrizen an, so ist die richtige Reihenfolge, wie in Formel 4 dargestellt, einzuhalten.

$$M = M_{Translation} \cdot M_{Rotation}$$
$$M^{-1} = M_{Rotation}^{-1} \cdot M_{Translation}^{-1}$$

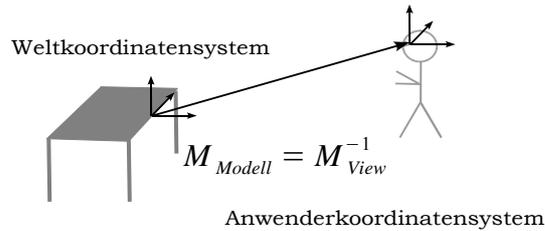
Formel 4: Reihenfolge invertierter Matrizen

4.3.3.2 Blickrichtungs-Transformation

Die Blickrichtung wird aus den gewonnenen Daten der Bildverarbeitung, die über die ViewPoint Software der Firma Arrington Research justiert werden kann, und den Daten der Betrachtungssituation, wie Bildschirmauflösung und Abstand zur Bildebene, sowie unter der Berücksichtigung von Kalibrierungswerten, errechnet. Die von der Eye-Tracking Software gelieferten geometrischen Daten sind also kalibriert, sofern zu Beginn die Kalibrierung durchgeführt wurde (siehe Abschnitt: 5.3 *Eye-Tracker Kalibrierung*). Das Source Development Kit (SDK) des Eye-Trackers bietet Zugriff auf eine Funktion, die die aktuelle Blickrichtung bzw. die Augenorientierung liefert, indem sie die beiden Winkel Yaw und Pitch übermittelt. Für die Visualisierung der aktuellen Blickrichtung im HMD genügen die relativen Daten der Augenorientierung, da die Anzeige, genau wie auch der Eye-Tracker, am Kopf des Anwenders befestigt ist. Für eine Anwendung hingegen wäre es wünschenswert, die aktu-

4.3 FRAMEWORK-INTEGRATION DES EYE-TRACKERS

elle Blickrichtung in absoluten Koordinaten zu erhalten, da sich der Anwender absolut zum Ursprung des Koordinatensystems bewegt. Auch für die Anwendung der Blickrichtung zur Interaktion wäre dieses Bezugssystem von Vorteil, das durch die Kombination der relativen Lage des Auges mit der absoluten Lage des Kopfs entsteht. Auf Grund dieser Überlegung entstand die Forderung nach der Möglichkeit die Daten unterschiedlicher Tracker miteinander kombinieren zu können. Die Synchronisations-Problematik die dadurch entsteht, wurde im Abschnitt: *4.3.2 Zeitnähe und Synchronisation* besprochen.



Formel 5: Modell-View-Matrix

Aus Sicht der Geometrie sind eine Reihe von Transformationen notwendig, um die relativen mit den absoluten Daten miteinander zu kombinieren. In der Abbildung von Formel 5 ist die Lagebeziehung zwischen dem Modell, das direkt im Welt- oder Ursprungskoordinatensystem entsteht, und der View, der Position und Orientierung vom Kopf des Rezipienten bzw. der „Anzeige“ des Viewers bildlich und durch Matrizen dargestellt. Wie schon besprochen, wird die Lage des Kopfs mittels eines Trackers in die virtuelle Realität übertragen (siehe *3.1 Mixed Reality Technologie*). Der Ursprung eines Mixed Reality Systems wird bei der Installation festgelegt. Die Matrizen in der Abbildung repräsentieren die mathematische Beziehung und liefern einen Ausgangspunkt für die weitere Berechnung der absoluten Blickrichtung des Rezipienten. Weiterhin ist der Vollständigkeit halber zu erwähnen, dass der Bezugspunkt der

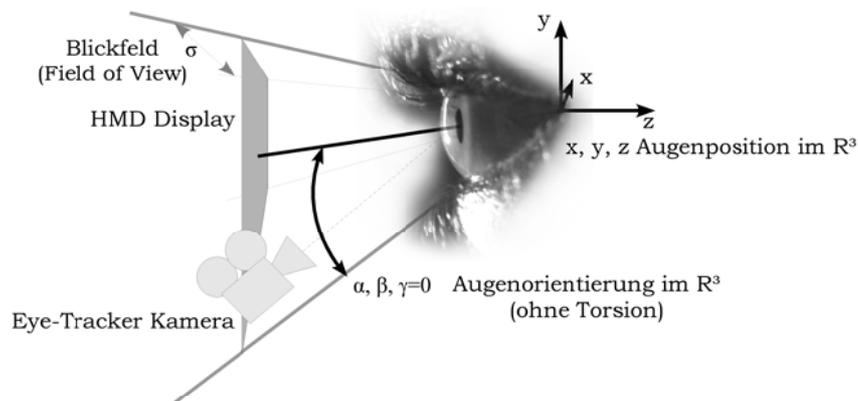


Abbildung 37: Augenometrie

AUSWAHL UND INTEGRATION DES EYE-TRACKING

View-Matrix genau mittig, zwischen den beiden Brennpunkten, die sich vor den Retinas des Rezipienten befinden, liegt.

Das Auge ist im Detail, mit überlagerter geometrischer Information, in Abbildung 37 zu sehen. Als geometrische Information ist neben der Orientierung auch die Position des Auges von Bedeutung, da in Hinblick auf die bevorstehende Kombination der absoluten und relativen Trackerdaten der Offset des Auges, zum Bezugspunkt des Kopfs, zu berücksichtigen bleibt. Er entspricht dem halben Augenabstand und ist im Mittel anatomischer Eigenheiten bei etwa 3 cm anzusiedeln.

In der Ausgangssituation werden die Daten kontinuierlich von den beiden Trackern nach Position und Orientierung getrennt geliefert. Hier werden sie von Matrizen repräsentiert:

$$\begin{aligned}M_{Kopf\ absolut} &= M_{Kopf\ Translation\ absolut} \cdot M_{Kopf\ Rotation\ absolut} \\M_{Blick\ relativ} &= M_{Blick\ Translation\ relativ} \cdot M_{Blick\ Rotation\ relativ}\end{aligned}$$

Formel 6: Absolute und relative Transformationen

Die Berechnung der absoluten Blickrichtung erfolgt durch die folgende Kombination der Matrizen:

$$M_{Blick\ absolut} = M_{Kopf\ absolut} \cdot M_{Blick\ relativ}$$

Formel 7: Kombination von absoluten und relativen Koordinaten

Im Einzelnen ergeben sich daraus die folgenden Translationen und Rotationen:

$$\begin{aligned}M_{Blick\ Translation\ absolut} &= M_{Kopf\ Translation\ absolut} \cdot M_{Kopf\ Rotation\ absolut} \cdot M_{Blick\ Translation\ relativ} \\M_{Blick\ Rotation\ absolut} &= M_{Kopf\ Rotation\ absolut} \cdot M_{Blick\ Rotation\ relativ}\end{aligned}$$

Formel 8: Kombination von absoluten und relativen Koordinaten getrennt nach Position und Orientierung

Die Daten werden vom Combiner, der sich wie ein Tracker verhält, nach Position und Orientierung getrennt weiter geschickt. Der Vollständigkeit wegen sei noch erwähnt:

$$M_{Blick\ absolut} = M_{Blick\ Translation\ absolut} \cdot M_{Blick\ Rotation\ absolut}$$

Formel 9: Ergebnis der Kombination

Die Kombination der absoluten und relativen Trackerdaten ist durch die Anwendung von einer Reihe an geometrischen Transformationen zu bewerkstelligen. Die Implementierung der folgenden Matrizen-Kombination, die zur gewünschten Transformation führt, erfolgt im Abschnitt 4.3.5 *Tracker Combiner*.

4.3.4 Eye-Tracker

Die Implementierung des Eye-Trackers bindet über das SDK die View-Point Software der Firma Arrington Research in das Morgan Framework, wie in Abbildung 34 zu sehen, ein. Dadurch ist es möglich, vom Framework aus den Zugriff auf wichtige Funktionen zu erhalten und die im Umfeld des Auges ermittelten Daten in vorhandene Mechanismen des Frameworks zu integrieren.

Dunit steht für Distributed Unit und bedeutet so viel wie verteilte Einheit. Sie entsteht zur Laufzeit der Anwendung in einer sogenannten Factory, in der sie bei Bedarf „hergestellt“ bzw. instantiiert wird, wie dies einleitend im Abschnitt: *4.3.1 CORBA und das Morgan Framework* festgestellt wurde. Eine Eye-Tracker Dunit ist eine Implementierung, die den Treiber und die damit verbundene Software für einen bestimmten Eye-Tracker abstrahiert. Die Arrington Eye-Tracker Dunit ist wie alle Dunits ein CORBA-Objekt. Sie greift auf die Daten, die von der Arrington Hard- und Software vom Auge bzw. Blick des Anwenders ermittelt werden über das SDK (Software Development Kit) mittels C++ zu. Das passiert in einem eigenen Thread, der sogenannten „Datapump“. Sie liest die Daten in einer Schleife kontinuierlich ein und schickt sie an alle Interessenten im Framework mit einer konfigurierbaren Frequenz weiter. Das allen Trackern zugrunde liegende Publish und Subscribe Prinzip sorgt dafür, dass die Empfänger der Trackerdaten bekannt sind und ermöglicht dadurch das „Pushen“ der Daten. Der Vorteil dieses Mechanismus ist, dass die zeitkritischen Daten ohne unnötige Verzögerung weiterverarbeitet werden können.

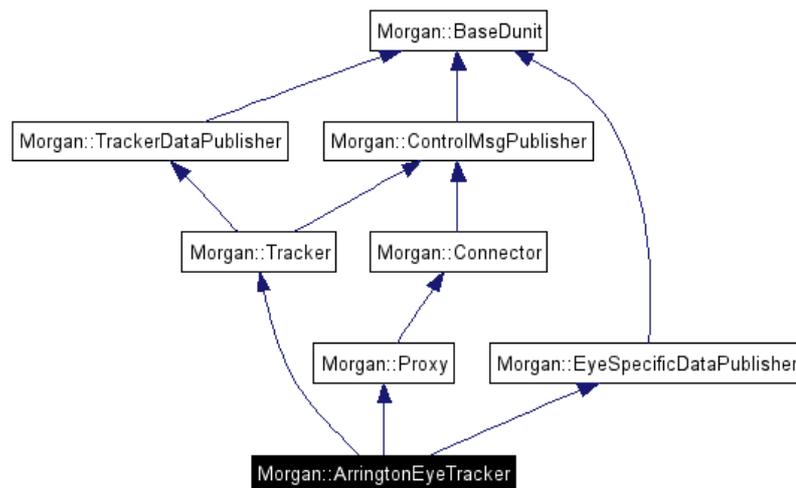


Abbildung 38: Vererbungsschema Arrington-Eye-Tracker

Die Vererbung von C++ Objekten vereinfacht die Implementierung, da das Erben vorhandener Mechanismen und implementierten Codes, eine Basis von Funktionalität liefert, auf die das neu zu implementierende

AUSWAHL UND INTEGRATION DES EYE-TRACKING

Objekt zurückgreifen kann. In der Abbildung 38 sind alle Vererbungen an den Arrington Eye-Tracker dargestellt. Der Eye-Specific-Data-Publisher wurde extra für den Einsatz von Eye-Tracking im Morgan Framework implementiert, weil neben den kontinuierlichen geometrischen Daten, die über die geerbten Mechanismen des Trackers an das Framework gelangen, auch kontinuierliche Daten, die aus dem für Eye-Tracking spezifischen Kontext hervorgehen, übertragen werden müssen. Es war also erforderlich einen Eye-Specific-Data-Publisher zu implementieren, von dem der Eye-Tracker zusätzliche Funktionalität erbt. Wie die Bezeichnung vermuten lässt, publiziert der Eye-Specific-Data-Publisher die diskreten kontinuierlichen Daten genau wie ein Tracker nach dem Publish und Subscribe Prinzip. Übermittelt wird die aktuelle Geschwindigkeit des Blicks, die Zeit der Fixierung des Auges auf eine Richtung und eine Kennzahl, aus der die Qualität der übermittelten Daten hervorgeht. Durch Auswerten der Kennzahl kann unter anderem das Zwinkern des Auges erkannt werden.

Zur Fehlersuche war es erforderlich geworden, weiterhin für die augenspezifischen Daten eine Dunit zu implementieren, die als Subscriber, also als Empfänger der Daten, dafür sorgt, dass sie für die Anzeige in einer Console auf dem Bildschirm umgewandelt werden.

Der Zugriff auf die Kalibrierung und Funktionen zur Fernsteuerung der „ViewPoint-Software“ von der Firma Arrington wurde so implementiert, dass über die IDL alle notwendigen Zugriffe erfolgen können. Auch eine Methode zum Senden neutraler Trackingdaten wurde implementiert. Neutral bedeutet in diesem Zusammenhang, dass sie auf geometrische Transformationen keinen Einfluss haben.

Damit die vom Eye-Tracker gelieferten Blickrichtungsdaten, die relativ zum Kopf des Anwenders von der Eye-Tracking Dunit publiziert werden, auch in das Weltkoordinatensystem der Szene transformiert werden können und um eine generische Dunit für ähnliche Anwendungsfälle zu erhalten, war die Implementierung eines Tracker Combiners notwendig geworden.

4.3.5 Tracker Combiner

Ein Tracker Combiner sorgt dafür, dass die geometrischen kontinuierlichen Daten unterschiedlicher Tracker miteinander kombiniert werden können. Dabei sind situationsbedingt verschiedene Strategien für die Art und Weise der Kombination erforderlich. Nachdem die Strategie bei der Konzeption einer MR-Anwendung festgelegt wird und in der Regel zur Laufzeit nicht verändert werden muss, wurde entschieden, für jede

4.3 FRAMEWORK-INTEGRATION DES EYE-TRACKERS

Strategie eine separate Dunit zu implementieren. Ein weiterer Aspekt, der zu dieser Entscheidung beitrug, waren die unterschiedlichen Anforderungen an die Schnittstelle der Tracker Combiner, die nur durch unterschiedlich definierte IDLs umgesetzt werden konnten und sich aus der angewandten Strategie ergeben.

In dem Abschnitt 4.3.2 *Zeitnähe und Synchronisation* wurde der Einsatz von einer Prädiktion zum Ausgleich von Latenzen besprochen, die zunächst direkt in der Tracker Combiner Dunit implementiert wurde. Durch diese Vorgehensweise konnte die Latenz, die zwischen Prädiktion und dem Kombinieren der Daten entsteht, auf ein Minimum reduziert werden. Da jedoch die Prädiktion nicht die gewünschten Ergebnisse lieferte, wurde sie in die „Tracker-NewtonPrediction“ Dunit ausgelagert. Bei Bedarf kann sie dadurch für beliebige Tracker genutzt werden und ist nicht mehr an die Tracker Combiner Dunit gebunden. Wie in Abbildung 39 des Trackervererbungsschemas zu sehen, ist die erste Implementierung des gleichnamigen „TrackerCombiner“ von Tracker abgeleitet.

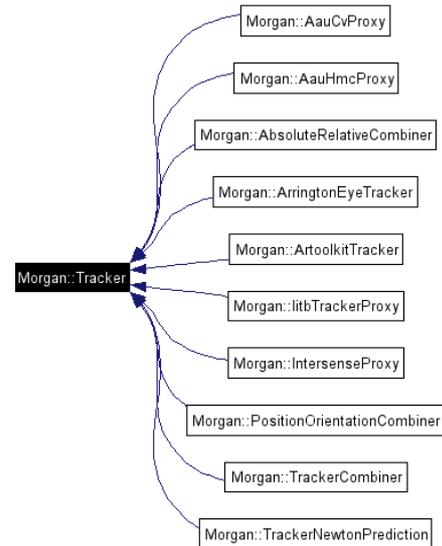


Abbildung 39: Trackervererbungsschema

Die darauffolgenden Implementierungen „*AbsoluteRelativeCombiner*“ und „*PositionOrientationCombiner*“ sind, wie der Vorgänger, Tracker. Die Bezeichnung der 2. Generation von Combinern gibt einen Hinweis auf die von ihnen angewandte Strategie zur Kombination der Trackerdaten. Da Combiner sowohl Tracker-Publisher wie auch Tracker-Subscriber sind, die auf der einen Seite von Trackern, die unter Umständen mit unterschiedlichen Frequenzen arbeiten, Daten gepusht bekommen und auf der anderen Seite mit einer vorgegebenen variierbaren Frequenz Daten kombinieren und publizieren, also pushen müssen, war es erforderlich, die eingehenden Daten erstmal zwischenspeichern. Zeitdifferenzen, die sich aus den unterschiedlichen Frequenzen ergeben werden dadurch ausgeglichen. Der Nachteil, der bei dieser Vorgehensweise verursacht wird, ist ein Fehler in den geometrischen Daten, der allerdings so gering ist, dass er von der Wahrnehmung des Anwenders nicht bemerkt wird. Bei der Konzeption des Speichers bzw. der Ablage für die eingehenden Daten wurde darauf geachtet, dass dieser kritische Bereich, der unterschiedlichen Threads simultan zur Verfügung stehen muss, keine inkonsistenten Zustände verursacht. Die Ablage enthält immer den aktuellsten Datensatz, auf den die Datapump, die für das Kombinieren und Publizieren der Daten mit variierbarer Frequenz zuständig ist, zugreift.

AUSWAHL UND INTEGRATION DES EYE-TRACKING

Da in einem Trackerdatenpaket des Morgan Frameworks Daten von mehreren Trackern zusammengefasst werden können, haben sie eine Id, um sie innerhalb des Pakets identifizieren zu können. Die Pakete von verschiedenen Trackern wiederum können über den Parameter `appData` unterschieden werden. Bei Eingang eines neuen Datenpakets in einen Combiner wird die jeweilige Id mit dem `appData` Parameter kombiniert. Dadurch entsteht ein eindeutiges Kennzeichen, das als Schlüssel mit den zugehörigen Trackingdaten in der Ablage zwischengespeichert wird. Die Ablage ist mit einer Map der C++ Standard Library [Jos02] realisiert. Über die IDL einer Combiner Dunit kann zur Laufzeit mittels des Kennzeichens festgelegt werden, welche Trackerdaten miteinander kombiniert werden sollen. Außerdem ist es auch notwendig, den durch die Kombination neu entstandenen Daten, eine neue Id zu geben. Der Zugriff auf die dafür zuständige Methode ist ebenfalls über die IDL möglich. Die generische Implementierung der Tracker Combiner, die sich nach außen hin wie Tracker verhalten, ermöglicht den Einsatz an beliebigen Stellen im Morgan Framework.

Die Subscriptionen, die den Datenfluss zwischen den unterschiedlichen Trackern regeln, werden im Eye-Manager vorgenommen, der auf Ebene des Morgan Frameworks implementiert ist und als nächstes besprochen wird.

4.3.6 Eye-Manager

Der Eye-Manager entwickelte sich aus der Applikation Trackermonitor, die im Morgan Framework zum Debuggen und Instantiieren von Dunits zur Verfügung steht. Nachdem sich seine Funktionalität immer mehr auf den Rahmen des Eye-Tracking konzentrierte und er außerdem für das Subskribieren zwischen allen beteiligten Trackern genutzt wurde, das den Fluss aller kontinuierlichen Daten regelt, kam er zu seinem Namen. In einer Konsole des Eye-Managers (siehe Abbildung 40) ist es möglich,

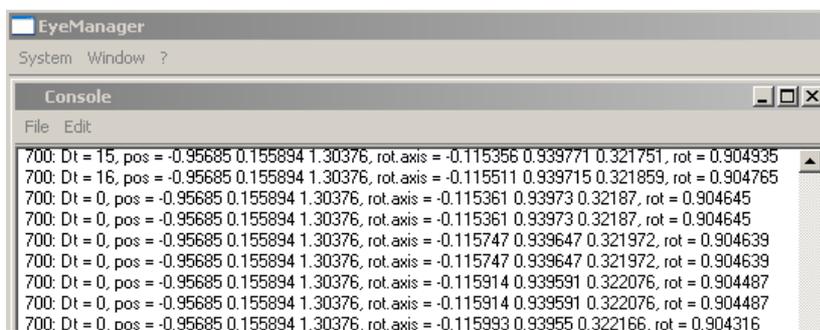


Abbildung 40: Screenshot Eye-Manager

4.3 FRAMEWORK-INTEGRATION DES EYE-TRACKERS

die kontinuierlichen Daten eines Trackers numerisch anzuzeigen. Zum Zeitpunkt der Entwicklung der Eye-Tracker Dunit, aber auch beim Debuggen der Combinern und der Prediction Dunit, war dies sehr hilfreich, da die Konsole erste Rohdaten lieferte und es ermöglichte, Laufzeitfehler aufzudecken, die innerhalb des Morgan Frameworks auftraten.

Um die Eye-Tracking Daten erstmals innerhalb des Frameworks zu sehen, genügte die Konsole des Eye-Managers. Da man sich aber den Punkt im Raum, der von endlosen Zahlenreihen, die seine Position und/oder die Orientierung im Raum beschreiben, kaum vorstellen kann, war eine ergonomischere Darstellung, die der Mensch aus seinem natürlichen Umfeld gewohnt ist, erforderlich. Die erste geometrische Visualisierung erfolgte durch die Fadenkreuzanzeige im HMD.

4.3.7 Fadenkreuzanzeige im Viewer

Die Fadenkreuzanzeige in SmallView wurde bereits für die Selektion virtueller Objekte in der Mixed Reality Stage eingesetzt. Dabei war das Fadenkreuz allerdings in der Mitte des Sichtfelds verankert und konnten nur durch die Kopfbewegung des Anwenders auf Objekte in der Mixed Reality Szene ausgerichtet werden. Die zusätzliche Bewegungsfreiheit des Fadenkreuzes, die durch das Eye-Tracking möglich wird, erfordert die Erweiterung der Implementierung des Viewers.

Der Offset von der Mitte des Sichtfelds im HMD, kann über die Orientierung des Auges berechnet werden. Damit diese Berechnung erfolgen kann, war es notwendig, die relativen kontinuierlichen Eye-Tracking Daten durch vorhandene Mechanismen innerhalb des Viewers zu transportieren. Der Offset für die Fadenkreuzanzeige wurde sowohl für die einfache Betrachtung in der „Monoview“, als auch für die Betrachtung in der „Stereoview“, realisiert. Die stereoskopische Anzeige, die dem Betrachter die räumliche Wahrnehmung der Mixed Reality Szene ermöglicht, war vorhanden und wurde mehrfach nach verschiedenen Verfahren implementiert. Abbildung 41 zeigt die Berechnung des Offsets vom Bildschirmmittelpunkt aus, mittels der relativen Eye-Tracker Daten. Die Formel wird angewendet um den Offset des Fadenkreuzes zu berechnen. Dabei ist \mathbf{x} der Offset und \mathbf{e} der Abstand einer Ebene vom Anwender, auf der das Fadenkreuz positioniert wird. Befindet sich das Fadenkreuz in der Mitte der Anzeige, so ist die Berechnung korrekt. Im äußeren Bereich des Blickfelds stimmt die Größe des Fadenkreuzes nur näherungsweise. Der Abschnitt *5.3 Eye-Tracker Kalibrierung* befasst sich etwas ausführlicher mit dieser Problemstellung. Der Winkel α entspricht der Orientierung des Auges in einer Achse. Die Berechnung wird für den horizontalen und vertikalen Offset einmal je Achse durchgeführt.

AUSWAHL UND INTEGRATION DES EYE-TRACKING

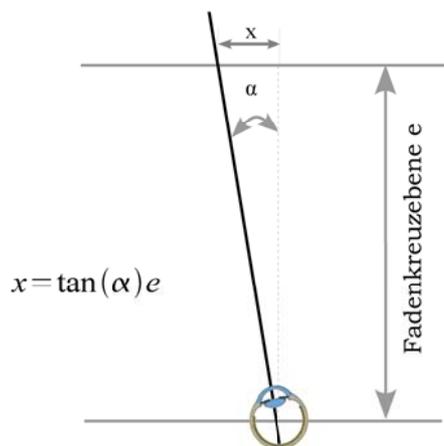


Abbildung 41: Berechnung des Fadenkreuzoffsets

Damit das Fadenkreuz, ohne bei schnellen Bewegungen zu verschwimmen, angezeigt werden kann, ging man davon aus, dass es notwendig ist, das Kreuz direkt mit OpenGL [url20] zu zeichnen. Später stellte sich heraus, dass die Anzeige des Fadenkreuzes auch mit der Darstellung über den Head-Up-Node, um den die VRML-Spezifikation erweitert werden könnte und der speziell für die Anzeige von VRML-Objekten, die sich relativ zum Kopf befinden, implementiert wurde, ein gutes Ergebnis liefert.

Das Fadenkreuz wird nur für ein Auge gerendert, da von dem monokularen Eye-Tracker nur die Orientierung des Auges zuverlässig ermittelt wird, ist dies die einzige Möglichkeit, eine näherungsweise korrekte Darstellung der aktuellen Blickrichtung für unterschiedliche Fokussierungsdistanzen zu realisieren. Diese Situation könnte man unter Einbeziehen der Torsion des Auges umgehen. Da jedoch in einem Versuch festgestellt wurde, dass das Ermitteln der Torsion über die Eye-Tracker Software grundsätzlich möglich ist, die zusätzliche Kalibrierung aber Aufwand erfordert, der aus zeitlicher Sicht die Nutzung unzumutbar machen würde, verzichtete man darauf.

Die Visualisierung der relativen Eye-Tracking Daten durch die Erweiterung der Viewer Implementierung wurde eben besprochen. Ein Versuch, die absoluten Eye-Tracking Daten mit VRML durch das Nutzen von Behaviours [Bro98] in SmallView zu visualisieren und sie zum „Picken“, das im Abschnitt 5.2 *Implementierung* näher beschrieben wird, in der Szene zu verwenden, ermöglichte es zunächst, das Implementieren einer auf der AR-API basierenden Anwendung zu umgehen. Es stellte sich jedoch schnell heraus, dass sich eine etwas komplexere Anwendungslogik zur Interaktion kaum mit den Möglichkeiten der Behaviours im erweiterten Rahmen von VRML realisieren lässt. Darum wurde es erforderlich, eine Benutzerschnittstelle auf Ebene der AR-API für die Blickinteraktion zu implementieren.

5 Benutzerschnittstelle und Blickinteraktion

Damit Interaktion stattfinden kann, ist eine Benutzerschnittstelle erforderlich: das sogenannte User Interface. In den Abschnitten 3.2 *Augenbasierte Interaktionstechniken* und 3.3 *Mixed Reality Interaktion* wurden bereits Probleme und Erkenntnisse im Kontext der Interaktion mittels des menschlichen Blicks besprochen, die hier nun als Grundlage für die Entwicklung einer multimodalen Mixed Reality Applikation zur Verfügung stehen. Die Applikation soll neben der Blickinteraktion in der Mixed Reality auch die Evaluation von augenbasierten Interaktionstechniken ermöglichen und darüber hinaus dazu beitragen, dass in anderen Mixed Reality Anwendungen der Blick zur Interaktion kalibriert werden kann.

Das Ziel von HCI ist eine multimodale und natürliche Benutzerschnittstelle, also ein Interface, das die menschlichen Interaktions- und Kommunikationsgewohnheiten so gut wie möglich berücksichtigt. Um durch das Schnittstellendesign eine hohe Bandbreite für den Informationsaustausch zwischen Mensch und Maschine zu erhalten und somit zu einem hohen Immersionsgrad der Mixed Reality beizutragen, ist die Evaluation möglicher multimodaler Interaktionsmechanismen für die weitere Entwicklung entscheidend. Zunächst wird das Design und anschließend die Implementierung einer Benutzerschnittstelle besprochen, welche die Evaluation multimodaler Interaktionsmechanismen in einer Mixed Reality Umgebung ermöglicht. Darauf folgt der Abschnitt 5.3 *Eye-Tracker Kalibrierung*, der sich mit Integrieren und Durchführen der Eye-Tracker Kalibrierung beschäftigt, die aus technischen Gründen in die Benutzerschnittstelle eingebunden werden muss. Danach wird in 5.4 *Applikationsübergreifender Einsatz* das Nutzen der stereoskopischen Kalibrierfunktionalität der Eye-Interact Applikation in anderen AR-API basierten Applikationen anhand eines Beispiels gezeigt.

5.1 Benutzerschnittstellen Design

Die Frage nach dem natürlichen Design der Benutzerschnittstelle stellt sich wiederholt an dieser Stelle. Führt man sich vor Augen, dass Multimodalität gefordert ist und bei MR-Anwendungen eher Abstand von Tastatur und Maus im klassischen Sinn genommen wird, so ergibt sich unter Berücksichtigung des Blicks und der Sprache eine Mischform zwischen kommandobasierender und direkt bedienbarer Benutzerschnittstelle. Das im Abschnitt 3.2 *Augenbasierte Interaktionstechniken* ermittelte Modell wird hier erweitert und mit konkreten Angaben ergänzt.

Interaktion					
Selektion		Modifikation			
Zeiger	Auslöser	Bewegung	Skalierung	Rotation	Eigenschaft
Auge	Zeigen	Auge	Auge	Auge	Auge
Kopf	Verweilzeit	Kopf	Kopf	Kopf	Kopf
Hand	Zwinkern	Hand	Hand	Hand	Hand
Auge & Kopf	Sprache	Auge & Kopf	Auge & Kopf	Auge & Kopf	Auge & Kopf
Hand & Kopf	Taste	Hand & Kopf	Hand & Kopf	Hand & Kopf	Hand & Kopf
Hand & Auge		Hand & Auge	Hand & Auge	Hand & Auge	Hand & Auge

Abbildung 42: Modell multimodaler Interaktionsmöglichkeiten

Die Selektion erfolgt durch Zeigen und Auslösen. In der ersten Spalte von Abbildung 42 ist die Bedienung des Zeigers durch Kopf, Hand und Auge sowie alle möglichen Kombinationen zwischen ihnen, zur Vollständigkeit aufgeführt. Die Zeiger, deren Einsatz keinen Sinn macht, sind hellgrau gekennzeichnet. In der Abbildung 43 ist das Resultat der sinnvollen Kombinationen zu sehen. Blick steht dabei für die Kombination von Auge- und Kopfbewegung, da sich auch der natürliche Blick des Menschen aus beiden Bewegungen zusammensetzt.

Interaktion					
Selektion		Modifikation			
Zeiger	Auslöser	Bewegung	Skalierung	Rotation	Eigenschaft
Kopf	Zeigen	Kopf	Kopf	Kopf	Kopf
Hand	Verweilzeit	Hand	Hand	Hand	Hand
Blick	Zwinkern	Blick	Blick	Blick	Blick
	Sprache				
	Taste				

Abbildung 43: Multimodales Interaktionsmodell

Die zweite Spalte fasst unterschiedliche Auslösemechanismen zusammen. **Zeigen** bedeutet in diesem Zusammenhang, dass ein Objekt, auf das der Zeiger ausgerichtet ist, sofort selektiert wird. Bei der **Verweilzeit** ist das Verweilen des Zeigers auf einem Objekt für eine bestimmte Dauer notwendig, um dessen Selektion zu bewirken. Mit **Zwinkern** ist das Augenzwinkern gemeint, **Sprache** löst über ein spezielles

5.1 BENUTZERSCHNITTSTELLEN DESIGN

Sprachkommando d.h. ein ausgesprochenes Wort aus, die Bedienung einer **Taste** ist bekannt.

Eine Interaktion setzt sich, wie in Abbildung 43 dargestellt, aus der Selektion und der Modifikation zusammen. Als Modifikation steht die **Bewegung** im Vordergrund, da sich durch sie die markanteste Eigenschaft eines Objekts im dreidimensionalen Raum, die Position, verändern lässt. Betrachtet man den Vorgang zum Bewegen eines Objekts, kann man feststellen, dass eine weitere Dimension betroffen ist nämlich der Lauf der Zeit. Sicherlich können auch andere Ziele verfolgt werden, um eine Veränderung in einer MR-Umgebung zu bewirken, darum sind weitere charakteristische Möglichkeiten aufgeführt, zu denen auch die **Eigenschaft** zählt, die stellvertretend für alle weiteren denkbaren Parameter eines Objekts steht. Da in der Implementierung nur das Bewegen von virtuellen Objekten ermöglicht wird, sind die anderen Möglichkeiten zur Modifikation grau dargestellt.

Die Benutzerschnittstelle soll einerseits, die in *6.1 Usability Szenarien* benötigten Interaktionen ermöglichen, andererseits könnte sie aber auch darüber hinaus zur weiteren Entwicklung und Evaluation der Interaktion in MR beitragen. Die Möglichkeit unterschiedliche Szenarien in Form von VRML-Szenen von einer Datei jederzeit in die aktuelle Szene der Applikation zur Laufzeit einfügen zu können und auch den eingefügten Code jederzeit wieder entfernen zu können, wäre nicht nur für die Durchführung von Benutzertests von Vorteil, sondern auch für die weitere Entwicklung der Benutzerschnittstelle. Damit die Modifikation auf bestimmte Objekte der Szene beschränkt werden kann und um gleichzeitig gezielt zwischen selektierten Objekt-Typen differenzieren zu können, ist ein geeigneter Mechanismus notwendig. Diese Anforderung ergibt sich aus der Tatsache, dass Objekte in der Szene verschiedene Aufgaben erfüllen, auf die die Benutzerschnittstelle unterschiedlich zu reagieren hat. Zum Beispiel könnte es Objekte geben, die zum Menü der Benutzerschnittstelle gehören, andere jedoch nicht. Weiterhin ist eine zuschaltbare Visualisierung von bestimmten Applikationsparametern sowie einer Stoppuhr wünschenswert, die zum Bewusstsein über die aktuelle Interaktionssituation beiträgt.

Für die Evaluation verschiedener Geräte, die als Beispiel zum Zeigen in der Szene verwendet werden können und deren simultane Nutzung keinen Sinn macht, wäre das Wechseln zwischen ihnen zur Laufzeit der Applikation eine weitere Anforderung an das Design der Benutzerschnittstelle. Die Kombination unterschiedlicher Geräte für Selektion und Modifikation sollte möglich sein, damit auch verschiedene Kombinationsmöglichkeiten evaluiert werden können.

Die genannten Anforderungen an die Benutzerschnittstelle spezifizieren ihr Design, das durch die Implementierung zum Leben erweckt wird.

5.2 Implementierung

Die Implementierung der Eye-Interact Anwendung, die auf der AR-API aufbaut und durch sie Zugriff auf Daten und Funktionen von Dunits des Morgan Frameworks erhält, erfolgte zunächst, um einen Eindruck von der Performanz zu erhalten, die sich aus der Implementierung des *Ar-ringtonEyeTrackers* und des *AbsoluteRelativeCombiners* ergibt. Die aus dem Sourcecode hervorgehende Funktionalität wird im Folgenden besprochen.

Im frühen Entwicklungsstadium entsprach die Anwendung eher einem Werkzeug, das die Aufgabe hatte, mittels AR-API Funktionen, erweiterte Funktionalität für das Debuggen zur Laufzeit anzubieten. Um die speziellen Funktionen zum Debuggen erhalten zu können und um die Performance der angestrebten Anwendung nicht unnötig negativ zu beeinflussen, wurde der Debugmode implementiert, der zur Laufzeit der Anwendung über die Tastatur aktiviert werden kann. Im Debugmode ist ein visuelles Feedback im dreidimensionalen Raum umgesetzt worden, das die simultane Anzeige der kontinuierlichen Blickrichtungsdaten in den folgenden drei Varianten ermöglicht:

- Fadenkreuzanzeige im Viewer
- Relative Eye-Tracker Daten im Head-Up-Node
- Absolute Eye-Tracker Daten in der Szene

Nachdem alle drei Varianten den gleichen Punkt im Raum visualisieren, sollten die zur Visualisierung verwendeten Objekte sich auch an einem identischen Punkt im Raum befinden. Ist das für alle möglichen Positionen im Raum der Fall, so kann es als Hinweis auf die mathematisch korrekte Implementierung des zur Visualisierung genutzten Codes be-

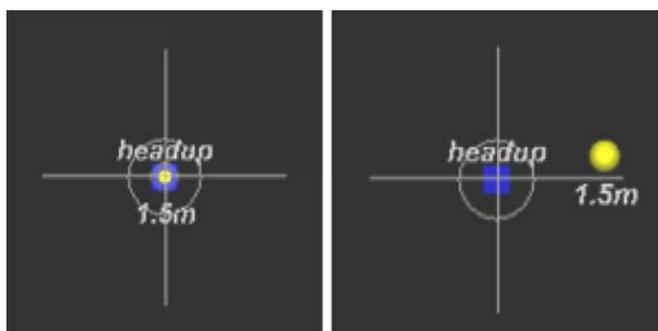


Abbildung 44: 3-fache Blickrichtungsvisualisierung

trachtet werden. Da durch die drei unterschiedlichen Varianten alle für die Blickrichtung relevanten Bereiche des Sourcecodes genutzt werden und außerdem auch viel Code der Implementierung in SmallView und dem Morgan Framework beteiligt ist, können zusätzlich Rückschlüsse über auftretende Latenzen im Sinne der Wahrnehmung gemacht werden.

In den beiden Screenshots in Abbildung 44 werden die relativen Blickrichtungsdaten einmal durch das Fadenkreuz direkt über OpenGL angezeigt und einmal über den blauen Würfel, der als Kind in der Hierarchie des Head-Up-Nodes in der Szene bewegt wird. Die gelbe Kugel zeigt die absoluten Daten an, die aus dem Combiner hervorgehen. Der linke Screenshot ist ohne und der rechte während einer schnellen Kopfbewegung aufgenommen worden.

In der Abbildung 44 ist zu sehen, dass auf Grund von Latenzen die kombinierten Daten, die in der Szene gerendert werden, bei schnellen Kopfbewegungen einen kurzzeitigen Offset haben. Der blaue Würfel, der im Head-Up-Node gerendert wird, ist hingegen auch bei schnellen Bewegungen absolut synchron mit dem Fadenkreuz. Das bedeutet, sie könnten durch ein beliebiges VRML-Objekt im Head-Up-Node ersetzt werden. Der Debugmode zeigt in einem Fenster zusätzlich nützliche Informationen über den Zustand der Eye-Interact Applikation an.

Die Anforderungen aus dem vorhergehenden Abschnitt *5.1 Benutzerschnittstellen Design* wurden alle umgesetzt. Im Anhang *C Interaction Command Guide* sind sämtliche Befehle strukturiert nach dem besprochenen Interaktionsmodell und nach Eingabegeräten getrennt in einer Übersicht auf englisch dokumentiert, da einerseits die Codierungsvorschriften des Morgan Frameworks die Implementierung und die Dokumentation in Englisch verlangen und andererseits die Sprachkommandos nur auf Englisch funktionieren.

Multimodalität ist dadurch gegeben, dass die Benutzerschnittstelle Kommandos unterschiedlicher Eingabekanäle jederzeit entgegennimmt. Ausgenommen davon sind Interaktionen, die keinen Sinn machen oder sich gegenseitig ausschließen. Um diesen Fällen aus dem Weg zu gehen, wurden zum Beispiel Schaltmechanismen implementiert, die die Wahl von bestimmten Auslösemechanismen ermöglichen.

Das Verhalten der Anwendung aus zeitlicher Sicht wird jetzt beschrieben. Sie hat vier verschiedene Interaktionsmodi, die sich automatisch gegenseitig ausschließen: Select, Move, Remove und Game. Game nimmt eine Sonderstellung ein, da das der einzige zeitabhängige Modus ist, der sich nicht abschalten lässt. Das Spiel endet nach einer festgelegten Dauer, danach wird selbständig in den vorhergehenden Modus gewechselt. Die weiteren Besonderheiten des Gamemodus werden nun kurz erläutert. Ist er einmal aktiviert, werden aus einem Pool verschieden animierter VRML-Szenen, zehn in zufälliger Reihenfolge nacheinander geladen. Sie enthalten alle einen Würfel, der mit einem Zeiger für ca. drei Sekunden kontinuierlich selektiert werden muss, damit der Spieler einen Punkt erhält und der Würfel mit einer Explosion, die durch Abspielen einer Animation umgesetzt wurde, aus der Szene verschwindet. Eine weitere Sonderstellung haben die beiden Kalibrierungszyklen des Eye-Trackers, die in jedem Modus aktiviert werden können und dann vollständig durchlaufen werden müssen. Im Gegensatz dazu ist das Wechseln zwischen den verschiedenen Zeigern

BENUTZERSCHNITTSTELLE UND BLICKINTERAKTION

zur Selektion in der Szene ist jederzeit möglich, auch im Movemode der das „simultane“ Nutzen unterschiedlicher Zeiger für Selektion und Modifikation erlaubt. Weiterhin sind Wechsel zwischen den verschiedenen Triggern, das Laden und Entfernen bestimmter Szenen sowie das Umschalten der Anzeige von Informationen über Interaktionsmodus und Zeit der Stoppuhr möglich.

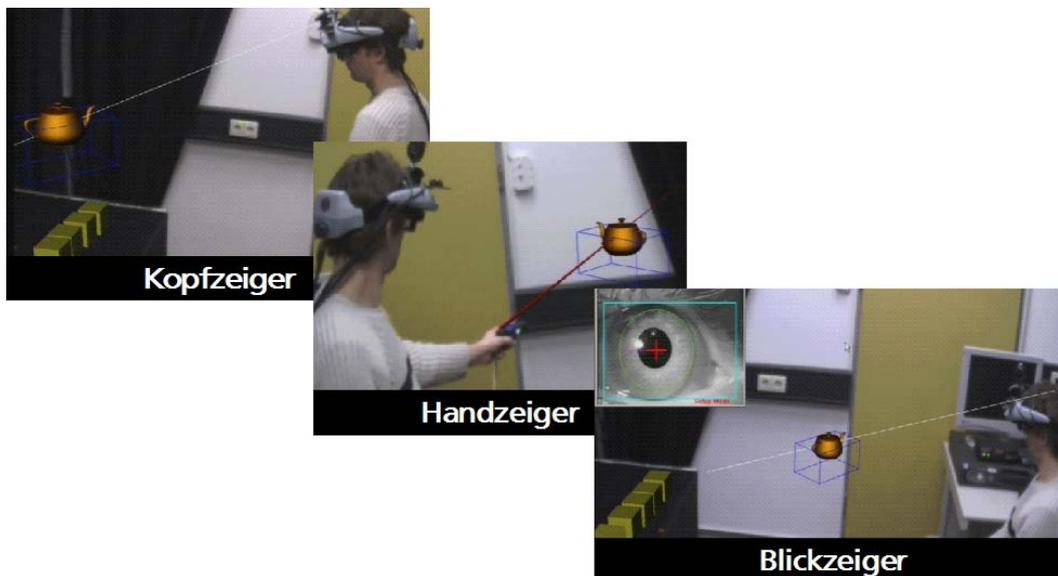


Abbildung 45: MR-Zeiger

Die Zeiger, die zur Selektion von Objekten in der Szene genutzt werden, existieren zunächst nur in Form von einem Punkt im Raum, der durch Position und Orientierung definiert ist. Damit mittels der Zeiger Objekte in der Szene selektiert werden können, ist das Picking notwendig. Gepickt wird mit Hilfe einer AR-API Funktion, die durch einen imaginären Strahl der von dem Zeiger ausgeht (weiße Linie in Abbildung 45), Information über das erste Objekt in der Szene liefert, auf das er trifft. Durch Auswerten der Information ist es möglich, Zugriff über die AR-API auf ein Objekt in der Szene zu bekommen, dessen Parameter dann manipuliert werden können. Der im vorhergehenden Abschnitt angesprochene Mechanismus, der es erlauben soll, zwischen verschiedenen Objekten zu differenzieren, wurde durch Einführen des Schlüsselwortes: GRIP umgesetzt. Das Wort muss am Anfang der Bezeichnung des Transformknotens aller VRML-Objekte enthalten sein, die für die Interaktionen zur Verfügung stehen sollen. Dadurch ist es beim Auswerten der gepickten Information möglich, zwischen verschiedenen Objektsorten zu differenzieren.

Die Kalibrierung des Eye-Trackers wird wegen ihres Stellenwerts und Umfangs im folgenden Abschnitt ausführlich beschrieben.

5.3 Eye-Tracker Kalibrierung

Die Kalibrierung des Eye-Trackers ist notwendig, damit die inhomogenen Daten, die aus der Bildverarbeitung hervorgehen, in einem homogenen Raum verwendet werden können. Bevor man den Kalibriervorgang startet, müssen Parameter sowie die Kamera und die IR-Beleuchtung, so justiert werden, dass ein hoher Kontrastumfang und eine gleichmäßig ausgeleuchtete Szene für den Bildverarbeitungsalgorithmus zur Verfügung stehen, der dann die Augenbewegung des Anwenders erfassen kann. In der Abbildung 46 ist die Benutzerschnittstelle in einem unkalibrierten Zustand zu sehen, das wird durch die sporadisch verteilten Punkte im Bild der Augenkamera deutlich und durch die im Eye Space angezeigten symmetrischen Punkte.

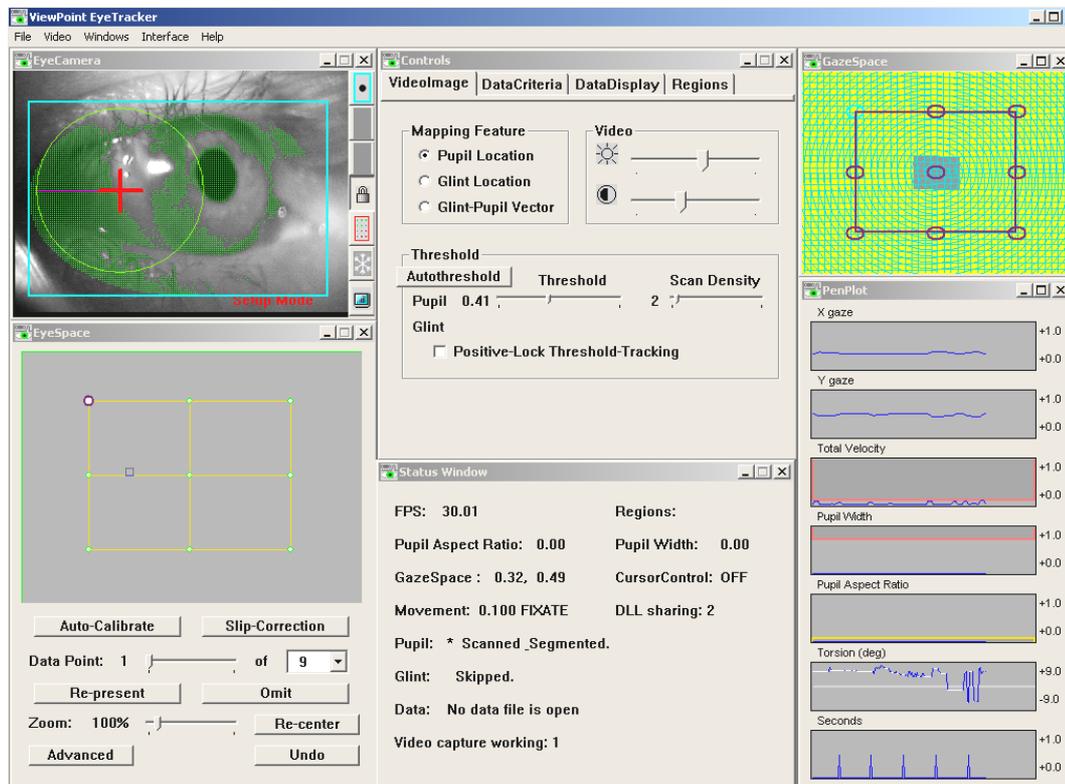


Abbildung 46: Screenshot Eye-Tracking Software

Ist das Bild der Kamera gut eingestellt, kann mit der eigentlichen Kalibrierung begonnen werden. Im einfachsten Fall kann man die Software des Eye-Tracking Systems nutzen, um den Mauszeiger mittels der Augenbewegung über den Bildschirm zu bewegen. Dazu ist es notwendig, Bildschirmparameter und den Abstand des Anwenders zum Bildschirm einzugeben. Darauf folgt der Kalibriervorgang, bei dem der Anwender nacheinander auf unterschiedliche Punkte, die auf dem Bildschirm angezeigt werden, schaut und die Position mit einem Mausklick

BENUTZERSCHNITTSTELLE UND BLICKINTERAKTION

bestätigt. Abbildung 47 zeigt eine abgeschlossene Kalibrierung dabei ist deutlich zu sehen, dass die Punkte im Bild der Augenkamera nur innerhalb der Pupille auftreten und die Punkte im Eye Space nun nicht mehr absolut symmetrisch angeordnet sind.

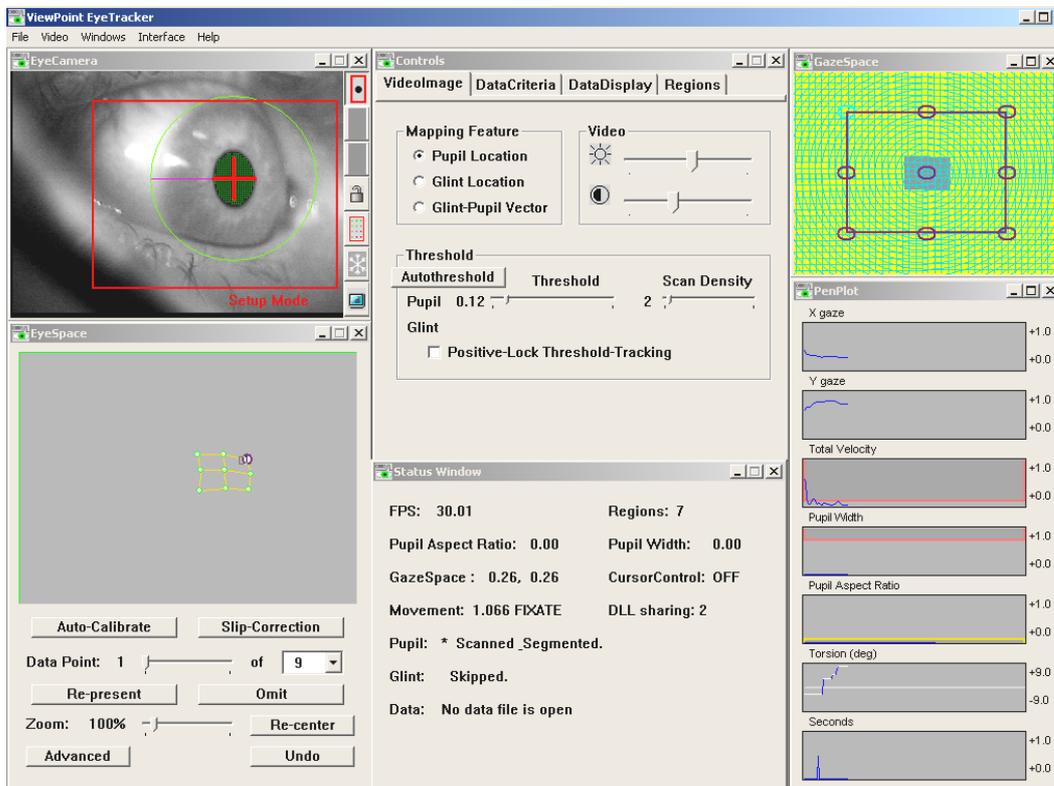


Abbildung 47: Screenshot Eye-Tracking Software kalibriert

Die Kalibrierung des monokularen Eye-Trackers für die Interaktion mit stereoskopisch angezeigten Objekten konnte nicht ausschließlich mit dem Mechanismus erfolgen, den die mitgelieferte Software bereitstellt, da er nur für die Kalibrierung am Bildschirm konzipiert wurde und bestenfalls zum Kalibrieren von einer Standard-Anzeige, bei der das Bild für beide Augen gleich angezeigt wird, geeignet ist. Im Abschnitt 2.1.1 *Anatomie und Physiologie* wurde festgestellt, dass die Abbilder der Umgebung auf den beiden Retinas erst in der Wahrnehmung des Menschen verschmelzen. Damit eine Kalibrierung mit stereoskopisch angezeigten Objekten möglich ist, ist es also notwendig, eine entsprechende Wahrnehmungssituation zu schaffen, damit die Wahrnehmung des Anwenders kalibriert werden kann. Um eine solche Situation optimal in Szene setzen zu können, ist zunächst der gezielte Einsatz von verschiedenen Tiefenkriterien notwendig, die im Abschnitt 1.3 *Wahrnehmung* besprochen wurden. Es stellte sich während einigen Experimenten mit der Anzeige im HMD heraus, dass zur Gestaltung der optimalen Kalibrierung eine leere Mixed Reality Szene, also ein Blickfeld, das keinerlei Ablenkung bietet, besonders gut für die Visualisierung der Kalibrierungsobjekte geeignet ist. Die Objekte wurden gezielt, unter Berücksichtigung von Tiefenkriterien, die aus monokularen wie auch bin-

5.3 EYE-TRACKER KALIBRIERUNG

okulare Signalen hervorgehen, modelliert. In Abbildung 48 ist der Kalibrierungswürfel zu sehen, der sich um eine Achse dreht, deren Orientierung sich laufend ändert, aber immer im Mittelpunkt des Würfels zentriert ist. Die Dynamik des kantigen Objektes erfüllt mehrere Aufgaben gleichzeitig. Auf der einen Seite sorgt die variierende Drehachse für besonderes Interesse beim Rezipienten und stellt somit ein notwendiges Maß an Aufmerksamkeit sicher, andererseits ist der Würfel beleuchtet und bewegt mit unterschiedlichen Abbildungen auf den beiden Retinas, vertreten. Dadurch wirken mehrere der bereits besprochen Tiefenkriterien gleichzeitig auf die Wahrnehmung ein und sorgen für ein räumlich wahrgenommenes Bild. Der Würfel ist so dimensioniert, dass sein Abbild genau auf die Fovea passt und er ohne bewusste Augenbewegung wahrgenommen werden kann (siehe *2.1 Visuelles System des Menschen*). Dies ist für die Kalibrierung wichtig, da sie durch den Abgleich mehrerer statischer Punkte, die sich auf dem Horopter im gleichen Fokussierungsabstand zum Benutzer im Raum befinden, für die inhomogenen Daten, die aus der Bildverarbeitung des Eye-Trackers hervorgehen einen Offset berechnet, der es erlaubt sie in einem homogenen Raum zu verwenden.

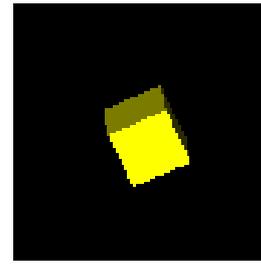


Abbildung 48: Kalibrierungswürfel

Die Anzeige des Kalibrierungswürfels in verschiedenen Punkten auf dem Horopter im Raum wurde in der Eye-Interact Anwendung umgesetzt. Dort stehen zwei Möglichkeiten zum Kalibrieren zur Verfügung: die vollständige mit neun Würfeln, die immer einmal vor der Nutzung des Eye-Trackers durchgeführt werden muss, die andere dient als Hilfsmittel und adressiert das zentrale Problem des Eye-Trackers, der nach der initialen Einstellung häufig nachjustiert werden muss. Die Rekalibrierung zeigt den Würfel nur einmal an. Der Kalibriervorgang läuft generell so ab, dass der Anwender den Würfel anschauen muss und dann zur Bestätigung eine Taste betätigt. Wenn kein Würfel mehr zu sehen ist, ist die Kalibrierung abgeschlossen.

In Abbildung 49 sind verschiedene Fokussierungssituationen aufgezeigt, die alle in einer stereoskopischen MR-Umgebung auftreten können. Nachdem die tatsächliche Fokussierungsdistanz nicht bekannt ist und sich laufend ändert, muss mit einer Näherung gearbeitet werden. Im Mittel wird eine Distanz von 1,5m angenommen, in der der Anwender interagiert und natürlich auch Objekte fokussiert. Für den Augenabstand wird der gängige Wert von 6cm verwendet. Diese Werte werden sowohl für die stereoskopisch gerenderte Szene eingesetzt als auch für die Kalibrierung und Visualisierung des Eye-Trackers. Ein kleiner Offset, der durch unterschiedliche Tiefen in den äußeren Bereichen des Horopters zwischen Fokussierungsdistanz und Fadenkreuzebene entsteht und sich minimal auf die dargestellte Größe, also die Skalierung, nicht aber auf die Position des Fadenkreuzes auswirkt, wird vernachlässigt.

BENUTZERSCHNITTSTELLE UND BLICKINTERAKTION

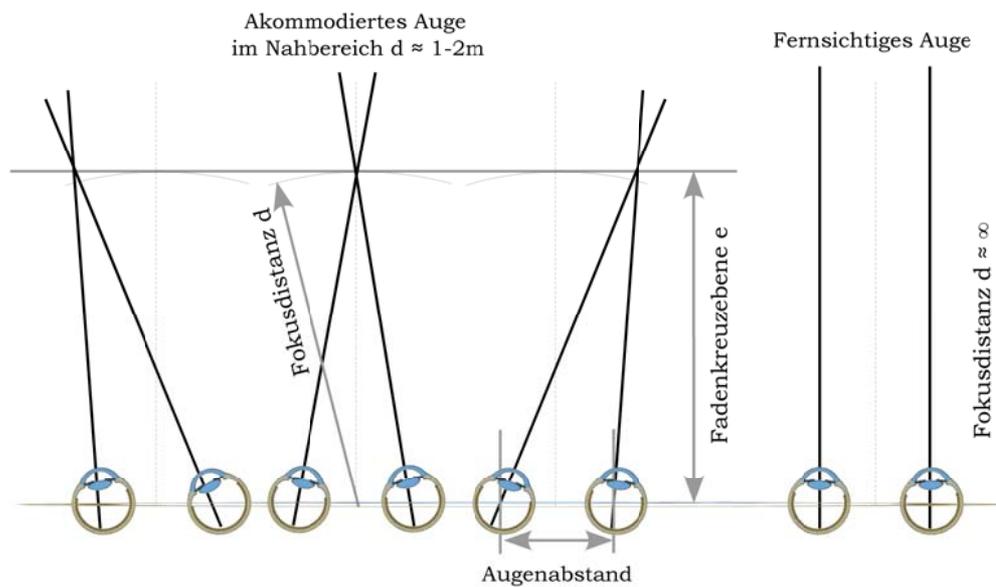


Abbildung 49: Fokussierungssituationen

Über die Qualität der Kalibrierung im Lauf der Zeit kann nur eine subjektive Aussage getroffen werden, da ihre Wirkung nur direkt vom Anwender wahrgenommen wird. In den Benutzertests ist das der Fall, da der Anwender in verschiedenen Szenarien interagiert und auf die Funktionalität der Kalibrierung angewiesen ist. Der Einsatz der stereoskopischen Kalibrierfunktionalität ist auch in anderen AR-API basierten Applikationen wünschenswert, deshalb wird im Folgenden Beispiel auf eine Möglichkeit eingegangen.

5.4 Applikationsübergreifender Einsatz

Der applikationsübergreifende Einsatz der Eye-Interact Anwendung zur stereoskopischen Kalibrierung des Eye-Tracking Systems wird durch das gleichzeitige Nutzen von zwei unterschiedlichen Mechanismen möglich, die eine Applikation um die Funktionalität einer anderen erweitern. In der Abbildung 50 ist das Funktionsschema anhand eines Beispiels dargestellt, wie die Applikation der Mixed Reality Stage, die für diesen Einsatz geringfügig angepasst werden musste, eingesetzt wird, damit in ihr die Blickinteraktion genutzt werden kann. Um den Eye-Tracker mit der stereoskopischen Anzeige der Mixed Reality Stage kalibrieren zu können, war die Kalibrierfunktionalität der Eye-Interact Applikation erforderlich geworden. Da der Kalibriervorgang eng mit der Applikation verbunden ist, schien es sinnvoll, einfach die notwendigen Ein- und Ausgabe Events [Bro98] der beiden Applikationen aufeinander abzustimmen, während sie mit identischen Eye-Tracking Daten aus dem

5.4 APPLIKATIONSÜBERGREIFENDER EINSATZ

Morgan Framework arbeiten. Dies wurde durch den Einsatz bestehender Mechanismen aus SmallView und Morgan realisiert. Die beteiligten Anwendungen wurden Abbildung 50 entsprechend konfiguriert.

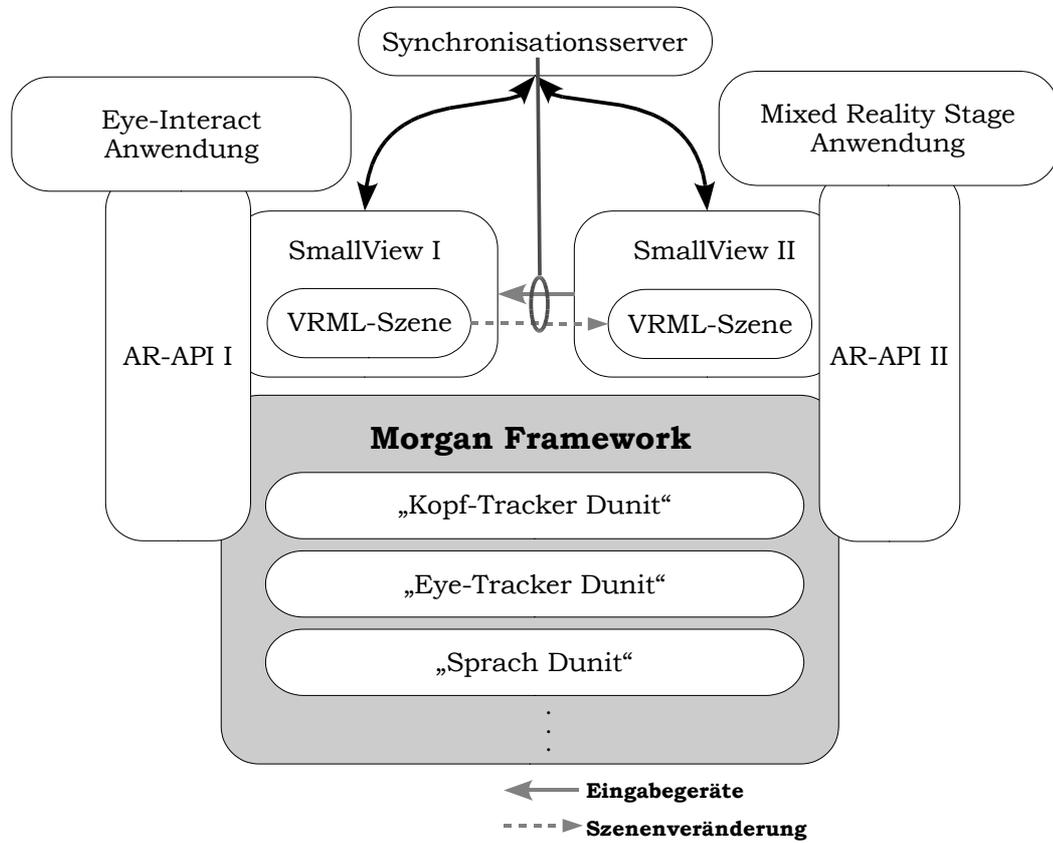


Abbildung 50: Applikationsübergreifende Synchronisation

BENUTZERSCHNITTSTELLE UND BLICKINTERAKTION

6 Evaluation

Für die Evaluation der Blickinteraktion im Kontext gewohnter und natürlicher Interaktionen definiert der Abschnitt *6.1 Usability Szenarien*, die Szenarien und die Aufgaben, mit denen sich die Probanden während der Benutzertests beschäftigten. Die Durchführung der Tests ist im Abschnitt *6.2 Benutzertests* festgehalten. Nachdem die praktischen Aufgaben erledigt waren, erhielten die Probanden einen Fragebogen (siehe Anhang *D Aufgaben und Fragen zur Evaluation*) mit Fragen zu den Aufgaben. Die Antworten und deren Begründungen liefern zusätzliches Feedback zur Evaluation. Die Abschnitte *6.3 Selektion statischer Objekte*, *6.4 Selektion dynamischer Objekte* und *6.5 Interaktion durch Bewegung* enthalten die Ergebnisse, die in *6.6 Resümee der Evaluation* nochmals zusammengefasst werden.

6.1 Usability Szenarien

Die Usability Szenarien liefern die Basis für Benutzertests und tragen dadurch zur weiteren Entwicklung von natürlichen multimodalen augenbasierten Interaktionstechniken in Mixed Reality Benutzerschnittstellen bei. Das Design der Benutzerschnittstelle, die speziell für das Testen von MR-Interaktionen und besonders für die Blickinteraktion entwickelt wurde, ermöglicht die Evaluation vieler denkbarer Szenarien.

Der Einsatz von Eye-Tracking zur Interaktion in Mixed Reality Umgebungen greift das natürliche Blickverhalten auf, indem die Blickrichtung des Menschen im Raum, die aus der Kombination von Augen- und Kopf-

EVALUATION

bewegung hervorgeht, in der aktuellen Wahrnehmungssituation zur Interaktion mit virtuellen Objekten in einem näherungsweise synchronen Raum evaluiert wird. Die Szenarien, in denen während der Benutzertests Aufgaben zu lösen sind, ergeben sich aus gesammelten Erfahrungen, sowie Erkenntnissen, die im Abschnitt 3.2 *Augenbasierte Interaktionstechniken* und im Abschnitt 5.1 *Benutzerschnittstellen Design* angesprochen wurden.

Für die Selektion kamen drei unterschiedliche Zeiger, deren Reichweite vergleichbar ist, zum Einsatz. Der selbst zusammengesetzte *Handzeiger*



Abbildung 51: Handzeiger mit Tasten

mit drei Maustasten ist in Abbildung 51 zu sehen. Der Kopfzeiger wird mit der Kopfbewegung ausgerichtet und wurde im Abschnitt 3.3 *Mixed Reality Interaktion* im Kontext der Mixed Reality Stage ausführlich vorgestellt. Der Blickzeiger unterscheidet sich vom Kopfzeiger dadurch, dass er innerhalb des gesamten Blickfelds bewegt werden kann und immer genau in die Richtung zeigt, in die man schaut. Abbildung 45 zeigt die drei Zeiger während der Anwendung.

Im Szenario der ersten Aufgabe wird die Interaktionsgeschwindigkeit für die drei Zeiger ermittelt. Der Proband selektiert jeweils fünf Objekte mit den Zeigern. Die dafür benötigte Zeit wird für den Vergleich notiert. In dieser Aufgabe kommen zwei gewohnte, natürliche Verhaltensweisen vor: einmal die Ausrichtung des Blicks und als zweites die Ausrichtung eines Hilfsmittels mit der Hand auf ein Objekt, mit dem interagiert werden soll. Die reine Kopfbewegung zur Selektion eines Objekts im Raum ist nicht natürlich und gewohnt, aber zum Vergleich auch vertreten, da mit ihr in der Vergangenheit positive Erfahrungen gesammelt wurden. Als Auslösemechanismus für die Selektion wurde die Taste gewählt, da sie weitverbreitet ist und die meisten Anwender gewohnt sind, sie als Auslöser einzusetzen. Ein ähnliches Szenario, in dem bereits die

Interaktion mittels Finger- und Blickzeiger in einer rein virtuellen Umgebung evaluiert wurde, ist in der Arbeit „*Interacting with Eye Movements in Virtual Environments*“ [Tan00] dokumentiert. Hier wird ein ähnliches Szenario bewusst eingesetzt, um die Ergebnisse tendenziell in Verbindung miteinander bringen zu können.

Die zweite Aufgabe der Benutzertests evaluiert verschiedene Auslösemechanismen, die in Kombination mit dem Blickzeiger zur Selektion von fünf Objekten eingesetzt werden. Die Mechanismen wurden im Abschnitt 5.1 *Benutzerschnittstellen Design* vorgestellt. Die benötigte Zeit für die Selektion der Objekte wird je Mechanismus ermittelt.

In der dritten Aufgabe erfolgt die Selektion bewegter Objekte mit jedem der drei verschiedenen Zeiger. Die Objekte bewegen sich nacheinander auf den Probanden zu. Der Zeiger muss für eine Dauer von ca. drei Sekunden auf das aktuelle Objekt gerichtet sein, damit es selektiert wird. Sofern der Zeiger korrekt auf das Objekt ausgerichtet ist, erhält der Proband eine visuelle Rückmeldung durch einen blauen Rahmen, der um das Objekt erscheint. Die Selektion wird durch das Abspielen einer kurzen Animation visualisiert. Dadurch erhält die Aufgabe den Charakter eines Spiels und erhöht den Spaßfaktor, der zum Wohlbefinden und der Motivation der Probanden beiträgt. Die Bewegung, mit der sich die Objekte auf den Probanden zu bewegen, variiert. Die Geschwindigkeit ist unterschiedlich und gelegentlich ist die Bewegung nichtlinear. Dieses Szenario wurde gewählt, damit die natürlichen Vorteile bei der gewohnten Verfolgung mit dem Blick zur Geltung kommen können. Um vergleichbare Ergebnisse zu erhalten, sind je Zeiger zehn mögliche Selektionen innerhalb eines festgelegten Zeitraums vorgesehen. Für die Auswertung wird die Anzahl der tatsächlichen Selektionen je Zeiger notiert.

Die vierte und letzte Aufgabe hat das Bewegen eines Objekts zum Ziel. Hier geht es darum, eine Aktion mit dem zuvor selektierten Objekt auszuführen. Die Veränderung der Position wurde als gewohnte und natürliche Interaktion mit einem Objekt im Raum ausgewählt. Für die Objektselektion kommt der Blickzeiger zum Einsatz und als Auslösemechanismus ist das Betätigen mit einer Taste festgelegt. Die Veränderung der Position erfolgt mit den drei unterschiedlichen Zeigern, die in Kombination mit dem Blickzeiger, der immer zur Selektion genutzt wird, evaluiert werden. Die benötigte Zeit für das Selektieren und Bewegen sowie das abschließende Deselektieren wird für die unterschiedlichen Zeiger festgehalten.

In den Benutzertests beschäftigten sich die Probanden mit dem Lösen der in den Szenarien beschriebenen Aufgaben. Die Durchführung der Tests ist im folgenden Abschnitt festgehalten.

6.2 Durchführung der Benutzertests

Die Szenarien und Aufgaben für die Probanden wurden im vorhergehenden Abschnitt festgelegt. Bevor die Probanden mit dem Lösen der verschiedenen Aufgaben bemüht wurden, erhielten sie eine Einweisung zum Justieren und Aufsetzen des HMD. Anschließend wurde die Eye-Tracking Hardware eingerichtet und die Parameter der Bildverarbeitungssoftware konfiguriert. Danach führte der Proband die Kalibrierung des Eye-Trackers über den Mechanismus, der im Abschnitt 5.3 *Eye-Tracker Kalibrierung* beschrieben wurde, durch.

Das Einrichten und Konfigurieren des Eye-Trackers sowie die anschließende Kalibrierung erfolgte in einem iterativen Prozess, da die Genauigkeit nur von dem Probanden wahrgenommen wird und ein ausreichendes Ergebnis nur in Kooperation erzielt werden konnte. Wie zu erwarten, gestaltete sich die Kalibrierung des Eye-Trackers als schwierig, da unterschiedliche Fehlerquellen auszumachen waren, die zunächst die erfolgreiche Kalibrierung verhinderten. Mit Abstand die häufigste Fehlerquelle war das Verrutschen des HMD auf dem Kopf des Probanden, da auch geringfügige Offsets zwischen dem Kopf und dem HMD zu einem erheblichen Offset des Eye-Trackers führten.

Das Verrutschen ist auf verschiedene Faktoren zurückzuführen. Dazu gehört die Beschaffenheit der Haare des Probanden. Lange und besonders glatte Haare begünstigen das Verrutschen außerordentlich. Eine Verbesserung konnte durch das Tragen von einem Tuch, das enganliegend am Kopf befestigt wurde und aus einem griffigen Stoff bestand, erreicht werden. Das Gewicht und die Lage des Schwerpunkts des verwendeten HMD sind die entscheidenden Faktoren, die bei allen Probanden als Ursache des Verrutschens identifiziert wurden und im Rahmen dieser Arbeit nicht beeinflusst werden konnten. Versuche, den Offset durch die Wahl des eingesetzten Bildverarbeitungsalgorithmus minimieren zu können, lieferten nicht den erhofften Erfolg, führten aber in manchen Situationen zu einer Verbesserung.

Nachdem die Kalibrierung abgeschlossen war, wurden die verschiedenen Interaktionsmöglichkeiten vorgestellt und getestet, um den Probanden etwas Erfahrung mit den Interaktionen sammeln zu lassen. Im Anschluss an die Einführungsphase wurden die verschiedenen Szenarien über die Benutzerschnittstelle konfiguriert und dem Probanden die jeweiligen Aufgabenstellungen erklärt.

In den Diagrammen der folgenden drei Abschnitte sind die Ergebnisse visualisiert. Sie repräsentieren die Mittelwerte aller Werte, die von 11 Probanden während der Benutzertests erhoben wurden. Die über die Fragebögen nach der praktischen Evaluation ermittelten Informationen sind im Text des jeweiligen Abschnitts zusammengefasst.

6.3 Selektion statischer Objekte

Bei der **ersten Aufgabe** stand der Unterschied zwischen verschiedenen Zeigern im Vordergrund. In Abbildung 52 ist klar zu erkennen, dass die Selektion mit dem Blickzeiger am längsten dauerte. Das entspricht nicht den Erwartungen, da ausgehend von vorhandenen Arbeiten im Abschnitt 3.2 *Augenbasierte Interaktionstechniken* und durch die klare Darlegung von Tatsachen festgestellt wurde, dass der Blick und somit auch der Blickzeiger im Vergleich zu anderen Zeigern als der schnellste einzuordnen ist. Überraschend erscheint das Ergebnis dennoch nicht, da bereits bei der Kalibrierung deutlich wurde, dass das Verrutschen des HMD nach kurzer Nutzungsdauer zu einer Ungenauigkeit des Eye-Trackers führt. Nachdem die fünf Objekte so im Raum verteilt waren, dass es notwendig war, den Kopf zu bewegen, um sie alle selektieren zu können, was ja auch dem natürlichen Blickverhalten entspricht, wurde natürlich auch das am Kopf befestigte HMD bewegt. Messungen, bei denen ein Proband das HMD mit der Hand stabilisierte und dadurch der geschilderten Problematik entgegenwirkte, lieferten ein Ergebnis, das sich tendenziell mit den besprochenen Erwartungen deckte. Auch Versuche, in denen die Objekte so angeordnet waren, dass ihre Selektion keine Kopfbewegung erforderte, bestätigten die eben genannte Tendenz. Die Zeiten dieser Vergleiche wurden außer Konkurrenz ermittelt, da sie in einem Szenario erfolgten, das nicht den Anforderungen für natürliche Interaktionen genügt.

Die Probanden wurden im Fragebogen nach dem intuitivsten Interaktionsmittel zur Bewältigung der gestellten Aufgabe befragt, die Antworten fielen auf Grund der Kalibrierprobleme unterschiedlich aus. Unter Einbeziehen der Begründung konnte aus den meisten Antworten der Blickzeiger als intuitivstes Interaktionsmittel für diese Aufgabe interpretiert werden. Die Begründungen der Probanden machten deutlich, dass man automatisch, ohne jedes Zutun, auf das Objekt von Interesse schaut. Die geringe kognitive Last wurde weiterhin als Vorteil genannt. Die subjektiv wahrgenommene Ungenauigkeit des Eye-Trackers durch das Verrutschen des HMDs wurde aber auch deutlich angesprochen. Der Handzeiger wurde wegen des gewohnten Umgangs mit der Maus in einem Fall als intuitives Interaktionsmittel genannt.

Selektion statischer Objekte

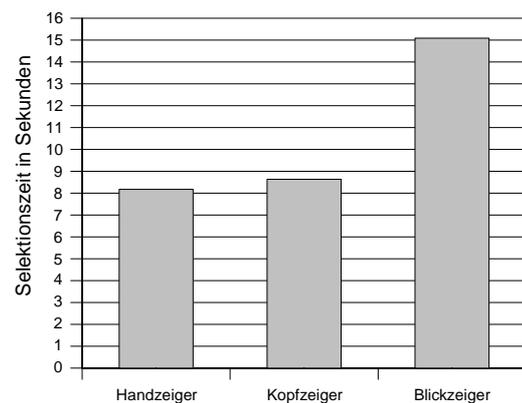


Abbildung 52: Zeigerselektionszeiten statischer Objekte

EVALUATION

In der **zweiten Aufgabe** ging es darum, unterschiedliche Auslöser für die Auswahl einzusetzen und zu vergleichen. Die Spracheingabe als Auslösemechanismus musste aus dem geplanten Vergleich gestrichen werden, da zwischen Aussprache und Wirkung des Befehls soviel Zeit verging, dass es anstrengend wurde, den Blick für diese Dauer in einer Richtung

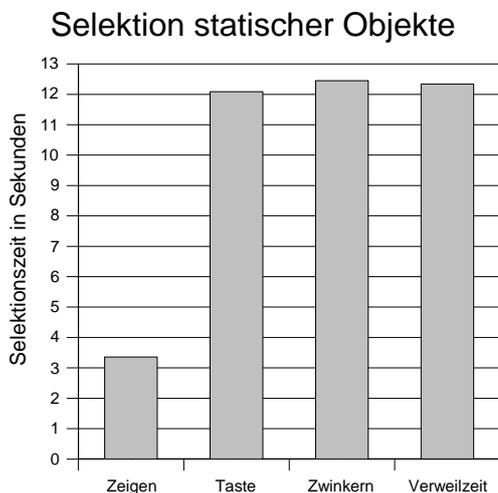


Abbildung 53: Auslöserselektionszeiten

durch die geringe Selektionszeit deutlich. Als Referenzwert ist die Erhebung interessant, aber der praktische Nutzen der Methode ist nicht besonders hoch, da der bereits erwähnte Midas Touch (siehe Abschnitt 3.2 *Augenbasierte Interaktionstechniken*) nicht zur natürlichen Interaktion beiträgt. Die Verweilzeit ist auch eine Methode, die prinzipbedingt zeitabhängig ist. Während der Tests wurde festgestellt, dass sie am Besten mit einer Möglichkeit zur individuellen Konfiguration des Schwellwertes ausgestattet werden müsste, da auf Grund von unterschiedlichen Wahrnehmungsgewohnheiten nur so ein näherungsweise natürlicher Umgang sichergestellt werden kann. Aus zeitlicher Sicht, kann zwischen der Taste, dem Zwinkern und der Verweilzeit keine deutliche Tendenz erkannt werden.

Bei der Frage nach der intuitivsten Methode zeichnete sich auch kein eindeutiger Favorit ab (das Zeigen wurde auch hier außer Konkurrenz betrachtet). Die Verweilzeit wurde im Zusammenhang mit längerer Interessensbekundung für eine Blickrichtung befürwortet. Weiterhin war auch das Zwinkern sehr beliebt. Die klassische und gewohnte Betätigung einer Taste wurde wegen der Zuverlässigkeit hervorgehoben. Je nach Anwendung können unterschiedliche Auslösemechanismen von Vorteil sein, aber auch persönliche Präferenzen spielen eine entscheidende Rolle, deshalb ist sicherlich die Wahlmöglichkeit zwischen verschiedenen Auslösemechanismen ein gutes Konzept, um eine natürliche Interaktion zu ermöglichen.

festzuhalten. Das natürliche Verhalten und der resultierende Blickpfad wurde dadurch entschieden gestört. Außerdem hätte die Latenz das Ergebnis verfälscht, da optimaler Weise eine umgehende Reaktion erwartet wird. Eine Implementierung, die dieses Problem berücksichtigt, ist mit etwas Aufwand möglich. Das implizite Auslösen durch Zeigen ist mit Abstand die schnellste Methode und außer Konkurrenz, da man nur auf das Objekt schaut und nichts weiter für die Selektion erforderlich ist. Die kognitive Last ist sehr gering, das wird

6.4 Selektion dynamischer Objekte

Die **dritte Aufgabe** profitiert vom Spieltrieb der Probanden. In der Aufgabe wurde die Anzahl der selektierten Objekte notiert, deshalb gelten im Gegensatz zu den Diagrammen der anderen Aufgaben höhere Werte als besser. Es ist deutlich zu sehen, dass der Blickzeiger schlecht abschneidet und auch hier nicht die erwarteten Ergebnisse liefert. Gerade

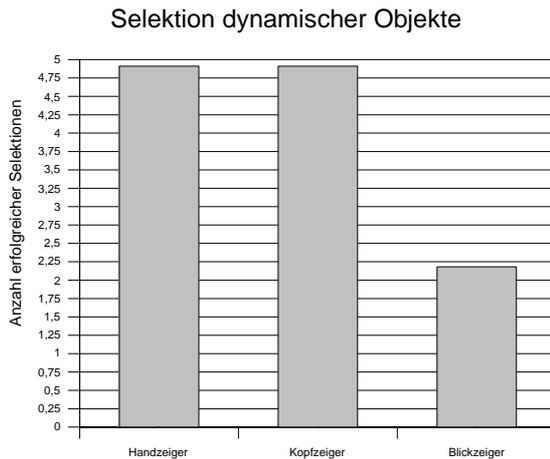


Abbildung 54: Selektion dynamischer Objekte

in dieser Aufgabe ist die Kopfbewegung beim Einsatz des Blickzeigers notwendig, die bereits als Ursache für die Ungenauigkeit des Eye-Trackers identifiziert wurde. Während die Probanden interagierten, konnte man deutlich sehen, dass der Blickzeiger, wenn auch mit Offset, meist parallel mit nahezu konstantem Abstand dem Objektpfad folgte. Deshalb könnte er theoretisch, ohne die Fehlerquellen, als der am Besten geeignete im Vergleich abschneiden.

Auch bei dieser Frage fielen die Antworten wegen des Kalibrierproblems etwas unterschiedlich aus. Die Frage nach dem intuitivsten Zeiger für die Aufgabe wurde unter Einbeziehen der Begründungen eindeutig mit Blickzeiger beantwortet. Allerdings ist das bekannte Problem mit der Kalibrierung betont worden. Der Einsatz des Handzeigers wurde als bedingt tauglich für die Verfolgung bewegter Objekte eingeschätzt, weil er näherungsweise nur in zwei Dimensionen bewegt werden kann und brauchbare Ergebnisse nur nach Eingewöhnung oder mit Erfahrung erzielt werden können. Es wurde außerdem beobachtet, dass manche Anwender den Handzeiger hauptsächlich über das Handgelenk bewegten, andere eher aus dem Schultergelenk. Auch die Kombination von Hand- und Schulterbewegung wurde beobachtet sowie auch das Zielen wie mit einer Waffe. Dabei wurde deutlich, dass die Art und Weise der Bedienung auch eine entscheidende Rolle für die Treffsicherheit spielt und individuell ist. Quantitative Erhebungen wurden in diesem Zusammenhang nicht gemacht.

6.5 Interaktion durch Bewegung

In der letzten und **vierten Aufgabe** kommt das besprochene Modell zur Interaktion erstmals vollständig zum Einsatz. Die Modifikation der Position eines Objekts durch **Zeigen, Auslösen** und **Bewegen** ist das Ziel. Außerdem wird die Kombination von verschiedenen Zeigern zur Interaktion getestet. Die Selektion erfolgte mit dem Blickzeiger, während das selektierte Objekt dann mit einem der drei unterschiedlichen Zeiger in das Ziel bewegt wurde. Im Diagramm der

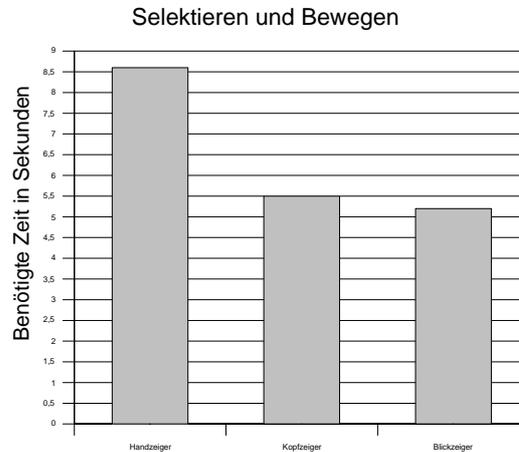


Abbildung 55: Bewegung

Abbildung 55 sind die Ergebnisse für die drei Fälle dargestellt. Dabei fällt auf, dass mit dem Handzeiger am meisten Zeit gebraucht wurde, um das Objekt durch den Raum zu bewegen. Interessant ist in diesem Zusammenhang das Ergebnis der folgenden Frage. Obwohl mit dem Handzeiger die meiste Zeit für die Aufgabe benötigt wurde, steht er, auf Grund der gewohnten Nutzung, als Favorit zum Bewegen von Objekten fest.

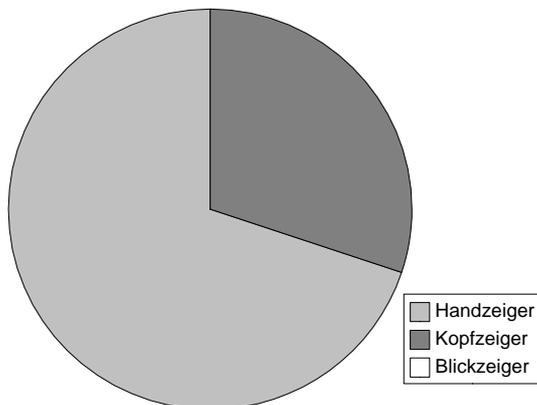


Abbildung 56: Bevorzugter Zeiger zum Bewegen

während des Positionierens auch dem natürlichen Explorationsverhalten nachkommt, das zur Lokalisierung des Ziels notwendig ist.

Die Antwort der Probanden auf die Frage nach dem bevorzugten Mittel zum Bewegen von Objekten ist in Abbildung 56 visualisiert. Den Begründungen zur Antwort war zu entnehmen, dass man naturgemäß Dinge oder Objekte in seiner Umgebung mit der Hand bewegt, außerdem kann im Gegensatz zum Handpointer die räumliche Tiefe zum Positionieren nur schwer mit Blick- oder Kopfzeiger eingeschätzt werden. Der Blickzeiger wurde weiterhin als zu unruhig eingestuft, da er

6.6 Resümee der Evaluation

Die Evaluation der Blickinteraktion im Kontext gewohnter und natürlicher Interaktionen hat deutlich gezeigt, wo die technischen Schwächen beim Eye-Tracking zur Interaktion in Mixed Reality Umgebungen besonders liegen, aber auch welche Einsatzmöglichkeiten in natürlichen Benutzerschnittstellen als geeignet erscheinen. Dabei wurden vorhandene Erkenntnisse bestätigt, aber auch spezielle Probleme identifiziert. Als Erstes ist das aufwendige Kalibrieren des Eye-Trackers zu nennen, zu dem auch das Rekalibrieren gezählt werden kann, das zum Ausgleich der Positions-Offsets notwendig ist, die vordergründig aus dem Verrutschen der AR-Anzeige auf dem Kopf des Anwenders resultieren. Als Ursachen dafür wurde das Gewicht, der Schwerpunkt, die Kabel und die Ergonomie der AR-Anzeige festgestellt.

Der Einsatz der Blickinteraktion in Mixed Reality Umgebungen ist primär für die statische und dynamische Selektion von Objekten geeignet, aber auch in Anwendungsbereichen, in denen die Hände und/oder die Sprache nicht zur Verfügung stehen, da durch das Zwickern mit dem Auge ein alternativer Auslösemechanismus umgesetzt werden kann. Zum Bewegen von Objekten eignet sich der Handzeiger besser, da der Mensch es von Natur aus gewohnt ist, Dinge mit den Händen zu positionieren.

EVALUATION

7 Zusammenfassung und Ausblick

Die Zusammenfassung bietet die zentralen Aspekte dieser Arbeit in komprimierter Form an. Darauf folgt der Ausblick, der sowohl auf erforderliche nächste Schritte für die weitere Entwicklung eingeht, wie auch mögliche Ziele für die Zukunft aufzeigt.

7.1 Zusammenfassung

Die Ausgangssituation für den Einsatz des Eye-Tracking zur Interaktion in Mixed Reality Umgebungen definiert sich durch den Stand von Technik und Forschung in den verschiedenen Disziplinen, die zur Realisierung der Blickinteraktion in einer Mixed Reality Umgebung beitragen. Darum ging der praktischen Arbeit eine umgehende Recherche voran. Zu den zentralen beteiligten Disziplinen zählen unter anderem die Wahrnehmungspsychologie, das Eye-Tracking, die Mensch-Maschine-Kommunikation und die Informatik.

Wahrnehmung steht am Anfang jeder Naturwissenschaft und ist elementar für unser Wissen über die Welt. In dieser Arbeit ist sie von besonderer Bedeutung, da Mixed Reality durch sie entsteht und der visuellen Sinn, als ein Teil von ihr, mittels des Eye-Tracking zur Interaktion mit der Mixed Reality eingesetzt wird. Die historische Entwicklung der Wahrnehmung von künstlichen Welten im Kontext gesellschaftlicher Strukturen sowie Erkenntnisse über die menschliche Wahrnehmung für die Mixed Reality Technologie und die technische Umsetzung von Blickinteraktion liefern einleitend einen Bezug zu möglichen und bestehenden Mixed Reality Anwendungen.

ZUSAMMENFASSUNG UND AUSBLICK

Die Voraussetzungen für die Auswahl und den erfolgreichen Einsatz von Eye-Tracking Systemen sind einerseits das Wissen über das visuelle System des Menschen und andererseits die fundierten Kenntnisse über Eye-Tracking und Mixed Reality Technologien. Die notwendigen Informationen wurden zusammengetragen und festgehalten. Sie dienen als Grundlage zur Definition von Anforderungen an Eye-Tracking Systeme, die sich für den Einsatz in Mixed Reality Umgebungen eignen. Die Eigenentwicklung eines Eye-Tracking Systems wurde aus Zeitgründen nicht in Betracht gezogen. Von 10 verschiedenen Herstellern wurde ein System anhand der definierten Vergleichskriterien ausgewählt und beschafft. Eine speziell konzipierte und realisierte Haltevorrichtung ermöglicht die Integration der Hardware des Systems in verschiedene HMDs. Die Software des Eye-Tracking Systems ist in bestehende Mechanismen des Morgan Frameworks unter zeitkritischem Betrachten der dynamischen Daten und durch Berücksichtigen von wahrnehmungsrelevanter Synchronität integriert worden. Diese Kriterien wurden auch bei der Entwicklung eines generischen Konzepts zur Kombination von geometrischen, kontinuierlichen Daten berücksichtigt, das sich ebenfalls in die bestehenden Mechanismen des Frameworks einfügt. Die geometrischen Daten des Eye-Tracking stehen dadurch in verschiedenen Bezugssystemen allen MR-Applikationen zur Verfügung, die im Framework oder auf Basis der API des Frameworks entwickelt werden.

Um den Blick zur Interaktion einsetzen zu können und um unter Berücksichtigung von intuitiven, natürlichen und gewohnten Handlungen des Menschen evaluieren zu können, wie und wo die Augenbewegung sinnvoll in Verbindung mit weiteren Modalitäten zur Eingabe eingesetzt werden kann, ist eine Anwendung mit einer geeigneten multimodalen Benutzerschnittstelle erforderlich. Für die Implementierung der Anwendung wurde ein Interaktionsmodell als Basis hergeleitet, aus der die MR-Anwendung „Eye Interact“ entstand. Sie berücksichtigt die Wahrnehmung für die stereoskopische Kalibrierung und Rekalibrierung des Eye-Trackers, ermöglicht die generische Anbindung von VRML Szenen zur Interaktion, besitzt spezielle Möglichkeiten zur Fehlersuche, ist applikationsübergreifend einsetzbar und bietet eine Basis zur Evaluation multimodaler Interaktionsmöglichkeiten.

Die Evaluation der Blickinteraktion hat ergeben, dass ein erheblicher Aufwand erforderlich ist, um den Eye-Tracker zu kalibrieren. Dazu zählt auch das häufige Rekalibrieren, das Positions-Offsets ausgleicht, die schon durch minimales Verrutschen des AR-Displays auf dem Kopf des Anwenders entstehen. Beim Bewegen von Objekten hat sich ergeben, dass ein von Hand bedienter Zeiger dafür am besten geeignet ist. Weiterhin stellte sich heraus, dass der Einsatz von Blickinteraktion in Mixed Reality Umgebungen sich primär für die Selektion statischer und dynamischer Objekte eignet, aber auch in Anwendungsbereichen, in denen die Hände und/oder die Sprache nicht zur Verfügung stehen, als alternatives Interaktionsmittel eingesetzt werden kann.

7.2 Ausblick

Die Evaluation hat gezeigt, dass die Blickinteraktion nur mit erheblichem Kalibrieraufwand möglich ist. Daraus folgt die Forderung nach der Vereinfachung und Automatisierung des kompletten Kalibriervorgangs. Die automatische Rekalibrierung, die keine Benutzerinteraktion erfordert und ein Eye-Tracking System, das einen äußerst robusten Bildverarbeitungsalgorithmus zum Extrahieren der Blickrichtung einsetzt, ist für den reibungslosen Einsatz wünschenswert. Unter dieser Voraussetzung könnten dann die Vorteile der Blickinteraktion, besonders im Rahmen einer natürlichen Benutzerschnittstelle, einen echten Mehrwert bringen. Die extreme Geschwindigkeit des Blicks und vor allem das Interesse des Anwenders, das in seinem Blickpfad codiert ist, sind vielversprechende Parameter für die Blickinteraktion der Zukunft.

Eine erneute Evaluation der Blickinteraktion im Kontext multimodaler und natürlicher Benutzerschnittstellen ist in einem größeren Rahmen über einen längeren Zeitraum unter den neuen Voraussetzungen notwendig. Der Anwender kann dann in einer ausgiebigen Eingewöhnungsphase Erfahrung mit völlig neuen Interaktionsmöglichkeiten sammeln und sich etwas an völlig fremde Interaktionsformen gewöhnen. Dadurch ist dann eine Evaluation möglich, die nachhaltige Ergebnisse liefert. Die Szenarien sollten klassische Interaktionssituationen bieten und das Design der Benutzerschnittstelle sollte dabei möglichst vielfältige Interaktionen ermöglichen und dem Benutzer Optionen zur Konfiguration bieten. Aus den verschiedenen Konfigurationsprofilen und den gesammelten Erfahrungen der Probanden könnten dann allgemeine Präferenzen abgeleitet werden, die in die Entwicklung von natürlichen, intuitiven und multimodalen Benutzerschnittstellen einfließen würden.

Eine gewinnbringende Einsatzmöglichkeit von bidirektionaler visueller MR ist die Blickinteraktion in Situationen, in denen weder Hände noch die Sprache zur Interaktion geeignet sind. Ein einfaches, rein visuelles MR-System könnte aus einem Display und einem Eye-Tracker bestehen. Virtuelle Informationen könnte man in die reale Umgebung einblenden, der Blick würde „selektieren“ und das Zwinkern „auslösen“. Der Anwender könnte dadurch zum Beispiel in einer Menüstruktur navigieren oder aus einer Liste Information auswählen. Sensible und nicht öffentliche Informationen könnten diskret selektiert und konfiguriert werden. Weiterhin ist die Auswahl von Kommandos zur Steuerung verteilter Geräte denkbar.

In progressiven, multimodalen Benutzerschnittstellen, die auf intuitive, natürliche und gewohnte Interaktion setzen, kann auf den Blick zum Selektieren nicht verzichtet werden. Durch Auswerten der Historie von betrachteten Objekten, in der die Fixierungsdauer und Häufigkeit des Blicks enthalten sind, sind Rückschlüsse auf das Interesse des

ZUSAMMENFASSUNG UND AUSBLICK

Anwenders denkbar. Dadurch könnten Informationen im Kontext der aktuellen Wahrnehmung angeboten werden. Ein entscheidender Faktor dafür ist das EWK (Extent of World Knowledge), das dem objektorientierten Intressenhistogramm Information über die Objekte und die Semantik in der Szene liefert. Ein ausgeprägtes semantisches Modell ist dabei die Basis, die es erlaubt Anwendungskontext fließend zu „augmentieren“.

Behinderte die auf Blick basierende Kommunikation und Interaktion angewiesen sind könnten einerseits praktische Erfahrung beitragen und andererseits von der weiteren Entwicklung enorm profitieren.

Der Rendervorgang für die Anzeige könnte durch den Einsatz von Eye-Tracking optimiert werden, da die Zerlegung von Kurven in Liniensegmente an die physiologischen Gegebenheiten des Auges angepasst werden kann, ähnlich wie bei der MPEG I Layer 3 Kodierung das Wissen über die hörpsychologischen Komponenten ausgenutzt wurde, um einen Kompressionsfaktor von 12:1 zu erreichen. Der Einsatz beim triangulierenden Rendern, dem gegenwärtigen Standard, wurde soeben beschrieben. Nachdem die Zukunft in Hinblick auf die Reproduktionsgenauigkeit RF (Reproduction Fidelity) dem Ray Tracing mit Echtzeit Radiosity gehört und leider noch keine entsprechende Hardware bezahlbar ist mit der dies Wohnzimmerauglich angeboten werden kann, könnte ein geeignetes Renderengine in Verbindung mit Eye-Tracking eine Lösung bieten. Die Rechenlast für die Anzeige würde dadurch in einem ersten Schritt im Sinne der Wahrnehmung optimiert werden. Schließlich könnte man durch berücksichtigen von Tiefenunschärfe eine Steigerung der EPM (Extent of Presence Metaphor) erreichen.

Auf EWK, RF und EPM wird im Abschnitt *1.1 Mixed Reality* näher eingegangen.

A Quellennachweise

A.1 Literaturverzeichnis

- [Azu97] Azuma, R. T., „A Survey of Augmented Reality“, In *Teleoperators and Virtual Environments*, Vol. 6, No. 4, 1997, 355-385
- [Azu01] Azuma, R. T., Bailiot, Y., Behringer, R., Feiner, S., Julier, S., Macintyre, B., „Recent Advances in Augmented Reality“, In *Computer Graphics and Applications, IEEE*, Vol. 21, No. 6, 2001, 34-47
- [Bal91] Balota, D. A., Rayner, K., *Word recognition processes in foveal and parafoveal vision: the range of influence of lexical variables*, D. Besner & G. W. Humphreys(Eds.), Basic Processes in Reading: Visual word recognition. Hillsdale, NJ, England: Lawrence Erlbaum Associates, 1991
- [Ben36] Benjamin, W., *Das Kunstwerk im Zeitalter seiner technischen Reproduzierbarkeit*, Suhrkamp Verlag, 2003
- [Ben04] Benko, H., Ishak, E. W., Feiner, S., „Collaborative Mixed Reality Visualization of an Archaeological Excavation“, In *ISMAR '04: Proceedings of the Third IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'04)*, IEEE Computer Society, 2004, 132-140
- [Bof88] Boff, K. R., Lincoln, J. E., *Engineering Data Compendium: Human Perception and Performance*, Armstrong Aerospace Medical Research Laboratory (AAMRL), Wright-Patterson AFB, OH, 1988
- [Bor94] Bormann, S., *Virtuelle Realität: Genese und Evaluation*, Addison-Wesley Verlag, 1994
- [Bro98] Broll, W., *Ein objektorientiertes Interaktionsmodell zur Unterstützung verteilter virtueller Umgebungen*, GMD - Forschungszentrum Informationstechnik, (GMD Research Series; 1998, No.11) Zugl.: Tübingen, Univ., Diss., 1998
- [Bro04b] Broll, W., Grünvogel, S., Herbst, I., Lindt, I., Maercker, M., Ohlenburg, J., Wittkämper, M., „Interactive Props and Choreography Planning with the Mixed Reality Stage“, In *Proc. of ICEC 2004*, , 2004, 185-192
- [Bro04a] Broll, W., Lindt, I., Ohlenburg, J., Wittkämper, M., Yuan, C., Novotny, T., Mottram, C., Fatah, A., Strothmann, A., „ARTHUR:

- A Collaborative Augmented Environment for Architectural Design and Urban Planning“, In *Proc. of the fourth International Symposium on Human and Computers (HC 2004)*, , 2004, 102-109
- [Bur03] Burdea, G. C., Coiffet, P., *Virtual Reality Technology*, John Wiley & Sons, Inc., Hoboken, New Jersey, 2003
- [Cam02] Camus, T. A., Wildes, R., „Reliable and Fast Eye Finding in Close-up Images“, In *Proceedings of the 16 th International Conference on Pattern Recognition (ICPR'02) Volume 1*, IEEE Computer Society, 2002, 10389
- [Doi03] Doil, F., Schreiber, W., Alt, T., Patron, C., „Augmented reality for manufacturing planning“, In *EGVE '03: Proceedings of the workshop on Virtual environments 2003*, ACM Press, 2003, 71-76
- [Dro96] Drosdowski, G., Müller, W., Scholze-Stubenrecht, W., Wermke, M., *Duden, Die deutsche Rechtschreibung*, Dudenverlag, 1996
- [Dur00] Duchowski, A. T., Vertegaal, R., „Course 05: Eye-Based Interaction in Graphical Systems: Theory & Practice“, In *SIGGRAPH 2000 Course Notes*, ACM SIGGRAPH, New York, NY, 2000
- [End05] Endres, C., Butz, A., Macwilliams, A., „A Survey of Software Infrastructures and Frameworks for Ubiquitous Computing“, In , Vol. 1, No. 1, Jan–mar 2005, IOS press, Amsterdam, 2005
- [Gam03] Gambke, J., *persönliche Kommunikation*, am 06.12.2004, 2004
- [Ike03] Ikeuchi, K., Nakazawa, A., Hasegawa, K., Ohishi, T., „The Great Buddha Project: Modeling Cultural Heritage for VR Systems through Observation“, In *ISMAR '03: Proceedings of the The 2nd IEEE and ACM International Symposium on Mixed and Augmented Reality*, IEEE Computer Society, 2003, 7
- [Ina03] Inamoto, N., Saito, H., „Immersive Observation of Virtualized Soccer Match at Real Stadium Model“, In *ISMAR '03: Proceedings of the The 2nd IEEE and ACM International Symposium on Mixed and Augmented Reality*, IEEE Computer Society, 2003, 188
- [Ish02] Ishii, H., Ben-Joseph, E., Underkoffler, J., Yeung, L., Chak, D., Kanji, Z., Piper, B., „Augmented Urban Planning Workbench: Overlaying Drawings, Physical Models and Digital Simulation“, In *ISMAR '02: Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR'02)*, IEEE Computer Society, 2002, 203
- [Jac93] Jacob, R. J. K., „Eye-gaze Computer Interfaces: What You Look At is What You Get“, In *IEEE Computer*, Vol. 26, No. 7, 1993, 65-67
- [Jac03] Jacob, R. J., Karn, K. S., „Eye tracking in human-computer interaction and usability research: Ready to deliver the

- promises“, In *The Mind's Eyes: Cognitive and Applied Aspects of Oculomotor Research*, Hyönä, J., Radach, R., Deubel, H., eds. Elsevier Science, Oxford, 2003, 573-605
- [Joo02] Joos, M., Rötting, M., Velichkovsky, B. M., „Bewegung des menschlichen Auges: Fakten, Methoden und innovative Anwendungen“, In *Handbuch der Psycholinguistik*, T. Herrmann, S. Deutsch & G. Rickheit (Hrsg.), Berlin: De Greyter, 2002
- [Jos02] Josuttis N., M., *The C++ Standard Library*, Addison-Wesley Verlag, 2002
- [Kai03] Kaiser, E., Olwal, A., Mcgee, D., Benko, H., Corradini, A., Li, X., Cohen, P., Feiner, S., „Mutual disambiguation of 3D multimodal interaction in augmented and virtual reality“, In *ICMI '03: Proceedings of the 5th international conference on Multimodal interfaces*, ACM Press, 2003, 12-19
- [Kat03] Kato, H., Tachibana, K., Tanabe, M., Nakajima, T., Fukuda, Y., „A City-Planning System Based on Augmented Reality with a Tangible Interface“, In *ISMAR '03: Proceedings of the The 2nd IEEE and ACM International Symposium on Mixed and Augmented Reality*, IEEE Computer Society, 2003, 340
- [Kli02] Klinker, G., Dutoit, A. H., Bauer, M., Bayer, J., Novak, V., Matzke, D., „Fata Morgana - A Presentation System for Product Design“, In *ISMAR '02: Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR'02)*, IEEE Computer Society, 2002, 76
- [Koy03] Koyama, T., Kitahara, I., Ohta, Y., „Live Mixed-Reality 3D Video in Soccer Stadium“, In *ISMAR '03: Proceedings of the The 2nd IEEE and ACM International Symposium on Mixed and Augmented Reality*, IEEE Computer Society, 2003, 178
- [Lev00] Levine, M. W., *Fundamentals of Sensation and Perception*, Oxford University Press, 2000
- [Lin03] Lindt I., Herbst I., Maercker M., „Interacting with the Mixed Reality Stage“, In , Nadia(Hrsg.), Magnenat-Thalmann, 2003
- [Mck04] Mckenzie, F. D., Garcia, H. M., Castelino, R. J., Hubbard, T. W., Ullian, J. A., Gliva, G. A., „Augmented Standardized Patients Now Virtually a Reality“, In *ISMAR '04: Proceedings of the Third IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'04)*, IEEE Computer Society, 2004, 270-271
- [Mil02] Milekic, S., „The more you look the more you get: Intention-based interface using gaze-tracking“, In *Museums and the Web 2002 Selected Papers from an International Conference*, Archives & Museum Informatics, D. Bearman, and Trant, J. (eds.), Pittsburgh, PA, 2002

- [Mil94] Milgram, P., Kishino, F., „A Taxonomy of Mixed Reality Virtual Displays“, In *IEICE Transactions on Information and Systems*, E77-D, 9, 1994
- [Nov04] Novak, V., Sandor, C., Klinker, G., „An AR Workbench for Experimenting with Attentive User Interfaces“, In *Proc. of IEEE and ACM International Symposium on Mixed and Augmented Reality*, Arlington, Va, Usa, 2004
- [Ohl04] Ohlenburg, J., Herbst, I., Lindt, I., Fröhlich, T., Broll, W., „The MORGAN Framework: Enabling Dynamic Multi-User AR and VR Projects“, In *Proc. of the ACM Symposium on Virtual Reality Software and Technology (VRST 2004)*, ACM, 2004, 166-169
- [Ohn02] Ohno, T., Mukawa, N., Yoshikawa, A., „FreeGaze: a gaze tracking system for everyday gaze interaction“, In *ETRA '02: Proceedings of the symposium on Eye tracking research & applications*, ACM Press, 2002, 125-132
- [Rog04] Rogers, Y., Price, S., Fitzpatrick, G., Fleck, R., Harris, E., Smith, H., Randell, C., Muller, H., O'malley, C., Stanton, D., Thompson, M., Weal, M. „Ambient wood: designing new forms of digital augmentation for learning outdoors“, In *IDC '04: Proceeding of the 2004 conference on Interaction design and children*, ACM Press, 2004, 3-10
- [Roh85] Rohen, J. W., *Funktionelle Anatomie des Nervensystems*, 4.Aufl. Stuttgart: Schattauer Verlag, 1985
- [Sch04a] Schraefel, M. C., Hughes, G. V., Mills, H. R., Smith, G., Payne, T. R., Frey, J., „Breaking the book: translating the chemistry lab book into a pervasive computing lab environment“, In *CHI '04: Proceedings of the 2004 conference on Human factors in computing systems*, ACM Press, 2004, 25-32
- [Sch04b] Schrempp, U., *Benutzerindividuelle Registrierung in Anwendungen der Erweiterten Realität*, Diplomarbeit, Fachhochschule Offenburg, 2004
- [Shn98] Shneiderman, B., *Designing the user interface: strategies for effective human-computer interaction / 3rd edition*, Addison Wesley Longman, Inc, 1998
- [Sib00] Sibert, L. E., Jacob, R. J. K., „Evaluation of eye gaze interaction“, In *CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM Press, 2000, 281-288
- [Sol04] Soler, L., Nicolau, S., Schmid, J., Koehl, C., Marescaux, J., Pennec, X., Ayache, N., „Virtual Reality and Augmented Reality in Digestive Surgery“, In *ISMAR '04: Proceedings of the Third IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'04)*, IEEE Computer Society, 2004, 278-279

- [Stö00] Stöcker, H., *Taschenbuch der Physik*, Harri Deutsch Verlag, 2000
- [Tan03] Tang, A., Owen, C., Biocca, F., Mou, W., „Comparative effectiveness of augmented reality in object assembly“, In *CHI '03: Proceedings of the conference on Human factors in computing systems*, ACM Press, 2003, 73-80
- [Tan00] Tanriverdi, V., Jacob, R. J. K., „Interacting with eye movements in virtual environments“, In *CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM Press, 2000, 265-272
- [Tho03] Thomas, B. H., Piekarski, W., „Outdoor virtual reality“, In *ISICT '03: Proceedings of the 1st international symposium on Information and communication technologies*, Trinity College Dublin, 2003, 226-231
- [Tim02] Timpe, K.-P., Jürgensohn, T., Kolrep, H., *Mensch-Maschine-Systemtechnik*, Symposion Publishing, 2002, 07.01.2005, http://www.symposion.de/mensch-maschine/mms_07.htm
- [Vog03] Vogt, S., Khamene, A., Sauer, F., Keil, A., Niemann, H., „A High Performance AR System for Medical Applications“, In *ISMAR '03: Proceedings of the The 2nd IEEE and ACM International Symposium on Mixed and Augmented Reality*, IEEE Computer Society, 2003, 270
- [Wil02] Wilder, J., Hung, G. H., Tremaine, M. M., Kaur, M., „Chapter 8. Eye Tracking in Virtual Environments“, In *Handbook of Virtual Environments*, Stanney, K. M., Lawrence Erlbaum Associates, Inc., 2002
- [Woh00] Wohlgenuth, W., Triebfürst, G. „ARVIKA: augmented reality for development, production and service“, In *DARE '00: Proceedings of DARE 2000 on Designing augmented reality environments*, ACM Press, 2000, 151-152

A.2 Internet Referenzen

Alle hier referenzierten Internetseiten waren am 06.01.2005 erreichbar.

- [url1] <http://www.well.com/user/jaron/>
<http://www.advanced.org/jaron/general.html>
- [url2] <http://www.fit.fraunhofer.de>
- [url3] <http://www.omg.org>
- [url4] <http://telelab.mi.fh-offenburg.de/Vorlesung/Wahrnehmungspsychologie/Psy-K2-Visuell-PhysiologieOT/index.htm>
- [url5] <http://rcswww.urz.tu-dresden.de/~cogsci/?/~cogsci/augenbewegungen.html>
- [url6] <http://vret.ces.clemson.edu/sigcourse/>
- [url7] <http://www.fit.fraunhofer.de/gebiete/usability/>
- [url8] <http://www.brl.ntt.co.jp/people/takehiko/freegaze/>
- [url9] <http://page.mi.fu-berlin.de/~vratisla/Bildverarbeitung/GeoTrans/GeoTrans.html>
- [url10] <http://www.edu.uni-klu.ac.at/~dpeintne/MobiLearn/MathematischeGrundlagen/GeometrischeTransformation.html>
- [url11] http://www.biols.susx.ac.uk/home/Georg_Mather/Linked%20/Pages/Physiol/LGN.html
- [url12] http://www.mathacademy.com/pr/mini-text/escher/big.asp?IMAGE=cube_with_ribbons
- [url13] <http://www.eyetoy.com>
- [url14] http://www.wearable.ethz.ch/fileadmin/edu/projects/ws01/vrdpup/vrdpup_bericht_2.pdf
- [url15] <http://www.web3d.org/x3d/vrml/index.html>
- [url16] <http://www.web3d.org/x3d/>
- [url17] <http://www.wintimesync.com>
- [url18] <http://www.faqs.org/rfcs/rfc2030.html>
- [url19] http://informatiksysteme.pt-it.de/vr-ar-3/projekte/mqube/paper_MQUBE.pdf
- [url20] <http://www.opengl.org/>
- [url21] <http://www.cs.tufts.edu/~jacob/papers/>
- [url22] <http://telelab.mi.fh-offenburg.de/Vorlesung>
- [url23] <http://www.3dgaming.de>
- [url24] <http://www.pervasive-gaming.org/>
- [url25] <http://www.fit.fraunhofer.de/projekte/netattack/>

B Vergleichstabelle Eye-Tracking Systeme

Hersteller	Produkt	FoV		T	Auflösung Track	Blickpos.	Sampling	Gewicht	Größe	Preis	AR/VR tauglich	Bauweise	Binokular	API
		H	V											
Optom		+/-15°	---		<0,1°		1kHz			6.620 €				Seriell
Polhemus	Visiontrack									12.246 €				no
Electronica		+/-30°	+/-20°	+/-45°	+/-0,016°	+/-0,1°	480/240/60Hz							
El-mar		+/-40°	+/-30°	---		+/-0,1°	120Hz	300g						
SMI	iView	+/-30°	+/-20°	---	<0,1°	0,5-1°	50/60 Hz	450g		20.000 € +Int. ca. 9.000 €	Teilweise	HMD integriert		UDP
Ect		+/-15°	+/-10°	---		1°	?	12g		~6.600 €	spec. Integr.	mini-HMD in- tegriert		ja
Eye Link	Eye Link II α	+/-20°	+/-18°	---	<0,022°	<0,5°	250Hz	420g				ext. vor. see through		
Eye Link	Eye Link II po	+/-30°	+/-20°	---	<0,01°	<0,5°	500Hz	420g		23.600 €	Teilweise	ext. vor. see through	X	C/Python
ASL	501	50°	40°	---	0,1°	0,5°	50/60 Hz	241g		24.432 €		ext. vor. see through		
ASL	501vr	50°	40°	---	0,1°	0,5°	60 Hz	35g	11/47mm	27.500 €	AR/VR	mini-HMD in- tegriert		C++/RS232
Arrington	ViewPoint	+/-44°	+/-20°	---	0,15°	0,25-1°	ähnlich as 501vr			5.500 € +Int. 3.800 €	spec. Integr.	(mm) - HMD integriert		C++
Arrington	ViewPoint SDK	---	---	---	---	---	---	---	---	3.400 €	Ohne Optik	---	---	C++
nVision	DV See Through							1,6 kg		~130.000 €				

Hersteller	Produkt	API Features	Kalibrierung	Infrarot	Realtime Parameter	Lieferumfang
Optom						
Polhemus	Visiontrack			X		
Electronica						
El-mar						
SMI	iView	Control important functions over UDP	über UDP initialisierbar	X	Blickposition, Pupillendurchmesser etc	Komplettsystem ind. PC
Ect					?	
Eye Link	Eye Link II cr	s.u.	s.u.	X	s.u.	s.u.
Eye Link	Eye Link II po	Fernsteuerung der meisten Funktionen der Software	3-13 Punkte		3-10ms: Blickrichtung, Pupillengröße 30ms: Saccaden, Fixationen, Blinzeln	Host: Application, C Developers Kit, Python Package, EyeLink II Board
ASL	501	s.u.	s.u.	X	s.u.	Series 5000 Control Unit, 2 Monitore, Headmounted Scene Camera, Operating and Analysis Software
ASL	501vr	get/access all Parameters of external control unit, blinkfilter	9 Punkte	X	Blickrichtung, Pupillendurchmesser	vermutlich s.o.
Arrington	ViewPoint	s.u.	User Profile, Auto-Calibrate, Slip-Correction	X	s.u.	Software, capture board, Camera, illuminator, filter, close lens, sok
Arrington	ViewPoint SDK	Fernsteuerung aller Funktionen der Kalibrierungs Software, Blinkdetection	User Profile, Auto-Calibrate, Slip-Correction	X	0,1-0,5ms: Pupillenposition, Geschwindigkeit, Fixationen, Pupillenabmaße, Torsion	Software, capture board, sok
nVision	DV See Through	1280x1024 @ 78°ort mit ASL 501vr				AR HMD only - tracker integration by ASL

C Interaction Command Guide

Command Guide for the Eye Interact Application

		Commands				
		Voice	Keyboard	Button (Mouse)	Description	
Calibration		calibrate	tab	middle	Initiates the calibration procedure for the eye tracker.	
		improve			To perform the calibration you've got to look at the spinning cube until it has moved to the next position (always be aware around which axis it turns)	
		setup				
		correct	'	right	Initiates the slip correction procedure.	
Interaction	Selection	correction			This can be used to correct a calibration offset after the initial full calibration.	
		slip			The slip might occur between the HMD and the head of the user.	
		user	u		Loads user specific data of the eye calibration.	
	Objective	Move	markus	i		Loads my calibration settings (thanks the guy who put this together :-))
			gaze	e		Sets selection device to gaze .
			head	w		Sets selection device to head .
			point	q		Sets selection device to pointer .
			pointer			
			touch	?		Switches to „touch“ triggering.
			lock			
Interaction	Trigger	dwell	,		Switches to „dwell“ triggering.	
		trigger	ü		Switches to „action“ triggering.	
		ok	spacebar	left	Executes the action of the interaction context.	
	Objective	Action	next	enter		
			execute			
			go			
			blink	.		Switches to „blink“ triggering. (blink with your eye)
			select	1		Switches to Selection Mode .
			move	2		Switches to Move Mode which allows to move objects in the scene.
			bind			
Interaction	Objective	attach				
		translate				
		hold				
	Objective	Game		,		Sets move device to gaze .
				+		Sets move device to head .
				#		Sets move device to pointer .
			dynamic	3, g		Switches to a game where boxes move towards the user who has got to focus on them so they'll be removed.
			game			
			balls	h		... some gravity balls for playing around ...
Interaction	Scenes	remove	4		Deletes Objects in the scene.	
		grid	j		... a grid of boxes for a task.	
		five	k		... five boxes for a task. (mode 1)	
	Awareness	Awareness	floor	l		... some boxes on the floor for a task.
			box	o		... a box for moving task. (mode 2)
			feedback	f		Toggles the visual feedback on and off.
			menu			
			reset	r		Resets the point and time counter.
			points			
			time	z		Toggles the time count on/off.
	d		Toggles the debug mode on/off.			
	a		Toggles showing the bounding box after delete on and off.			
	m		Changes the crosshairs style			

D Aufgaben und Fragen zur Evaluation

Aufgabe 1

Selektieren Sie bei jedem Durchlauf die 5 virtuellen Objekte in der Szene. Zur Bestätigung der erfolgreichen Selektion erscheint ein blauer Rahmen. Die Selektion erfolgt durch zeigen und drücken der Taste.

Als Zeiger kommt zuerst der **Handzeiger**, dann der **Kopfzeiger** und abschließend der **Blickzeiger** zum Einsatz.

Frage: *Welchen Zeiger empfanden Sie als intuitivstes Interaktionsmittel?*

Antwort:

Begründung:

Aufgabe 2

Wie die erste Aufgabe, als Zeiger wird der **Blickzeiger** eingesetzt. Die Auswahl erfolgt mit den folgenden Auslösern:

Berührung, Sprache (ok, next, go, execute), **Zwinkern, Verweilzeit, Taste**

Frage: *Welchen Auslöser empfanden Sie am intuitivsten?*

Antwort:

Begründung:

Aufgabe 3

Bei dieser Aufgabe geht es darum möglichst viele der Objekte zu selektieren, die sich nacheinander aus zufälligen Richtungen durch den Raum bewegen. Für die erfolgreiche Selektion ist das Objekt mit dem Zeiger zu verfolgen.

Als Zeiger kommt zuerst der **Handzeiger**, dann der **Kopfzeiger** und abschließend der **Blickzeiger** zum Einsatz.

Frage: *Welchen Zeiger empfanden Sie als intuitivstes Interaktionsmittel?*

Antwort:

Begründung:

Aufgabe 4

In der letzten Aufgabe geht es darum einen Ordner auf die Ablage zu stellen.

Zum Zeigen wird der Blickzeiger angewandt. Die Auswahl erfolgt mit der Taste. Abgelegt bzw. deselektiert wird das Objekt durch drücken der Taste.

Die Bewegung des Objekts erfolgt zuerst durch den **Handzeiger**, dann mittels des **Kopfzeigers** und abschließend durch den **Blickzeiger**.

Frage: *Welche Kombination von Interaktionsmitteln würden Sie zum Bewegen von Objekten bevorzugen?*

Antwort:

Begründung:

Frage: *Wie empfanden Sie die Genauigkeit zur Positionierung im Ziel?*

Antwort:

Frage: *Für welche Anwendungen könnte man die Interaktionsmittel nutzen?*

Antwort: