

# Vereinfachung und Automatisierung der Erzeugung von virtuellen, stereoskopischen Inhalten

Fachbereiche IEM und MND der  
Fachhochschule Gießen-Friedberg

## Bachelorarbeit

vorgelegt von

**Matthias Clasen**

geb. in Leinefelde

Referent der Arbeit: Prof. Dr. Cornelius Malerczyk  
Korreferentin der Arbeit: Dipl.-Math. (FH) Sabine Langkamm



Fachbereiche

Informationstechnik-Elektrotechnik-Mechatronik IEM  
und  
Mathematik, Naturwissenschaften und Datenverarbeitung MND

Friedberg, 2010



# Danksagung

An dieser Stelle möchte ich all den Menschen danken, durch die diese Arbeit erst ermöglicht wurde.

Besonderer Dank geht dabei zunächst an den Betreuer und Referenten dieser Arbeit: Prof. Dr. Cornelius Malerczyk. Erst durch ihn wurde mir ein Praktikum ermöglicht, bei dem ich mein Hobby zur Arbeit machen konnte, woraus eine Erweiterung zum Thema der Bachelorarbeit wurde. Außerdem stand er in dieser Zeit für Ratschläge zur Verfügung und konnte mir bei Problemen helfen. Weiterhin danke ich Sabine Langkamm, die ebenfalls im Laufe dieser Arbeit zur Verfügung stand, sobald Fragen anfielen.

Ein weiterer großer Dank gilt Yvonne Jung, welche beim Fraunhofer IGD arbeitet. Sie stand mir bei Fragen zum InstantPlayer mittels eines Forums stets zur Seite und ermöglichte mir dadurch einen leichteren Einstieg in die Programmierung des Systems.

Ich danke auch meiner Familie, die mir meinen bisherigen Werdegang ermöglichte und besonders dazu meinem Bruder Michael Clasen, welcher ein sehr aufmerksamer und guter Korrekturleser dieser Arbeit war.

Letztlich danke ich an dieser Stelle meiner Freundin Frauke Schneider. Sie unterstützte in dieser Zeit stets mein mentales Wohl und besaß sehr viel Verständnis für die Zeit, welche ich für dieses Thema opferte.



# Selbstständigkeitserklärung

Ich erkläre, dass ich die eingereichte Bachelorarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die von mir angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Werken wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Friedberg, September 2010

Matthias Clasen



# Inhaltsverzeichnis

<b>Danksagung</b>	<b>i</b>
<b>Selbstständigkeitserklärung</b>	<b>iii</b>
<b>Inhaltsverzeichnis</b>	<b>v</b>
<b>Abbildungsverzeichnis</b>	<b>ix</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Problemstellung und Zielsetzung . . . . .	3
1.3 Organisation der Arbeit . . . . .	4
1.4 Zusammenfassung der wichtigsten Ergebnisse . . . . .	5
<b>2 Grundlagen</b>	<b>7</b>
2.1 Räumliches Sehen . . . . .	7
2.1.1 Akkommodation . . . . .	7
2.1.2 Konvergenz . . . . .	8
2.1.3 Querdisparation . . . . .	9
2.2 Stereoskopische Begriffe in der virtuellen Realität . . . . .	10
2.2.1 Interaxial Separation . . . . .	10
2.2.2 Zero Parallax Plane . . . . .	12
2.2.3 Parallaxenwinkel . . . . .	14
2.2.4 Parallaxentypen . . . . .	14
2.2.4.1 Null Parallaxe . . . . .	14
2.2.4.2 Negative Parallaxe . . . . .	15
2.2.4.3 Positive Parallaxe . . . . .	15
2.2.4.4 Divergente Parallaxe . . . . .	16
2.2.5 Besonderheiten stereoskopischer Kameras in der virtuellen Realität . . . . .	16
2.2.5.1 Parallele Kameras . . . . .	17
2.2.5.2 Konvergierende Kameras . . . . .	17
2.2.5.3 Off-Axis Kamera . . . . .	18
2.2.6 Zusammenfassung . . . . .	19

<b>3</b>	<b>Stereoskopische Anzeigeverfahren</b>	<b>21</b>
3.1	Freie Betrachtungsmöglichkeiten	21
3.1.1	Parallelblick	21
3.1.2	Kreuzblick	22
3.1.3	Autostereogramme	23
3.2	passive Anzeigeverfahren	24
3.2.1	Anaglyphen	25
3.2.2	Chromadepth	28
3.2.3	Polarisation	29
3.2.3.1	lineare Polarisation	29
3.2.3.2	zirkulare Polarisation	30
3.2.4	Interferenzfiltertechnologie	32
3.2.5	KMQ	33
3.3	Aktive Anzeigeverfahren	34
3.3.1	Shutterbrillentechnik	34
3.3.2	Head Mounted Displays	36
3.4	Autostereoskopische Displays	38
3.5	Zusammenfassung	40
<b>4</b>	<b>Mathematische Grundlagen für Optimierungen</b>	<b>41</b>
4.1	Formel „Native Pixel Parallax“	41
4.2	Formel nach John Bercovitz	42
4.3	Formel „1.5° maximaler Winkel“	43
4.4	Zusammenfassung	46
<b>5</b>	<b>Grundlagen Programmierung</b>	<b>47</b>
5.1	X3D	47
5.2	ECMAScript	49
5.3	MEL	50
5.4	Zusammenfassung	52
<b>6</b>	<b>Stand der Technik</b>	<b>53</b>
6.1	Darstellungsverfahren	53
6.1.1	Autostereoskopische Displays	53
6.1.1.1	Magnetic 3D 57" ALLURA ENABL3D™ LCD DISPLAY	53
6.1.1.2	Alioscopy 3D HD 47"	54
6.1.2	HEyeWall 2.0	55
6.1.3	Head Mounted Displays	56
6.1.3.1	Virtual Research VR1280	56
6.1.3.2	NVIS nVisor SX111	57
6.1.4	Geräte für den Heimgebrauch	57
6.1.4.1	Acer H5360	57
6.1.4.2	Sony KDL-60LX905	58
6.1.4.3	Nintendo 3DS	59

6.1.4.4	Fujifilm FinePix REAL 3D W3 . . . . .	60
6.2	Optimierte stereoskopische Projektionen . . . . .	61
6.2.1	Free2C Desktop Display . . . . .	61
6.2.2	Seereal C-i 18 . . . . .	62
6.2.3	IMSYS . . . . .	63
6.2.4	zukünftige Systeme . . . . .	65
6.2.4.1	Microvision Color Eyewear . . . . .	65
6.2.4.2	Retinal Image Display . . . . .	66
6.3	Zusammenfassung und Ausblick . . . . .	66
<b>7</b>	<b>Konzept stereoskopischer Vereinfachung in Autodesk Maya</b>	<b>69</b>
7.1	Beschreibung des Aufbaus vom Script . . . . .	69
7.2	Beschreibung der Funktionalitäten des Scripts . . . . .	70
7.3	Anforderungen an das Script . . . . .	71
7.4	Zusammenfassung . . . . .	71
<b>8</b>	<b>Implementierung des MEL Scripts</b>	<b>73</b>
8.1	Die Oberfläche des Scripts . . . . .	73
8.1.1	Überblick über die Funktionalitäten . . . . .	74
8.1.2	Hilfesystem . . . . .	74
8.2	Programmierung der Oberfläche . . . . .	75
8.3	Programmierung der Funktionalitäten . . . . .	76
8.3.1	Erstellung einer stereoskopischen Kamera . . . . .	76
8.3.2	Ersetzen einer vorhandenen Kamera . . . . .	78
8.3.3	Verfolgung eines Objekts der Szene . . . . .	79
8.3.4	Rendern einer Mehrfachansicht für autostereoskopische Displays . . . . .	80
8.3.5	Projektionsmaße bearbeiten . . . . .	82
8.3.6	Laden stereoskopischer Kameras aus der Szene . . . . .	83
8.4	Ergebnisse . . . . .	84
8.5	Zusammenfassung und Ausblick . . . . .	85
<b>9</b>	<b>Konzept stereoskopischer Automatisierung im InstantPlayer</b>	<b>87</b>
9.1	Beschreibung des Menüaufbaus . . . . .	87
9.2	Funktionalitäten des Menüs . . . . .	88
9.3	Anforderung an die Szene . . . . .	89
9.4	Zusammenfassung . . . . .	89
<b>10</b>	<b>Implementierung der X3D-Szene und des dazugehörigen Menüs</b>	<b>91</b>
10.1	Oberfläche des Menüs . . . . .	91
10.2	Aussehen der Funktionalitäten im Menü . . . . .	92
10.3	Aussehen des Webinterfaces . . . . .	93
10.4	Struktur der X3D-Szene . . . . .	94
10.5	Implementierung der Menüfunktionalitäten . . . . .	96
10.6	Implementierung der Optimierung stereoskopischer Parameter . . . . .	99

10.7 Ergebnisse . . . . .	101
10.8 Zusammenfassung und Ausblick . . . . .	102
<b>11 Zusammenfassung und Ausblick</b>	<b>103</b>
11.1 Ausblick . . . . .	104
<b>Glossar</b>	<b>106</b>
<b>Literaturverzeichnis</b>	<b>113</b>

# Abbildungsverzeichnis

1.1	Stereofoto im Anaglyphenformat . . . . .	2
1.2	Modulares System von ICIDO . . . . .	2
2.1	Anatomie des menschlichen Auges . . . . .	8
2.2	Akkommodation und Konvergenz . . . . .	9
2.3	Querdisparation . . . . .	10
2.4	keine interaxial Separation . . . . .	11
2.5	kleine interaxial Separation . . . . .	11
2.6	große interaxial Separation . . . . .	12
2.7	Zero Parallax Plane auf dem Würfel . . . . .	13
2.8	Zero Parallax Plane auf der Kugel . . . . .	13
2.9	Zero Parallax Plane auf dem Kegel . . . . .	14
2.10	Null Parallaxe . . . . .	15
2.11	Negative Parallaxe . . . . .	15
2.12	Positive Parallaxe . . . . .	16
2.13	Divergente Parallaxe . . . . .	16
2.14	Parallele Kameras . . . . .	17
2.15	Konvergierende Kameras . . . . .	18
2.16	Off-Axis Kameras . . . . .	19
3.1	Parallelblick . . . . .	22
3.2	Kreuzblick . . . . .	23
3.3	Autostereogramm Tiefenbild . . . . .	23
3.4	Autostereogramm . . . . .	24
3.5	Anaglyphenformat Rot-Cyan . . . . .	26
3.6	Anaglyphenformat Gelb-Blau . . . . .	26
3.7	Anaglyphenformat Grün-Magenta . . . . .	27
3.8	Anaglyphenformat Rot-Grün . . . . .	27
3.9	Chromadepth Funktionsweise . . . . .	28
3.10	Chromadepth Beispiel . . . . .	29
3.11	lineare Polarisisation . . . . .	30
3.12	zirkulare Polarisisation . . . . .	31
3.13	3D-Monitor basierend auf zirkularer Polarisisation . . . . .	31

3.14	Interferenzfiltertechnologie . . . . .	32
3.15	KMQ Brille . . . . .	33
3.16	KMQ Beispiel . . . . .	34
3.17	Shutterbrillentechnik . . . . .	36
3.18	Head Mounted Display . . . . .	37
3.19	lentikulares autostereoskopisches Display . . . . .	38
3.20	parallax-barrieren autostereoskopisches Display . . . . .	39
4.1	maximaler Öffnungswinkel . . . . .	43
4.2	optimierte Formel . . . . .	45
5.1	MouseOver Quader X3D . . . . .	50
6.1	autostereoskopisches Display von Magnetic 3D . . . . .	54
6.2	autostereoskopisches Display von Alioscopy . . . . .	54
6.3	Stereoprojektion auf HEyeWall 1.0 . . . . .	55
6.4	Stereoprojektor Cube3D <sup>2</sup> . . . . .	56
6.5	Head Mounted Display VR1280 . . . . .	56
6.6	Head Mounted Display nVisor SX111 . . . . .	57
6.7	Stereoprojektor H5360 . . . . .	58
6.8	3D-Fernseher KDL-60LX905 . . . . .	59
6.9	Spielekonsole 3DS . . . . .	60
6.10	Digitalkamera 3D W3 . . . . .	61
6.11	Stereodisplay Free2C Desktop . . . . .	62
6.12	Stereodisplay C-i 18 . . . . .	63
6.13	modulares System von IMSYS . . . . .	63
6.14	Brillentarget . . . . .	64
6.15	Navigator . . . . .	64
6.16	Brille Color Eyewear . . . . .	65
6.17	Retinal Image Display von Brother . . . . .	66
8.1	Oberfläche MEL Script . . . . .	73
8.2	Übersicht MEL Script Funktionen . . . . .	74
8.3	Übersicht MEL Script Hilfe . . . . .	75
8.4	Optimierte stereoskopische Maya-Szene . . . . .	84
10.1	Oberfläche X3D Konfigurationsmenü . . . . .	91
10.2	Oberfläche Menüfunktionalitäten . . . . .	92
10.3	Oberfläche Webinterface . . . . .	93
10.4	Optimierte stereoskopische X3D-Szene . . . . .	102

# Kapitel 1

## Einleitung

### 1.1 Motivation

Die Existenz zweier Augen bildet die Grundlage des räumlichen Sehens. Dadurch sind wir in der Lage, Objekte in unserem Umfeld dreidimensional wahrzunehmen und in der Tiefe einzuordnen.

Mit Hilfe der Stereoskopie wird versucht, diese natürliche Sichtweise des Menschen virtuell nachzubilden, sodass sich ein räumliches Sehen einstellen kann. Das Interesse an der Stereoskopie existierte bereits in den Anfängen der Fotografie im 19. Jahrhundert.

Seitdem versuchen die Menschen, monoskopisch-fotografische Abbildungen auch stereoskopisch herzustellen, was später im Zeitalter des Kinos zu einem regelrechten Boom der Stereoskopie führte. Auch im Bereich der Fotografie entstand das Gebiet der Stereofotografie, bei der oftmals zwei Kameras in paralleler Anordnung fotografierten. Zur Darstellung der Fotos wurden in Zeiten der analogen Fotografie Diaprojektoren benutzt. Dass dieser Boom der Stereoskopie nach und nach abflaute, lag lediglich daran, dass die damaligen Anzeigeformate in keinster Weise ausgereift waren und somit hatten Betrachter oftmals mit den Schwierigkeiten einer stereoskopischen Darstellung zu kämpfen.



**Abbildung 1.1:** Um 1906 entstandene Stereofotografie im bekannten Anaglyphenformat. Zur Betrachtung bitte die Rot-Cyan Brille aufsetzen. ©Los Hawlos

Heutzutage ist diese Stereoskopie allgegenwärtig und wird in vielen Bereichen eingesetzt. Sei es in der Forschung bei der Entwicklung von Autos oder in der Medizin für bessere Darstellbarkeit bereits modellierter Objekte.



**Abbildung 1.2:** Präsentation einer stereoskopischen Szene auf einem System von ICIDO. Inklusive Brillen und Navigator. ©ICIDO

In der Filmindustrie erreichte die stereoskopische Darstellung in den letzten Jahren eine zweite Renaissance, wodurch ein Bedarf für viele andere Unternehmen entstanden ist, eigene stereoskopische Inhalte zu erzeugen oder bereits vorhandene stereoskopische Inhalte optimiert präsentieren zu können.

Neben der Erstellung virtueller, stereoskopischer Inhalte für die spätere Betrachtung im 3D-Kino oder heutzutage schon am 3D-Bildschirm zu Hause, werden immer öfter Echtzeit-Rendering-Systeme für eine stereoskopische Visualisierung bereits modellierter Welten genutzt, um die vorhandenen dreidimensionalen Inhalte plastisch darstellen zu können. Dabei fehlt es oftmals an Optimierungen für diese Darstellungen, sodass sich ein angenehmes räumliches Sehen einstellen kann. Um dieses natürliche, räumliche Sehen simulieren zu können, ist es notwendig, alle stereoskopischen Parameter auf die Projektionsgröße und Betrachterabstand anzupassen.

Echtzeit-Rendering-Systeme, welche diese Eigenschaften besitzen, sind zwar bereits vorhanden, allerdings auch sehr stark abgekapselt, sodass man z.B. daran gebunden ist, vorgegebene Modellformate zu benutzen. Da diese Systeme oftmals sehr teuer sind, sind sie für die meisten Unternehmen wirtschaftlich nicht rentabel.

## 1.2 Problemstellung und Zielsetzung

Für die optimierte Darstellung stereoskopischer Inhalte bzw. die Erstellung optimierter stereoskopischer Darstellungen existieren mehrere Voraussetzungen.

Sollen stereoskopische Inhalte dargestellt werden, so ist ein fundiertes Wissen aus dem Bereich der Fotografie eine Grundvoraussetzung. Ohne dieses Wissen ist es sehr schwer, in die Materie der Stereoskopie einzutauchen, da hier die Gesetze der Optik aus der Fotografie erweitert und angewandt werden. Neben dieser Grundvoraussetzung spielt Mathematik eine weitere große Rolle in der Stereoskopie. Der Umgang mit den stereoskopischen Parametern setzt voraus, dass ein Verständnis für mathematische Beziehungen vorhanden ist, da jeder einzelne Parameter einen Einfluss auf die stereoskopische Wirkung hat und nicht stets logisch nachvollziehbar geändert werden kann. Eine spielerische Herangehensweise an das Ändern dieser Parameter erweist sich ebenfalls schnell als nicht zufriedenstellend, weil die Auswirkung der Parameter auf die Darstellung sehr schnell sehr groß wird und dadurch ein Ausprobieren nahezu unmöglich wird.

Wenn nun ohne diese Voraussetzungen daran gearbeitet wird, stereoskopische Inhalte zu erzeugen, so sind die ersten Ergebnisse meist enttäuschend und stören die eigentliche Sicht so sehr, dass die Umsetzung einer stereoskopischen Szene nicht stattfindet. Soll die stereoskopische Darstellung zudem noch so optimiert werden, dass sie möglichst barrierefrei eingesetzt werden kann, fehlt oftmals das nötige Wissen, um dies zu bewerkstelligen. Hierbei kommt es darauf an, dass alle vorhandenen stereoskopischen Parameter auf die aktuelle Projektion der Darstellung so angepasst werden, dass ohne Schwierigkeiten eine stereoskopische Sicht der Darstellung geschehen kann.

Diese Bachelorarbeit hat zum einen das Ziel, dem Anwender im Bereich der Erstellung virtueller Inhalte eine möglichst einfache und intuitive Möglichkeit zu geben, eine bereits auf

die Projektion optimierte stereoskopische Darstellung mit gegebenen Inhalten anzubieten. Diese Optimierungen können mit dem Modellierungswerkzeug Maya vom Entwickler Autodesk genutzt werden. Dazu lädt der Anwender lediglich ein Script für die Optimierung der stereoskopischen Parameter und kann somit seine vorhandenen Inhalte direkt stereoskopisch umwandeln, sodass keinerlei Inhalte geändert werden müssen.

Im Bereich der Echtzeit-Rendering-Systeme hat diese Bachelorarbeit das Ziel, Anwendern die Möglichkeit zu geben, bereits modellierte Welten mit dem vom Fraunhofer IGD entwickeltem System InstantPlayer stereoskopisch anzeigen zu lassen, sodass diese stereoskopische Darstellung optimiert auf die Projektionsparameter geschieht. Da dieses Echtzeit-Rendering-System kostenlos ist und Modelle hierbei auf den Standards X3D und VRML basieren, hat der Anwender die Möglichkeit, aus seinem gewählten Modellierungswerkzeug komplette Szenen direkt mit InstantPlayer anzeigen zu lassen. Weiterhin besteht die Möglichkeit, dieses offene Echtzeit-Rendering-System mit verschiedensten Programmiersprachen selbst zu erweitern.

### 1.3 Organisation der Arbeit

Diese Bachelorarbeit beginnt mit der Motivation und der Zielsetzung. Darin soll verdeutlicht werden, warum es zur Zeit an stereoskopischen Optimierungen mangelt und dass der Bedarf einer solchen technischen Umsetzung notwendig ist. Das 2. Kapitel befasst sich mit den Grundlagen des räumlichen Sehens und geht dabei auf verschiedene Probleme ein, welche bei stereoskopischen Projektionen entstehen können. Im Anhang dieses Kapitels befindet sich noch eine kurze Einführung in die Stereoskopie der virtuellen Welten und deren besonderen Gegebenheiten gegenüber der realen Stereoskopie.

Daraufhin folgen die Erläuterungen zu den gängigsten, heutzutage vorhandenen Möglichkeiten, stereoskopische Darstellungen zu präsentieren. Hier wird auch auf die verschiedensten Funktionen und Probleme der jeweiligen Systeme eingegangen. Für eine korrekte Optimierung in der Stereoskopie sind mathematische Grundlagen notwendig. Deshalb wird in Kapitel 4 das notwendige Grundwissen vermittelt, aufgezeigt welche bisherigen mathematischen Optimierungsmöglichkeiten es gibt und warum diese an verschiedensten Problemen scheitern. Kapitel 5 beinhaltet dann die Grundlagen der Programmierung. Dort werden verschiedene Programmiersprachen vorgestellt und anhand von Beispielen demonstriert. Danach ist der Leser in der Lage, die nachfolgenden Kapitel zu verstehen und kann die in der Bachelorarbeit entwickelten Konzepte nachvollziehen. Der Stand der Technik wird in Kapitel 6 dargestellt. Hier wird auf aktuelle stereoskopische Systeme eingegangen und aufgezeigt, inwiefern diese zur Zeit Optimierungsmöglichkeiten bieten und ob diese für den Markt relevant sind.

Danach folgt der praktische Teil dieser Arbeit. Zunächst wird hierbei in Kapitel 7 ein Konzept für die stereoskopische Vereinfachung in Autodesk Maya präsentiert. Dort werden auch Anforderungen an die Oberfläche und die Funktionalitäten des Scripts gestellt. Kapitel 8 deckt die Implementierung dieses Konzepts ab. Dabei wird auch aufgezeigt, ob das vorgestellte Konzept korrekt umgesetzt werden konnte und ob eine Optimierung stereoskopischer Darstellungen erreicht werden konnte.

Der zweite Teil der praktischen Arbeit wird ab Kapitel 9 bearbeitet. Auch hier wird zunächst ein Konzept vorgestellt, welches umgesetzt werden soll. Kapitel 10 zeigt daraufhin die Im-

plementierung dieses Konzeptes und geht dabei auf vereinzelte Funktionalitäten an Hand des Quellcodes ein.

Das letzte Kapitel bildet den Schluss dieser Arbeit und befasst sich mit der Zusammenfassung der Ergebnisse der Arbeit und bietet einen Ausblick auf eventuelle Erweiterungen der in dieser Arbeit entwickelten Konzepte.

### 1.4 Zusammenfassung der wichtigsten Ergebnisse

Diese Bachelorarbeit zeigt, dass eine Entwicklung eines Konzeptes für die Vereinfachung und Automatisierung der Generierung virtueller, stereoskopischer Inhalte an verschiedenen Stellen deutliche Herausforderungen mit sich bringt. Es wird ersichtlich, dass zur Zeit kein verfügbares Konzept für diese Vereinfachung und Automatisierung existiert. Dafür wird im Rahmen dieser Bachelorarbeit eine eigene mathematische Herangehensweise vorgestellt. Erst dadurch wird es möglich, für die zwei verwendeten Umgebungen Autodesk Maya und InstantPlayer eine Vereinfachung und Automatisierung mittels der Entwicklung eines Scripts und einer Oberfläche zu entwickeln.

Vorteil hierbei ist, dass sowohl Autodesk Maya als auch der InstantPlayer bereits fundierte stereoskopische Eigenschaften bieten, so dass hier eine Optimierung der stereoskopischen Parameter ermöglicht werden kann.

Eine Implementierung beider vorgestellter Konzepte ermöglicht daraufhin eine korrekte und einfache Optimierungsmöglichkeit für stereoskopische Darstellungen. Diese Optimierungen können dabei auf verschiedene Darstellungsverfahren angewandt werden und besitzen zusätzliche Funktionalitäten. Im Bereich des Modellierungswerkzeuges sollen diese zusätzliche Möglichkeiten bieten, in bereits vorhandene Szenen stereoskopische Kameras zu integrieren. Dadurch wird ein weiteres Benutzen der vorhandenen Szenen ermöglicht. Auch die Implementierung einer Möglichkeit zum Rendern für ein autostereoskopisches Display wird hier stattfinden. Bei der Implementierung für das Echtzeit-Rendering-System InstantPlayer wird ein Menü und ein externes Webinterface angefertigt. Mit Hilfe dessen und einer bereitgestellten X3D-Szene soll dem Benutzer ein Einstieg für die stereoskopische Automatisierung im InstantPlayer geboten werden. Die X3D-Szene dient dabei zur ersten Visualisierung der Funktionalitäten des Menüs. Die zusätzliche Implementierung der Steuerung dieser Szene mit einem USB-Eingabegerät unterstützt dabei die intuitive Benutzung und braucht somit keine weiteren Eingabegeräte.

Die in dieser Arbeit vorgestellten Optimierungen werden von Testpersonen getestet und bewertet sodass eine Korrektheit dieser Optimierungen subjektiv bestätigt werden kann. Objektiv wird dies zudem durch eine Messreihe nachgewiesen, die im Laufe der Implementierungen stattfinden wird. Die Objektivität wird dadurch gezeigt, dass beim Messen die gleichen horizontalen Trennungen wie bei der später gezeigten Formel erreicht werden.

Mit den erreichten Zielen der Konzepte und Implementierungen gelingt es, eine kostenlose und frei verfügbare Optimierung stereoskopischer Darstellungen zu entwickeln. Diese kann dann sowohl in Autodesk Maya als auch im InstantPlayer des Fraunhofer IGDs verwendet werden.



# Kapitel 2

## Grundlagen

In diesem Kapitel werden die Grundlagen des räumlichen Sehens erklärt und mit den Grundlagen der Stereoskopie verglichen. Weiterhin finden sich hier Erklärungen zu den meisten heutzutage vorkommenden Darstellungsverfahren in der Stereoskopie, sowie der Einstieg in die mathematischen Grundlagen der Optimierungsmöglichkeiten stereoskopischer Visualisierungen und die Programmierung der für die Bachelorarbeit relevanten Programmiersprachen. Für alle dargestellten Beispiele liegen dieser Bachelorarbeit verschiedene Brillen bei. Je nach Symbol sollte die entsprechende Brille zur korrekten Betrachtungsweise der Beispielbilder aufgesetzt werden.

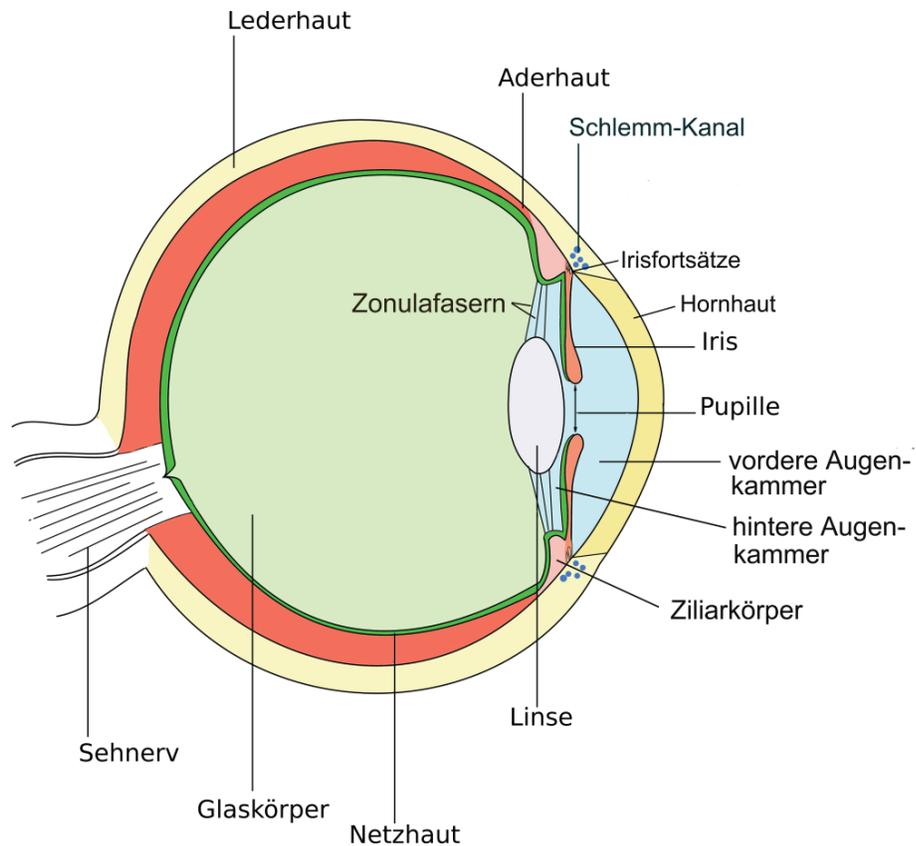
### 2.1 Räumliches Sehen

Die Möglichkeit des räumlichen Sehens besitzt der Mensch dank binokularer Sicht. Dieses Kapitel soll erklären, wie aus dieser binokularen Sicht ein räumliches Sehen entsteht und welche Fähigkeiten des menschlichen Auges dafür notwendig sind.

#### 2.1.1 Akkommodation

Wird ein Objekt in einer Landschaft betrachtet, so ist das Auge in der Lage, alle betrachteten Objekte scharf zu sehen. Seien es Blumen in der Nähe oder andere, sich in der Ferne befindlichen Strommasten und Bäume. Diese Fähigkeit nennt sich Akkommodation[Kuh99][Fri07]. Wird versucht, ein Objekt in der Ferne scharf zu sehen, so ziehen die Zonulafasern [Abbildung 2.1] die elastische Linse des Auges so auseinander, dass diese abgeflacht wird. Dadurch verringert sich die Brechkraft und vergrößert sich der Durchmesser der Linse. Die sich zwischen der Linse und den Zonulafasern befindlichen Ziliarkörper [Abbildung 2.1] sind in diesem Zustand entspannt[Fri07]. Soll hingegen ein Objekt in der Nähe betrachtet werden, so ziehen sich die Ziliarkörper zusammen, verringern damit den Zug auf die Zonulafasern und die elastische Linse verformt sich zu einer ovalen Kugel, wodurch die Brechkraft dieser Linse wiederum steigt. Die Verformung der Linse bei unterschiedlichen Betrachtungsabständen

wird in Abbildung 2.2 ersichtlich. Diese Fähigkeit wurde bereits 1952 von Hermann von Helmholtz entdeckt[vH52].



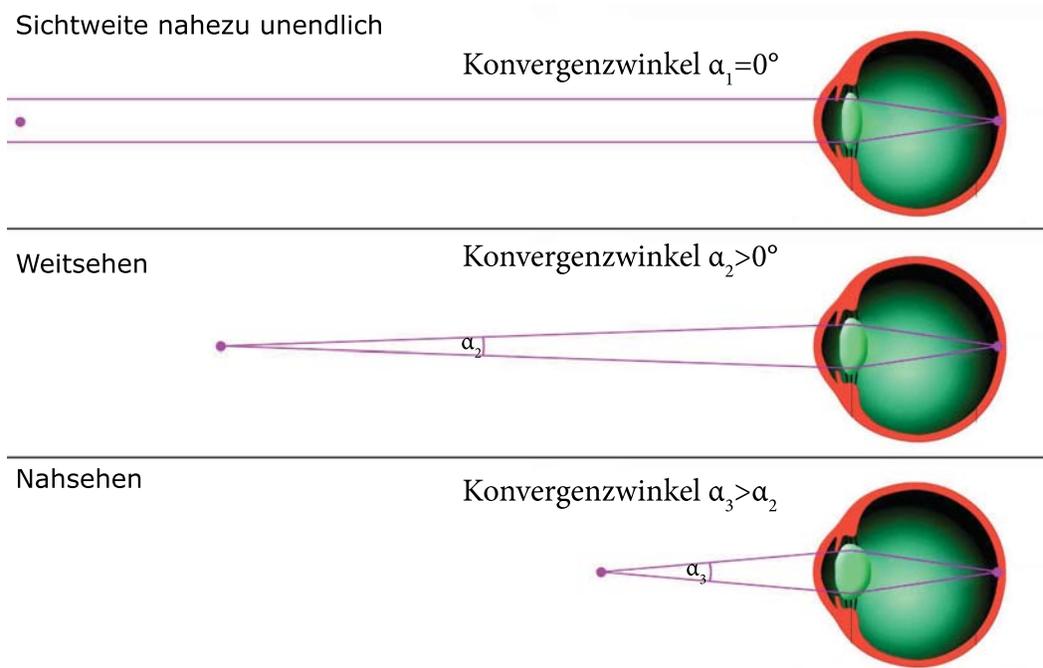
**Abbildung 2.1:** Die Anatomie des menschlichen Auges<sup>1</sup>.

### 2.1.2 Konvergenz

Verändert sich der Betrachtungsabstand zu einem Objekt, so ändert sich auch die Blickrichtung, welche jedes Auge einnehmen muss. Während bei einer Betrachtung sehr weit entfernter Objekte die Augen nahezu parallel ausgerichtet sind, führt ein Betrachten näher liegender Objekte zum Konvergieren der Augen. Diese Fähigkeit des menschlichen Auges nennt sich deshalb Konvergenz[Kuh99][Fri07]. Der Winkel, den beide Sehachsen am betrachteten Objekt dabei einschließen nennt sich Konvergenzwinkel, nachfolgend in Abbildung 2.2 zu sehen.

---

<sup>1</sup>Online verfügbar unter <http://de.academic.ru/dic.nsf/dewiki/280519>. Besucht am 13.07.2010.

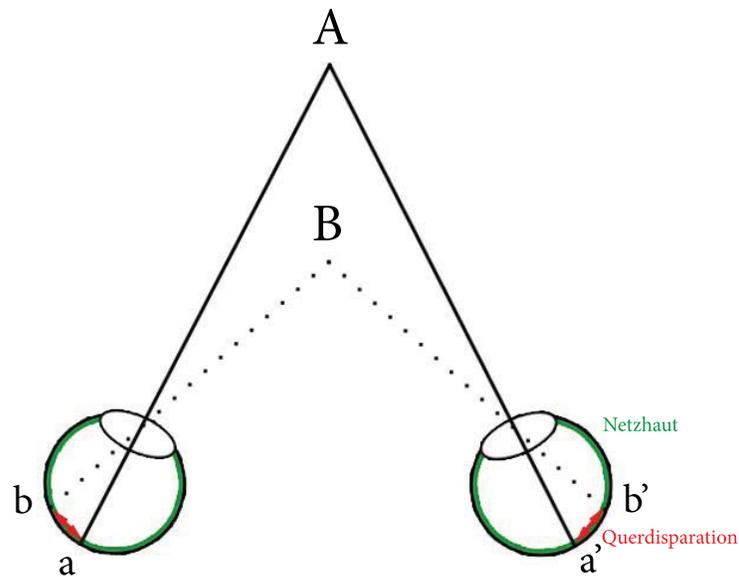


**Abbildung 2.2:** Demonstration von Akkommodation und Konvergenz<sup>2</sup>.

### 2.1.3 Querdisparation

Den größten Einfluss auf die Tiefenwahrnehmung hat die so genannte Querdisparation [Kuh99] [Fri07]. Dabei handelt es sich um die leicht versetzte Bildlage von Abbildungen auf der Netzhaut beider Augen, welche in Abbildung 2.3 zu sehen ist. Diese versetzten Abbildungen entstehen durch den Augenabstand, welcher im Durchschnitt 6,3cm beträgt. Durch diesen Abstand nimmt jedes Auge eine unterschiedliche Perspektive eines betrachteten Objektes wahr, wodurch das Gehirn eine Tiefenwahrnehmung erzeugt. Damit eine korrekte Tiefenwahrnehmung möglich ist, darf diese Querdisparation nur in Grenzen variieren. Wird diese Grenze überschritten, so entstehen, durch die verhinderte Fusionierbarkeit des betrachteten Objekts, Doppelbilder.

<sup>2</sup>Abbildung nach Vorlage von Bertelsmann BKK. Online verfügbar unter <http://www.bertelsmann-bkk.de/meine-gesundheit/lexika/gesundheitslexikon/a.html>. Besucht am 13.07.2010.



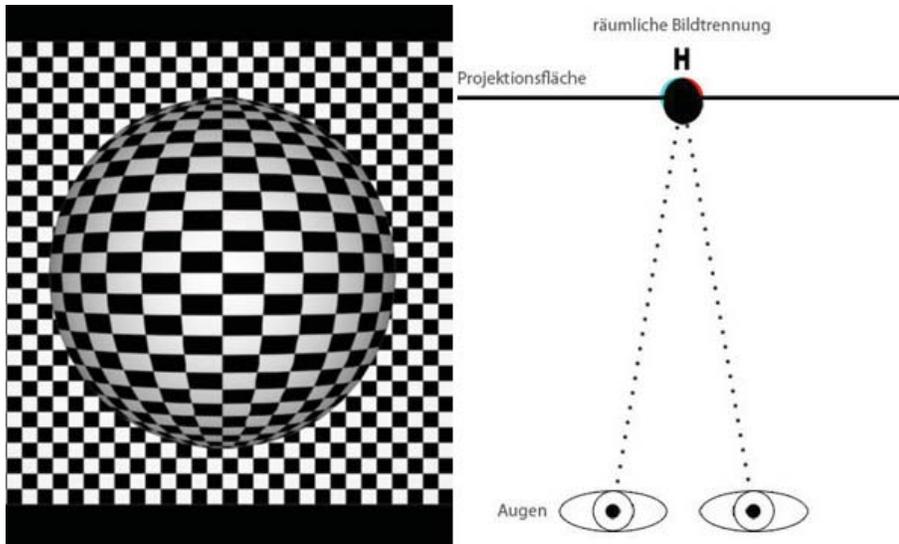
**Abbildung 2.3:** Darstellung der Querdisparation im menschlichen Auge.

## 2.2 Stereoskopische Begriffe in der virtuellen Realität

In diesem Kapitel werden die Beziehungen zwischen den Begriffen des räumlichen Sehens und der Stereoskopie erklärt. Es wird auch aufgezeigt, welche Schwierigkeiten die Abbildung des räumlichen Sehens auf die virtuelle Realität mit sich bringt und wie diese vermieden werden können.

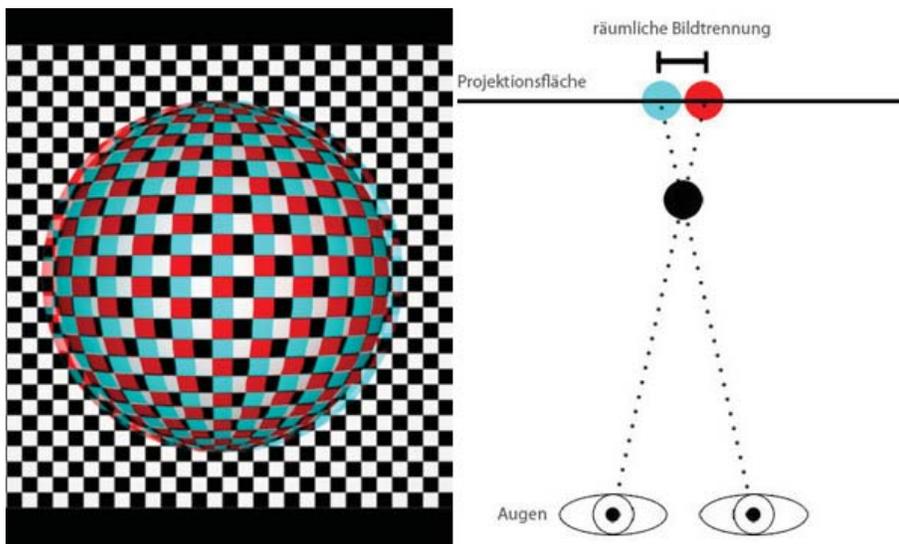
### 2.2.1 Interaxial Separation

Damit auch in der Stereoskopie unterschiedliche Bildlagen von Abbildungen auf der Netzhaut entstehen, werden Bilder für das linke und rechte Auge horizontal so verschoben, dass die getäuschte Wahrnehmung des Menschen einen Tiefeneindruck erzeugt. Der horizontale Versatz wird in der Stereoskopie interaxial Separation genannt [Aut08]. Da auch wie bei der Querdisparation des Auges Grenzen für den Versatz von Bildern gelten, wird hier die Grenze des Augenabstands gesetzt. Ist der Versatz beider Bilder größer als der Augenabstand, richten sich die Augen entweder nach außen oder man wird zum starken Schielen gezwungen, was eine enorme Anstrengung der Muskeln erfordert.



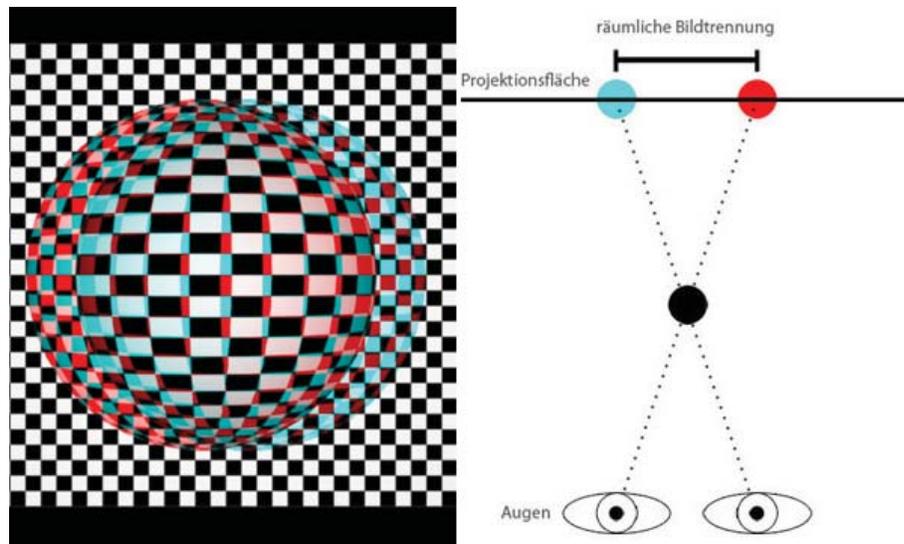
**Abbildung 2.4:** Eine nicht vorhandene interaxiale Separation führt dazu, dass Objekte auf der Tiefe der Projektionsfläche erscheinen.

Bei diesem ersten Bild wurde die interaxiale Separation auf 0 gesetzt. Dadurch erscheint das angezeigte Objekt auf der selben Tiefe wie die dahinter liegende Ebene zu liegen.



**Abbildung 2.5:** Eine kleine interaxiale Separation führt zur Hervorhebung des Objektes gegenüber dem Hintergrund.

Die interaxiale Separation dieser Grafik ist größer als 0, allerdings noch gering gehalten. Dadurch scheint die Kugel vor der Ebene zu schweben.

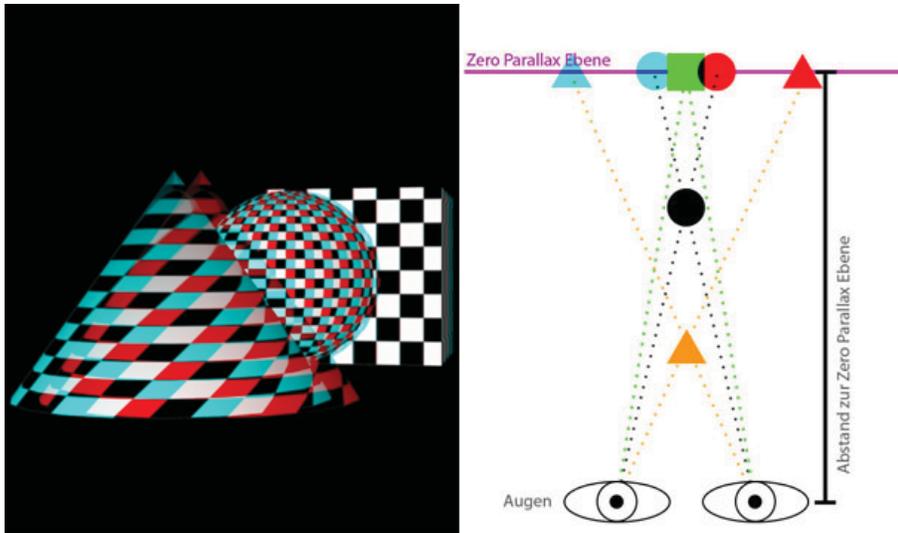


**Abbildung 2.6:** Eine verstärkte interaxial Separation verstärkt auch den Hervorhebungseffekt.

Diese Grafik besitzt eine große interaxial Separation. Dadurch verändert sich nicht nur die Stärke der Hervorhebung, sondern der Betrachter bekommt dabei das Gefühl, dass das gezeigte Objekt zusätzlich näher an ihm ist als bei der vorherigen Grafik.

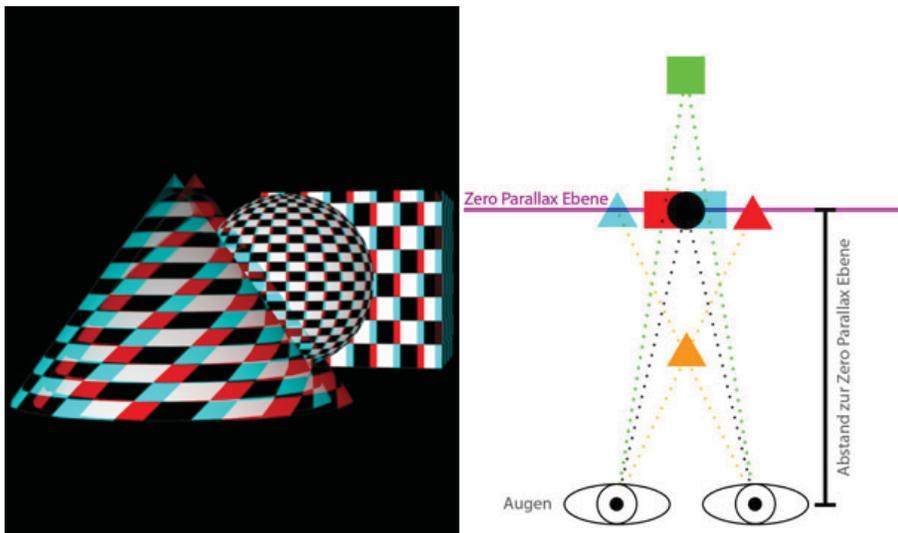
### 2.2.2 Zero Parallax Plane

Eine Nachahmung der Akkommodation existiert innerhalb der Stereoskopie nicht. Denn im Gegensatz zum räumlichen Sehen befindet sich bei einer stereoskopischen Projektion das betrachtete Objekt immer im selben Abstand zum Betrachter. Allerdings wird durch das Setzen einer Ebene, der so genannten Zero Parallax Plane [Aut08], versucht, den Blick des Betrachters auf einen bestimmten virtuellen Abstand zu lenken. Definiert wird diese Zero Parallax Plane durch die Stelle im Raum, an der keinerlei horizontaler Versatz beider Bilder vorhanden ist. Somit fusioniert der Betrachter ein Objekt dieser Ebene, da die Abbildung auf die Netzhaut beider Augen an der gleichen Stelle projiziert wird. Sowohl vor als auch hinter der Zero Parallax Plane steigt der horizontale Versatz beider Bilder wieder an, wodurch letztendlich der getäuschten visuellen Wahrnehmung des Menschen eine Tiefe vorgespielt wird.



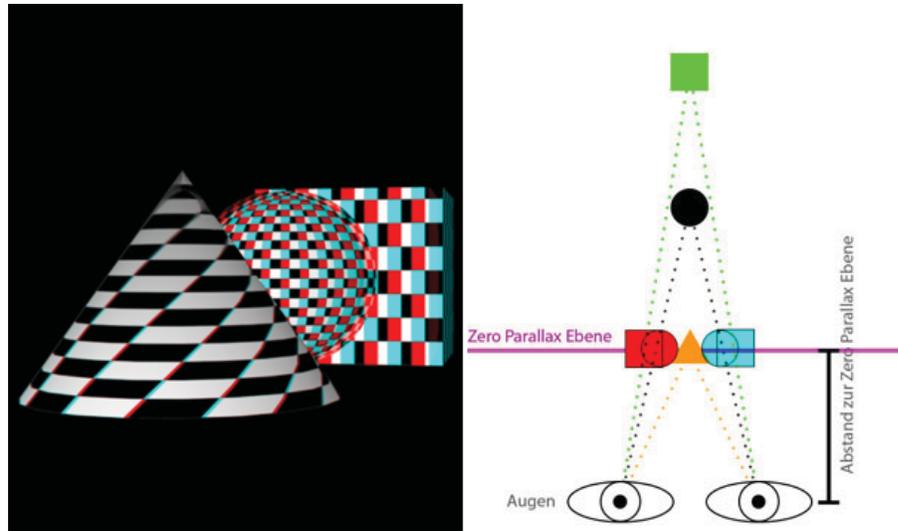
**Abbildung 2.7:** Die Zero Parallax Plane liegt auf dem hintersten Objekt, somit liegen alle anderen Objekte scheinbar davor.

Hier wurde die Zero Parallax Plane auf den hintersten Würfel gesetzt. Dadurch scheinen alle Objekte tatsächlich vor diesem Würfel zu sein.



**Abbildung 2.8:** Die Zero Parallax Plane liegt auf der Kugel, der Kegel scheint davor und der Würfel scheint dahinter zu liegen.

Die Zero Parallax Plane befindet sich hier auf dem mittleren Objekt. Der Kegel davor scheint auch tatsächlich davor zu schweben, während der Würfel dahinter zu sein scheint.



**Abbildung 2.9:** Die Zero Parallax Plane liegt auf dem Kegel, alle Objekte scheinen deshalb dahinter zu liegen.

Die Zero Parallax Plane befindet sich hier auf dem vordersten Objekt. Dadurch scheinen alle anderen Objekte hinter diesem Objekte zu liegen.

### 2.2.3 Parallaxenwinkel

Bei einer stereoskopischen Projektion wird die Fähigkeit der Konvergenz durch den so genannten Parallaxenwinkel realisiert. Die Veränderung dieses Winkels geht mit der unterschiedlichen Lage von Objekten einher. Das bedeutet, dass in der Praxis das Ändern dieses Winkels durch das Ändern der interaxial Separation geschieht. Durch die Verschiebung der Bilder sowohl für das linke als auch das rechte Auge müssen beide Augen auf die gezeigten Bilder konvergieren, um deren virtuelle Tiefe einordnen zu können. Damit ein horizontaler Versatz größer als der Augenabstand vermieden wird, sollte der resultierende Parallaxenwinkel nicht größer als  $1,5^\circ$  sein, da sonst eine angenehme Betrachtungsweise der stereoskopischen Projektion nicht gewährleistet werden kann[Leh05].

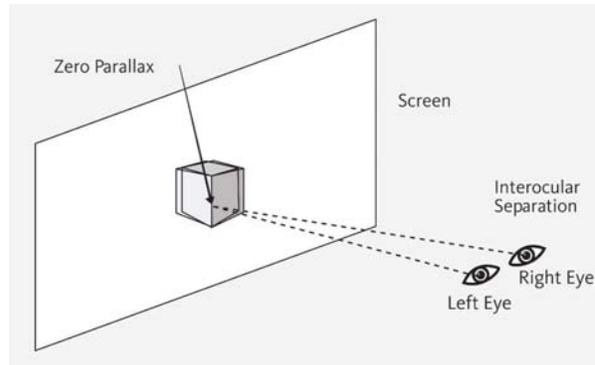
### 2.2.4 Parallaxentypen

Aus den verschiedenen horizontalen Verschiebungen, welche die unterschiedlichen Objekte innerhalb einer stereoskopischen Visualisierung besitzen, resultieren auch unterschiedliche räumliche Eindrücke dieser. In der Stereoskopie nennt man dies Parallaxentypen. Nachfolgend werden alle Typen von Parallaxen aufgelistet und anhand von Bildern dargestellt.

#### 2.2.4.1 Null Parallaxe

Von Null Parallaxe[Aut08], auch Zero Parallax, wird gesprochen, wenn das Objekt keinerlei interaxial Separation auf der Projektionsebene besitzt. Damit erscheint das Objekt im glei-

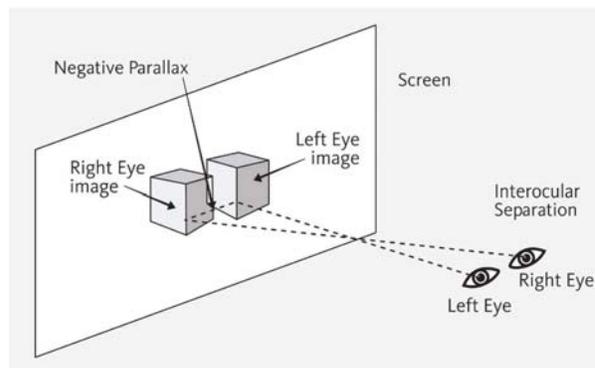
chen Abstand der Projektion zum Betrachter und wird deshalb auch keiner Tiefe oder Nähe zugeordnet.



**Abbildung 2.10:** Objekte mit keinerlei Trennung erscheinen auf der selben Tiefe wie die Projektionsebene. ©Autodesk

#### 2.2.4.2 Negative Parallaxe

Die negative Parallaxe [Aut08] beschreibt alle Objekte, die sich scheinbar vor der Projektionsebene befinden. Dabei besitzen diese Objekte eine interaxial Separation größer als 0cm, jedoch maximal dem Augenabstand entsprechend und sind so getrennt, dass die Augen zum Schielen neigen, wodurch der Konvergenzwinkel größer ist, als Objekte die eine Null Parallaxe besitzen (bei gleicher stereoskopischer Projektion mit selben Betrachterabstand und Projektionsgröße).

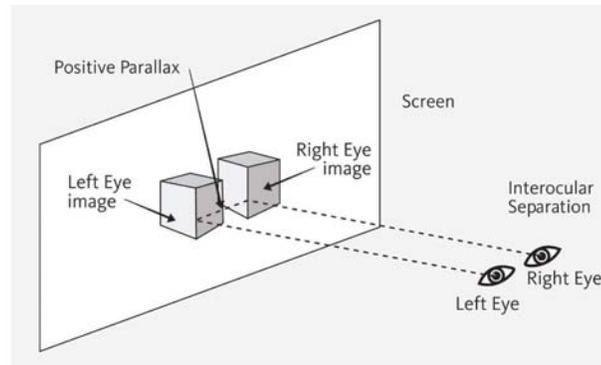


**Abbildung 2.11:** Objekte mit einer Trennung größer als 0 erscheinen bei korrekter Anordnung im Vordergrund der Projektionsebene. ©Autodesk

#### 2.2.4.3 Positive Parallaxe

Eine positive Parallaxe [Aut08] besitzen alle Objekte, die hinter der Projektionsebene zu liegen scheinen. Dabei ist auch hier die interaxial Separation größer als 0cm, maximal dem

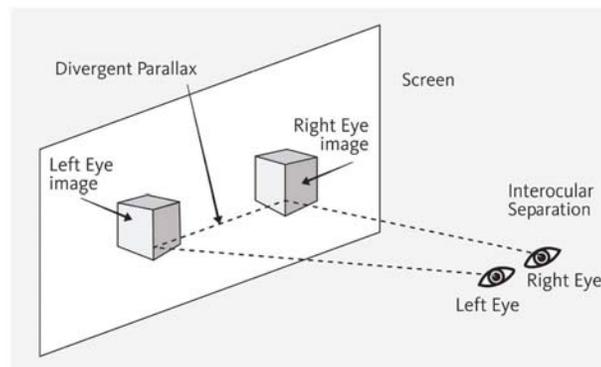
Augenabstand entsprechend, allerdings so getrennt, dass die Augen zum parallel schauen neigen, wodurch der Konvergenzwinkel gegen  $0^\circ$  geht. Objekte mit einer interaxial Separation die dem Augenabstand entspricht, liegen somit im Unendlichen.



**Abbildung 2.12:** Objekte mit einer Trennung größer als 0 erscheinen bei korrekter Anordnung im Hintergrund der Projektionsebene. ©Autodesk

### 2.2.4.4 Divergente Parallaxe

Dieser Typ Parallaxe sollte möglichst vermieden werden, da hierbei die Augen dazu neigen, nach außen zu drehen, was dem natürlichen Sehen nicht entspricht. Die divergente Parallaxe [Aut08] tritt dann auf, wenn die interaxial Separation größer als dem entsprechenden Augenabstand wird.



**Abbildung 2.13:** Objekte mit einer Trennung größer als dem Augenabstand führen dazu, dass die Augen sich nach außen drehen müssen. ©Autodesk

### 2.2.5 Besonderheiten stereoskopischer Kameras in der virtuellen Realität

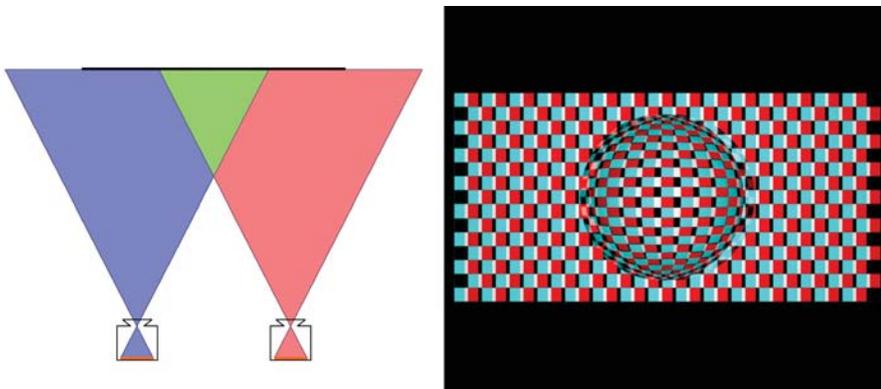
Um für jedes Auge eine eigene Perspektive zu gewährleisten, gibt es verschiedene Möglichkeiten, beide Kameras anzuordnen. Zusätzlich dazu gibt es in der virtuellen Realität die

Möglichkeit, die beiden Kameras so anzuordnen, dass keinerlei perspektivische Verzerrungen entstehen, wodurch eine exakte Nachbildung der menschlichen Sichtweise ermöglicht wird. In diesem Unterkapitel werden diese verschiedenen Kameraanordnungen erklärt, mit der Stereoskopie in der realen Welt verglichen und aufgezeigt, wo eventuelle Probleme entstehen.

### 2.2.5.1 Parallele Kameras

Eine parallele Anordnung beider Kameras führt dazu, dass der Konvergenzwinkel  $0^\circ$  ist [Men09] [Aut], was einem Blick in die weite Ferne, z.B. Himmel, entspricht. Daraus resultiert allerdings auch, dass alle Objekte, welche vor dieser unendlichen Ferne liegen, eine negative Parallaxe besitzen und somit im Vordergrund erscheinen. Dementsprechend bietet sich diese Kamerastellung in der virtuellen Realität nur an, wenn eine Landschaftsszene mit sehr weit entfernten Objekten visualisiert werden soll.

Diese Kameraanordnung wird in der stereoskopischen Abbildung der realen Welt, z.B. in der Stereofotografie, am häufigsten verwendet, da hierbei keine perspektivischen Verzerrungen entstehen und durch ein simples Verschieben beider Bilder der Parallaxenpunkt bestimmt werden kann. Dadurch entsteht ein realistischer stereoskopischer Effekt, bei dem allerdings Bildinformationen sowohl an der linken als auch rechten Seite verloren gehen.

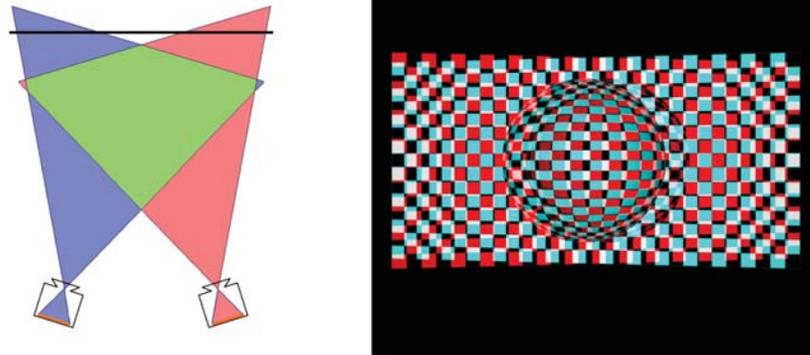


**Abbildung 2.14:** Parallele Kameras besitzen keine Zero Parallax Plane. Deshalb befinden sich alle Objekte scheinbar im Vordergrund.

### 2.2.5.2 Konvergierende Kameras

Die konvergierende Anordnung zweier Kameras wird benutzt, um zusätzlich zum räumlichen Eindruck den Parallaxenpunkt direkt bestimmen zu können. Dabei werden beide Kameras zueinander gedreht, während sie eine parallaktische Verschiebung besitzen [Men09] [Aut]. Dadurch fokussieren beide Kameras ein und denselben Abstand, was dem Fusionieren des menschlichen Auges entspricht. Somit ist im Gegensatz zur parallelen Anordnung der Kameras schon beim Schritt der Aufnahme bzw. Erstellung einer Szene klar, auf welchen Punkt die Zero Parallax Plane gesetzt ist. Der Nachteil dieser Anordnung ist allerdings, dass durch das Eindrehen beider Kameras perspektivische Verzerrungen entstehen, die bei einem zu

großen Konvergenzwinkel zum Bildzerfall führen und nicht korrigierbar sind. Deshalb ist diese Anordnung beider Kameras sowohl in der virtuellen als auch in der realen Welt nur dann bedingt nutzbar, wenn der Konvergenzwinkel sehr klein gehalten wird.

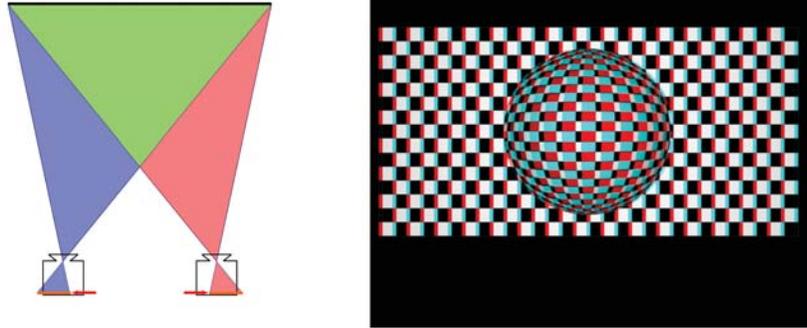


**Abbildung 2.15:** Konvergierende Kameras werden zueinander geneigt. Dadurch entstehen geometrische Verzerrungen im Bild.

### 2.2.5.3 Off-Axis Kamera

Die Besonderheit bei der Kamerastellung Off-Axis ist, dass beide Kameras parallel angeordnet sind, wobei die Sensoren nach außen verschoben werden [Men09] [Aut]. Dadurch wird es möglich, die Schwachstellen der parallelen und konvergierenden Kamerastellungen zu eliminieren. Durch den Versatz beider Sensoren entstehen zwei verschiedene Perspektiven mit asymmetrischen Abbildungen des selben Raums. Dies hat zur Folge, dass beide Kameras die selben Begrenzungen in ihrer Abbildung besitzen, d.h. dass keine perspektivische Verzerrungen auftreten und zusätzlich die Zero Parallax Plane festgelegt werden kann. In der virtuellen Realität bietet sich deshalb diese Anordnung beider Kameras für alle Gegebenheiten an und wird von dem für diese Bachelorarbeit verwendeten Modellierungswerkzeug Autodesk Maya sowie auch von dem Echtzeit-Rendering-System InstantPlayer unterstützt.

Bei der Stereoskopie in der realen Welt existiert solch eine Möglichkeit bisher nicht, da das Verschieben beider Sensoren um den selben Abstand, welcher sehr genau eingehalten werden muss, sehr komplex umzusetzen wäre. Eine Realisierung solcher Verschiebungen würde bedeuten, dass die Sensoren frei im Gehäuse der Kamera beweglich, aber auch feststellbar sein müssten.



**Abbildung 2.16:** Bei Off-Axis Kameras werden die Sensoren verschoben, wodurch eine Zero Parallax Plane entsteht und dennoch keine Verzerrungen auftreten.

### 2.2.6 Zusammenfassung

In diesem Kapitel wurden alle grundlegenden physikalischen Zusammenhänge erklärt, warum der Mensch räumliches Sehen besitzt. Weiterhin wurde die Nachbildung des räumlichen Sehens in der Stereoskopie aufgezeigt und erklärt, anhand welcher verschiedener Parameter sowohl Tiefe als auch Vordergrund in einer stereoskopischen Darstellung entstehen können und welche Grenzen bei diesen Parametern eingehalten werden müssen, um stereoskopische Darstellungen für das menschliche Auge als noch angenehm empfindbar zu machen.



## Kapitel 3

# Stereoskopische Anzeigeverfahren

In diesem Kapitel werden verschiedenste stereoskopische Anzeigeverfahren aufgezeigt, die sowohl in der Vergangenheit als auch heutzutage und in naher Zukunft benutzt werden. Dabei unterteilt sich das Kapitel zunächst in jene Anzeigeverfahren, welche keinerlei Hilfsmittel zur Betrachtung benötigen. Daraufhin folgen passive Anzeigeverfahren und im Anschluss dazu dann aktive Anzeigeverfahren. Als letztes werden gesondert autostereoskopische Anzeigen aufgeführt. Für diese werden zwar auch keinerlei Hilfsmittel zur Betrachtung benötigt, allerdings entsteht hierbei ein stereoskopischer Eindruck direkt durch die Anzeige und nicht dadurch, dass der Betrachter in einer besonderen Weise schauen muss.

### 3.1 Freie Betrachtungsmöglichkeiten

#### 3.1.1 Parallelblick

Wie schon aus dem vorherigen Kapitel ersichtlich, stellen sich die Augen nahezu parallel, wenn ein Mensch in die Ferne blickt. Diese Fähigkeit machen sich stereoskopische Abbildungen im parallelen Format zu nutze. Dabei werden zwei verschiedene Perspektiven des selben Bildinhaltes nebeneinander so positioniert, dass die linke Ansicht links und die rechte Ansicht rechts abgebildet wird[Kuh99]. Der Betrachter muss dann zunächst über diese Abbildungen hinweg in die Ferne oder durch die Abbildungen „hindurch“ blicken, behält diesen Blick bei, und schaut dann auf die parallel angeordnete Ansicht. Dabei sollte nach kurzer Zeit zwischen beiden positionierten Ansichten eine dritte Abbildungen erscheinen, welche dann als dreidimensional empfunden wird. Bei korrekter Betrachtungsweise sieht das linke Auge nur die linke Abbildung und das rechte Auge nur die rechte Abbildung. Da das Gehirn eine Fusionierung beider Ansichten vornimmt, können dann ohne Probleme mehrere Objekte auf diesem Bild betrachtet werden und die Fusionierung des dreidimensionalen Bildes bleibt erhalten. Diese Art der stereoskopischen Betrachtung erfordert einiges an Übung und muss mehrfach wiederholt werden, um schnellere und bessere Ergebnisse zu erzielen. Allerdings ist dieses Verhalten des menschlichen Auges vom natürlichen Sehen bereits vorgesehen und ist deshalb von den freien Betrachtungsmöglichkeiten immernoch diejenige, welche von den Augen die geringste Anstrengung benötigt und weniger zu Kopfschmerzen führt.

Wie bei allen anderen stereoskopischen Abbildungen müssen auch hier die Grenzen der Fusionierbarkeit des menschlichen Auges eingehalten werden, da ansonsten kein stereoskopischer Effekt entstehen kann.



**Abbildung 3.1:** Beim Parallelblick muss in die Ferne geschaut werden, um einen stereoskopischen Effekt zu sehen.

Anleitung zum Betrachten: Hier sollte zunächst am besten über das Blatt hinweg in den Raum geschaut werden, so dass eine Fokussierung auf ein weit entferntes Objekt stattfindet. Erst danach mit Beibehaltung dieser Augenstellung auf das Blatt schauen. Dabei sollte dann ein stereoskopischer Effekt entstehen.

#### 3.1.2 Kreuzblick

Bei dieser Art einer stereoskopischen Betrachtungsweise werden beide verschiedenen Perspektiven einer Abbildung so angeordnet, dass sich links die rechte Ansicht und rechts die linke Ansicht befindet [Kuh99]. Das erfordert vom Betrachter, dass dieser für einen stereoskopischen Effekt so stark schielen muss, bis, wie auch beim Parallelblick, eine dritte Abbildung zwischen der linken und rechten Abbildung entsteht, welche auch diesmal die dreidimensionale Ansicht repräsentiert. Dabei ist diese Betrachtung vom menschlichen Auge in der realen Welt nur sehr selten erforderlich und dementsprechend auch anstrengender. Besonders überfordernd ist bei anfänglichen Versuchen, solche Bilder korrekt zu betrachten, dass zusätzlich zum Schielen die dritte, mittlere Abbildung scharf gestellt werden muss, damit sich ein stereoskopischer Effekt einstellen kann. Gelingt dieses Vorhaben, kann man nach einiger Übung sehr schnell alle möglichen Bilder in dieser Ansichtsweise betrachten. Ist ein Mensch in der Lage, stärker zu schielen, als es die üblichen Grenzen stereoskopischer Darstellungen erfordern, so kann dieser auch größere Abbildungen korrekt sehen, obwohl hierbei normalerweise der horizontale Versatz zu groß für die Fusionierung ist.



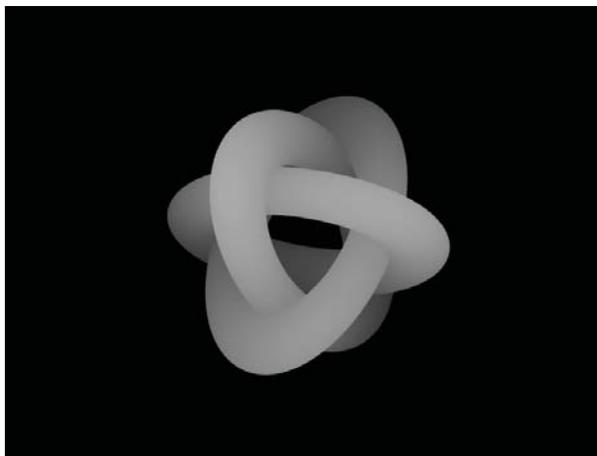
**Abbildung 3.2:** Beim Kreuzblick muss geschickt werden, um einen stereoskopischen Effekt zu erzielen.

Anleitung zum Betrachten: Ein Betrachten des Bildes ist hier sofort möglich. Danach sollte versucht werden zu schielen, so dass ein drittes Bild zwischen den zwei real abgebildeten sichtbar wird. Die Scharfstellung dieses dritten Bildes führt dann zum stereoskopischen Effekt.

#### 3.1.3 Autostereogramme

Eine besondere Form der freien Betrachtungsweise stellen Autostereogramme dar. Dabei handelt es sich um Abbildungen, in denen zunächst keinerlei stereoskopische Bildinformation erahnt werden kann. Üblicherweise wird bei diesen Abbildungen ein Parallelblick angewandt, um aus den generierten Mustern ein dreidimensionales Objekt erkennen zu können. Deshalb ist die Betrachtung solcher Bilder für alle Menschen vorteilhaft, die bereits den Parallelblick geübt haben und gezielt einsetzen können.

Für eine Erstellung eines Autostereogramms ist stets ein Tiefenbild notwendig. Für das gezeigte Beispiel wurde das nachfolgende Tiefenbild verwendet.



**Abbildung 3.3:** Tiefenbild, mit dem das Autostereogramm erstellt wurde. ©BG & BRG Wien

Verrechnet wird letztendlich ein Muster mit dem Tiefenbild, so dass an den geeigneten Stellen Verschiebungen entstehen, welche einen stereoskopischen Effekt erzeugen. Da das Muster sehr oft in der Grafik auftaucht, ist ohne einen Parallelblick hierbei kein dreidimensionales Objekt zunächst erkennbar.



**Abbildung 3.4:** Autostereogramm mit Parallelblick: zu sehen sind 3 zusammenhängende Ringe. ©BG & BRG Wien

## 3.2 passive Anzeigeverfahren

Passive Anzeigeverfahren zeichnen sich dadurch aus, dass die stereoskopische Projektion selbst durch das Ausgabegerät geschieht und der Betrachter zur Wahrnehmung dieser stereoskopischen Bilder eine Brille benutzen muss. Dies bedeutet, dass jedes Ausgabegerät selbst die Trennung beider Ansichten für die Augen übernimmt und entsprechend aufzusetzende Brillen dabei ausschließlich Filter im jeweiligen Farbformat besitzen, wodurch wiederum die Ansicht für jedes Auge zurückgewonnen werden kann.

### 3.2.1 Anaglyphen

Anaglyphen sind die am weitesten verbreitete Methode stereoskopische Bilder zu betrachten. Bei dieser Technik handelt es sich um eine reine Farbfiltertechnik, bei der durch farbige Folien, welche auf der Brille sitzen, jedes Auge möglichst nur die eigene Perspektive des stereoskopischen Bildes wahrnehmen kann [Sch07]. Dementsprechend muss auch das stereoskopische Bild beide Farben dieser Folien enthalten.

Vorteile dieser Technik sind unter anderem, dass diese am meisten eingesetzt wird, um schnell dreidimensionale Bilder sichtbar zu machen. Weiterhin ist diese Methode der Anzeigeverfahren sehr günstig und lässt sich dank ausschließlicher Farbtrennung auch in Zeitschriften abdrucken. Für die Darstellung solcher Abbildungen ist hierbei auch nur ein Ausgabegerät notwendig, während die meisten anderen Verfahren zwei Geräte benötigen, um eine stereoskopische Ansicht darstellen zu können. Heutzutage wird diese Technik immernoch bei stereoskopischen Filmproduktionen eingesetzt um schnelle Zwischenergebnisse sichtbar zu machen.

Ein sehr großer Nachteil dieser Technik ist, dass wegen der Farbfiltertechnik nur bedingt farbige Abbildungen erkennbar sind. Enthält ein Bild bei einem Anaglyphenformat von Rot-Cyan z.B. eine rote Rose, so wird diese nur flimmerartig wahrgenommen, da der Filter der Farbe Cyan die Blüte der Rose komplett schwarz färbt, während der rote Filter diese Blüte der Rose komplett weiß färbt. Bilder in anaglyphem Format haben auch oftmals Schwierigkeiten mit so genannten „Geisterbilder“. Dabei handelt es sich um Darstellungsfehler, bei denen z.B. das linke Auge durch einen Filter Teile des rechten Bildes erkennen kann, so dass diese Teile als schwach erkennbar, und deshalb als „geisterartig“ wahrgenommen werden können.

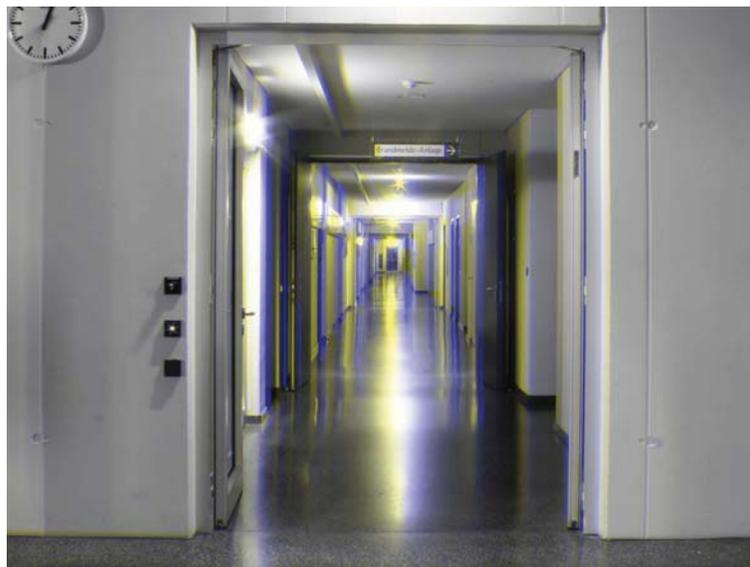
Wegen der nur sehr bedingten Farbwahrnehmung anaglyph gestalteter stereoskopischer Bilder sind oftmals diese Bilder im Schwarz-Weiß-Format. Hierbei kann der Tiefeneindruck sehr gut nachvollzogen werden, da keinerlei Farben im Bild stören können. Zur Optimierung solcher Bilder gibt es seit längerem Farboptimierungsverfahren, bei denen oftmals die Sättigung der Farben geschwächt wird und Farbtöne angepasst werden, sodass jeweilige Filter möglichst gut die jeweils andere Farbe heraus filtern können.

Diese Technik gibt es bereits in sehr vielen Farbvarianten. Während anfangs die Rot-Grün- und Rot-Blau-Anaglyphen am bekanntesten waren, gibt es heutzutage zusätzlich dazu auch Gelb-Blau-, Rot-Cyan- und Grün-Magenta-Anaglyphen. Die Entscheidung, welches dieser Formate letztendlich eingesetzt wird, hängt oftmals von den verwendeten Farben in der Szene ab. Im Gegensatz zu den alten Varianten besitzen alle heutzutage vorkommenden Varianten das Spektrum aller 3 additiven Farben Rot, Grün und Blau. Besonders an der Gelb-Blau-Variante ist, dass das Unternehmen ColorCode vor einigen Jahren die Technik „ColorCode3D“ patentiert hat, welche dafür sorgen soll, dass eine deutlich bessere Farbtreue der Bilder beibehalten wird.

Die nachfolgenden Abbildungen zeigen Bilder in verschiedenen anaglyphen Formaten, deshalb sollte jeweils die entsprechend angegebene Brille aufgesetzt werden.



**Abbildung 3.5:** Zum Betrachten dieses Bildes bitte die Rot-Cyan-Brille aufsetzen.



**Abbildung 3.6:** Zum Betrachten dieses Bildes bitte die Gelb-Blau-Brille aufsetzen.



**Abbildung 3.7:** Zum Betrachten dieses Bildes bitte die Grün-Magenta-Brille aufsetzen.



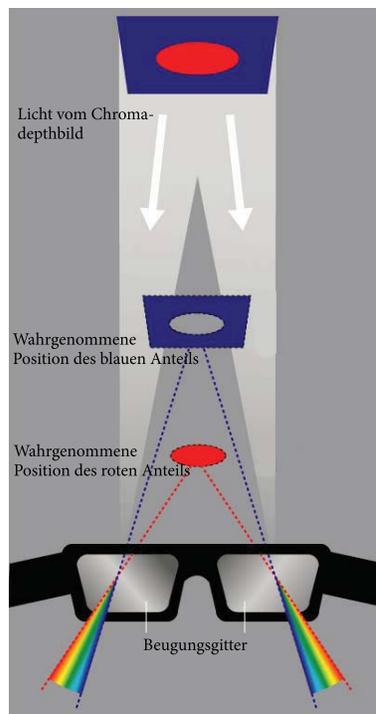
**Abbildung 3.8:** Zum Betrachten dieses Bildes bitte die Rot-Grün-Brille aufsetzen.

Alle abgebildeten Darstellungen sind in einem Schwarz-Weiß-Format, so dass ein möglichst guter stereoskopischer Effekt erzielt werden kann. Auffällig hierbei wird, dass mit der letzten Variante Rot-Grün dabei der beste Effekt entsteht. Der Grund dafür ist, dass diese Variante lediglich zwei der drei verfügbaren Farbkanäle zur Farbtrennung nutzt, wodurch jegliche Geisterbilder unterdrückt werden können.

#### 3.2.2 Chromadepth

Wird sichtbares Licht durch ein Prisma geleitet, spreizt sich das Spektrum dieses Lichts auf und es wird die ganze Bandbreite der im sichtbaren Licht vorhandenen Lichtfarben erkennbar. Hierbei werden längere Wellenlängen (z.B. die Farbe Rot) stärker abgelenkt als kürzere Wellenlängen (z.B. Blau). Diese Tatsache wird bei Chromadepth genutzt, um mit einer Ansicht eines Bildes einen dreidimensionalen Eindruck zu erzeugen. Dabei werden stets rote Farbbestandteile als Vordergrundfarben verwendet und blaue Farbbestandteile als Hintergrundfarben verwendet. Alle anderen Farben des Spektrums liegen dazwischen, von wärmeren Farben im Vordergrund zu kälteren Farben im Hintergrund.

Bei den Brillen, die für die Betrachtung solcher Bilder benötigt werden, ist ein Brillenglas eine normale, transparente Folie, während eine Augenseite eine Prismenfolie besitzt, welches die Farbbestandteile ablenkt.

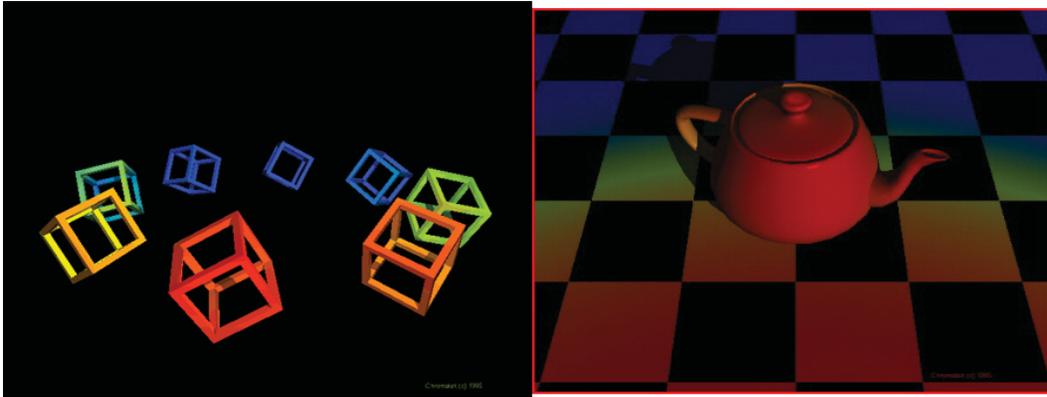


**Abbildung 3.9:** Eine Prismenfolie lenkt das sichtbare Licht unterschiedlich nach Wellenlängen ab. Dadurch entsteht der stereoskopische Effekt. ©Oregon State University

Diese Technik hat einen großen Vorteil, wenn es darum geht, möglichst schnell abstraktere Darstellungen dreidimensional erfassen zu können. So wird dieses Verfahren z.B. bei der Gestaltung eines Storyboards für 3D-Filme verwendet, um einen guten Eindruck davon wiederzugeben, wie später diese Szenen wirken könnten. Entsprechende Bilder können auch ohne Brille betrachtet und gedreht werden. Allerdings wird die Farbigkeit der Bilder immer bereits durch die auf dieser Technik beruhende Ablenkung der Farben vorbestimmt, wodurch

die Farbigkeiten der Abbildungen eingeschränkt wird.

Da allerdings die bei diesem Verfahren verwendeten Brillen optisch keine hochwertige Qualität aufwiesen, hatte dieses Verfahren im Mainstream nie eine wirkliche Bedeutung.



**Abbildung 3.10:** Beim linken Bild erscheinen die roten Würfel im Vordergrund, die blauen im Hintergrund. Beim rechten Bild befindet sich der bekannte Teapot auf einer dreidimensionalen Fläche. ©Chromatek

### 3.2.3 Polarisation

Bei sichtbarem Licht handelt es sich um elektromagnetische Transversalwellen. Dabei beschreibt die Polarisation genau die Ebene, in der eine dieser Lichtwellen schwingt. Bei ungefiltertem sichtbarem Licht schwingen die Lichtwellen in allemöglichen Richtungen und bilden dabei elliptische Bewegungen ab. Um diese Gegebenheit des Lichts für die Stereoskopie nutzbar zu machen, muss das Licht polarisiert werden[Sch07]. Dazu unterscheidet man die Polarisation in die lineare und zirkulare Polarisation. Für die Nutzung dieser Technologie sind oftmals zwei Ausgabegeräte notwendig, um beide Perspektiven einer Ansicht übereinander zu projizieren.

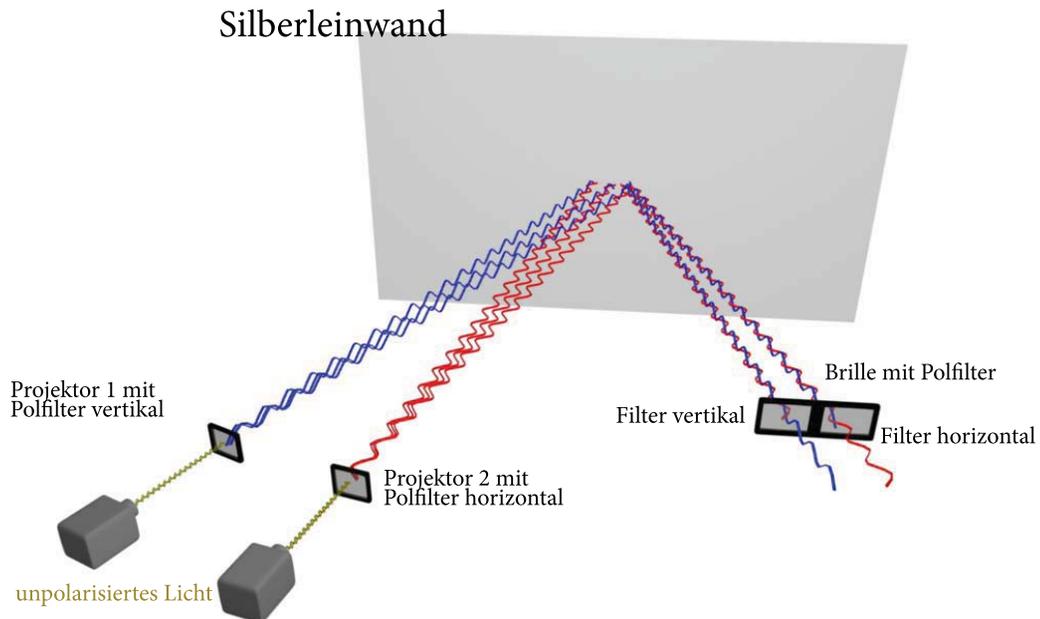
Die Technik der Polarisation wird heutzutage sehr weit verbreitet eingesetzt und befindet sich deshalb in den meisten digitalen 3D-Kinos, da die Produktion benötigter Brillen sehr günstig ist und damit schnell sehr viele Betrachter eingebunden werden können, ohne größere Ausgaben tätigen zu müssen. Zusätzlich dazu ist diese Technik heutzutage so ausgereift, dass fehlerhafte Darstellungen mit „Geisterbildern“ nahezu überall vermieden werden können. Nachteil dieser Technologie ist, dass zwei Projektoren auf die gleiche Fläche projizieren müssen, was die Kosten für die Ausgabegeräte direkt verdoppelt und eine perfekte Projektion dadurch zwar möglich, aber nur mit Aufwand sehr genauer Justage möglich ist.

#### 3.2.3.1 lineare Polarisation

Lineare Polarisation bedeutet, dass sichtbares Licht in genau eine Ebene für jeweils eine Ansicht gefiltert wird[Mah05]. Dabei wird z.B. vor dem linken Projektor, welcher für die linke Ansicht des Bildes zuständig ist, ein horizontaler Filter verwendet, während der rechte

Projektor einen vertikalen Filter besitzt. Diese beiden Polarisationen müssen dann auf den selben Punkt projiziert werden. Um die Polarisation dabei nicht aufzuheben, ist eine metallische Projektionsfläche notwendig. Dazu wird in der Praxis oftmals eine Silberleinwand eingesetzt, welche beide Polarisationen des Lichtes beibehält. Damit nun der Betrachter aus diesen übereinander liegenden Bildern eine stereoskopische Ansicht erhält, muss auch dieser eine Brille tragen, die für die linke Seite einen horizontalen und für die rechte Seite einen vertikalen Filter besitzt.

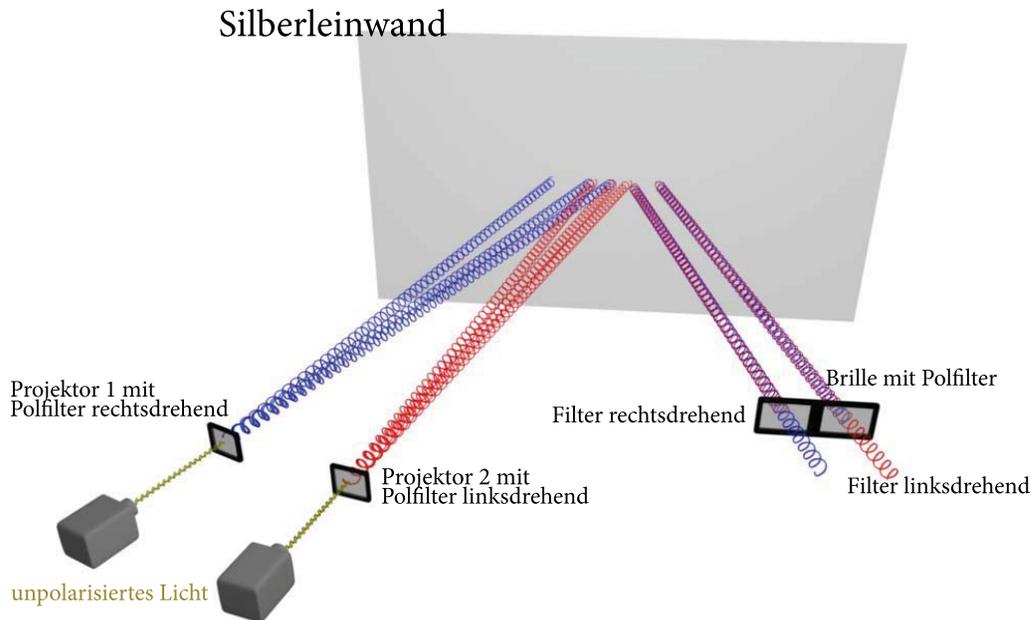
Nachteil der linearen Polarisation ist, dass sobald der Kopf mit aufgesetzter Brille gedreht und geneigt wird, die Filtereigenschaft der Polarisationsfilter aufhebt und somit kein stereoskopisches Ergebnis sichtbar ist.



**Abbildung 3.11:** Beispiel für eine lineare Polarisation mittels zwei Projektoren, Silberleinwand und Brillen.

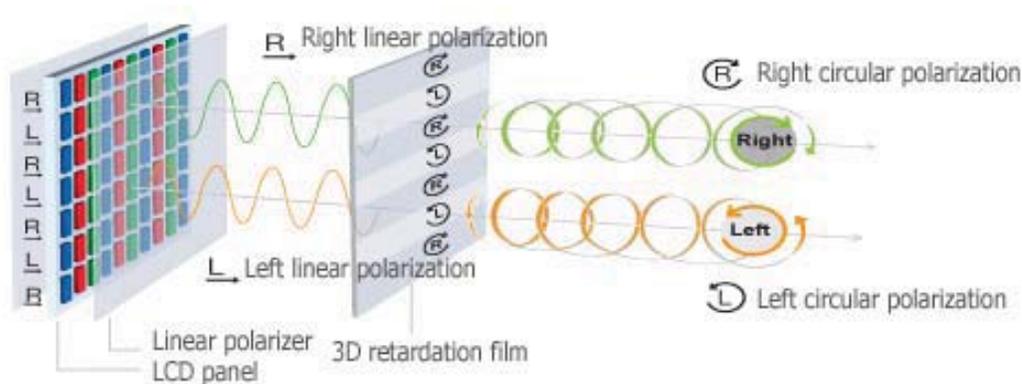
#### 3.2.3.2 zirkulare Polarisation

Zirkulare Polarisation bedeutet, dass die Eigenschaft des Lichtes genutzt wird, elliptische Bewegungen abzubilden. Für entsprechende Filter besitzen diese jeweils die Eigenschaft, nur links- bzw. rechtsdrehendes Licht durchzulassen [Mah05]. Der Betrachter muss, wie schon bei linearer Polarisation, genau die gleichen Filter auf der Brille besitzen, um diese stereoskopische Ansicht sehen zu können. Vorteil gegenüber der linearen Polarisation ist, dass der Kopf dabei geneigt werden kann, da weiterhin links- und rechtsdrehendes Licht gefiltert werden kann.



**Abbildung 3.12:** Beispiel für eine zirkulare Polarisation mittels zwei Projektoren, Silberleinwand und Brillen.

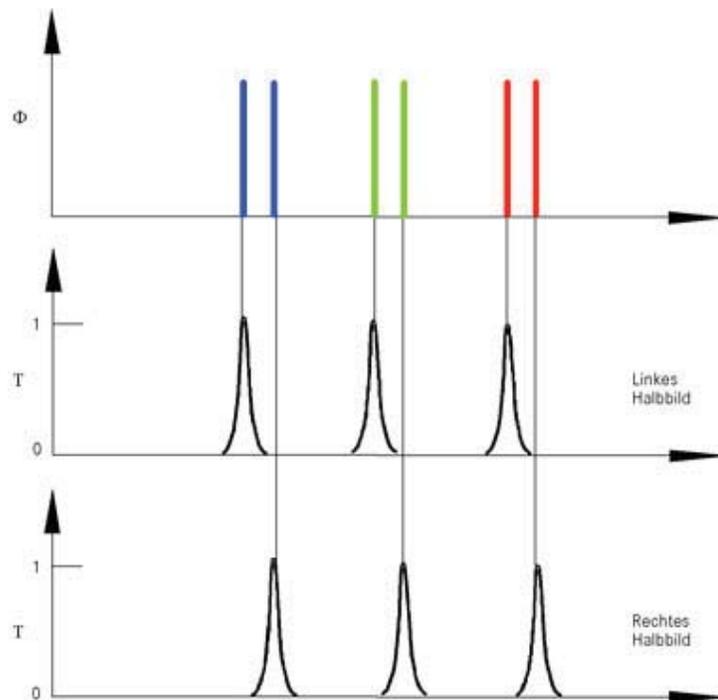
Im Gegensatz zur linearen Polarisation wird diese Technik heutzutage auch in Monitoren eingesetzt, welche bereits heute zu Preisen um die 200€ erhältlich sind. Nachteil dieser Monitore ist, dass dabei die Betrachterposition nur in sehr geringen Rahmen nach oben und unten abweichen darf, da diese Monitore darauf beruhen, dass die Polarisation durch Filter, welche bereits in den Panels integriert sind, vorgenommen wird. Auch hier muss der Betrachter die gleichen Brillen tragen, wie sie bei anderer zirkularer Polarisation üblich sind.



**Abbildung 3.13:** Funktionsweise eines 3D-Monitores mit zirkularer Polarisation von Zalman.  
©Zalman Tech Co., Ltd.

### 3.2.4 Interferenzfiltertechnologie

Die Interferenzfiltertechnologie beruht darauf, mit Hilfe des Wellenlängenmultiplex-Verfahrens die für eine stereoskopische Darstellung notwendigen Ansichten zu übertragen [Sch07]. Bei einem Wellenlängenmultiplex-Verfahren wird versucht, mehrere Wellenlängen in einer optischen Übertragung zusammen zu fassen. Dabei werden durch trennscharfe Interferenzfilter vor den zwei notwendigen Projektoren die Wellenlängen der jeweils drei Farbbestandteile Rot, Grün und Blau um wenige Nanometer verschoben. Hauptbestandteil dieser Filter sind aufgedampfte dielektrische Beschichtungen auf Trägern wie Glas oder Kunststoff. Wie bei anderen passiven stereoskopischen Anzeigeverfahren besitzt auch hier der Betrachter die entsprechend gleichen Filter auf seiner Brille. Diese trennen wiederum ihre Wellenlängenbestandteile heraus, wodurch eine stereoskopische Sicht ermöglicht wird.



**Abbildung 3.14:** Verschiebung der Wellenlängen mittels Filter und Wellenlängentripel der beiden Halbbilder. ©infitec

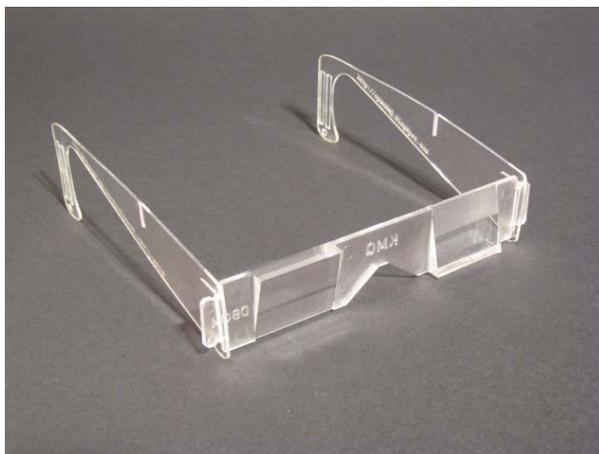
Vorteil dieser Technologie ist eine sehr gute Trennbarkeit beider Kanäle, wodurch Fehler wie „Geisterbilder“ nahezu unmöglich sind. Allerdings hat die Besonderheit dieser Technologie auch ihre Nachteile. Zum einen ist im ursprünglichen Filterformat dieser Technologie wegen einer scheinbar sehr starken Beschichtung die Farbtreue nicht ausreichend einhaltbar. Dazu kommt, dass schon bei geringer Beleuchtung des Projektionsraumes diese Farbtreue weiter abnimmt. Im Gegensatz zur Polarisierungstechnik sind hier sowohl die Filter vor den Projektoren, als auch die Filter in den Brillengestellen sehr teuer, was eine Massenbetrachtung im

Heimkinobereich nicht erschwinglich macht.

Patentiert wurde dieses Verfahren von der Firma infitec GmbH, welche heutzutage alle Filter dieser Art vertreibt. Auch im 3D-Kino hat diese Technologie Einsatz gefunden und wurde dadurch auch weiterentwickelt. Kinos mit dem Verfahren Dolby 3D benutzen von infitec produzierte Brillen, welche eine deutlich bessere Farbtreue aufweisen, wodurch ein 3D-Film in diesen Kinos nahezu ohne negative Aspekte erlebt werden kann. Diese Brillen werden oftmals nur gegen Pfand ausgegeben und müssen nach der Vorstellung zurückgegeben werden.

### 3.2.5 KMQ

Darstellungen, welche für die KMQ-Technik produziert sind, enthalten beide Ansichten im ÜberUnter-Format. Das bedeutet, die rechte Perspektive der Ansicht befindet sich oben und die linke Ansicht unten. Für diese Technik ist eine so genannte KMQ-Brille notwendig. Hierbei besitzt eine entsprechende Brille auf beiden Seiten jeweils ein Prisma. Dadurch wird der Blick durch das linke Brillenglas nach unten abgelenkt, während der Blick durch das rechte Brillenglas nach oben abgelenkt wird.



**Abbildung 3.15:** KMQ-Brille mit beiden ablenkenden Prismen. ©Robomotic

Die Entwicklung dieser Technik stammt von der Universität Hohenheim. Das Kürzel KMQ steht dabei für die Nachnamen der Entwickler Dr. Christoph Koschnitzke, Rainer Mehnert und Dr. Peter Quick.

Da entsprechende Brillen eine hohe Güte besitzen ist der stereoskopische Eindruck angezeigter Bilder sehr gut und da beide Blickwinkel nur abgelenkt werden, bleibt die Farbtreue erhalten. Diese Technik wird in vielen Bereichen eingesetzt, aber hauptsächlich findet man diese Art stereoskopischer Darstellungen in Museen auf großen Plakaten.



**Abbildung 3.16:** Für das Betrachten dieses Bildes bitte die KMQ-Brille aufsetzen.

## 3.3 Aktive Anzeigeverfahren

Aktive Anzeigeverfahren sind dadurch beschrieben, dass ein Ausgabegerät die stereoskopische Ansicht mit einer Optik übermittelt. Die Brillen, durch welche ein räumlicher Eindruck entsteht, übernehmen dabei das Trennen der in einer Darstellung enthaltenen Perspektiven für das linke und rechte Auge.

### 3.3.1 Shutterbrillentechnik

Bei dieser Technik werden die notwendigen Perspektiven einer stereoskopischer Ansicht nacheinander durch ein Ausgabegerät angezeigt. Deshalb handelt es sich dabei um ein Zeitmultiplex-Verfahren[Sch07]. Die für die Betrachtung aufzusetzenden Brillen werden dabei mit dem Ausgabegerät über verschiedene Übertragungsmöglichkeiten synchronisiert, so dass jedes Auge nur die für sich bestimmte Ansicht erhält. Um dies zu realisieren, dunkelt ein Brillenglas, bestehend aus Liquid Crystal Displays, ab, wenn das andere Brillenglas eine

Ansicht erhält. Dadurch „flackern“ diese Brillen förmlich, wenn sie im Betrieb sind. Da bei dieser Technik jedes Auge nur die Hälfte der Bildwiederholrate sieht, muss eine gesamte Bildwiederholrate mindestens 120Hz betragen, damit jedes Auge mindestens 60Hz wahrnimmt und kein Flackern sichtbar ist.

Diese Technik hat mehrere Vorteile. Zunächst ist hierfür stets nur ein Ausgabegerät notwendig, was die Kosten hierfür relativ niedrig hält. Dadurch entfallen auch Notwendigkeiten wie das Übereinanderprojizieren von zwei Anzeigegeräten. Da dieses Verfahren auf keinerlei Lichteigenschaften beruht, kann der Betrachter den Kopf drehen und wenden und trotzdem bleibt der stereoskopische Effekt erhalten.

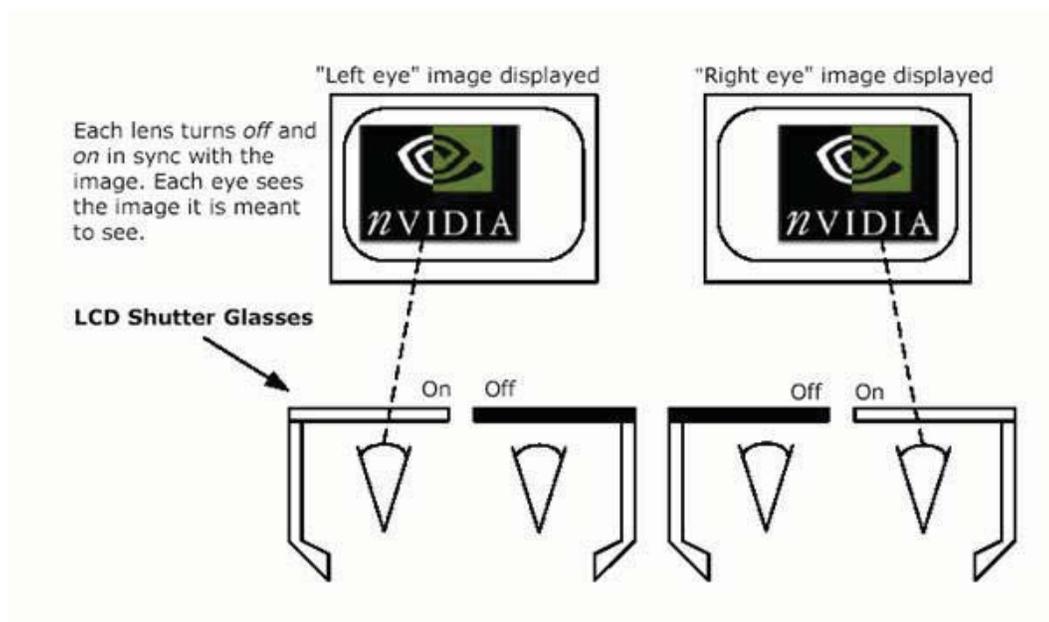
Nachteil dieser Technik ist unter anderem, dass ein entsprechendes Ausgabegerät erst seit ein paar Jahren wirklich nutzbar ist. In der Vergangenheit wurden für die Shutterbrillentechnik Röhrenmonitore benutzt, welche wegen nachleuchtender Phosphorbestandteile in der Röhre stets „Geisterbilder“ produzierten. Ein anderer Nachteil ist das „Flackern“ der Brillen, was bei manchen Menschen schneller zu Ermüdungen der Augen führt als es andere stereoskopische Anzeigeverfahren tun. Die für diese Technik notwendigen Brillen benutzen Batterien, um das abwechselnde Abdunkeln der Brillengläser zu ermöglichen, weshalb hierfür immer nur eine gewisse Betrachtungsdauer gegeben ist, bis diese Batterien aufgeladen bzw. erneuert werden müssen. Die in diesen Brillen vorhandenen LC-Displays sind auch der Grund, warum diese Brillen so teuer sind. Deshalb findet diese Technik auch in 3D-Kinos nicht allzu oft Verwendung.

Auf dem heutigen Markt gibt es verschiedene Ansätze die gewährleisten, dass Shutterbrillen stets den korrekten Teil des Lichts ausblenden und sich somit auf das stereoskopische Bild synchronisieren. Diese Ansätze können in emitterbasierende und emitterfreie Synchronisation unterteilt werden. Ein Emitter stellt hierbei sicher, dass die Brille nur die entsprechende Ansicht zeigt.

Den Ansatz der emitterbasierenden Synchronisation wird in den letzten Jahren z.B. von dem Grafikhersteller nVidia verfolgt. Dabei wird ein pyramidenförmiger Infrarotsender über USB mit dem PC verbunden. Mit Hilfe von systeminternen Grafiktreibern wird über diese USB-Verbindung stets ein abwechselndes Signal an den IR-Emitter gesendet. Dieser sendet dann das entsprechende Infrarotsignal, welches von der Shutterbrille empfangen und erkannt werden kann. Emitter dieser Art unterstützen je nach Größe und Anzahl der IR-LEDs und damit verbunden die Reichweite des infraroten Lichts, bis zu mehrere hundert Brillen. Allerdings hat der Hersteller nVidia bereits angekündigt, in naher Zukunft Brillen für den professionellen Grafikbereich zu vermarkten, welche über Funk synchronisiert werden und der entsprechende Emitter ausschließlich über einen 3-Pin-VESA-Stecker angeschlossen werden kann. Damit vermeidet nVidia die Fehler ihrer bisherigen Technologie mit dem Namen „3D Vision“, welche Probleme bei der Synchronisation hatte, weil über eine USB-Schnittstelle jederzeit eine zeitliche Verzögerung entstehen kann, wodurch die Brillen das falsche Signal erhalten können. Weiterhin braucht die bisherige Technik der Synchronisation stets ein Sichtfeld zum IR-Emitter oder wird über Umwege wie Decken und Wände reflektiert.

Einen völlig anderen Ansatz verfolgt die Technologie DLP-Link. Dahinter versteckt sich eine Technik, welche von zertifizierten Projektoren eingesetzt wird. Entsprechende Projektoren senden dabei zwischen jedem anzuzeigenden Bild einen kurzen Weißblitz, welcher von einer DLP-Link-Brille dank einer Photodiode am Bügel erkannt werden kann. Dadurch synchro-

nisiert sich die Brille mit dem System und ist auf keinerlei Emitter angewiesen. Der Träger muss lediglich in die Richtung des Bildes schauen. Problem hierbei ist bisher allerdings, dass ein Projektor lediglich ein abwechselndes Videosignal z.B. vom Computer erhält. Dadurch kann der Projektor nicht analysieren, welche Ansicht gerade angezeigt wird und sendet somit den Weißblitz zu einer zufälligen Zeit. Die Shutterbrille kann daraus ebenfalls nicht analysieren, ob diese Ansicht nun die rechte oder linke ist und dunkelt mit der angezeigten Frequenz des Weißblitzes nacheinander die LC-Displays des linken und rechten Auges ab. Allerdings ist auch hierfür bereits Abhilfe geschaffen. Zertifizierte Geräte können hierbei entweder über das Service-Menü den Weißblitz „umkehren“, wodurch eine einmalige kurze Verzögerung entsteht und damit die Brillen in ihrer Ansicht gewechselt werden. Damit auch dieses bald nicht mehr nötig ist, wurde im Jahre 2009 der Standard HDMI, Kurzform für High Definition Multimedia Interface, in der Version 1.4 verabschiedet. Dabei ist vorgesehen, das komplette stereoskopische Bild an den Beamer zu senden, z.B. im Übereinander-Untereinander- oder im Nebeneinanderformat. Dadurch besitzt der Beamer Informationen, welche Ansicht rechts und welche links ist und stellt den Weißblitz auf die entsprechende Einsatzzeit ein.



**Abbildung 3.17:** Die Bilder werden abwechselnd gezeigt während die Brille die jeweils andere Ansicht mit schwarzen LC-Displays sperrt. ©nVidia

#### 3.3.2 Head Mounted Displays

Die in deutsch übersetzten, so genannten „Kopf befestigten Bildschirme“ stellen exakt das dar, was der Name vermuten lässt. Bei diesen Brillen hat der Betrachter direkt vor seinen Augen zwei kleine LC-Displays in einem Gehäuse gefasst, welches er auf dem Kopf trägt [Sch07]. Für eine bessere Lichtabschattung sind diese Gehäuse meist geschlossen, es

gibt allerdings auch Varianten mit halbtransparenten Displays, welche wie normale Brillen getragen werden. Bei diesen Varianten unterstützt die stereoskopische Wiedergabe lediglich die reale Welt, weshalb hierbei auch von „Augmented Reality“, also der erweiterten Realität, gesprochen werden kann.

Die Brillen geschlossener Art bieten für den Betrachter exakt zwei Displays, wodurch die stereoskopische Sicht entstehen kann. Dadurch gibt es Probleme, die bei anderen Darstellungsverfahren entstehen, wie z.B. ein maximales Kopfnicken und -drehen oder Geisterbilder nicht. Durch das Vorhandensein zweier Displays direkt vor den Augen erscheinen diese als große Leinwand, welche je nach Preisklasse in „virtueller“ abgebildeter Größe und Auflösung variieren. Mit zusätzlichen Erweiterungen kann bei diesen Brillen auch eine verbesserte, lokalisierbare Stereoskopie erreicht werden. Kopfbewegung und -drehung kann dabei verfolgt werden, wodurch wiederum die Sicht, die der Betrachter hat, ebenfalls bewegt werden kann. Dadurch entsteht der Eindruck, dass sich der Betrachter tatsächlich in einer virtuellen Welt bewegt.

Neben diesen Vorteilen besitzen HMDs allerdings auch mehrere Nachteile. Dazu gehört unter anderem das schnelle Ermüden der Augen, da hier eine für das Gehirn völlig untypische Visualisierung der Umgebung stattfindet. Daneben gibt es auch heutzutage noch den Nachteil des hohen Preises für gute Head Mounted Displays. Während Geräte mit schlechterer Auflösung von gerade einmal 640x480 Pixeln circa 450€ kosten, gibt es Geräte mit heute üblicheren Auflösungen von 1280x1024 erst ab 17.000€. Allein deshalb etablieren sich entsprechende Geräte nur in Forschungseinrichtungen oder werden bei der militärischen Ausbildung genutzt.

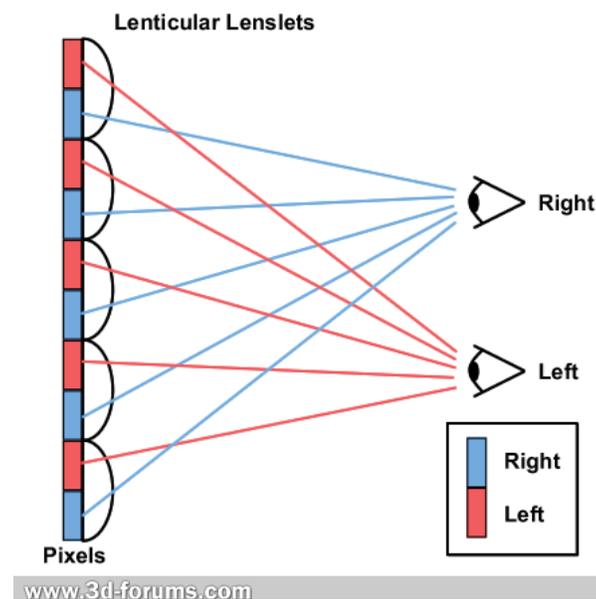


**Abbildung 3.18:** Beispiel eines Head Mounted Displays. ©Fifth Dimension Technologies

### 3.4 Autostereoskopische Displays

Unter dem Namen Autostereoskopie werden alle Geräte zusammengefasst, bei denen der Betrachter keinerlei Hilfsmittel benötigt, um eine stereoskopische Sicht zu erhalten. Trotz Fehlen dieser Hilfsmittel wird hier versucht, einen bestmöglichen Eindruck zu bieten [Sch07]. Am leichtesten wird diese Technologie verstanden, wenn man sich Postkarten und Schaubilder mit kippbaren Linsenrasterfolien betrachtet. Wird dieses Kärtchen ein wenig geneigt, sieht der Betrachter ein anderes Bild oder eine andere Ansicht der vorherigen Grafik. Auf dem gleichen Grundprinzip beruhen auch heutige stereoskopische Displays. Hierbei wird durch die parallele Anordnung schräg gestellter Streifenmasken oder Linsenraster sichergestellt, dass jedes Auge nur die jeweilige Ansicht bekommt. Dazu muss allerdings auch die stereoskopische Ansicht in Streifen und jedes Pixel in Subpixel zerlegt und stets hinter eine Streifenmaske gelegt werden. Steht nun der Betrachter in einem als vorgegeben optimalen Abstand, sieht jedes Auge eine andere Sicht, wodurch ein räumlicher Eindruck entsteht.

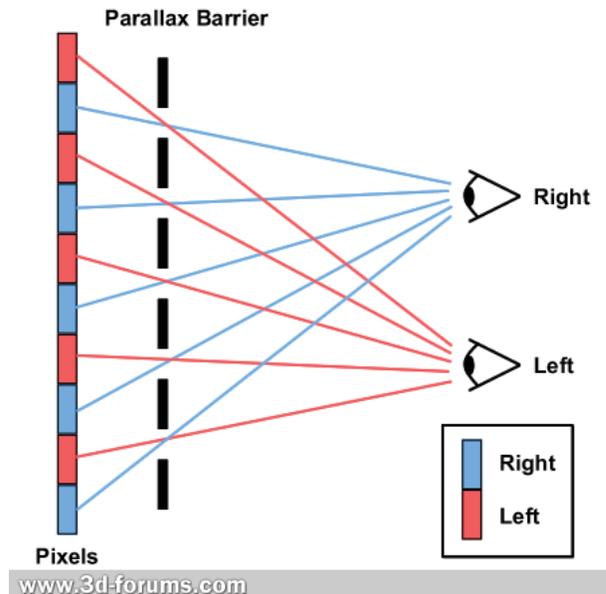
Bezogen auf die Gewährleistung, dass jedes Auge nur die entsprechende Ansicht bekommt, unterscheiden sich die Geräte in der Technik „Lenticular“ und „Parallax Barrier“ [Jia10][Bla08]. Der Unterschied hierbei ist lediglich, dass bei lenticularen Displays in einem festgelegten Abstand über dem Panel Linsen angeordnet sind, die das Licht so ablenken, dass ein Betrachter im festgelegten Betrachtungsbereich, dem so genannten „Sweet Spot“, die korrekte Ansicht für jedes Auge bekommt.



**Abbildung 3.19:** Funktionsweise eines lenticularen autostereoskopischen Displays.  
©www.3D-forums.com

Bei der Parallaxen Barriere basierenden Technik befindet sich in einem gewissen Abstand vor dem Panel eine Streifenmaske, sodass der Betrachter im vorgegebenen Abstand jeweils

nur die korrekt Ansicht pro Auge sehen kann.



**Abbildung 3.20:** Funktionsweise eines Parallaxen Barriere basierendem autostereoskopischen Displays. ©www.3D-forums.com

Weiterhin werden diese Displays in so genannte „Single User Displays“ und „Multi User Displays“ unterschieden. Bei Single User Displays handelt es sich um jene, bei denen nur ein Betrachter eine Ansicht aus einem bestimmten Abstand und Betrachtungswinkel sehen kann. Ist ein Display in der Lage, mehr als 2 Perspektiven für mehr als eine Person darzustellen, wird es Multi User Display genannt. Dabei besitzen beide Technologien Vor- und Nachteile. Vorteil eines Single User Displays ist, dass ein Betrachter im vorgegeben Abstand und Winkel eine sehr gute stereoskopische Sicht erhält und dabei die horizontale Auflösung des Geräts lediglich halbiert wird, während bei Multi User Displays die Qualität dieser Ansichten leidet, sodass aus manchen Winkeln auch andere, nicht für das linke oder rechte Auge bestimmte, Ansichten erkennbar werden. Weiterhin sinkt die horizontale Auflösung des Displays mit jedem anzeigbaren Blickwinkel proportional, wodurch eine qualitativ gute Darstellung bei mehreren Ansichten heutzutage noch nicht garantiert werden kann. Vorteil der Multi User Displays ist allerdings, dass mehrere Betrachter nebeneinander stehen können, dabei auch mehrere Ansichten sehen und auch ein einzelner Betrachter förmlich um die Objekte herum sehen kann, wenn er aus der mittigen Position tritt.

Erweitert werden autostereoskopische Displays heutzutage um die Möglichkeit, Augen- und Kopfbewegungen zu verfolgen, wodurch auch Single User Displays eine bessere Beweglichkeit des Betrachters ermöglichen. Displays dieser Art finden sich heutzutage hauptsächlich in der Forschung und Medizin, da diese Technologie immernoch sehr teuer ist. Allerdings gibt es bereits Ansätze, größere Displays dieser Art für den Heimkinobereich käuflich zu machen. Jetzige Geräte mit Diagonalen ab 50“ kosten dabei heute noch über 40.000€.

Alle bisherigen Modelle haben allerdings die Schwierigkeit, einen ähnlich guten räumlichen Eindruck wie andere stereoskopische Anzeigeverfahren zu gewährleisten. Entscheidend für einen guten räumlichen Eindruck ist die Möglichkeit, möglichst viele Subpixel für einen Pixel bereitzustellen. Damit ein entsprechender Eindruck gut ist, sollten dabei mehr als 100 Subpixel zur Verfügung stehen. Allerdings ist es heutzutage noch nicht möglich, so kleine Subpixelmasken in hoher Qualität herzustellen und zu verbauen. Damit würde auch eine entsprechende Bandbreite einher gehen, die ebenfalls noch nicht zu Verfügung steht. Verbunden mit den hohen Kosten sind deshalb diese Geräte auch nicht allzu häufig zu finden.

#### **3.5 Zusammenfassung**

Dieses Kapitel hat einen sehr umfangreichen Überblick über bisherige Möglichkeiten gegeben, einen räumlichen Eindruck für Betrachter zu erzeugen. Inhalte dieses Kapitels sind später für den Stand der Technik und Wahl der für diese Bachelorarbeit notwendigen VR-Systeme InstantPlayer und das Modellierungswerkzeug Autodesk Maya notwendig.

## Kapitel 4

# Mathematische Grundlagen für Optimierungen

Dieses Kapitel soll aufzeigen, auf welchen mathematischen Grundlagen bisherige Optimierungsverfahren basieren und auch begründen, warum nur eins dieser Verfahren für diese Bachelorarbeit in Frage kommt. Diese mathematischen Grundlagen werden in verschiedenen Bereichen der Stereoskopie zur Zeit eingesetzt und beziehen sich ausschließlich in einem Fall auf einen trigonometrischen Bezug zwischen Betrachter und Darstellung.

### 4.1 Formel „Native Pixel Parallax“

Die erste hier beschriebene Formel für Optimierungsmöglichkeiten bei stereoskopischen Darstellungen beruht auf reinen Erfahrungswerten in der Filmindustrie[Men09] und bietet sich dann an, wenn bei der Produktion stereoskopischer Inhalte vorher nicht klar ist, auf welche Größe die spätere Projektion stattfinden wird und wie weit die Betrachter davon weg sitzen. Deshalb sind Werte, die bei dieser Formel entstehen, als absolute maximale Grenzen zu verstehen und sollten daher nicht vollständig ausgereizt werden.

$$NPP = \frac{eyedist \cdot res_{hor}}{width_{output}} \quad (4.1)$$

$NPP$	= maximale interaxial Separation in Pixel
$eyedist$	= Augenabstand in cm (üblicherweise 6,5cm)
$res_{hor}$	= Auflösung des Ausgabegerätes in Pixel
$width_{output}$	= Breite des Ausgabegerätes in cm

Diese Formel besteht, wie auch die nachfolgenden, aus einem einfachen Bruch. Hierbei wird für den Zähler die Augendistanz in Zentimetern multipliziert mit der horizontalen Auflösung des Ausgabegerätes eingesetzt. Geteilt durch die Projektionsbreite, ebenfalls in Zentimetern,

ergibt sich der Wert der maximalen Trennung in Pixel, auch NPP genannt. Für eine optimale Projektion darf die in einem Bild vorhandene Trennung hierbei diese genannte Pixelzahl nicht überschreiten.

Bei dieser Formel mit dem Resultat „Native Pixel Parallax“, kurz NPP, wird schnell klar, dass sie nur bedingt für große Projektionen einsetzbar ist, wenn verschiedene Beispiele durchgerechnet werden. Bereits die Tatsache, dass die horizontale Auflösung in einem Faktor entscheidet, wie groß die interaxial Separation ausfällt, erweist sich bei Ausgabegeräten als falsche Annahme. Würde z.B. auf einem projizierten Bild mit einer Breite von 3m und einer horizontalen Auflösung von nur 1024 Pixel laut Formel eine interaxial Separation von rund 22 Pixel genutzt, wäre dies bei gleicher projizierten Größe mit einer horizontalen Auflösung von 1920 Pixel direkt 42 Pixel. Dadurch wird klar, dass diese Formel keine winkelabhängige Komponente besitzt, welche allerdings für eine korrekte Berechnung bezogen auf das menschliche Auge notwendig wäre.

Weitaus auffälliger ist hierbei jedoch, dass kleine Ausgabegeräte mit z.B. 40cm Breite laut Formel viel höhere Trennungen erfordern als viel größere Ausgabegeräte. Diese Tatsache ist logisch nicht nachvollziehbar, wenn das Verhalten des menschlichen Auges bezogen auf die Winkelabhängigkeit stereoskopischer Darstellungen betrachtet wird. Dieser auffällige Fehler entsteht deshalb, weil bei dieser Formel der Betrachterabstand nicht berücksichtigt wird und diese Formel lediglich auf Erfahrungswerten beruht.

## 4.2 Formel nach John Bercovitz

Diese Formel, entwickelt von John Bercovitz, ist in der Stereofotografie die am meisten benutzte Formel für die Berechnung der Stereobasis zwischen zwei Kameras[Kuh99]. Virtuell betrachtet ist das der gleiche Abstand, welcher von der interaxial Separation beschrieben wird. In dieser Formel werden viele Parameter berücksichtigt, wodurch verhindert werden soll, dass an irgendeiner Stelle im stereoskopischen Bild eine Überschreitung der interaxial Separation so geschieht, dass das Bild nicht mehr fusioniert werden kann.

$$b = d \cdot \left( \frac{af \cdot an}{af - an} \right) \cdot \left( \frac{1}{f} - \frac{1}{a} \right) \quad (4.2)$$

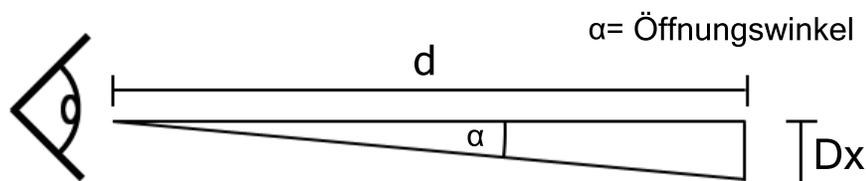
- $b$  = Stereobasis der Kameras; entspricht interaxial Separation in mm
- $d$  = gewünschter Bildabstand zw. Nah- und Fernpunkt; 0,0012m bei 35mm Film
- $af$  = am weitesten entferntes Objekt in m
- $an$  = nahestes Objekt in m
- $f$  = Brennweite Objektiv in m
- $a$  = Fokusabstand in m; entspricht Zero Parallax Plane

Hauptbestandteil dieser Formel sind zwei große Terme, die jeweils verschiedene Multiplikatoren mit enthaltenen Parametern darstellen. Während der erste Teil der Formel ein vordefinierter Wert des gewünschten Bildabstandes zwischen Nah- und Fernpunkt mit einer Angabe

von 0,0012m ist, bestehen alle restlichen aus frei festlegbaren Zahlen. Der erste große Term stellt einen Bruch dar, bei dem im Zähler der Abstand zum nächsten Objekt (in m) mit dem Abstand zum weitesten Objekt (in m) multipliziert wird. Dieses Produkt wird durch die Differenz dieser beiden Abstände dividiert und bildet damit den ersten berechneten Wert der Formel. Der zweite große Term besteht aus zwei einzelnen Brüchen. Der erste Bruch wird hierbei durch einen Quotienten dargestellt, welcher aus der Teilung von 1 geteilt durch die Brennweite des Objektivs (in m) resultiert. Von diesem Bruch wird danach ein weiterer Bruch, bestehend aus dem Quotienten aus 1 geteilt durch den Fokusabstand (in m), subtrahiert und bildet damit den zweiten berechneten Wert dieser Formel. Zwar ergibt diese Formel für viele reelle abzubildende Szenen gute Ergebnisse, allerdings wird es schwierig, diese Formel in der virtuellen Realität einzusetzen, da hierbei kein Betrachterabstand und keine Projektionsgröße mit einfließen. Deshalb ist auch diese Formel nur bedingt nutzbar, da sie nur die aufzunehmende Szene mit einbezieht, aber logischerweise auf einem kleinen Monitor und gleichzeitig auf einer riesigen Kinoleinwand nicht die gleichen stereoskopischen Parameter verwendet werden können. Den hauptsächlichen Einsatz findet diese Formel deshalb auch heute nur in der Stereofotografie. Der Grund dafür ist, dass diese Formel aus den Zeiten analoger Fotografie stammt, bei der eine Diaprojektion der aufgenommenen Bilder notwendig war, um die Ergebnisse zu präsentieren.

### 4.3 Formel „1,5° maximaler Winkel“

Bereits in den Grundlagen des räumlichen Sehens wurde klar, dass Menschen einen maximalen Winkel von 1,5° abweichend vom betrachteten Punkt als noch wahrnehmbar empfinden können. Das heißt, dass in der Praxis abhängig vom Betrachterabstand und der Projektionsbreite ein Winkel größer als 1,5° nicht überschritten werden sollte. Daher wird schnell ersichtlich, dass mit Bezug von Winkel und Betrachterabstand die Definition der maximalen interaxial Separation notwendig wird.



**Abbildung 4.1:** Der Öffnungswinkel des Auges sollte maximal 1,5° betragen. In Anlehnung an ©Thomas M. Lehmann

$Dx$  = interaxial Separation an der Anzeige  
 $d$  = Betrachterabstand

Aus dieser trigonometrischen Beziehung folgt eine bereits entickelte Formel. Zur Feststellung dieser Formel wird sowohl der Winkel alpha mit einer Größe von  $1,5^\circ$ , als auch die Gegenkathete, entspricht  $Dx$ , und die Ankathete, entspricht  $d$ , in ein Dreieck eingezeichnet. Mit Anwendung der trigonometrischen Beziehung des tangens erhält man hierdurch die Möglichkeit, eine maximale interaxial Separation auf einem dargestellten stereoskopischen Bild festzulegen.

$$Dx = \tan(1,5) \cdot d \quad (4.3)$$

$Dx$  = maximale interaxial Separation in cm  
 $d$  = Betrachterabstand in cm

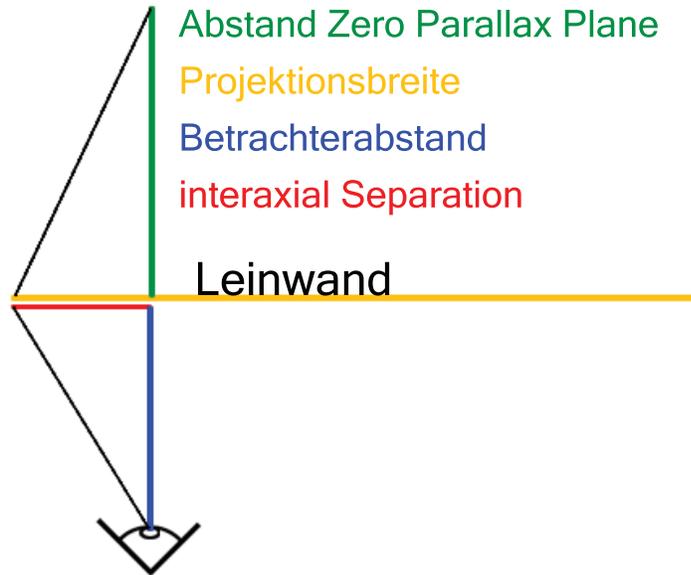
Damit bietet diese Formel bereits die Notwendigkeit der Abhängigkeit von Betrachterabstand und interaxial Separation. Dadurch wird auch klar, dass bei größerem Betrachtungsabstand eine größere interaxial Separation ermöglicht wird, was auch logisch nachvollzogen werden kann. Allerdings hat diese Formel, wie auch die vorangegangene Formel, das Problem, dass sowohl die Projektionsbreite als auch ein wichtiger stereoskopischer Parameter, die Zero Parallax Plane, darin nicht vorkommen. Deshalb ist auch diese Formel in dieser Form nur bedingt für die virtuelle stereoskopische Projektion einsetzbar.

Im Rahmen dieser Bachelorarbeit wurde deshalb versucht, diese Formel so zu erweitern, dass sie auch für die virtuelle Stereoskopie genutzt werden kann. Hierbei wurde darauf geachtet, dass die beiden fehlenden Parameter, die Zero Parallax Plane und die Projektionsbreite, mit in die Formel aufgenommen werden. Dafür fand eine Versuchsreihe statt, bei der in Autodesk Maya mehrere stereoskopische Bilder erstellt wurden. Diese Bilder hatten jeweils eine unterschiedliche interaxial Separation und Zero Parallax Plane.

Für eine korrekte Erweiterung wurden dazu auch die Maße der Projektion festgehalten und notiert. Nach der Erstellung der Bilder wurden diese betrachtet und auf der Projektion die interaxial Separation abgemessen und ebenfalls notiert. Dadurch entstand nach einer Weile eine große Anzahl an Messwerten, aus denen eine Funktion ersichtlich war. Jedoch hatte diese Funktion zu diesem Zeitpunkt keine konkreten Parameter, sondern bestand zunächst aus den gemessenen Werten.

Nach einiger Zeit wurden in die vorhandene Formel die Zero Parallax Plane in den Zähler und die Projektionsbreite in den Nenner übernommen. Der daraus berechnete Wert für die interaxial Separation wurde nach dem Programm, aus dem die erweiterte Formel entstand, benannt. Somit konnte nun stets die korrekte maximale interaxial Separation gemessen werden.

Die nachfolgende Grafik beschreibt dabei die visuellen Zusammenhänge der einzelnen Bestandteile der Formel:



**Abbildung 4.2:** Verbesserte Formel für maximale interaxiale Separation.

Hier wird ersichtlich, dass eine Vermischung von virtuellen und reellen Parametern stattfindet. Während die interaxiale Separation und die Zero Parallax Plane zunächst virtuelle Werte darstellen, bildet die Projektionsbreite und der Betrachterabstand den reellen Teil.

Zur Überprüfung dieser erweiterten Formel fanden weitere Messreihen statt. Dabei wurde mit der vorherigen Ausgangsformel für verschiedene Projektionsgrößen mit verschiedenen Betrachtungsabständen eine optimale interaxiale Separation berechnet. Danach folgte die Berechnung mit der erweiterten Formel, welche zunächst einen anderen Wert ergab. Dieser bezog sich jedoch auf einen virtuellen Wert und musste deshalb durch Messen an der Projektion als korrekt nachgewiesen werden. Da dort der Wert der erweiterten Formel exakt dem Wert der Ausgangsformel entsprach, wurde hiermit objektiv die Richtigkeit der erweiterten Formel nachgewiesen.

Zusätzlich dazu fanden subjektive Tests mit 2 Testpersonen statt, welche die angezeigten stereoskopischen Darstellungen ebenfalls als gut bezeichneten, was die Korrektheit dieser erweiterten Formel weiter bestärkte.

$$Dx_{maya} = \frac{\tan(1,5) \cdot Zeroparallax \cdot d}{AusgabegeraetBreite} \quad (4.4)$$

$Dx_{maya}$  = der in einer Software einzugebende Wert für interaxiale Separation in cm

$Z_p$  = Zero Parallax Plane-Distanz in cm

$d$  = Betrachterabstand in cm

$AusgabegeraetBreite$  = Breite des Ausgabegerätes in cm

Diese nun erweiterte Formel hat zusätzlich zur Ausgangsformel im Zähler einen weiteren Faktor, den Abstand der Zero Parallax Plane in cm. Geteilt wird der gesamte Bruch dann durch die Breite der Projektionsgröße (in cm), wodurch insgesamt wieder eine interaxial Separation in der Einheit Zentimeter entsteht.

Wichtig hierbei ist, dass dieser Wert stets in dem jeweiligen Programm eingegeben wird. Dieser dort virtuelle Wert ergibt beim Abmessen den realen Wert der Ausgangsformel. Da diese Formel für die meisten Anwendungen entsprechend einfach einzusetzen ist, wurde sie für den in dieser Bachelorarbeit vorkommenden Praxisteil eingesetzt.

### **4.4 Zusammenfassung**

In diesem Kapitel wurden verschiedene Ansätze aufgezeigt, mit denen es möglich ist, Optimierungen im Bereich virtueller stereoskopischer Inhalte zu realisieren. Nach den verschiedenen Ansätzen wurde allerdings schnell klar, dass nur eine dieser Optimierungen auch tatsächlich geeignet ist, da nur hierbei eine Winkelabhängigkeit zwischen Auge des Betrachters und Betrachtungsabstand vorhanden ist. Diese Formel wurde dann für die virtuellen Welten erweitert, sodass diese Formel auch alle Parameter für reelle und virtuelle Eigenschaften besitzt. Diese Formel wurde dann letztendlich für die Bachelorarbeit ausgewählt.

## Kapitel 5

# Grundlagen Programmierung

In den letzten Kapiteln wurden alle nötigen Begriffe der Stereoskopie in der virtuellen Realität erklärt und anhand von Beispielen verdeutlicht. In diesem Kapitel werden alle Begrifflichkeiten, die den Praxisteil dieser Bachelorarbeit betreffen, erklärt, sodass der Leser ohne Schwierigkeiten in der Lage ist, die Methodik dieser Bachelorarbeit nachvollziehen zu können.

### 5.1 X3D

eXtensible3D, kurz X3D, ist eine Sprache zur Beschreibung von 3D-Szenen[Klo10]. Dabei baut sie vollkommen auf die eXtensible Markup Language, kurz XML, auf. X3D ist der Nachfolger der Beschreibungssprache VRML und wurde 2001 vom W3C Konsortium verabschiedet. Ein sehr großer Vorteil von X3D ist, dass es eine offene und freie Sprache ist und somit von jedem genutzt werden kann. X3D-Dokumente bestehen dabei aus Elementen und Attributen. Während Elemente stets durch Tags dargestellt sind, stellen Attribute zusätzliche Informationen innerhalb dieser Elemente dar. Zur Betrachtung mit X3D erstellter Szenen benötigt es zusätzlich noch einen Viewer, welcher in dieser Bachelorarbeit durch den vom Fraunhofer IGD entwickeltem InstantPlayer repräsentiert wird. Auch dieser Viewer ist frei und kann, wenn nicht kommerziell eingesetzt, kostenlos verwendet werden. Ein weiterer Grund für die Nutzung dieses Viewers ist die Tatsache, dass dieser verschiedenste stereoskopische Darstellungsweisen unterstützt und dabei die gleichen Parameter verwendet wie das verwendete Modellierungswerkzeug Autodesk Maya. Da auch die Beschreibung der 3D-Modelle innerhalb von X3D wie XML aufgebaut ist, fällt das Erlernen dieser Sprache nicht schwer, wenn XML Kenntnisse vorhanden sind.

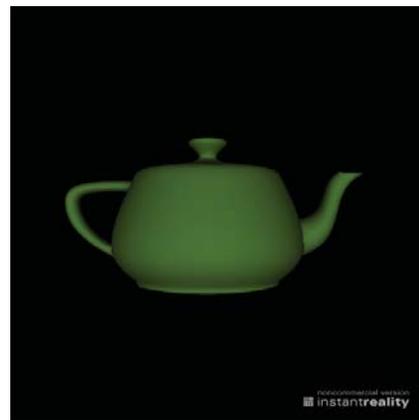
Im nachfolgenden Beispiel wird ein einfacher Text als String angezeigt:

```
<?xml version="1.0" encoding="UTF-8"?>
<X3D profile='Immersive'>
  <Scene>
    <Shape>
      <Text string="Hello world!" />
    </Shape>
  </Scene>
</X3D>
```



Der nachfolgende Code hingegen erzeugt den bekannten Teapot mit einem vordefinierten Material:

```
<?xml version="1.0" encoding="UTF-8"?>
<X3D profile='Immersive'>
  <Scene>
    <Shape>
      <Appearance>
        <Material diffuseColor='0.3 0.6 0.3' />
      </Appearance>
      <Teapot />
    </Shape>
  </Scene>
</X3D>
```



Der Teapot gehört in der Sprache X3D zu den so genannten Grundprimitiven und kann deshalb mit dem Element `<Teapot />` direkt erzeugt werden.

Aus diesen gezeigten Beispielen wird ersichtlich, welche Struktur ein X3D-Dokument aufweist. Jedes X3D-Dokument besteht aus einer XML-Deklaration in der Kopfzeile und kann, wenn erforderlich, durch eine Dokumenttyp-Deklaration erweitert werden. Besitzt ein X3D-Dokument die angeführte XML-Deklaration und hält dabei zusätzlich folgende Regeln ein:

- es existiert exakt ein oberstes Wurzelement, z.B. `<X3D>...</X3D>`
- jedes Element besitzt ein Start- und End-Tag, `<element1>...</element1>` mit Ausnahme von in sich geschlossenen Elementen ohne Inhalt, gekennzeichnet als `<element2/>`
- ein Element darf nur ein Attribut gleichen Namens besitzen

,so ist es *wohlgeformt*[Klo10]. Besitzt ein X3D-Dokument allerdings zusätzlich dazu eine Dokumenttyp-Deklaration, so ist es dann auch als *gültig* zu bezeichnen[Klo10]. Dabei verweist die Dokumenttyp-Deklaration, auch DTD genannt, auf ein Dokument, in der die Grammatik definiert ist, welche das X3D-Dokument erfüllen muss, um als *gültig* zu gelten.

## 5.2 ECMAScript

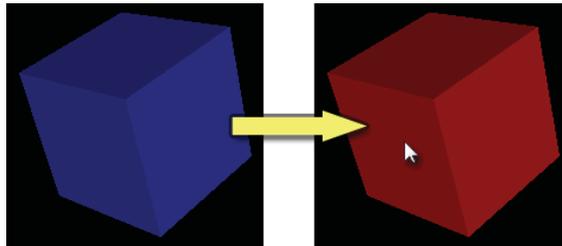
Um eine Interaktivität zwischen Benutzer und 3D-Szene zu bieten, ist eine weitere Programmiersprache nötig. Für den in der Bachelorarbeit gebrauchten Funktionsumfang bietet sich das ECMAScript an, welches direkt in die Architektur von X3D-Szenen eingesetzt werden kann. Mit Hilfe dieser Skriptsprache ist es möglich, die angezeigte Szene und deren Objekte zu beeinflussen[DB07]. Jedoch ist für diese Bachelorarbeit viel wichtiger, dass mit dieser Skriptsprache auch Eigenschaften der Render-Engine geändert werden können. Erst hierdurch ist es möglich, Optimierungen im Bereich stereoskopischer Darstellungen umzusetzen. Notwendig dafür ist allerdings auch ein Viewer, welcher stereoskopische Darstellungen beherrscht und mit einer Skriptsprache die Möglichkeit bietet, auf Elemente dieser Darstellung zuzugreifen.

In einem X3D-Dokument ist es möglich, ECMAScript sowohl intern als auch extern einzubinden. Das nachfolgende Beispiel zeigt, wie eine Interaktion zwischen Benutzer und X3D-Szene realisiert wurde. Fährt der Benutzer hierbei mit der Maus über den Würfel, ändert dieser seine Farbe von Blau zu Rot:

```
<?xml version='1.0' encoding='UTF-8'?>
<X3D profile='Interchange'>
<Scene>
<!-- Quader mit TouchSensor und Farbwechsel -->
  <Shape>
<Appearance>
  <Material DEF="Farbe" diffuseColor='0 0 1' />
</Appearance>
<Box />
  </Shape>
  <TouchSensor DEF="TouchMe" />

<!-- ECMAScript zum beruehrungsabhaengigen Farbwechsel -->
<Script DEF="FarbScript">
  <field name="aktiv" accessType="inputOnly" type="SFBool" />
  <field name="farbe_gewechselt" accessType="outputOnly" type="SFColor" />
  <![CDATA[ecmascript:
function aktiv(Wert, Zeitstempel) {
  if (Wert == true) {
farbe_gewechselt = new SFColor(1, 0, 0); // rot
  } else {
farbe_gewechselt = new SFColor(0, 0, 1); // blau
  }
}
  ]]>
</Script>
```

```
<ROUTE fromNode='TouchMe' fromField='isOver'  
  toNode='FarbScript' toField='aktiv' />  
<ROUTE fromNode='FarbScript' fromField='farbe_gewechselt'  
  toNode='Farbe' toField='set_diffuseColor' />  
</Scene>  
</X3D>
```



**Abbildung 5.1:** Abbildung der MouseOver-Funktionalität in X3D mit ECMAScript.

Um die Funktionalität des Beispiels zu ermöglichen, ist zum einen ein *TouchSensor* und zum anderen *Routes* notwendig. Bei einem *TouchSensor* handelt es sich um einen Sensor, welcher innerhalb der ausgeführten Szene jede Mausektion über dem ihm zugewiesenen Bereich erkennen kann. Im gezeigten Beispiel liegt der *TouchSensor* in der Hierarchie direkt im Element `<Scene>...</Scene>`, d.h. er agiert auf alle Mausektionen der sich dort befindlichen Objekte. Da hier allerdings nur ein Würfel genutzt wird, reagiert der Sensor auch nur auf Mausektionen über diesem. Fährt der Benutzer nun mit der Maus über den Würfel, so ändert der *TouchSensor* das Attribut `isOver` von `FALSE` nach `TRUE`. Die *Routes* sind dafür zuständig, einen Weg zwischen dem programmierten Script und den X3D Szenen zu bilden. Im gezeigten Beispiel gibt es also einen Weg vom Attribut `isOver` des Elements `TouchMe`, also dem *TouchSensor*, zum Attribut `aktiv` des Elements `FarbScript`. D.h., sobald sich das Attribut `isOver` ändert, ändert sich auch der dazu abgebildete Wert vom Attribut `aktiv` innerhalb des ECMAScripts. Ist dieser Wert dort dann `TRUE`, so wird eine neue Farbe definiert. Damit nun auch der Würfel diese entsprechende Farbe übernehmen kann, gibt es eine zweite Route vom Attribut `farbe_gewechselt` des Elements `FarbScript` zum Attribut `set_diffuseColor` des Elements `Farbe`. Eine Besonderheit der *Routes* wird hier klar, wenn der Teil *toField* bei der zweiten Route betrachtet wird. Hier wird nicht das eigentliche Attribut des Würfels, welches `diffuseColor` heißt, aufgerufen, sondern lediglich eine Funktion, welches dieses Attribut beim Aufrufen setzen kann. Diese Art des Umleitens ist in X3D üblich und bereits in der API vorgesehen und deshalb für den Programmierer nicht direkt nachvollziehbar, sondern meist nur durch Beispiele erlernbar.

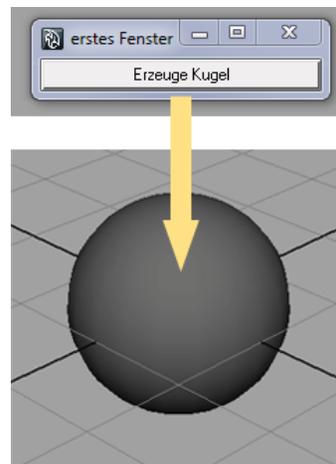
### 5.3 MEL

Die Maya Embedded Language, kurz MEL, ist eine von Autodesk entwickelte Skriptsprache zur Steuerung von Maya[MRW05]. Besonders hieran ist, dass als Benutzer die Möglichkeit

besteht, zu sehen, welche MEL-Befehle Maya selbst intern ausführt, um z.B. eine Kugel zu erstellen. Beispiel: Klickt der Benutzer auf einen entsprechenden Button in der Benutzeroberfläche von Maya, so zeigt der Script-Editor bei korrekter Einstellung an, welcher MEL-Befehl beim Anklicken des Buttons ausgeführt wurde. Diesen Befehl kann man nun direkt im Script Editor kopieren, ihn dort ausführen und erhält das gleiche Ergebnis wie beim Klicken auf den Button.

Da Autodesk Maya seit der Version 2009 bereits stereoskopische Kameras mit spezialisierten Funktionen und Kameratypen wie „Off Axis“ besitzt, wird es mit Hilfe von MEL möglich, Optimierungen stereoskopischer Parameter vorzunehmen. Zusätzlich zu diesen Befehlen ist durch MEL die Möglichkeit gegeben, ganze Layouts zu erzeugen, wodurch der Benutzer eine komplette, eigenständige Oberfläche erhält, in der er agieren kann. Im folgenden Code-Beispiel wird ein einfaches Layout erstellt, in dem sich ein Button befindet. Wird dieser Button geklickt, so erstellt Maya eine neue Kugel:

```
window -title "erstes Fenster" -width 200
-height 20 wnd1;
columnLayout -adj true;
button -label "Erzeuge Kugel"
      -command "polySphere";
showWindow wnd1;
```



Hier wird ersichtlich, dass MEL einen für andere Programmier- und Skriptsprachen untypischen Aufbau zeigt. Jedes Script beginnt hierbei mit der Definition der Fenstergröße und der Festlegung des späteren Layouts inklusive Buttons, Optionsmenüs, Eingabefeldern und untergeordneten Spalten- und Zeilenlayouts. Erst danach wird dieses durch den Befehl *showWindow* angezeigt. Die eigentliche Funktion der Erstellung einer Kugel übernimmt im Beispiel der Code-Teil *-command "polySphere"*.

Wird die Funktionalität erweitert, wie es auch für das endgültige Script dieser Bachelorarbeit nötig ist, bieten sich die so genannten „procedures“ an. Dabei handelt es sich um Funktionen, welche mehrere Aktionen oder gar andere Funktionen nacheinander aufrufen können. Alle Definitionen dieser procedures finden nach der Festlegung des Layouts statt. Zum besseren Verständnis ist nachfolgend ein Script mit derselben Funktionalität wie oben angeführtes dargestellt, allerdings diesmal mit Hilfe einer procedure.

```
window -title "erstes Fenster" -width 200 -height 20 wnd1;
columnLayout -adj true;
button -label "Erzeuge Kugel" -command "createSphere()";
```

```
showWindow wnd1;

proc createSphere()
{
polySphere;
}
```

Eine geeignete Programmierumgebung für MEL wird von Autodesk nicht mitgeliefert. Dementsprechend fehlen Fehleranzeigen beim Schreiben des Codes oder eine Art „Debugging“. Entweder wird das Script direkt im Scripteditor geschrieben, wobei hier auch keinerlei Highlighting und Tabbing des Codes stattfindet, oder ein erweiterter Texteditor wie Notepad++ mit Plugins für das Highlighting von MEL wird eingesetzt. Problematisch ist hier nur, dass jedes geschriebene Script zunächst in Maya kopiert oder geladen werden muss und weiterhin keine Fehleranzeigen stattfinden. Bei der Ausführung dieser kopierten Codefragmente kann es zu Instanzproblemen kommen. Diese entstehen, wenn das zuvor ausgeführte Script z.B. für ein Fenster bereits einen Instanznamen definiert hatte. Stürzt dieses Script ab, kann es passieren, dass es lediglich unsichtbar ist, allerdings noch existiert. Wird nun das Script erneut ausgeführt, kann das bereits vorhandene Fenster nicht erneut angezeigt werden. Fehler dieser Art müssen im Script-Editor per Hand korrigiert werden, in dem entsprechenden Instanznamen aus der Laufzeit gelöscht werden.

### 5.4 Zusammenfassung

In diesem Kapitel wurden alle nötigen Begriffe für den Praxisteil dieser Bachelorarbeit vorgestellt. Es wurde weiterhin ersichtlich, dass sich die Programmierung genannter Systeme nicht immer als einfach gestaltet, da es an vorhandenen Entwicklungsumgebungen und entsprechender Dokumentationen der Programmiersprachen mangelt. Dennoch sollte der Leser einen Eindruck von dem Aufbau vorgestellter Programmiersprachen bekommen haben und somit in der Lage sein, die Beschreibung der für diese Bachelorarbeit erstellten Praxisarbeiten zu verstehen.

# Kapitel 6

## Stand der Technik

Um die in diesem Kapitel vorhandenen Verfahren und Geräte der Stand der Technik zu verstehen sind Grundlagen der vorangegangenen Kapitel notwendig. Nachfolgend werden verschiedene Geräte der Stand der Technik aufgezeigt die hauptsächlich in den Forschungs- und Wirtschaftssektoren eingesetzt werden. Unterschieden werden diese Beispiele in verbesserte Darstellungsverfahren und optimierte Anpassungen stereoskopischer Parameter für die Betrachtung.

### 6.1 Darstellungsverfahren

Die Geräte, welche in dieser Kategorie aufgezeigt werden, besitzen verbesserte Möglichkeiten, stereoskopische Projektionen zu erzeugen. Dabei wird hauptsächlich darauf geachtet, dass hierbei der Betrachter möglichst fehlerfrei die angezeigten Inhalte betrachten kann. Diese Geräte verbessern allerdings nicht die stereoskopische Inhalte in ihren stereoskopischen Parametern selbst.

#### 6.1.1 Autostereoskopische Displays

##### 6.1.1.1 Magnetic 3D 57“ ALLURA ENABL3D™ LCD DISPLAY

Dieses Display, produziert von dem Unternehmen Magnetic 3D[Mag], stellt eines der zur Zeit größten autostereoskopischen Displays dar. Wie auch bei allen anderen autostereoskopischen Displays muss der Betrachter in einem optimalen Abstand zum Bild stehen, um einen stereoskopischen Effekt zu sehen. Bei diesem Gerät beträgt dieser Abstand zwischen 5m und 8,5m. Zusätzlich handelt es sich bei diesem Display um ein Multi User Display, da es bis zu neun Ansichten bereitstellen kann. Die volumetrische Wahrnehmung bei diesem Gerät ist mit 91cm angegeben und soll dadurch einen besonders beeindruckenden dreidimensionalen Effekt erzielen. Da es sich hierbei um ein sehr aktuelles Modell handelt, besitzt dieses Display eine Auflösung von 1920x1080 Pixeln, was umgangssprachlich als FullHD interpretiert wird. Der derzeitige Preis für dieses Gerät beträgt 16.000€.



**Abbildung 6.1:** Autostereoskopisches 57" Allura Display von Magnetic 3D. ©Magnetic

#### 6.1.1.2 Alioscopy 3D HD 47"

Wie auch das von Magnetic 3D vorgestellte Gerät besitzt das von Alioscopy entwickelte Gerät [Ali] derzeit eine Auflösung von 1920x1080 Pixeln. Im Gegensatz zum vorher genannten Bildschirm besitzt das Alioscopy 3D HD 47" 8 verschiedene Ansichten und einen Betrachterabstand von 3m bis 9m mit einer optimalen Position bei 4,4m. Der Preis für dieses Gerät wird zur Zeit mit 10.000€ angegeben.



**Abbildung 6.2:** Autostereoskopisches 3D HD 47" Display von Alioscopy. ©Alioscopy

### 6.1.2 HEyeWall 2.0

Bei der HEyeWall handelt es sich um eine 5x2,5m große Leinwand, welche von 24 Cube3D<sup>2</sup>-Projektoren der Firma digital-IMAGE[[dig](#)] beleuchtet werden. Bei einer Auflösung von 1400x1050 Pixeln pro Projektor wurde hier eine gesamte Auflösung von 8400x4200 Pixeln erreicht, was ungefähr 35 Megapixeln entspricht[[Boc09](#)]. Dies ermöglicht auch ein Betrachten der Projektion von nahen Distanzen, ohne ein zu niedrigauflösendes Bild zu erzeugen. Damit ist die HEyeWall ein einzigartiges Produkt im Bereich der Stereoprojektion.



**Abbildung 6.3:** Stereoskopische Projektion einer geografischen Karte auf der HEyeWall 1.0  
. ©Fraunhofer IGD

Besonders daran ist auch, dass die Projektoren dabei nur eine Optik verwenden, um ein stereoskopisches Bild zu erzeugen. Dabei basiert die Technik in den Projektoren auf der Shutterbrillentechnik, verwendet allerdings im Gegensatz zu den heute erhältlichen Stereoprojektoren mit Shutterbrillentechnik direkt zwei Eingänge, sodass eine spezielle Grafikkarte nicht nötig wird und systemunabhängig verwendet werden kann. Für ein flackerfreies Bild sorgen hier mit 120Hz projizierte Bilder, sodass jedes Auge stets 60Hz wahrnehmen kann.



**Abbildung 6.4:** Aktiver Stereoprojektor Cube3D<sup>2</sup> des Unternehmens digital-IMAGE.  
©digital-IMAGE

Hinter dieser Projektion steht ein Cluster von 48 Rechnern zur Verfügung, welcher die dreidimensionalen Daten verarbeitet, aufbereitet und für die Projektion neu generiert.

### 6.1.3 Head Mounted Displays

#### 6.1.3.1 Virtual Research VR1280

Dieses Head Mounted Display, hergestellt von dem Unternehmen Virtual Research[Vir], ist eines der wenigen topaktuellen Produkte in diesem Bereich. Ausgestattet mit einer Auflösung von 1280x1024 Pixel pro Display bietet es in heutiger Zeit eine ausreichende Auflösung für das Auge. Wichtiger Bestandteil dieses HMDs sind zwei SXGA auflösende FLCOS (ferroelectric liquid crystal on silicon) Displays. Diese Technik sorgt für eine noch schärfere Darstellung von Bildern auf den kleinen Bildschirmen im Gegensatz zu LCOS (liquid crystal on silicon) Displays. Neben diesen Merkmalen besitzt dieses Gerät einen diagonalen Bildwinkel von 60° und liegt preislich bei ca. 13.000€.



**Abbildung 6.5:** Head Mounted Display VR1280 des Unternehmens Virtual Research.  
©Virtual Research

### 6.1.3.2 NVIS nVisor SX111

Wie auch das VR1280 bietet dieses Gerät eine Auflösung von 1280x1024 Pixeln pro Display, basiert allerdings auf LCOS-Displays. Im Gegensatz zum VR1280 bietet dieses Gerät allerdings einen diagonalen Bildwinkel von 111°[NVI], wodurch ein natürlicheres Umsehen simuliert werden kann. Das SX111 bietet optionale Merkmale wie Head-Tracking ebenfalls an, sodass beim Umsehen in der virtuellen Welt eine realistische Kopfbewegung nachgeahmt werden kann. Zusätzlich benötigt ein Besitzer dieses Geräts noch eine „Video Control Unit“, welche die Informationen für die Bilddarstellung auf den Displays verarbeiten kann. Wird ein top ausgestattetes Modell dieser Serie bestellt, steigt der Preis von 19.350€ für das Gerät allein auf stolze 43.450€.



**Abbildung 6.6:** Head Mounted Display nVisor SX111 von NVIS. ©NVIS

## 6.1.4 Geräte für den Heimgebrauch

Nicht nur im Bereich von Wirtschaft, Forschung und Militär haben sich stereoskopische Systeme verbessert. Gerade in den letzten Jahren sind auch Systeme für private Endanwender immer weiterentwickelt worden und heutzutage an einem Stand angekommen, welcher bereits als professionell betrachtet werden kann. Neben verschiedenen Systemen für die Projektion virtueller und filmischer stereoskopischer Inhalte gibt es nun auch mobile Geräte für die Betrachtung stereoskopischer Bilder, sei es zum Fotografieren, Filmen oder auch zum Spielen. Nachfolgend sind einige dieser Geräte mit Beschreibung der Funktionsweise aufgelistet.

### 6.1.4.1 Acer H5360

Der von Acer produzierte Projektor mit der Bezeichnung H5360 liefert für einen günstigen Preis von 600€ bereits stereoskopische Bilder in sehr guter Qualität. Dabei setzt dieser Beamer auf die Shutterbrillentechnik und projiziert die Bilder mit einer Frequenz von 120Hz[Ace09]. Benutzer dieses Systems haben die Möglichkeit, entweder auf die von nVidia

bereitgestellte Technologie 3D Vision, welche optional für 130€ erwerbbar ist, oder auf die offene Technik DLP-Link zurückzugreifen. Bei dieser Technik synchronisieren entsprechende Brillen direkt mit dem angezeigten Bild und sind erst jetzt auf dem Markt zu Preisen von ca. 80€ je Stück erhältlich.



**Abbildung 6.7:** Aktiver Stereoprojektor H5360 von Acer. © Acer

### 6.1.4.2 Sony KDL-60LX905

Auch Hersteller im Bereich von Fernsehgeräten bieten seit dem letztem Jahr Geräte an, welche dreidimensionalen Inhalt anzeigen können. Bei diesen Geräten kommen verschiedene Techniken zum Einsatz, genau wie im 3D-Kino. An dieser Stelle wird ein Gerät des Unternehmens Sony beschrieben, welches mit der Shutterbrillentechnik arbeitet, da dieses sowohl ein sehr gutes stereoskopisches Feeling ohne Ghosting produzieren kann, als auch Fehler beseitigt hat, die oftmals mit Shutterbrillen entstehen[Son10]. Dazu gehört auch, dass wenn sich ein Betrachter mit aufgesetzter Brille im Raum bewegt und umherschaut, andere Lichtquellen das Erlebnis beeinträchtigen können und es dort zum Flackern kommt, wenn der Betrachter z.B. in eine Lampe schaut. Mit verbesserter Technologie ist es hier Sony gelungen, dieses Problem nahezu vollständig zu beseitigen.



**Abbildung 6.8:** Der 3D-Fernseher KDL-60LX905 des Unternehmens Sony beruht auf der Shutterbrillentechnik. ©Sony

Weiterer Grund für die Nennung dieses Fernsehers gegenüber anderen auf Polarisation basierenden Geräten ist, dass der Betrachter mit der Brille stehen kann, wo er möchte, und dennoch stellt sich eine dreidimensionale Wahrnehmung ein. Bei Fernsehern mit der Polarisationstechnik bestehen die gleichen Probleme wie bei Monitoren mit gleicher Technik: Der Bewegungsraum ist vorgegeben und entspricht im Vertikalen meistens maximal  $10^\circ$  Spielraum.

Bei diesem Fernseher werden alle Inhalte in FullHD angezeigt und er ist bereits für 3D-BluRays gerüstet, unterstützt somit auch den HDMI 1.4 Standard. Preislich ist das Gerät mit 5.000€ angegeben und weitere 3D-Brillen kosten hierbei ca. 150€.

#### 6.1.4.3 Nintendo 3DS

Auch in der Spieleindustrie ist der Boom der 3D-Technologie angekommen und nicht nur die Sony Playstation 3 unterstützt in Verbindung mit einem 3D-fähigen Ausgabegerät stereoskopische Inhalte. Auch der Hersteller Nintendo hat sich mit der 3D-Technik beschäftigt und will im März 2011 eine neue Generation der mobilen Konsole Nintendo DS mit dem neuen Kürzel 3DS auf den Markt bringen. Der Nintendo 3DS besitzt ein autostereoskopisches Display mit einer Auflösung von  $800 \times 240$  Pixeln, was einer Auflösung von  $400 \times 240$  Pixeln bei eingeschaltetem 3D-Modus für jedes Auge bedeutet.



**Abbildung 6.9:** Die Spielekonsole Nintendo 3DS bietet ein autostereoskopisches Display.  
©Nintendo

Der Benutzer kann hier über das Gerät selbst die Tiefenwirkung des 3D-Displays über einen eingebauten Schieber kontrollieren. Mit dem eingebauten Display bietet Nintendo erstmals eine neue Generation des DS mit einem größeren Display an und auch weitere Eigenschaften des Geräts sind hier verbessert worden.

#### 6.1.4.4 Fujifilm FinePix REAL 3D W3

Im Bereich der Fotografie bot Fujifilm mit der Digitalkamera FinePix REAL 3D W1 bereits Mitte 2009 eine Kamera an, welche Stereoaufnahmen sowohl in Film als auch Bild umsetzen konnte. Während Filme nur in 640x480 Pixeln aufgenommen werden konnten, bietet die ab September erhältliche 3D W3 bereits 1280x720 Pixel an und verfügt erstmals über einen 3D-Modus für Makroaufnahmen und soll Vorgängerprobleme bezogen auf die Stereofotografie behoben haben[Fuj10]. Auf der Rückseite dieser Kamera befindet sich ein vergrößertes 3,5“ großes autostereoskopisches Display mit einer Auflösung von 1.150.000 Bildpunkten. Der Vorgänger 3D W1 besaß lediglich ein 2,8“ Display mit 230.000 Bildpunkten. Preislich liegt sie auf dem gleichen Niveau wie die 3D W1 und soll für 450€ erhältlich sein.



**Abbildung 6.10:** Die Digitalkamera FinePix REAL 3D W3 von Fujifilm. ©Fujifilm

## 6.2 Optimierte stereoskopische Projektionen

Innerhalb dieser Sektion befinden sich jene Geräte, welche die eigentlichen stereoskopischen Parameter so bearbeiten, dass der Betrachter ein angenehmeres Gefühl bei der Betrachtung hat. Diese Geräte optimieren allerdings nur nebensächlich die Darstellung stereoskopischer Inhalte, d.h. sie basieren auf bereits in den Grundlagenkapiteln vorgestellten Möglichkeiten, stereoskopische Projektionen sichtbar zu machen, ohne dabei diese Projektion zu verbessern.

### 6.2.1 Free2C Desktop Display

Das vom Fraunhofer Heinrich-Hertz-Institut entwickelte autostereoskopische Display mit der Bezeichnung Free2C Desktop Display kommt in den Anwendungsfeldern der Medizin, Architektur, Simulation, Design und Konstruktion zum Einsatz[Fra]. Es bietet neben einer auf Linsenrasterfolien basierenden stereoskopischen Ansicht mit einer Auflösung von 1200x1600 Pixeln die Möglichkeit, die Augenbewegung zu verfolgen. Dabei können bei schlechteren Lichtverhältnissen zusätzlich mehrere Infrarot-LED's zugeschaltet werden. Die Verfolgung der Augen geschieht hierbei mit bis zu 120Hz. Dadurch ist der Betrachter, im Gegensatz zu anderen autostereoskopischen Displays, nicht gezwungen, in einem bestimmten Bereich zu stehen, um eine Tiefenwahrnehmung zu erhalten.



**Abbildung 6.11:** Autostereoskopisches Display Free2C Desktop vom Fraunhofer HHI.  
©Fraunhofer HHI

Außergewöhnlich an diesem Display ist zudem der hochformatige Aufbau, welcher nur selten bei Bildschirmen verwendet wird. Mit Rücksicht auf die diagonale Größe von nur 21,3“ stellt dieses Display mit einer Auflösung von 600x1600 Pixeln pro Auge eines der hochauflösendsten autostereoskopischen Displays in der Forschung dar. Der große Nachteil ist allerdings, dass durch die Verfolgung der Augen nur ein Betrachter die Szene betrachten kann, dieses Gerät nicht käuflich erwerbbar ist und somit dem wirtschaftlichen Markt vorenthalten bleibt.

### 6.2.2 Seereal C-i 18

Auch hierbei handelt es sich um ein autostereoskopisches Display, welches in der Lage ist, die Kopf- und Augenbewegungen des Betrachters zu verfolgen[See]. Mit einer Größe von 18,1“ besitzt dieses Gerät eine Auflösung von 1280x1024 Pixel, wodurch ein Betrachter pro Auge 640x1024 Pixel sehen kann. Wie auch beim Gerät vom Heinrich-Hertz-Institut, befinden sich die Trackerkameras im oberen Teil des Gehäuses. Im Gegensatz zum Free2C Desktop Display ist dieses Gerät zwar auf dem Markt für ca. 10.000€ erhältlich, aber da dieses Gerät nicht mehr produziert wird, gibt es nur noch eine begrenzte Anzahl und wird somit in naher Zukunft durch einen Nachfolger, welcher noch nicht vorgestellt wurde, ersetzt.



**Abbildung 6.12:** Autostereoskopisches Display C-i 18 von Seereal inklusive Tracker-Kameras. ©Seereal

### 6.2.3 IMSYS

Die Firma imsys GmbH & Co. KG, Kurzform von immersive systems, bietet software- und/oder hardwarebasierende Virtual Reality-Systeme an, welche hauptsächlich für Präsentationszwecke genutzt werden. Besonders hervorzuheben sind hierbei die modularen Komplettlösungen[IMS10], bei denen auch die stereoskopischen Inhalte dank Kombination von Software und verfolgten Objekten optimiert werden können.



**Abbildung 6.13:** Modulares System für stereoskopische Projektion von IMSYS. ©IMSYS

Eines dieser trackbaren Objekte ist ein so genanntes Brillentarget[ICI], welches der Hauptbetrachter der Projektion trägt. Dieses Brillentarget ist mit retroreflektierenden Kugeln ausgestattet und wird durch Infrarotkameras verfolgt.



**Abbildung 6.14:** Trackbares Brillentarget, entwickelt von IMSYS. ©IMSYS

Daraus ergibt sich eine maximale interaxial Separation und Vergrößerung der angezeigten Objekte. Zusätzlich dazu besitzt der Hauptbetrachter einen Navigator[ICI], mit dem er virtuelle Objekte anfassen, drehen, skalieren und verschieben kann.



**Abbildung 6.15:** Trackbarer Navigator, entwickelt von IMSYS. ©IMSYS

Auch dieser Navigator besitzt retroreflektierende Kugeln, die zur Positionsbestimmung der Hand im virtuellen Raum dienen und somit eine Interaktion zwischen Mensch und Maschine erleichtern.

## 6.2.4 zukünftige Systeme

In naher Zukunft wird es möglich sein, Bilder sehr nah direkt vor dem Auge zu projizieren. Dieser kurzer Abschnitt befasst sich mit zwei verschiedenen Methoden, dies zu ermöglichen.

### 6.2.4.1 Microvision Color Eyewear

Das zur Zeit von der Firma Microvision, Inc. entwickelte optische System erscheint zunächst als eine einfache Brille. In Wahrheit steckt dahinter allerdings der Versuch, direkt Bilder auf die Brille zu projizieren und für den Menschen sichtbar zu machen[Mic]. Grundvoraussetzung für die Entwicklung eines solchen Systems bietet Microvision, welche bereits durch kleine, an den Apple iPod anschließbare Projektoren bekannt geworden sind. Genau wie bei diesen Projektoren kommen auch bei der Microvision Color Eyewear kleinste elektronische Bestandteile zum Einsatz. Dies sind unter anderem ein MEMS, Kurzform für Micro-Electro-Mechanical-System Scanning Mirror, welcher eine Einheit darstellt, die ankommendes Licht durch einen nur 1x1mm großen Spiegel in alle Richtungen ablenkt. Dadurch entsteht bei schnellen Kippbewegungen des Spiegels, ähnlich DLP, ein aufgespanntes Bild mit einer bestimmten Auflösung. Dazu kommen weitere Bestandteile, die Bilder, z.B. ein Eingangssignal für die Brille, in elektrische Lichtinformationen umwandeln und über kleine Laserdioden an eine Kombinationsoptik weiterleiten, die wiederum aus allen einzelnen Lichtbestandteilen der Farben Rot, Grün und Blau gemischte Farben kombiniert und an den MEMS weiterleitet. Zwar ist diese Technik bisher nur als monokular vorgesehen, könnte jedoch binokular erweitert werden und würde dann einen neuen Trend der stereoskopischen Sicht ermöglichen[Sch10].



**Abbildung 6.16:** Brille Color Eyewear von Microvision mit projizierten Bildern auf den Gläsern. ©Microvision

### 6.2.4.2 Retinal Image Display

Einen Schritt weiter geht das aus dem Bereich der Drucker bekannte Unternehmen Brother[Bro08]. Im Jahr 2009 gelang es Brother, ein Retinal Image Display, nachfolgend als RID bezeichnet, einen aus der Drucktechnik stammenden Laser für die Projektion von Bildern direkt auf der Netzhaut zu nutzen. Weitere Technik basiert hierbei, wie auch Color Eyewear, auf dem MEMS. Die Auflösung des Displays entspricht zur Zeit 800x600 Pixel, welches dem Betrachter wie ein Bildschirm mit einer Diagonale von 10cm einen Meter vor dem Auge zu schweben scheint. Dadurch erreicht das Display eine virtuelle Größe von 16". Würde auch dieses System erweitert werden, wären stereoskopische Visualisierungen in der Augmented Reality möglich, die eine noch nie gesehene Qualität erreichen würden[Sch10]. Problematisch sind solche Displays allerdings für Menschen, die auch in ihrer normalen Sicht bereits Schwierigkeiten haben. So kann es hier zu Schwindelgefühlen und Reisekrankheit kommen.



**Abbildung 6.17:** Von Brother entwickeltes Retinal Image Display. ©Brother

## 6.3 Zusammenfassung und Ausblick

In diesem Kapitel wurde sehr deutlich, wie weit der Stand der Technik ist. Sehr auffällig dabei ist, dass hauptsächlich im Bereich der Forschung, Wirtschaft, Militär und Industrie an Geräten gearbeitet wird, welche verbesserte, stereoskopische Ansichten bieten. Dabei setzen viele Institute immer mehr auf autostereoskopische Displays, sodass alle Betrachter auf jegliche Brillen verzichten könnten. Genau das ist der Punkt, bei dem viele potenzielle Betrachter von der stereoskopischen Technik Abstand nehmen, weil es unnatürlich ist, entsprechende Brillen dauerhaft zu tragen, welche unter Umständen dazu führen können, dass

der Betrachter Kopfschmerzen oder ähnliche Beschwerden bekommt. Die bisherige Technik reicht allerdings noch nicht aus, um mehreren Betrachtern eine hohe Anzahl von Blickwinkeln in sehr guter Qualität zu bieten. Weiterhin ist hier die Problematik des festgelegten Betrachterabstands bei mehreren Betrachtern zur Zeit nicht lösbar, da ein entsprechendes System nur einen Betrachter verfolgen kann. Und gerade in diesem Bereich sind Geräte in letzter Zeit immer seltener geworden und es scheint bisher keine Nachfolger dafür zu geben. Aber auch im Heimkinobereich kommt nach und nach 3D-Technik zum Einsatz, sei es bei Projektoren, Fernsehern, Kameras oder mobilen Spielekonsolen. Damit ist es nun auch dem Endanwender zu Hause möglich, stereoskopische Inhalte in sehr guter Qualität betrachten zu können. Für die Verbreitung dieser Technik sorgen auch immer neuere Standards wie z.B. der 2009 überarbeitete BluRay-Standard mit Erweiterung zur heutigen 3D-BluRay, von denen es bereits einige Filme auf dem Markt gibt. Die Kosten solcher System sinken immer weiter und ermöglichen somit immer mehr Leuten, sich entsprechende Geräte käuflich zu erwerben.

Neben diesen bekannten Möglichkeiten stereoskopische Inhalte zu präsentieren gibt es einige wenige zukünftige Modelle, welche in ein paar Jahren eine völlig neue Art der Stereoskopie präsentieren könnten. Vor allem in Bereichen der Augmented Reality wären diese Geräte sehr hilfreich und könnten die menschliche Sicht durch Zusatzinformationen erweitern.

Trotz all dieser Neuheiten, verbesserten Möglichkeiten der stereoskopischen Darstellungsweisen und optimierter stereoskopischer Projektionen fehlt es immernoch an Lösungen, welche tatsächlich den Inhalt von stereoskopischen Darstellungen verbessern, sodass der Betrachter eine angenehmere Sichtweise auf die präsentierten Abbildungen bekommt. Der Markt für solche Geräte ist bereits vorhanden und lediglich ein paar Hersteller vermarkten entsprechende Systeme, schotten sich dabei aber von außen ab. Hier ist es für jeden möglich, das erwerbbar System zu nutzen, allerdings bisher nicht, es zu erweitern oder eigenen Bedürfnissen anzupassen. Zusätzlich dazu kosten entsprechende Systeme zur Zeit sehr viel Geld und bieten sich deshalb nur sehr großen oder wirtschaftlich rentablen Firmen an. Diese Systeme basieren oftmals auf eigenen Formaten für die Beschreibung dreidimensionaler Modelle, unterstützen mit ihren eigenen Visualisierungsprogrammen lediglich eine Hand voll systemfremder Formate und dabei oftmals nicht alle Funktionen dieser Modelle.

Schlussfolgernd wird mit diesem Stand der Technik klar, warum an einer frei verfügbaren und kostenlosen Variante dieser Optimierung der stereoskopischen Inhalte gearbeitet werden sollte. Das bereits vorgestellte Programm InstantPlayer vom Fraunhofer IGD bietet dabei alle Funktionalitäten, die in diesem Bereich nötig sind und es bietet dazu genügend Freiheit, eigene Modelle, Welten und ganze Visualisierungen zu erschaffen.



## Kapitel 7

# Konzept stereoskopischer Vereinfachung in Autodesk Maya

Das Modellierungswerkzeug Autodesk Maya bietet bereits fundierte Möglichkeiten, stereoskopische Visualisierungen zu rendern. Mit dem aus dem Kapitel der stereoskopischen Sicht des Menschen sowie den stereoskopischen Begriffen der virtuellen Welt und verbunden mit den mathematischen Grundlagen hinter den Optimierungen der Stereoskopie ist es möglich, bereits ausgereifte Inhalte zu erzeugen, welche auch in der Praxis überzeugen und nicht zu Kopfschmerzen führen.

Mit der Programmierung eines MEL Scripts für diese Vereinfachung soll es dem Anwender von Autodesk Maya möglich sein, diese Optimierungen der Stereoskopie ohne große Hürden und Vorwissen erstellen zu können. In diesem Kapitel wird dargestellt, welchen Aufbau eine Visualisierung dieses Scripts besitzen soll und welche Funktionalitäten dieses Script besitzen wird.

### 7.1 Beschreibung des Aufbaus vom Script

Das zu erstellende Script soll für den Benutzer ein unterteiltes Layout darbieten, bei dem dieses Layout anhand der Funktionalitäten des Scripts unterteilt wird. Während im oberen Teil zunächst ein Bild platziert wird, folgt danach der Teil des Scripts, welcher für die Erstellung einer stereoskopischen Kamera zuständig ist.

In diesem Teil soll der Benutzer die Möglichkeit bekommen, die Brennweite der stereoskopischen Kamera und die zwei wichtigsten stereoskopischen Parameter, die interaxial Separation und Zero Parallax Plane, zu definieren. Dabei wird ebenfalls implementiert, dass bei nicht festgelegter interaxial Separation eine Abfrage der Projektionsgröße und des Betrachterabstands erfolgt. Damit ist gegeben, dass die erstellte stereoskopische Kamera exakt auf die angegebenen Parameter optimiert ist.

Im zweiten Teil des Layouts soll der Benutzer in der Lage sein, bereits erstellte stereoskopische Kameras in ihren Eigenschaften zu ändern. Mit einem entsprechenden Klick auf einen platzierten Button sollen diese Änderungen dann übernommen werden können.

Der dritte Teil der Oberfläche dient dazu, bereits eingegebene Projektionsmaße zu über-

schreiben.

### 7.2 Beschreibung der Funktionalitäten des Scripts

Die gesamten Funktionalitäten des Scripts sollen wie folgt programmiert werden:

- Erstellung einer stereoskopischen Kamera
  - Festlegung der Brennweite
  - Festlegung der interaxial Separation
  - bei Nicht-Festlegung der interaxial Separation Angabe der Projektionsmaße und Optimierung darauf
  - Festlegung der Zero Parallax Plane
- Bearbeitung der stereoskopischen Kameras
  - Verfolgung in der Szene vorhandener Objekte
  - Verfolgung neu erstellbarer Objekte
  - Abgleichen der gewählten Kamera mit Parametern anderer Kameras
  - Rendermöglichkeit für autostereoskopische Displays mit bis zu 24 Ansichten
  - Festlegung der Zero Parallax Plane auf ein gewähltes Objekt
- Bearbeitung der Projektionsmaße einer gewählten stereoskopischen Kamera
- Lademöglichkeit der in einer Szene bereits vorhandenen, von anderen Scriptinstanzen erstellter stereoskopischer Kameras
- Einbau eines kompletten Hilfesystems für Benutzer mit wenig Vorwissen zur Erleichterung in den Einstieg des Scripts

Hier wird ersichtlich, dass die Funktionalität dieses Scripts die einfache Optimierung stereoskopischer Kameras bereits übersteigt. Die Funktionalitäten rühren unter anderem daher, da mit bereits vorhandenem Vorwissen, eine Optimierung stereoskopischer Parameter innerhalb von Autodesk Maya sehr schnell möglich ist. Alle weiteren Funktionalitäten bieten zusätzlichen Spielraum für den Benutzer, das Script für seine Zwecke angepasst zu nutzen. So ist es häufig der Fall, dass eine Szene bereits komplett fertig modelliert ist inklusive Kamerafahrten. Die Möglichkeit, dass der Benutzer mit dem Script diese bereits vorhandenen Kameras durch auf die Projektion optimierte stereoskopische Kameras mittels eines Knopfdrucks ersetzen kann, erhöht die Wahrscheinlichkeit, dass dieses Script auch tatsächlich genutzt wird. Aber auch Verfolgungen von Objekten in der Szene sowie vom Script erstellter, so genannter „Null-Objekte“, wird möglich sein, um eine Kamerafahrt anhand eines Objekts oder eines „Null-Objektes“ zu simulieren. Dabei stellt ein „Null-Objekt“ eine leere Gruppe in Autodesk Maya dar, welche keine untergeordneten Objekte besitzt. Im verwendeten Fall dient ein

Null-Objekt damit lediglich der Führung der Kamera. Da auch in der FH Gießen Friedberg unter anderem ein autostereoskopisches Display zur Verfügung steht, wird dieses Script so erweitert werden, dass es möglich ist auch für Geräte dieser Art optimierte stereoskopische Bilder in bereits korrekt benannter Reihenfolge zu rendern. Damit fällt ein umständliches Umbenennen und Sortieren der gerenderten Bilder vollständig weg. Weiterhin sind die für diese autostereoskopischen Displays zusätzlichen Ansichten fest an die erstellte stereoskopische Kamera gebunden, so dass der Anwender diese völlig frei im dreidimensionalen Raum bewegen kann. Möchte ein Anwender eine stereoskopische Kamera haben, die allerdings eine variable Zero Parallax Plane bieten soll, kommt die Festlegung dieser auf ein gewähltes Objekt zum Einsatz. Somit bleibt der Fokus der Tiefenwahrnehmung stets auf dem gleichen Objekt und die Kamera kann dennoch frei bewegt werden.

Da sich allerdings die Umstände durchaus ändern können, müssen die optimierten stereoskopischen Parameter auch an diese Umstände angepasst werden können. Im dritten Teil der Übersicht kann der Benutzer eine erstellte stereoskopische Kamera in einem Listenfeld wählen und deren Projektionsmaße abändern. Bei der Erstellung einer Szene ist es vollkommen alltäglich, dass ein Benutzer das Programm schließt und später an der zu erstellenden Szene weiterarbeiten möchte. Dazu soll auch dieses Script in der Lage sein, von vorherigen Instanzen des Scripts alle erstellten stereoskopischen Kameras auszulesen, sodass auch diese nachträglich bearbeitet werden können.

Zu guter Letzt soll dieses Script ein umfassendes Hilfesystem mit angegebenen Fragezeichen-Buttons an den Optionen bieten. Damit soll auch unerfahrenen Benutzern die Möglichkeit gegeben werden, dieses Script zu benutzen. In diesem Hilfesystem werden alle Funktionalitäten und die stereoskopischen Begriffe nochmals erläutert und anhand von Beispielen visualisiert.

### **7.3 Anforderungen an das Script**

Mit den Beschreibungen des Aufbaus und der Funktionalitäten des Scripts wurden die Anforderungen an das Script definiert. Demnach soll jeder Benutzer eine leichte Möglichkeit besitzen, sich in die Stereoskopie innerhalb von Autodesk Maya einzufinden und Gefallen daran finden, verschiedenste Funktionalitäten dieses Scripts aus zu testen. Dies wird durch eine eingebaute Hilfe ebenfalls umgesetzt werden, da ohne diese Hilfe unerfahrene Nutzer schnell den Überblick über die Funktionalitäten verlieren können und stereoskopische Parameter eventuell falsch einstellen könnten, was wiederum zu unzufriedenen Ergebnissen führt und damit eine weitere Verwendung des Scripts unwahrscheinlich wird.

### **7.4 Zusammenfassung**

Die Umsetzung dieses Konzepts erfordert ein großes Wissen an den Umgang mit Autodesk Maya, der MEL Programmierung und die Einbindung stereoskopischer Grundlagen in dieses Script. Wird dieses Konzept allerdings realisiert, kann dem Benutzer ein deutlich leichter Umgang mit stereoskopischen Parametern innerhalb von Autodesk Maya geboten werden.

Mit den zusätzlichen Funktionalitäten wird weiterhin realisiert, dass ein Benutzer nach Belieben seine bereits erstellten Szenen in optimierte, stereoskopische Szenen umwandeln.

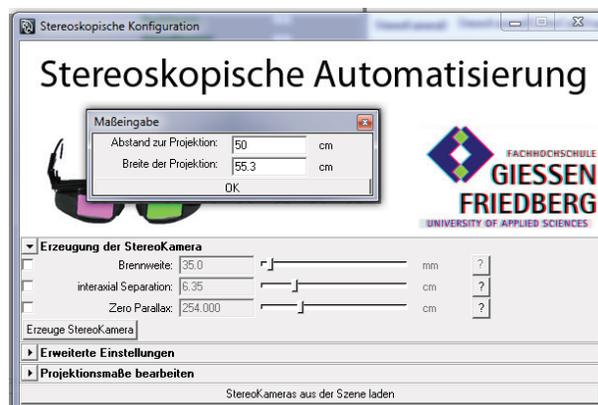
# Kapitel 8

## Implementierung des MEL Scripts

Im vorherigen Kapitel wurde erläutert, wie ein Konzept für die Realisierung eines interaktiven MEL Scripts aussehen könnte. Für die technische Umsetzung dieses Konzepts wird bei der Programmierung auf den frei erhältlichen Texteditor namens Notepad++ mit MEL Highlighting Plugins gesetzt. Dadurch fällt die Programmierung leichter, ist zudem strukturierter und kann stets neu in Autodesk Maya kopiert werden, falls ein Fehler beim Ablauf des Scripts sichtbar wird.

### 8.1 Die Oberfläche des Scripts

Im Konzept für die Realisierung wurde ein geeigneter Aufbau für die Umsetzung dieses Scripts vorgestellt. Da dieser Aufbau die logische Abfolge der Schritte bei der Erstellung stereoskopischer Kameras beinhaltet, wurde er exakt nach der Beschreibung im Konzept umgesetzt.



**Abbildung 8.1:** Übersicht über die Oberfläche des implementierten MEL-Scripts inklusive der Eingabe der Projektionsmaße.

### 8.1.1 Überblick über die Funktionalitäten

Auch die Anordnung der Funktionalitäten folgt exakt nach diesem Konzept. Bei jeder Funktionalität befindet sich ein so genannter Radiobutton, damit stets nur eine Funktionalität aktiviert werden kann. Ausnahme hierbei bildet die letzte Option, welche unabhängig von den anderen gewählt werden kann. Jede Funktionalität bietet ein oder mehrere Auswahlmensüs und jeweils einen Button für die Aktualisierung der vorhandenen Objekte in der Szene, Erstellen eigener neuer Objekte oder dem Rendern der Szene für ein autostereoskopisches Display.

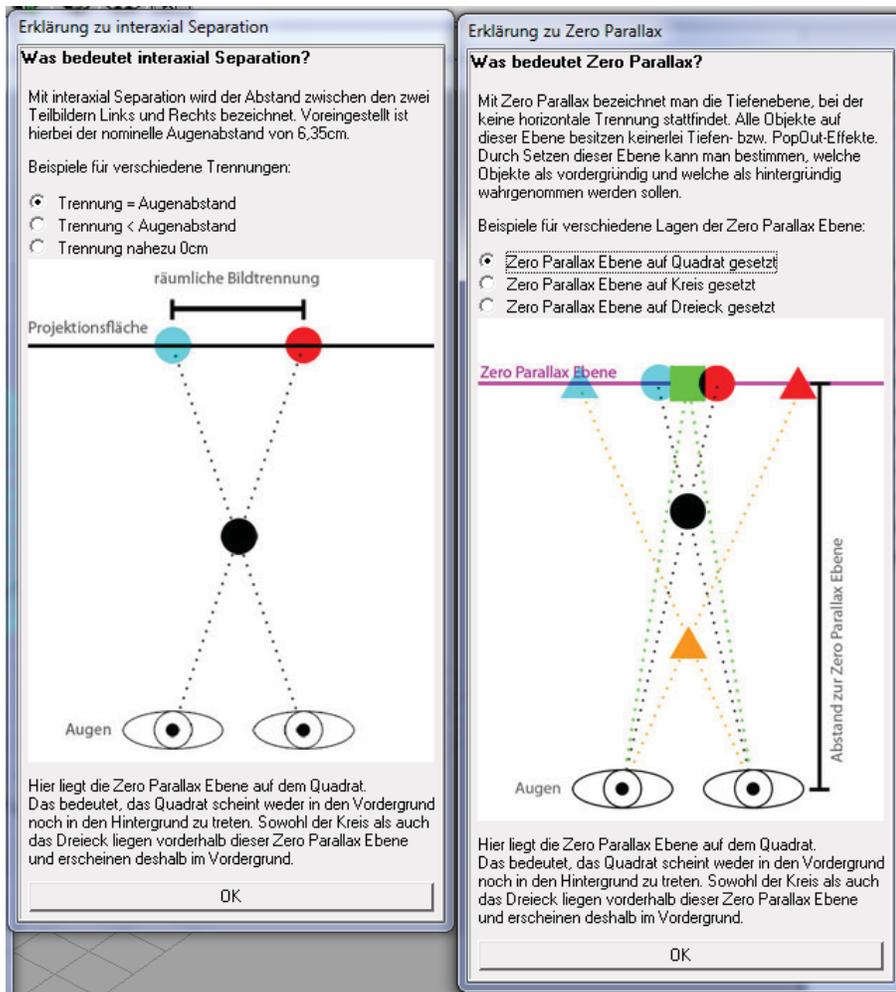


Abbildung 8.2: Alle vorhandenen Funktionalitäten des MEL-Scripts werden hier ersichtlich.

### 8.1.2 Hilfesystem

Hinter jeder Funktionalität befindet sich ein Button mit der Beschriftung eines Fragezeichens. Klickt der Benutzer darauf, erscheint ein neues, separates Fenster mit dem jeweiligen Hilfetext. Diese Texte sind im oberen Bereich der stereoskopischen Parameter durch Radiobuttons und Bilder erweitert, damit dem Anwender visualisiert wird, welche Auswirkungen

verschiedene Einstellungen dieser Parameter haben. Klickt ein Benutzer nach dem Lesen auf OK innerhalb des Hilfetextfensters, so wird dies geschlossen und kann erst dann wieder erneut vom Benutzer über die Hauptoberfläche aufgerufen werden.



**Abbildung 8.3:** Beispielhafte Hilfetexte zu dem implementierten Hilfesystem des MEL-Scripts.

## 8.2 Programmierung der Oberfläche

Die Oberfläche des Scripts besteht zum Hauptteil aus einem Fenster mit dem Titel „Stereo-skopische Konfiguration“. Untergeordnet dazu befinden sich mehrere Layouts, die ineinander verschachtelt angeordnet sind. Das oberste Layout, welches für das ganze Fenster zuständig ist, ist ein Column-Layout. Darin enthalten sind sowohl ein Row-Layout, welches das Headerbild enthält, sowie mehrere Frame-Layouts, in denen sich die ein- und ausklappbaren Bereiche für die vertikale, auf die Funktionalitäten basierende Unterteilung befinden. Jedes

dieser Frame-Layouts enthält je nach Funktionalität weitere Column- und Row-Column-Layouts.

Während ein Column-Layout die Anordnung mehrere Spalten ermöglicht, bietet ein Row-Layout die Möglichkeit, mehrere Zeilen untereinander mit jeweils einem Element zu befüllen. Die Kombination daraus, ein Row-Column-Layout, erlaubt hingegen die Anordnung mehrere Elemente in mehreren Spalten auf einer Zeile und die zusätzliche Aufteilung dieser Elemente auf mehrere Zeilen. Ein Frame-Layout zeichnet sich im Gegensatz dazu aus, dass ein Button mit einem Pfeil zum ein- und ausklappen untergeordneter Bereiche neben dem Titel des Frame-Layouts steht. Dies erlaubt eine variable Oberfläche, bei welcher mehrere Bereiche sichtbar und unsichtbar gemacht werden können.

Das oberste Frame-Layout mit dem Namen „Erzeugung der StereoKamera“ enthält ein Row-Column-Layout, sodass die entsprechenden Checkboxes neben den Eigenschaften zur Erstellung einer stereoskopischen Kamera platziert werden können. Zuletzt befindet sich oberhalb des Frame-Layouts ein Column-Layout mit dem Button zur Erzeugung der stereoskopischen Kamera.

Die zweite große Unterteilung des Fensters mit dem Namen „Erweiterte Einstellungen“ enthält mehrere untergeordnete Row-Column-Layouts, welche alle Buttons und Auswahlmensüs sowie die Hilfebuttons für erweiterte Einstellungen erstellter stereoskopischer Kameras besitzen.

Die letzte große Unterteilung mit dem Namen „Projektionsmaße bearbeiten“ besteht aus einem Auswahlmensü für die erstellten stereoskopischen Kameras, sowie zwei Eingabefeldern für die Eingabe des Abstands zur Projektion sowie die Breite der Projektion.

Der letzte Button im Fenster ermöglicht das Laden bereits vorhandener stereoskopischer Kameras aus der Szene, sofern diese von anderen Scriptinstanzen erstellt wurden.

Bei der Programmierung dieses Fenster wurde darauf geachtet, dass Felder, welche vor der Erstellung einer stereoskopischen Kamera keinen Wert besitzen, deaktiviert sind. Das gleiche gilt für die Hilfebuttons. Wird ein Hilfefenster geöffnet, so ist der jeweilige Button für dieses Fenster deaktiviert, da sonst beliebig viele Fenster mit dem gleichen Inhalt geöffnet werden könnten.

### 8.3 Programmierung der Funktionalitäten

Da dieses Script sehr umfangreich ist und viele zusätzliche Hilfetexte enthält, wird beispielhaft an ausgewählten Funktionalitäten die eigentliche Implementierung derer erklärt.

#### 8.3.1 Erstellung einer stereoskopischen Kamera

Eine der wichtigsten Funktionen dieses Scripts ist die Erstellung einer stereoskopischen Kamera. Hierbei wird auch auf die Optimierung des stereoskopischen Parameters interaxial Separation geachtet. Das Klicken auf den Button „Erzeuge StereoKamera“, ohne dabei den Haken bei interaxial Separation zu setzen, führt zu einer Eingabemaske, bei welcher sowohl der Projektionsabstand als auch die Projektionsbreite eingegeben werden sollen. Gibt der Benutzer diese Daten ein und drückt auf OK, so läuft im Hintergrund die Berechnung der optimalen interaxial Separation mit Bezug auf diese Eingaben und der Zero Parallax ab. Diese

Berechnung basiert dabei ausschließlich auf der Umsetzung der im Kapitel „Mathematische Grundlagen für Optimierungen“ gewählten Formel.

```

global proc createStereoRig ()
{
    //Initialisierung der Variable nummer für Iterationen der folgenden Kameras
    global int $nummer;

    //Python-Code Injection für die Erstellung der StereoKamera
    python "from maya.app.stereo import stereoCameraUtil;stereoCameraUtil.loadPlugin();" ;
    python "from maya.app.stereo import
        stereoCameraRig;stereoCameraRig.createStereoCameraRig(rigName='StereoCamera');" ;
    rename "stereoCamera" ("StereoKamera"+$nummer);

    //Überprüfung, welche Checkboxes aktiviert sind und welche nicht
    if('checkBox -query -value fLCheck'==1)
    {
        setattr ("StereoKamera"+$nummer+"CenterCamShape.focalLength")
            'floatSliderGrp -query -value focalLength';
    }

    if('checkBox -query -value iSepCheck'==1)
    {
        setattr ("StereoKamera"+$nummer+"CenterCamShape.interaxialSeparation")
            'floatSliderGrp -query -value iSeparation';
    }
    //Ist die Checkbox der interaxial Separation nicht aktiviert, wird das Fenster zur
    //Maßeingabe aufgerufen
    else
    {
        window -title "Maßeingabe" -tlb true iscalcwnd;
        columnLayout -adj true;
        floatFieldGrp -label "Abstand zur Projektion: " -el "cm" abstandproj;
        floatFieldGrp -label "Breite der Projektion: " -el "cm" breiteproj;
        // Prozedur zur Berechnung der optimalen interaxial Separation wird
        // aufgerufen sobald OK geklickt wird
        button -label "OK" -command "setOptimalIS";
        setParent ..;
        window -edit -width 300 -height 95 iscalcwnd;
        showWindow iscalcwnd;
    }
    //Übernahme des Abstandes der Zero Parallax Plane, wenn die Checkbox aktiviert ist
    if('checkBox -query -value zParCheck'==1)
    {
        setattr ("StereoKamera"+$nummer+"CenterCamShape.zeroParallax")
            'floatSliderGrp -query -value zPar';
    }

    //Erstellte Kamera wird in die entsprechenden Auswahlmenüs hinzugefügt
    menuitem -p oriCam -label ("StereoKamera"+$nummer);
    menuitem -p srcCam -label ("StereoKamera"+$nummer);
    menuitem -p nTileCam -label ("StereoKamera"+$nummer);
    menuitem -p srcCamOwn -label ("StereoKamera"+$nummer);
    menuitem -p optCam -label ("StereoKamera"+$nummer);
    menuitem -p zeroCam -label ("StereoKamera"+$nummer);

    //Teil der Bearbeitung der Maßeingaben wird aktiviert
    enableEditArea();
}

```

```

proc setOptimalIS()
{
    //Variablendeklaration
    ...
    global int $nummer;
    $nummer--;

    //Bei zu kleinen Angaben der Maße werden diese auf 0.001cm jeweils gesetzt
    if ($abstand[0]<=0.0) $abstand[0]=0.001;
    if ($breite[0]<=0.0) $breite[0]=0.001;

    //Die Prozedur zur Speicherung der Maßeingaben wird aufgerufen
    camStorage($stereocam,$abstand[0],$breite[0],0);
}

```

```

//Die Prozedur für die Erstellung der Expression wird gestartet; siehe unten
setISExpr($stereocam,$abstand[0],$breite[0]);

//Das Fenster zur Maßeingabe wird gelöscht
deleteUI -wnd iscalcwnd;
$nummer++;
}

proc setISExpr(string $stereocam, float $abstand, float $breite)
{
//Expression mit der Berechnung der interaxial Separation wird aufgerufen; Formel
ist dabei 1:1 die Formel aus den Grundlagen der Mathematik
expression -s
($stereocam+"CenterCamShape.interaxialSeparation=2*tan(deg_to_rad(0.75))*"+
$stereocam+"CenterCamShape.zeroParallax*"+($abstand/$breite))-n
($stereocam+"optimalis");
}

```

### 8.3.2 Ersetzen einer vorhandenen Kamera

Für viele Anwender kann es in Frage kommen, bereits fertige Szenen mit stereoskopischen Kameras auszustatten. Damit auch dies ermöglicht wird, kann das Script vorhandene Kameras in der Szene erkennen und ersetzt auf Wunsch die Parameter einer gewählten stereoskopischen Kamera mit denen der ausgewählten monoskopischen Kamera. Ist dieser Vorgang abgeschlossen, kann weiterhin die monoskopische Kamera verwendet werden und die stereoskopische Kamera folgt dieser bei jeder Bewegung.

```

global proc editStereoCam()
{
//Variablendeklaration
...

//Ist der Radiobutton für das Ersetzen einer aktuellen Kamera mit einer StereoKamera
aktiviert, wird folgende Funktion ausgeführt
if('radioCollection -query -sl chooseAdv=="rCamCheck")
{
//Falls bereits vorhanden, werden Expression und Constraint für gewählte
Kamera gelöscht
if('objExists($stereocam+"expr")')
{
delete($stereocam+"expr");
}
if('objExists($stereocam+"constraint")')
{
delete($stereocam+"constraint");
}
//Danach wird eine neue Expression erstellt, welche die Koordinaten der
Zielkamera auf die StereoKamera anwendet
expression -s ($stereocam+.translateX =
"+$destcam+.translateX;\r\n"+$stereocam+.translateY =
"+$destcam+.translateY;\r\n"+$stereocam+.translateZ =
"+$destcam+.translateZ;\r\n\r\n"+$stereocam+.rotateX =
"+$destcam+.rotateX;\r\n"+$stereocam+.rotateY =
"+$destcam+.rotateY;\r\n"+$stereocam+.rotateZ =
"+$destcam+.rotateZ;") -n ($stereocam+"expr");
setAttr($stereocam+.overrideEnabled) 1;
setAttr($stereocam+.overrideDisplayType) 2;
}
//weitere Funktionen
...
}

```

### 8.3.3 Verfolgung eines Objekts der Szene

In manchen Szenen ist es erforderlich, dass eine Kamera Objekte verfolgen kann, um zu simulieren dass die Kamera eine Art Verfolgungsjagd mit dem Objekt betreibt. Um dies zu realisieren, kann der Benutzer eine stereoskopische Kamera und ein sich in der Szene befindliches Objekt aus dem jeweiligen Auswahlmenü wählen. Rechts neben dem Auswahlmenü für das zu verfolgende Objekt befindet sich ein Update-Button, falls nach Ausführen des Scripts weitere Objekte zur Szene hinzugefügt wurden. Bestätigt der Benutzer diese Funktionalität, so wird ein so genanntes Aim-Constraint erstellt. Dieses ist dafür zuständig, dass sich Objekte zu anderen Objekten stets korrekt ausrichten, sofern zusätzliche Parameter dafür eingestellt werden. Möchte der Benutzer nach dieser Aktion ein anderes Objekt verfolgen, so wählt er einfach ein anderes im Auswahlmenü aus und der vorherige Aim-Constraint wird automatisch gelöscht.

```
global proc editStereoCam()
{
    //Variablendeklaration
    ...

    //Ist die Funktion des Verfolgens eines Objekts aktiviert, wird folgendes ausgeführt
    if ('radioCollection -query -sl chooseAdv'=="obTrackCheck")
    {
        //Existieren Expression und Constraint bereits, werden beide gelöscht
        if ('objExists ($srccam+" expr" )')
        {
            delete ($srccam+" expr");
        }
        if ('objExists ($srccam+" constraint" )')
        {
            delete ($srccam+" constraint");
        }

        //Dann wird das ausgewählte Objekt markiert und ein Constraint mit
        //Verbindung zur Kamera hergestellt
        select -r 'optionMenu -query -value trackObj' $srccam;
        aimConstraint -offset 0 -90 0 -weight 1 -aimVector 1 0 0 -upVector 0 1 0
            -worldUpType "vector" -worldUpVector 0 1 0 -name ($srccam+" constraint");
        select -r $srccam;
        setAttr ($srccam+".overrideEnabled") 0;
        setAttr ($srccam+".overrideDisplayType") 0;

        //weitere Funktionen
        ...
    }
}
```

In Verbindung mit der Funktionalität „ZeroParallax auf Objekt anwenden“ kann ein Effekt erreicht werden, der auch heutzutage oftmals im 3D-Kino verwendet wird. Dabei wird ein Objekt mit der Kamera verfolgt und die Zero Parallax Plane bleibt stets auf diesem Objekt, sodass an dieser Stelle dauerhaft der Fokus der Kamera zu sein scheint. Wird diese zusätzliche Option des Setzens der Zero Parallax verwendet, so erstellt das Script eine Expression, welche zu jeder Zeit die Zero Parallax Plane setzt und berechnet den dafür notwendigen Abstand zwischen stereoskopischer Kamera und gewähltem Objekt mittels Pythagoras im dreidimensionalen Raum.

```
global proc editStereoCam()
{
    //Variablendeklaration
    ...

    //Ist die Checkbox für der Setzen der Zero Parallax Plane checked, wird die Prozedur
    setZeroP() aufgerufen
}
```

```

if ('checkBox -query -value zeroCamCheck'==1)
{
    setZeroP ();
}

//weitere Funktionen
...
}

```

```

proc setZeroP ()
{
    //Variablendeklaration , um herauszufinden , welche Kamera und welches Objekt gerade
    //im Auswahlmenü gewählt ist
    string $stereocam = 'optionMenu -query -value zeroCam';
    string $selobj = 'optionMenu -query -value zeroObj';

    //Existiert eine Expression für die Zero Parallax Plane bereits , wird sie gelöscht
    if ('objExists ($stereocam+"ZeroP")')
    {
        delete ($stereocam+"ZeroP");
    }

    //Dann wird eine neue Expression erstellt , welche den Abstand zwischen Kamera und
    //Objekt permanent berechnet
    expression -s ($stereocam+"CenterCamShape.zeroParallax = sqrt(
        pow(("+$stereocam+" . translateX -"+$selobj+" . translateX),2)+
        pow(("+$stereocam+" . translateY -"+$selobj+" . translateY),2)+
        pow(("+$stereocam+" . translateZ -"+$selobj+" . translateZ),2))" ) -n ($stereocam+"ZeroP");
}

```

### 8.3.4 Rendern einer Mehrfachansicht für autostereoskopische Displays

Eine Besonderheit des Scripts ist es, für autostereoskopische Displays die Möglichkeit zu bieten, die entsprechende Anzahl der nötigen Ansichten rendern zu können. Dazu wählt der Benutzer die Funktion „nTile Rig“ und trägt dahinter eine Zahl zwischen 1 und 11 ein. Hierbei bedeutet 1-11 die jeweilige Anzahl der zusätzlichen stereoskopischen Kameras. So können mit Hilfe des Scripts bis zu 24 Ansichten gerendert werden. Alle zusätzlichen Kameras werden ebenfalls in der Szene eingeblendet, um eine bessere Nachvollziehbarkeit der Ansichten dieser Kameras zu ermöglichen. Allerdings sind all diese zusätzlich erstellten Kameras nicht anfass- oder veränderbar. Der stereoskopische Parameter interaxial Separation wird hier bei jeder zusätzlichen Kamera verdoppelt, während die Zero Parallax Plane von der gewählten erste stereoskopischen Kamera übernommen wird. Ändert der Benutzer hiernach die interaxial Separation, so wird automatisch für alle zusätzlichen Kameras die interaxial Separation je nach Ordnung verdoppelt.

```

global proc nTileCreate()
{
    //Variablendeklaration
    ...

    //Für die eingegebene Anzahl werden alle entsprechenden Kameras mittels
    //Python-Injection erstellt und die interaxial Separation und Zero Parallax jeder
    //dieser Kameras erhöht
    for ($i=1; $i<=$nTileSum; $i++)
    {
        python " stereoCameraRig.createStereoCameraRig (rigName='StereoCamera');" ;
        rename " stereoCamera" (" nTileRig"+$ntilecamnumber+$i);
        expression -s ((" nTileRig"+$ntilecamnumber+$i)+" . translateX =
            "+$ntilecam+" . translateX;\r\n"+" nTileRig"+$ntilecamnumber+$i)+
            ". translateY =
            "+$ntilecam+" . translateY;\r\n"+" nTileRig"+$ntilecamnumber+$i)+

```

```

    ". translateZ =
      "+$ntilecam+". translateZ;\r\n\r\n"+(" nTileRig"+$ntilecamnumber+$i)+
    ". rotateX =
      "+$ntilecam+". rotateX;\r\n"+(" nTileRig"+$ntilecamnumber+$i)+". rotateY =
    "+$ntilecam+". rotateY;\r\n"+(" nTileRig"+$ntilecamnumber+$i)+". rotateZ =
      "+$ntilecam+
    ". rotateZ;" -n ((" nTileRig"+$ntilecamnumber+$i)+" expr");
    expression -s
      (" nTileRig"+$ntilecamnumber+$i+" CenterCamShape.interaxialSeparation =
      "+$ntilecam+" CenterCamShape.interaxialSeparation*" +($i+1));
    setAttr ((" nTileRig"+$ntilecamnumber+$i)+". overrideEnabled") 1;
    setAttr ((" nTileRig"+$ntilecamnumber+$i)+". overrideDisplayType") 2;
    expression -s (" nTileRig"+$ntilecamnumber+$i+" CenterCamShape.zeroParallax =
      "+$ntilecam+" CenterCamShape.zeroParallax");
  }
select -r $ntilecam;
}

```

Der Button hinter dieser Option mit dem Namen „Render nTile“ ermöglicht nach dem Erstellen zusätzlicher Kameras das Rendern der nötigen Ansichten. Dafür wird im Hintergrund eine weitere Funktion gestartet, die alle Ansichten rendert, der Reihe nach sortiert und umbenent. Diese gerenderten Bilder befinden sich dann im selben Ordner wie die anderen des Projekts und können mit den entsprechenden Programmen der autostereoskopischen Displays zu einer Datei zusammengefügt werden.

```

proc renderNTile()
{
    //Variablen zur Festlegung des Speicherortes und Erfassen aller Kameras werden
    //deklariert
    string $imageDir = 'workspace -q -fullName';
    string $allcams[] = 'listCameras';
    string $rendercams[];
    int $i=0;

    //Jede Kamera, welche eine RECHTE Kamera repräsentiert, wird aufsteigend der Nummer
    //nach in das Array überführt
    for($cam in $allcams)
    {
        if('gmatch $cam "*Right" ')
        {
            $rendercams[$i]= $cam;
            $i++;
        }
    }
    $i=size($rendercams)*2-1;
    //Erst danach werden alle LINKEN Kameras dem Array zugeführt. Dadurch wird eine
    //korrekte Reihenfolge gewährleistet
    for($cam in $allcams)
    {
        if('gmatch $cam "*Left" ')
        {
            $rendercams[$i]= $cam;
            $i--;
        }
    }

    lookThroughModelPanel persp StereoPanel;
    int $b=1;
    //Nun wird jede vorhandene Kamera im Array aktiviert und hindurchgesehen, sodass
    //beim Rendering auch nur diese berücksichtigt wird
    //So befinden sich am Ende alle Bilder aller Kameras in der richtigen Reihenfolge im
    //Projektunterordner
    for($cam in $rendercams)
    {
        lookThru $cam;
        renderIntoNewWindow render;
        renderWindowSavelImageCallback " renderView" ($imageDir+"/images/Ansicht"+$b)
        "image";
        $b++;
    }
}

```

}

### 8.3.5 Projektionsmaße bearbeiten

Je nach Ausgabegerät und Größe des Präsentationsraumes können sich die Maße der Projektion ändern. Damit auch diese Variabilität im Script Einzug findet, werden bei jeder Erstellung einer stereoskopischen Kamera die eingegebenen Maße in zusätzliche Felder eines Arrays überführt. Wählt ein Anwender dann im Auswahlmenü eine stereoskopische Kamera aus, werden die Parameter geladen.

```

proc updateSizes()
{
    //Die gewählte Kamera aus dem Auswahlmenü wird erfasst
    string $stereocam = 'optionMenu -query -value optCam';

    //Existiert für die gewählte Kamera die entsprechende Expression, wird der Text
    //gesetzt
    if ('objExists ($stereocam+"optimalis")')
    {
        camStorage($stereocam,0.0, 0.0,1);
        text -edit -label "optimale Trennung vorhanden -> wird aktualisiert"
        projsizelbl;
    }
    //Ansonsten werden die Werte in den Feldern bei beiden auf 0.0 gesetzt und der Text
    //entsprechend gesetzt
    else
    {
        floatFieldGrp -edit -value1 0.0 abstandprojadv;
        floatFieldGrp -edit -value1 0.0 breiteprojadv;
        text -edit -label "optimale Trennung nicht gefunden -> wird erstellt"
        projsizelbl;
    }
}

```

Ein zusätzlicher Text unter dem Auswahlmenü zeigt dabei an, ob bereits für die gewählte stereoskopische Kamera optimierte Parameter berechnet wurden oder nicht. In jedem Fall wird nach Eingabe neuer Maße die vorhandene Expression gelöscht und daraufhin neu erstellt.

```

proc camStorage(string $camname, float $value1, float $value2, int $call)
{
    //Variablendeklaration
    ...

    //Je nach Aufrufparameter wird eine andere Funktion aufgerufen
    //Bei 0 werden die Werte einfach in ein Array gespeichert
    //Bei 1 existiert eine optimale Trennung bereits und wird aktualisiert
    //Bei 2 existiert noch keine optimale Trennung und wird neu erstellt
    if ($call==0)
    {
        $camnames[$arraynummer]=$camname;
        $camvalue1[$arraynummer]=$value1;
        $camvalue2[$arraynummer]=$value2;
    }
    if ($call==1)
    {
        for($i=0;$i<size($camnames);$i++)
        {
            if ('gmatch $camname $camnames[$i]')
            {
                floatFieldGrp -edit -value1 $camvalue1[$i] abstandprojadv;
                floatFieldGrp -edit -value1 $camvalue2[$i] breiteprojadv;
            }
        }
    }
}

```

```

if ($call==2)
{
    string $selectedcam = 'optionMenu -query -value optCam';
    float $newcamv1='floatFieldGrp -query -value1 abstandprojadv';
    float $newcamv2='floatFieldGrp -query -value1 breiteprojadv';

    if ($newcamv1<=0.0) $newcamv1=0.001;
    if ($newcamv2<=0.0) $newcamv2=0.001;

    for ($i=0;$i<size($camnames);$i++)
    {
        if ('gmatch $selectedcam $camnames[$i]')
        {
            $camvalue1[$i]=$newcamv1;
            $camvalue2[$i]=$newcamv2;
        }
    }

    if ('objExists ($selectedcam+" optimalis"')')
    {
        delete ($selectedcam+" optimalis");
        confirmDialog -t "optimale Trennung" -m "Optimale Trennung wurde
        aktualisiert!" -b "OK";
    }
    else
    {
        confirmDialog -t "optimale Trennung" -m "Optimale Trennung wurde
        erstellt!" -b "OK";
    }
    setIExpr($selectedcam,$newcamv1,$newcamv2);
}

$arraynummer++;
}

```

### 8.3.6 Laden stereoskopischer Kameras aus der Szene

Um das Script alltagstauglich zu machen, muss eine Funktion implementiert sein, die das Laden bereits vorhandener stereoskopischer Kameras aus älteren Scriptinstanzen ermöglicht. Ansonsten müssten bei jedem neuen Öffnen der Szene alle stereoskopischen Kameras neu erstellt werden. Diese Funktion beruht darauf, dass eine vom Script erstellte stereoskopische Kamera mit speziellen Namen versehen wird. Auch jede einzeln erstellte Expression erhält eindeutig identifizierbare Namen. Drückt der Benutzer den Knopf zum Laden dieser Kameras wird überprüft, ob in der Szene vorhandene Kameras dem Namenmuster des Scripts entsprechen und falls dies der Fall ist, werden alle Auswahlmenüs im Script mit den Namen der stereoskopischen Kamera ergänzt. Somit kann der Benutzer auf Kameras älterer Scriptinstanzen zurückgreifen und diese jederzeit bearbeiten.

```

proc loadcams()
{
    string $existcams[] = 'listCameras';

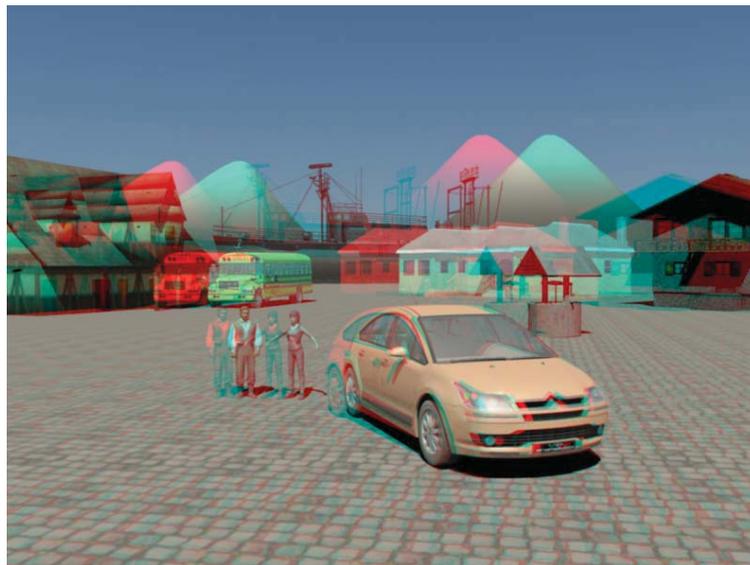
    //Alle Kameras, welche entsprechende Kürzel in den Namen besitzen, werden in alle
    //Auswahlmenüs überführt und aktivieren das Feld des Bearbeitens der
    //Projektionsmaße
    for ($excam in $existcams)
    {
        if ('gmatch $excam "StereoKamera*"')
        {
            if (!'gmatch $excam "*Right"')
            {
                if (!'gmatch $excam "*Left"')
                {

```

```
string $excamsub ='substitute "CenterCamShape" $excam ""';  
print (" alte scriptcam gefunden!");  
menultem -p oriCam -label $excamsub;  
menultem -p srcCam -label $excamsub;  
menultem -p nTileCam -label $excamsub;  
menultem -p srcCamOwn -label $excamsub;  
menultem -p optCam -label $excamsub;  
menultem -p zeroCam -label $excamsub;  
print ($excamsub);  
enableEditArea();  
}  
}
```

### 8.4 Ergebnisse

Die nachfolgende Abbildung zeigt ein optimiertes Ergebnis einer Beispielszene. Bei dieser Abbildung wurde die Optimierung auf eine Breite von 10cm mit einem Betrachterabstand von 50cm optimiert. Bestandteil dieser Beispielszene sind Objekte, welche frei verfügbar von Free 3D Models<sup>1</sup> erhältlich sind. Die gesamte für Autodesk Maya notwendige Szene inklusive eingebautem Script befindet sich ebenfalls auf der im Anhang beigelegten CD.



**Abbildung 8.4:** Darstellung einer optimierten stereoskopischen Szene in Maya. Zum Betrachten bitte die Rot-Cyan-Brille aufsetzen.

---

<sup>1</sup><http://gfx-3d-model.blogspot.com/>

## 8.5 Zusammenfassung und Ausblick

Die Implementierung des Scripts erfolgte strikt nach Vorgabe des Konzepts und soll dem Benutzer eine für die Erstellung stereoskopischer Kameras erleichterte Bedienung gegenüber den herkömmlichen Möglichkeiten in Autodesk Maya bieten. Hierbei war es möglich, alle vorgestellten Funktionalitäten wie erfordert einzubauen und eine fehlerfreie Handhabung des Scripts zu gewährleisten.

Da jedoch in naher Zukunft mit hoher Wahrscheinlichkeit neue Funktionen in nachfolgenden Versionen von Autodesk Maya erscheinen werden, wäre es möglich, das Script auf viele dieser zu erweitern. Zusätzlich dazu wäre es möglich, das Script in Autodesk Maya automatisch starten zu lassen und in die Benutzerleiste oben einzufügen, sodass kein Benutzer mehr das Script mit Hilfe des Scripteditors laden muss.



## Kapitel 9

# Konzept stereoskopischer Automatisierung im InstantPlayer

Während das vorherige Kapitel den Praxisteil für die Optimierung virtueller, stereoskopischer Inhalte innerhalb eines Modellierungswerkzeuges behandelte, folgt nun das Konzept für die Umsetzung des Praxisteils bezogen auf das Echtzeit-Rendering-System InstantPlayer.

Auch hier gibt es nahezu unbegrenzte Möglichkeiten, virtuelle Welten stereoskopisch zu präsentieren. InstantPlayer gibt dazu auch die Möglichkeit, sowohl auf passiven als auch auf aktiven stereoskopischen Systemen aufzubauen. Die Beschränkung dieser wird daher nur vom verwendeten System gestellt. Wie auch beim Konzept des Scripts wird hier auf dieselben Grundlagen der Stereoskopie und Mathematik zurückgegriffen. Als eine Besonderheit ist herauszustellen, dass die stereoskopische Funktionalität innerhalb des InstantPlayers direkt den Typ der „Off-Axis“-Kamera beinhaltet. Somit ist die angezeigte stereoskopische Darstellung, bezogen auf visuelle Fehler stereoskopischer Kamera, bereits optimiert.

Das Konzept für diese Umsetzung muss im Gegensatz zur Umsetzung bei Autodesk Maya allerdings auf andere Aspekte abgestimmt sein. Da der InstantPlayer selbst die virtuelle Umgebung für vorhandene virtuelle Welten bietet, bietet sich eine Visualisierung, welche auf einem Menü basiert, eher an, als eine zusätzliche, externe Oberfläche zum eigentlichen Programm. Daher wird dieses Menü eine direkte Interaktivität mit Objekten der Szene ermöglichen und bietet somit die direkte Einschätzung der eingestellten Parameter.

Das Produkt dieser Umsetzung wird eine X3D-Szene sein, welche zusätzlich andere X3D-Szenen laden kann, jedoch deren Render-Parameter überschreibt. Zusätzlich wird es eine Beispielszene geben, welche mit der Hauptszene geladen werden kann, um das stereoskopische Menü zu testen.

### 9.1 Beschreibung des Menüaufbaus

Das zu implementierende Menü soll sich innerhalb der Hauptszene befinden und dem Benutzer alle nötigen Konfigurationen zum Einstellen der stereoskopischen Parameter ermöglichen. Dieses Menü soll in verschiedene Tabs unterteilt werden. Diese Tabs sollen dabei in die Teile des Darstellungsformates und der stereoskopischen Konfiguration unterteilt werden. Im

Bereich des Darstellungsformats gibt es ein weiteres Tab, welches die Unterscheidung in passive und aktive Darstellung der stereoskopischen Visualisierung bietet. Die stereoskopische Konfiguration hingegen wird in zwei weitere Tabs unterteilt werden: Die des Bearbeitens der stereoskopischen Parameter und die der Optimierung der stereoskopischen Parameter.

### 9.2 Funktionalitäten des Menüs

Das Menü für das Einstellen der stereoskopischen Parameter soll die folgenden Funktionalitäten bieten:

- Darstellungsformat (Tab)
  - Einstellen von passivem und aktivem Anzeigeformat
    - \* passiv (Tab): rot-cyan / grün-magenta / blau-gelb
    - \* aktiv (Tab): aktivieren und deaktivieren des Anzeigeformat
  - Abstellen des stereoskopischen Effekts
- stereoskopische Konfiguration (Tab)
  - Optimierung (Tab)
    - \* Eingabe von Größe und Betrachtungsabstand der Projektion
    - \* beide Optionen als Slider realisiert und zusätzlich als Textbox
- Bearbeitung (Tab)
  - Setzen des Abstandes der Zero Parallax Plane mit Slider und Textbox
  - prozentuale Anpassung der interaxial Separation mit Slider und Textbox
- Umkehrung des stereoskopischen Effekts (Tauschen von Links und Rechts)

Diese Funktionalitäten richten sich nach den Einstellungen, welche nötig sind, um eine optimierte, stereoskopische Visualisierung zu ermöglichen. Mit den Einstellmöglichkeiten unter dem Reiter Darstellungsformat kann der Anwender wählen, ob eine passive oder aktive Betrachtungsweise bevorzugt wird. Während für eine passive Betrachtungsweise lediglich eine Brille im beschriebenen Farbformat nötig ist, braucht die aktive Betrachtungsweise bestimmte Hardware zur Projektion. Somit ist diese Art der Betrachtung nur mit entsprechender Ausstattung möglich. Im Bereich der stereoskopischen Konfiguration befindet sich der Benutzer zunächst im Teil, wo die Projektionsbreite und Abstand zur Projektion angegeben werden sollen. Danach kann im Teil der Bearbeitung die Zero Parallax Plane verschoben werden und die interaxial Separation notfalls angepasst werden. Zuletzt kann dort auch die Umkehrung des stereoskopischen Effektes erfolgen, falls das projizierende System ein Vertauschen von Links und Rechts erfordert.

Zur Erweiterung dieses Menüs soll ein externes Webinterface bereitgestellt werden, sodass stereoskopische Parameter von einem anderen Gerät mit Zugriff auf das projizierende System eingestellt werden können.

### **9.3 Anforderung an die Szene**

Die Hauptszene mit enthaltenem Menü soll dem Benutzer eine einfache Navigation und ein einfaches Einstellen der stereoskopischen Parameter ermöglichen. Die Beispielszene, welche integriert werden soll, soll dabei ohne Schwierigkeiten navigierbar sein und einen ersten Eindruck einer stereoskopischen Projektion mit optimierten Parametern geben.

### **9.4 Zusammenfassung**

Die Realisierung dieses Konzepts erfordert, trotz einfachen Aufbaus, ein hohes Maß an die Programmierfähigkeit virtueller Welten. Ist dieses Konzept allerdings vollständig umgesetzt, kann ein Benutzer ohne große Hürden in die Stereoskopie im Bereich von Echtzeit-Rendering-Systemen einsteigen und lernt dabei mit verschiedenen Parametern umzugehen. Das Konzept bietet hier die Möglichkeit, zunächst alle nötigen Parameter automatisiert zu präsentieren, was einen erleichterten Einstieg ermöglichen soll.



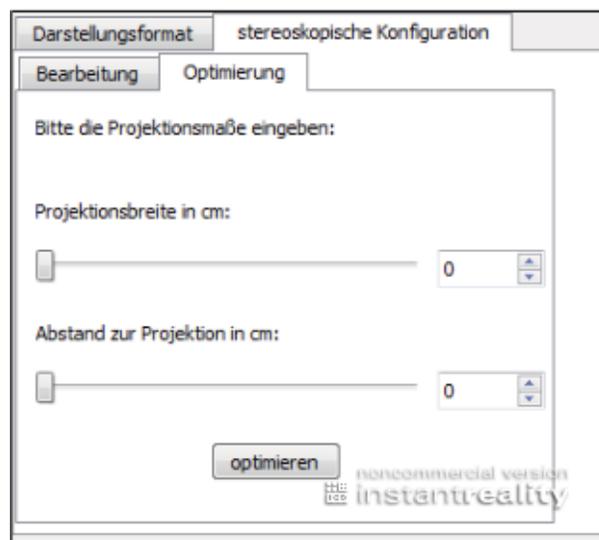
## Kapitel 10

# Implementierung der X3D-Szene und des dazugehörigen Menüs

Dieses Kapitel behandelt die technische Umsetzung des Konzepts vom vorherigen Kapitel. Dabei findet die gesamte Implementierung mit dem freien Texteditor Notepad++ und dem nötigen Designer Qt Creator statt. Mit Hilfe dieses Designers kann das Menü gestaltet werden, welches später in der virtuellen, stereoskopischen Szene angezeigt wird.

### 10.1 Oberfläche des Menüs

Das fertig impementierte Menü sieht nach der Erstellung wie folgt aus:

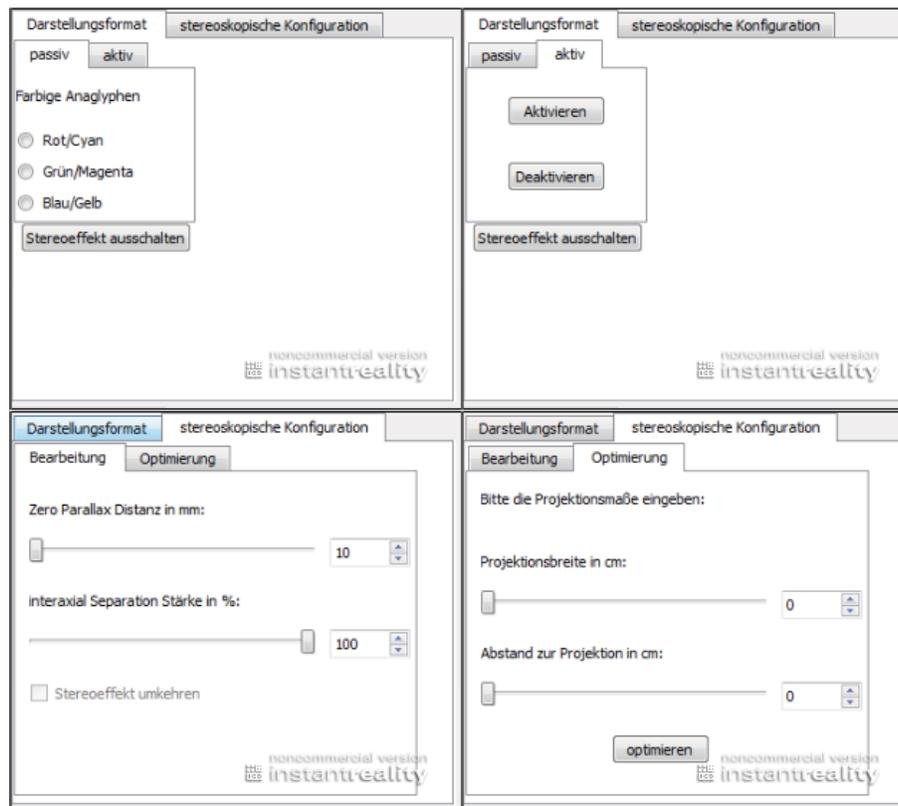


**Abbildung 10.1:** Übersicht über die Oberfläche des implementierten Menüs innerhalb der X3D-Szene.

Die Gestaltung des Menüs fand strikt nach dem Konzept statt. Dazu wurde auf das Programm Qt Creator, welches von Nokia entwickelt wurde, gesetzt. Damit ist es möglich, User Interface Dateien zu erstellen, welche wiederum vom InstantPlayer in X3D als Textur verwendet werden können. Zur verbesserten Interaktivität besitzt jede Einstellung eines stereoskopischen Parameters einen Slider mit angehangener Textbox, damit ein stufenloses Einstellen dieser Parameter ermöglicht wird, sodass sich das Auge stets an die neuen Werte anpassen kann. Im Gegensatz zum MEL Script besitzt dieses Menü auch die Möglichkeit, die interaxial Separation, welche aus der Optimierungsformel resultiert, zu justieren. Dies ist für den Fall gedacht, dass Benutzer mit stereoskopischen Projektionen nicht allzu vertraut sind und zuerst kleinere Trennungen der stereoskopischen Sicht bevorzugen.

## 10.2 Aussehen der Funktionalitäten im Menü

Alle Funktionalitäten wurden nach dem Konzept implementiert und angeordnet. Nachfolgend werden die einzelnen Tabs dargestellt.



**Abbildung 10.2:** Darstellung aller einzelnen Funktionalitäten des implementierten Menüs.

Hier wird ersichtlich, dass alle Funktionen inklusive der Slider und Textboxen ebenfalls nach dem Konzept entwickelt wurden. Dadurch hat der Benutzer nun die Möglichkeit, alle Para-

meter durch simple Mausaktionen abzuändern und direkt in der dargestellten X3D-Szene zu evaluieren.

## 10.3 Aussehen des Webinterfaces

Wie zu sehen ist, besitzt das Webinterface die gleichen Funktionalitäten wie das Menü. Allerdings kann dieses Interface auch von einem externen Rechner gestartet und somit die X3D-Szene fern eingestellt werden.

### Stereoskopische Konfiguration

#### Optimierung:

Breite der Projektion:

cm Wert:

Betrachterabstand zur Projektion:

cm Wert:

Optimieren

#### Bearbeitung:

Zero Parallax Distance:

mm Wert:

interaxial Separation Prozentsatz:

% Wert:

Stereoeffekt umkehren

#### Darstellungsformat:

passiv

aktiv

Farbige Anaglyphen:

- Rot/Cyan
- Grün/Magenta
- Blau/Gelb
- Aus

aktivieren

deaktivieren

Starten

**Abbildung 10.3:** Darstellung aller einzelnen Funktionalitäten des entwickelten Webinterfaces.

## 10.4 Struktur der X3D-Szene

Die Struktur der zu visualisierenden X3D-Szene ist essentiell für die Interaktivität zwischen Benutzer und Szene. Dabei ist wichtig, dass alle nötigen Variablen deklariert und benannt werden, sodass ein eindeutiger Zugriff darauf ermöglicht werden kann. Da es sich bei dieser Szene um eine stereoskopische Visualisierung handelt, besitzt sie eine spezielle Form des Engine-Elementes, welche die für die Darstellung der Szene zuständige Render-Engine steuert. Dieses Engine-Element wird hierbei im obersten Teil der X3D-Szene definiert. Das für die Bachelorarbeit notwendige Engine-Element sieht dabei wie folgt aus:

```
<Engine DEF='engine' >
  <RenderJob DEF='render' >
    <WindowGroup>
      <Window DEF='mainwindow' size='1000 1000' buffer='4' >
        <Viewarea DEF='view1' view='normalview' leftBuffer='true' rightBuffer='true' >
          <ShearedStereoViewModifier DEF='stereoL' leftEye='true' rightEye='false'
            eyeSeparation='1' zeroParallaxDistance='1' />
        </Viewarea>
        <Viewarea DEF='view2' view='normalview' leftBuffer='true' rightBuffer='true' >
          <ShearedStereoViewModifier DEF='stereoR' leftEye='false' rightEye='true'
            eyeSeparation='1' zeroParallaxDistance='1' />
        </Viewarea>
        <Viewarea lowerLeft='0.8375 0.75' upperRight='1 1' view='hudview' />
      </Window>
    </WindowGroup>
  </RenderJob>
</Engine>
```

Zunächst wird das Engine-Element mit einem Namen versehen. Dieser wird später notwendig sein, um aus dem zweiten großen Teil der X3D-Szene, des Scene-Elementes, Zugriff zu gewährleisten. Innerhalb dieses Engine-Elementes befindet sich das Window-Element. Dieses hält ein wichtiges Attribut, welches für die aktive, stereoskopische Darstellung der Szene notwendig ist. Dieses Attribut mit dem Namen `buffer` enthält die Anzahl der für die Visualisierung verwendeten Buffer-Einheiten, welche von der Grafikkarte bereitgestellt werden. Da die aktive, stereoskopische Darstellung mittels einer Quadbuffered OpenGL fähigen Grafikkarte angezeigt wird, muss dieses Attribut auf 4 gesetzt werden. Innerhalb dieses Window-Elementes sind zwei so genannte `Viewareas` mit jeweils einem so genannten `ShearedStereoViewModifier` vorhanden. Nur in Kombination von `Viewarea` mit `ShearedStereoViewModifier` bietet sich die Möglichkeit an, stereoskopische Parameter der X3D-Szene zu bearbeiten. Da für die stereoskopische Darstellung zwei verschiedene Ansichten nötig sind, befinden sich zwei unterschiedliche `Viewareas` innerhalb des Window-Elements. Zusammengefasst steuern diese beiden `Viewareas` alle nötigen Eigenschaften zur Optimierung der stereoskopischen Visualisierung. Je nach Parameter muss hierbei entweder auf den in einer `Viewarea` enthaltenen `ShearedStereoViewModifier` oder auf die `Viewarea` selbst zugegriffen werden.

Weiterhin befindet sich innerhalb des Window-Elements eine weitere `Viewarea`. Diese ist später für die Darstellung des Menüs, separiert von der stereoskopischen Darstellung, in der oberen rechten Ecke des Fensters zuständig.

Alle definierten `Viewareas` enthalten das Attribut `view`. Damit kann festgelegt werden, welche Ansichten durch diese `Viewarea` später repräsentiert werden. Da die Darstellung des stereoskopischen Teils der Szene von dem Menü unabhängig sein soll, besitzen die `Viewareas` verschiedene Attribute `view`. Damit definiert eine `Viewarea` lediglich die Art und Weise,

wie eine Ansicht später dargestellt werden soll. Die eigentliche Ansicht wird von dem Attribut `view`, welches auf einen so genannten `Viewpoint` verweist, bereitgestellt. Nach dieser Definition folgt die eigentliche Szene mittels dem Element `<Scene>...</Scene>`, in der sich alle weiteren Elemente befinden:

```

<!-- vorherige Deklaration der Render-Engine-Elemente -->
...
<Scene>
<!-- Deklaration beider nötiger Viewpoints für Menü und stereoskopische Darstellung -->
<OrthoViewpoint DEF='hudview' centerOfRotation='0 0 0' orientation='0 0 1 0' position='500
500 0' />
<Viewpoint DEF='normalview' zNear='1' zFar='10000' retainUserOffsets='false' />

<!-- Deklaration des Prototypen für das HeadsUpDisplay -->
<ExternProtoDeclare name='HeadsUpDisplay' url='' HeadsUpDisplayPrototype.x3d#HeadsUpDisplay' >
  <field accessType='inputOutput' name='children' type='MFNode' />
  <field accessType='inputOutput' name='screenOffset' type='SFVec3f' />
  <field accessType='outputOnly' name='position_changed' type='SFVec3f' />
  <field accessType='outputOnly' name='orientation_changed' type='SFRotation' />
</ExternProtoDeclare>

<!-- Festlegung der Elemente des HeadsUpDisplays -->
<ProtoInstance DEF='HeadsUpDisplay' name='HeadsUpDisplay'>
  <fieldValue name='screenOffset' value='0 0 -1' />
  <fieldValue name='children' >
    <Transform DEF='hudscale' scale='0.65 0.65 0.65' >
      <Shape>
        <Appearance>
          <UITexture DEF='stereowidget' minFilter='linear'
            magFilter='linear' url='widget.ui'
            updateMode='always' alpha='1' >
            <!-- Feldelemente für Zugriff auf die Textur
            ..
            -->
          </UITexture>
          <TextureTransform scale='1 -1' />
        </Appearance>
        <Rectangle2D size='3.45 3.05' />
      </Shape>
      <TouchSensor DEF='hudSensor' />
    </Transform>
  </fieldValue>
</ProtoInstance>

<!-- Einbettung der externen Beispielszene -->
<Inline url='' Beispielszene.x3d' />

<Script DEF='ConScript'>
  // ECMAScript Funktionen
  /*
  ...
  */
</Script>
<!-- Deklaration der Routes
... -->
</Scene>

```

Dies umfasst alle darstellbaren Elemente, die Viewpoints, die Verbindungen zwischen Elementen und das enthaltene ECMAScript. Während diese Szene zunächst das darzustellende Menü enthält, befinden sich in den darauffolgenden Teilen der Szene die Funktionen für das ECMAScript und die so genannten Routes. Die definierten Viewpoints im Element `<Scene>...</Scene>` repräsentieren die nötigen Ansichten für Menü und stereoskopische Darstellung. Da sich das Menü ebenfalls innerhalb dieser Szene befindet, allerdings von der stereoskopischen Darstellung getrennt sein soll, besitzt der zuständige Viewpoint einfachheitshalber eine Position, welche außerhalb der sonstigen angezeigten Elemente liegt. Die Struktur dieser Szene mit enthaltenem Menü ist nachfolgend abgebildet: Das implemen-

tierte Menü soll dem Benutzer stets zweidimensional erscheinen und nicht von Navigation beeinflusst werden. Dazu wird ein externer Prototyp, welcher die Grundfunktionen eines Heads-Up-Displays bereitstellt, verwendet. Dieser Prototyp simuliert diese Funktionalität allerdings nur, indem sich beim Bewegen im Raum das Menü stets an den selben Abstand zur Kamera positioniert. Dadurch hat der Benutzer das Gefühl, dass sich das eingebettete Menü stets am gleichen Punkt innerhalb der X3D-Szene befindet.

## 10.5 Implementierung der Menüfunktionalitäten

Damit die jeweiligen Menüpunkte auch im Bereich der X3D Szene genutzt werden können, müssen Verbindungen zwischen den Elementen des Menüs und der X3D Szene hergestellt werden. Dazu muss zunächst die Textur eines Shapes innerhalb der Szene auf die extern gestaltete Menüdatei verweisen. Zum besseren Verständnis wird nachfolgend die Implementierung der Verbindung zwischen dem Menü als Textur und der Interaktion des Benutzers beispielhaft an den Radio Buttons für die passive Darstellungsweise demonstriert:

```

<!-- vorherige Erstellung von X3D-Elementen -->
...
<Shape>
<Appearance>
  <UITexture DEF='stereowidget' minFilter='linear' magFilter='linear' url='widget.ui'
    updateMode='always' alpha='1'>
    <!-- Anaglyphen Radio Buttons -->
    <field accessType='outputOnly' name='redcyan/toggled' type='SFBool' />
    <field accessType='outputOnly' name='greenmagenta/toggled' type='SFBool' />
    <field accessType='outputOnly' name='blueyellow/toggled' type='SFBool' />
    ...
  </UITexture>
  <TextureTransform scale='1 -1' />
</Appearance>
<Rectangle2D size='3.45 3.05' />
</Shape>
...
<!-- nachfolgende Festlegung von X3D-Elementen -->

```

Der Verweis der Form auf die externe Menüdatei befindet sich in `<UITexture url='widget.ui' />`. Zur Interaktion zwischen Textur Elementen und der X3D-Szene besitzt die Textur einen definierten Namen und enthält für jedes einzelne, auf dem Menü befindliche Element einen Verweis des Typs `<field ... />`. Dadurch wird ermöglicht, dass Elemente der X3D-Szene auf diese Menüelemente zugreifen können. Jedes Feld des Typs `<field ... />` muss dabei den selben Namen wie das entsprechende Element auf dem Menü besitzen und deren Funktion, auf welche zugegriffen werden möchte, implementieren. Zusätzlich dazu muss definiert sein, auf welche Art Zugriff zu diesem Element gestattet wird. Während `accessType='outputOnly'` ausschließlich Werte vom referenzierten Element ausliest, übergibt `accessType='outputOnly'` Werte. Der Typ `accessType='inputOutput'` kann sowohl Werte annehmen als auch abgeben. Zuletzt muss in diesem Feld noch definiert werden, von welchem Typ der übertragende Wert ist. Im dargestellten Ausschnitt soll dabei abgerufen werden, ob die Radio Buttons für die passive Darstellungsweise betätigt werden. Damit das Menü auch im virtuellen Raum als ein Menü erkennbar ist, wurde es auf eine rechteckige Form des Typs `<Rectangle2D />` gesetzt.

Für eine Interaktion zwischen der Maus und dem Menü ist hier noch erforderlich, dass ein entsprechender Sensor innerhalb des Menüs eine Verbindung zwischen Maus und Menü in

der X3D-Szene herstellt. Bei diesem Sensor handelt es sich um einen TouchSensor. Dieser ermöglicht das Abrufen von Koordinaten auf dem Menü, sobald der Mauszeiger darüber fährt.

```
<!-- vorherige Erstellung von X3D-Elementen -->
...
<Transform DEF="hudscale" scale='0.65 0.65 0.65' >
  <Shape>
    <Appearance>
      <UITexture DEF='stereowidget' minFilter='linear' magFilter='linear'
        url='widget.ui' updateMode='always' alpha='1'>
        ...
      </UITexture>
      <TextureTransform scale='1 -1' />
    </Appearance>
    <Rectangle2D size='3.45 3.05' />
  </Shape>
  <TouchSensor DEF="hudSensor" />
</Transform>
```

Dieser Sensor mit dem Namen hudSensor benötigt zusätzlich dazu noch so genannte Routes, welche Verbindungen zwischen dem Sensor und dem Menü für die X3D-Szene ermöglichen.

```
...
<ROUTE fromNode='hudSensor' fromField='hitTexCoord_changed' toNode='stereowidget'
  toField='set_pointer' />
<ROUTE fromNode='hudSensor' fromField='isActive' toNode='stereowidget'
  toField='set_button1' />
...
```

Während die erste Route die Koordinaten der Maus über dem Menü abrufen, ist die zweite Route dafür notwendig, abzufragen, ob auf dem Menü geklickt wurde. Ohne diese Routes wird der Sensor nicht aktiv und mit der angezeigten Textur auf dem Rechteck kann nicht interagiert werden.

Im gezeigten Beispiel soll bei Klick auf eine der Radio Button Elemente des Menüs die entsprechende Darstellung aktiviert werden. Somit sind auch hier Routes notwendig. Allerdings diesmal als Verbindung zwischen der X3D-Szene und dem ECMAScript. Die Routes für das Beispiel sind wie folgt definiert:

```
<!-- Darstellungsformate -->
<ROUTE fromNode='stereowidget' fromField='redcyan/toggled' toNode='ConScript'
  toField='redcyan' />
<ROUTE fromNode='stereowidget' fromField='greenmagenta/toggled' toNode='ConScript'
  toField='greenmagenta' />
<ROUTE fromNode='stereowidget' fromField='blueyellow/toggled' toNode='ConScript'
  toField='blueyellow' />
```

Dabei greifen die Routes sowohl auf Felder der eingebundenen Textur als auch auf Felder des ECMAScripts zu. Die Verbindung läuft dabei von dem Feld der Textur zum Feld des ECMAScripts. Hierdurch ist der Weg von der X3D-Szene zum Script bereits definiert. Damit eine Reaktion des Scripts erfolgen kann, müssen auch dort die entsprechenden Felder mit Funktionsverweisen definiert werden. Im nachfolgenden Code wird diese Funktionalität beispielhaft gezeigt:

```
<Script DEF="ConScript">
  <!-- Deklaration der nötigen Felder zum Zugriff der X3D-Szene auf das Script -->
  <field name="redcyan" accessType="inputOnly" type="SFBool"/>
  <field name="greenmagenta" accessType="inputOnly" type="SFBool"/>
  <field name="blueyellow" accessType="inputOnly" type="SFBool"/>
  <![CDATA[ecmascript:
    //Deklaration der ShearedStereoViewModifier und ViewAreas
```

```

var view1 = Browser.currentScene.getNamedNode('engine::view1');
var view2 = Browser.currentScene.getNamedNode('engine::view2');
var stereoview1 = Browser.currentScene.getNamedNode('engine::stereoL');
var stereoview2 = Browser.currentScene.getNamedNode('engine::stereoR');

// Anaglyphen an/aus
function redcyan(value) { if(value){ Browser.println("redcyan activated!");
    setAnaglyphs(true, false, false, false, true, true);}}
function greenmagenta(value) { if(value){ Browser.println("greenmagenta
    activated!"); setAnaglyphs(false, true, false, true, false, true);}}
function blueyellow(value) { if(value){ Browser.println("blueyellow activated!");
    setAnaglyphs(true, true, false, false, false, true);}}

// Funktion zum Resetten der Anaglyphen
function setNormal(){
    setAnaglyphs(true, true, true, true, true, true);
    activateinvert=false;
    if (inverted)
    {
        checkinvert=false;
    }
}

//globale Anaglyphenfunktion
function
    setAnaglyphs(boolleft1, boolleft2, boolleft3, boolright1, boolright2, boolright3){
    view1.red=boolleft1; view1.green=boolleft2; view1.blue=boolleft3;
    view2.red=boolright1; view2.green=boolright2; view2.blue=boolright3;
    activateinvert=true;
}
}}>
</Script>

```

Der Zugriff von diesem Script auf ein Element innerhalb der Szene geschieht hierbei über die Anweisung `Browser.currentScene.getNamedNode('...')`. Da dieses Script jedoch auf den Bereich `<Engine>...</Engine>` zugreifen muss, muss der zusätzliche Namespaceverweis `'engine::DEF'` verwendet werden. Somit kann gewährleistet werden, dass bei jedem Klick eines Radio Buttons die tatsächliche Eigenschaft innerhalb des Engine-Unterelementes geändert werden kann.

Erst mit diesen ganzen Implementierungen ist eine einfache Interaktion zwischen Benutzer und dem eingebauten Menü möglich.

Im nachfolgenden Teil wird der gesamte Prozess von der Benutzeraktion zum Ändern eines stereoskopischen Parameters dargestellt: Klickt der Benutzer auf den Radio Button Rot/Cyan, so erhält das Feld `'redcyan/toggled'` in der X3D-Szene den Wert `TRUE`. Die Route `<ROUTE fromNode='stereowidget' fromField='redcyan/toggled' toNode='ConScript' toField='redcyan' />` leitet permanent den Wert dieses Feldes in das Feld „redcyan“ im ECMAScript mit dem Namen „ConScript“ um. In diesem Script muss somit dieses Feld, welches den Wert erhalten soll, ebenfalls definiert sein: `<field name="redcyan" accessType="inputOnly" type="SFBool"/>`. Im ECMAScript repräsentiert der Feldname zugleich eine Funktion, welche ebenfalls permanent aufgerufen wird.

```

function redcyan(value)
{
    if (value)
    {
        Browser.println("redcyan activated!");
        setAnaglyphs(true, false, false, false, true, true);
    }
}

```

Diese wiederum besitzt eine if-Anweisung, welche entscheidet, welche Aktion beim Aufruf ausgeführt werden soll. Der eigentliche Wert des Feldes „redcyan“ wird mit der Variable „value“ übertragen. Ist dieser Wert FALSE, wird permanent die Funktion verlassen und keine Aktion ausgeführt. Ist der Wert jedoch TRUE, so gibt es eine Ausgabe in der Konsole und die Funktion „setAnaglyphs“ mit den Parametern zur Angabe der Farbe der Anaglyphen wird aufgerufen. Diese aufgerufene Funktion enthält alle nötigen Anweisungen, um die stereoskopischen Parameter im Engine-Element festzulegen:

```
var view1 = Browser.currentScene.getNamedNode('engine::view1');
var view2 = Browser.currentScene.getNamedNode('engine::view2');

function setAnaglyphs( boolleft1, boolleft2, boolleft3, boolright1, boolright2, boolright3){
    view1.red=boolleft1; view1.green=boolleft2; view1.blue=boolleft3;
    view2.red=boolright1; view2.green=boolright2; view2.blue=boolright3;
}
```

## 10.6 Implementierung der Optimierung stereoskopischer Parameter

Die Optimierung der stereoskopischen Parameter geschieht über die gleichen Wege, welche bereits für die Menüfunktionalitäten notwendig waren. Der Unterschied hierbei ist allerdings, dass, sobald ein Benutzer die Projektionsmaße eingegeben hat, die Optimierung dieser Parameter vollautomatisch geschieht und lediglich der Parameter der Zero Parallax Plane angepasst werden muss. Zusätzlich dazu gibt es die Möglichkeit, dass die interaxial Separation nur zu einem gewählten Prozentsatz aus der Optimierungsformel übernommen wird. Dies bietet sich vor allem für Benutzer an, welche weniger Erfahrungen mit der Stereoskopie haben und bei größere Trennungen zwischen linker und rechter Perspektiven auch schneller zu Erscheinungen wie Kopfschmerzen neigen.

Die dazu notwendigen Parameter befinden sich, wie bereits erläutert, im obersten Teil der X3D-Szene, also im Engine-Element:

```
<Viewarea DEF='view1' view='normalview' leftBuffer='true' rightBuffer='true'>
  <ShearedStereoViewModifier DEF='stereoL' leftEye='true' rightEye='false' eyeSeparation='1'
    zeroParallaxDistance='1' />
</Viewarea>
<Viewarea DEF='view2' view='normalview' leftBuffer='true' rightBuffer='true'>
  <ShearedStereoViewModifier DEF='stereoR' leftEye='false' rightEye='true' eyeSeparation='1'
    zeroParallaxDistance='1' />
</Viewarea>
```

Für eine optimale Anzeige der stereoskopischen Szene müssen die Parameter `eyeSeparation` und `zeroParallaxDistance` in den Elementen `<ShearedStereoViewModifier .../>` auf die Projektionsmaße angepasst werden. Die Eingabe dieser Maße erfolgt im Reiter „Optimierung“ des oberen Reiters „stereoskopische Konfiguration“. Dort werden entweder über den Slider oder über die Textbox beide Werte eingegeben und danach auf „Optimieren“ geklickt.

Innerhalb der X3D-Szene werden diese Maße über die Rechteckform, welche auf die externe Menüdatei verweist, in die Felder übergeben. Diese Felder sind dabei wie folgt deklariert:

```
<!-- vorheriger X3D-Szenen Code -->
```

```

<Shape>
<Appearance>
  <UITexture DEF='stereowidget' minFilter='linear' magFilter='linear' url='widget.ui'
    updateMode='always' alpha='1'>
    <!-- Deklaration vorheriger Felder der Textur -->
    <!-- Stereo Optimierung -->
    <field accessType='outputOnly' name='setoptimize' type='SFBool' />
    <field accessType='inputOutput' name='projectionwidthslider' type='SFInt32'
      value='0' />
    <field accessType='inputOutput' name='projectiondistanceslider'
      type='SFInt32' value='0' />
    <field accessType='inputOutput' name='projectionwidthbox' type='SFInt32'
      value='0' />
    <field accessType='inputOutput' name='projectiondistancebox' type='SFInt32'
      value='0' />
  </UITexture>
  <TextureTransform scale='1 -1' />
</Appearance>
<Rectangle2D size='3.45 3.05' />
</Shape>
<!-- nachfolgender X3D-Szenen-Code -->

```

Damit die Möglichkeit der Eingabe sowohl über Slider als auch über Textbox gegeben ist, müssen alle jeweiligen Elemente als Felder repräsentiert werden. Im weiteren Teil der X3D-Szene kommen die Routes zum Einsatz:

```

<!-- vorheriger X3D-Szenen Code -->
<!-- Stereoskopische Optimierung -->
<ROUTE fromNode='stereowidget' fromField='setoptimize' toNode='ConScript'
  toField='setoptimization' />
<ROUTE fromNode='stereowidget' fromField='projectionwidthslider_changed' toNode='ConScript'
  toField='projectionwidth' />
<ROUTE fromNode='stereowidget' fromField='projectionwidthslider_changed'
  toNode='stereowidget' toField='set_projectionwidthbox' />
<ROUTE fromNode='stereowidget' fromField='projectionwidthbox' toNode='stereowidget'
  toField='set_projectionwidthslider' />
<ROUTE fromNode='stereowidget' fromField='projectiondistanceslider_changed'
  toNode='ConScript' toField='projectiondistance' />
<ROUTE fromNode='stereowidget' fromField='projectiondistanceslider_changed'
  toNode='stereowidget' toField='set_projectiondistancebox' />
<ROUTE fromNode='stereowidget' fromField='projectiondistancebox' toNode='stereowidget'
  toField='set_projectiondistanceslider' />
<!-- nachfolgender X3D-Szenen-Code -->

```

Hier werden verschiedene Verbindungen definiert. Zunächst befindet sich dort die Weiterleitung des Drückens des Knopfes zum ECMAScript. Die danach folgenden Routes verbinden sowohl die Slider direkt zu den Textboxen, als auch umgekehrt und ebenfalls die Slider zum ECMAScript. Diese mehrfachen Verbindungen sind deshalb nötig, damit auch während des Schiebens eines Slider's die Textbox ihren Wert übernimmt und auch beim Editieren der Textbox dieser Wert auf dem Slider repräsentiert werden kann. Das Kürzel „changed“ hinter den Feldern der Slider greift dabei auf die Eigenschaft der Slider zu, welche den aktuellen Wert übergeben, sobald sich dieser vom vorherigen Stand geändert hat. Bei den Attributen „toField“ hingegen ist es notwendig das Kürzel „set“ zu verwenden, sodass die Werte auf die danach folgende Definition gesetzt werden können.

Haben diese Routes die Werte weitergeleitet und es wurde der Knopf „Optimieren“ gedrückt, kommt das ECMAScript zum Einsatz:

```

<Script DEF="ConScript">
  //... vorheriger Script-Code
  <field name="setoptimization" accessType="inputOnly" type="SFBool" />
  <field name="projectionwidth" accessType="inputOutput" type="SFInt32" value="0" />
  <field name="projectiondistance" accessType="inputOutput" type="SFInt32" value="0" />
  <![CDATA[ecmascript:

```

```

/*
    mehrere Funktionen und Felddefinition
*/
var stereoview1 = Browser.currentScene.getNamedNode('engine::stereoL');
var stereoview2 = Browser.currentScene.getNamedNode('engine::stereoR');
var projectionwidth=0;
var projectiondistance=0;

// Optimierung setzen
function setoptimization(value) {
    var optis =
        (Math.tan(1.5*Math.PI/180)*zeroparallax*projectiondistance)/projectionwidth;
    stereoview1.eyeSeparation=optis*ispercent;
    stereoview2.eyeSeparation=optis*ispercent;
}
]]>
</Script>

```

Hier ist zu sehen, dass die von den Routes definierten Werte im Bereich „toField“ deklariert wurden. Nach dem Drücken des Knopfes „Optimieren“ erhalten die entsprechenden Felder ihre Werte und werden innerhalb vom Script auf die gleichnamigen Variablen übergeben. Zusätzlich dazu führt der Knopfdruck dazu, dass das Feld „setoptimization“ den Wert 1 erhält, worauf die entsprechende Funktion ausgeführt wird. Hierbei wird mit bereits bekannter Formel die optimale interaxial Separation berechnet und mittels Verweis auf das Engine-Element die Attribute „eyeSeparation“ gesetzt. Nun besitzt die betrachtete Szene bereits optimierte Parameter und kann über den Slider der Zero Parallax im Menü weiter angepasst werden.

## 10.7 Ergebnisse

Die nächste Abbildung zeigt eine optimierte Darstellung für eine X3D-Szene. Diese Szene besteht dabei aus den selben Objekten wie schon bei der Implementierung des MEL-Scripts. zur Benutzung dieser X3D-Szene wurden diese allerdings in X3D umgewandelt und verschiedenste Anpassungen vorgenommen, sodass die Darstellungen übereinstimmen. Für eine bessere Betrachtung bietet sich die Verwendung des entwickelten Web-Interfaces an. Dadurch kann in der Szene, unabhängig vom Einstellen der stereoskopischen Parameter, eine Navigation stattfinden. Zur verbesserten intuitiven Bewegung wurde diese Szene so erweitert, dass sie mit einem handelsüblichen Steuergerät über USB gesteuert werden kann. Der Benutzer repräsentiert dabei eine virtuelle Person aus der Ich-Perspektive und kann sich frei in der Szene bewegen. Auch ein Springen des virtuellen Charakters und das Einstellen der Zero Parallax Distance sowie die prozentuale Anpassung der interaxial Separation wurden ebenfalls implementiert.



**Abbildung 10.4:** Darstellung einer optimierten stereoskopischen X3D-Szene im Instant-Player. Zum Betrachten bitte die Rot-Cyan-Brille aufsetzen.

### 10.8 Zusammenfassung und Ausblick

Auch diese Implementierung fand strikt nach dem Konzept statt und konnte vollständig realisiert werden. Alle Funktionalitäten agieren in der im Rahmen der Bachelorarbeit erstellten Implementierung ohne Probleme, allerdings mit vorher beschriebenen Einschränkungen.

Als Ausblick gibt es für diesen Bereich eine Menge weiterer Möglichkeiten. So wäre der Einbau verschiedener Hilfen möglich, mit denen z.B. der Abstand zur Zero Parallax Plane stets in der Szene angezeigt werden. Auch könnte eine zusätzliche Hilfe eingebaut werden, welche einen ersten Einstieg in das Menü bietet und somit intuitiver wird.

Weiterhin könnten mit neuen Funktionen in der Stereoskopie und neuen Funktionen neuerer InstantPlayer-Versionen weitere Features eingebaut werden.

# Kapitel 11

## Zusammenfassung und Ausblick

Das vorrangige Ziel dieser Bachelorarbeit war es, eine Optimierung stereoskopischer Parameter intuitiv zu ermöglichen. Dazu war es notwendig, dem Leser einen Einstieg in die Stereoskopie zu bieten. Zusätzlich zu den Grundlagen des räumlichen Sehens wurden hierfür auch die stereoskopischen Begriffe virtueller Welten vermittelt. Mit denen in dieser Bachelorarbeit entwickelten Implementierungen ist es möglich, einen Großteil aller heutzutage gängigen Anzeigeverfahren zu nutzen, da beide Umgebungen universell für die Stereoskopie eingesetzt werden können. Grundlage dieser Optimierungen bilden stets mathematische Zusammenhänge. Dazu wurden verschiedene, heutzutage benutzte mathematische Ansätze verglichen und einer ausgewählt. Bei diesem Ansatz stellte sich die gegebene Formel als bereits optimal heraus. Diese Formel wurde dann im Rahmen dieser Bachelorarbeit erweitert, wodurch erst die Optimierung im Bereich von virtuellen, stereoskopischen Inhalten ermöglicht wurde. Erst durch diese Erweiterung bezogen sich die einzelnen Bestandteile der Formel sowohl auf virtuelle als auch auf reelle Parameter. Die Implementierungen wurden mit mehreren Programmiersprachen umgesetzt. Die dafür notwendigen Grundkenntnisse wurden dazu ebenfalls vorgestellt und anhand von Beispielen demonstriert.

Es fiel später auf, dass es heutzutage noch immer an der Verfügbarkeit optimierter stereoskopischer Darstellungen mangelt, deshalb wurden bereits vorhandene Systeme in Geräte für verbesserte Anzeige, dementsprechend fehlerfreiere Präsentation, und Geräte für die Optimierung der stereoskopischen Inhalte, welche eher auf bereits bekannten stereoskopischen Darstellungen aufbauen, eingeteilt. Dabei wurde klar, dass eine universelle, kostenfreie Möglichkeit der Optimierung stereoskopischer Parameter noch nicht existiert, was die Ziele dieser Bachelorarbeit weiter bestärkte. Für die Implementierungen fand eine Konzeptentwicklung für jedes der eingesetzten Systeme statt. Jedes dieser Konzepte sollte dem Benutzer eine intuitive Möglichkeit geben, einen Einstieg in die Stereoskopie zu finden. Für die Implementierungen innerhalb dieser Bachelorarbeit wurde im Bereich der Modellierung auf das Werkzeug Autodesk Maya gesetzt. Dieses bietet alle nötigen stereoskopischen Konfigurationen, welche für eine Optimierung nötig sind. Im Bereich von Echtzeit-Rendering-Systemen wurde hier auf den vom Fraunhofer IGD entwickelten InstantPlayer gesetzt, da auch dieser alle nötigen Freiheiten lässt, stereoskopische Optimierungen zu entwickeln. Weiterhin ist diese Umgebung für den nicht kommerziellen Gebrauch kostenfrei und baut auf eine offene

Beschreibungssprache namens X3D auf und bietet sich daher für viele Benutzer an. Mit der Umsetzung der Konzepte und dem Optimieren der virtuellen, stereoskopischen Inhalte wurden die Ziele dieser Bachelorarbeit erreicht.

### 11.1 Ausblick

Diese Bachelorarbeit hat gezeigt, dass eine Optimierung stereoskopischer Darstellung heutzutage noch nicht überall genutzt wird. Entsprechend vorgestellte Systeme bieten zwar die Möglichkeit der Optimierung, sind allerdings nicht frei verfügbar oder gar nicht käuflich erwerbbar. Daraus entstand die Idee von Konzepten einer neuen, eigenen Implementierung dieser Optimierungen für frei erhältliche oder zumindest frei erweiterbarer Systeme. Dabei war es sogar möglich, vorhandene Optimierungen zu erweitern und deren Beständigkeit nachzuweisen.

Auf dem Markt bereits vorhandene Systeme bieten allerdings Möglichkeiten, welche von Implementierungen dieser Bachelorarbeit nicht geboten werden können. Daher ist der Ausblick auf eine Erweiterung dieser Implementierungen sehr groß und könnte durch mehrere neue Fähigkeiten erweitert werden. Eine davon ist, dass der statische Parameter des Betrachterabstandes im Bereich des Echtzeit-Rendering-Systems InstantPlayer völlig abgeschafft wird. Ein Ersatz dessen bietet sich deshalb an, da der InstantPlayer bereits die Möglichkeit bietet, retroreflektierende Kugeln mit Kameras zu verfolgen. Dadurch wäre es möglich, den Abstand des Betrachters zur Projektion zu berechnen, wenn er, wie beim vorgestellten System von IMSYS, eine Brille mit den entsprechenden Kugeln an deren Oberseite tragen würde.

Diese Bewegungsverfolgung könnte auch dazu benutzt werden, die Perspektive der Ansicht zu ändern, wodurch eine realistischere Bewegung im virtuellen Raum ermöglicht würde.

Doch auch im Bereich der Begrenzung dieser Optimierungen auf einen Betrachter gäbe es mehrere Ansätze, dies zu verbessern und zu erweitern. Ein Ansatz davon könnte sein, dass mehrere Benutzer jeweils eine Brille tragen, welche getracked werden kann. Zusätzlich dazu könnte ein System, wie das vorgestellte Wellenlängenmultiplex-Verfahren von infitec, so erweitert werden, dass mehrere Interferenzfilter, genauer gesagt jeweils zwei pro Betrachter, genutzt werden könnten. Somit hätte bei korrekter Justage jeder Benutzer sein völlig eigenes optimiertes stereoskopisches Bild und könnte sich frei im Raum bewegen.



# Glossar

Akkomodation	Beschreibt die Fähigkeit des Auges, die Augenlinse auf betrachtete Objekte je nach Entfernung anzupassen.
Augmented Reality	Beschreibt die erweiterte Realität, welche computergesützt eine Erweiterung der Realitätswahrnehmung darstellt. Eine typische Anwendung dessen ist die Überlagerung von reellen Darstellungen mit computergenerierten Darstellungen.
Bluray	Ist eine Technik, bei der mit Hilfe eines blauen Lasers Daten auf ein Speichermedium geschrieben und gelesen werden können.
Cluster	Ein Rechnerverbund von Computern zur Steigerung der Rechenleistung oder Ausfallsicherheit.
Divergente Parallaxe	Entspricht jener horizontalen Verschiebung innerhalb einer stereoskopischen Darstellung, die größer als der Augenabstand ist. Dadurch drehen sich die Augen nach außen, was schnell zu Kopfschmerzen führen kann.
DLP	Digital Light Processing. Die Technologie wird vorwiegend bei DLP-Projektoren eingesetzt. Hauptbestandteil ist ein mikromechanischer Baustein, der mit Hilfe einzeln beweglicher Spiegel zur gezielten Lichtlenkung genutzt werden kann.
DLP-Link	Beschreibt ein Verfahren, bei denen Brillen für eine stereoskopische Darstellung mittels einem projizierten Bild synchronisiert werden können. Diese Synchronisation findet dabei mit einem so genannten Weißblitz statt.
ECMAScript	Beschreibt eine moderne, schlanke, dynamisch typisierte, objektorientierte, aber klassenlose Skriptsprache.
FLCOS	Ferroelectric Liquid Crystal On Silicon. Unterscheidet sich zum LCOS in der Geschwindigkeit, in der diese Anzeigen Darstellungen visualisieren können.
FullHD	Beschreibt die Auflösung mit 1920x1080 Pixel.
Geisterbilder	Bezeichnet die Sichtbarkeit einer Ansicht, welche fälschlicherweise in einer zweiten Ansicht auftaucht. Typischerweise beschreibt dies in der Stereoskopie den Fall, dass die rechte Ansicht in der linken Ansicht trotz Filter sichtbar wird, was den stereoskopischen Effekt zerstören kann.
HDMI	High Definition Multimedia Interface ist eine ab Mitte 2003 entwickelte Schnittstelle für die volldigitale Übertragung von Audio- und Video-Daten in der Unterhaltungselektronik.

---

HMD	Head-Mounted Display. Ist ein visuelles Ausgabegerät, welches wie ein Helm auf dem Kopf getragen wird und auf einem augennahem Bildschirm Visualisierungen darstellt.
HUD	Head-Up-Display. Ist ein Anzeigesystem, bei dem die für den Nutzer wichtigen Informationen in sein Sichtfeld projiziert werden.
InstantPlayer	Ein vom Fraunhofer IGD entwickelter Viewer für VRML und X3D.
Interferenz KMQ	Beschreibt die Überlagerung von zwei oder mehr Wellen. Abkürzung der Initialien der Erfinder Koschnitzke, Mehner und Quick. Ist ein stereoskopisches Anzeigeverfahren im Übereinander-Untereinander-Format.
Konvergenz	Die Fähigkeit des Auges zu konvergieren, wenn dies erforderlich ist.
Konvergierende Kameras	Kamerastellung in der Stereoskopie, bei der sich beide Kameras zueinander drehen.
LCOS	Liquid Crystal On Silicon. Ist ein Anzeigegerät ähnlich einem LCD. Im Gegensatz zu diesem lässt es das Licht allerdings nicht durch, sondern reflektiert es.
Lentikular	Bezeichnet das so genannten Linsenrasterverfahren. Hierbei befinden sich mehrere Ansichten auf einer Darstellung und werden durch Linsen in die jeweiligen Richtungen abgelenkt.
Maya MEL	Ein von Autodesk entwickeltes Modellierungswerkzeug. Maya Embedded Language. Ist eine Skriptsprache zur Steuerung von Autodesk Maya.
MEMS	Micro-Electro-Mechanical System. Ist ein miniaturisiertes Gerät, dessen Komponenten kleinste Abmessungen (im Mikrometerbereich) haben und als System zur Projektion von Signalen zusammenwirken.
Negative Parallaxe	Stellen einer stereoskopischen Darstellungen, die scheinbar vor der Projektionsfläche liegen. Sie besitzen eine horizontale Verschiebung und führen zum Konvergieren der Augen.
Null/Zero Parallaxe	Ist die Stelle einer stereoskopischen Darstellung, an der keine horizontale Verschiebung vorhanden ist.
Off-Axis Kameras	Kamerastellung in der Stereoskopie, bei der die Kameras parallel stehen, allerdings die Sensoren verschoben werden.
Parallaxe Barriere	Bezeichnet ein Verfahren, welches bei autostereoskopischen Displays eingesetzt wird. Hier befindet sich eine Streifenmaske vor der eigentlichen Ansicht, wodurch für den Betrachter das Sehen nicht relevanter Blickwinkel gesperrt werden soll.
Parallele Kameras	Typische Kamerastellung in der Stereoskopie. Dabei sind beide Kameras parallel angeordnet und filmen eine Szene.
Positive Parallaxe	Stellen einer stereoskopischen Darstellung, die scheinbar hinter der Projektionsfläche liegen, besitzen eine horizontale Verschiebung und führen bei den Augen zum parallel stellen.

Qt Creator	Von Nokia entwickelte Umgebung zur Erstellung interaktiver Oberflächen.
Quad Buffered OpenGL	Vierfach gepuffertes OpenGL. Beschreibt die Möglichkeit, in die linken, rechten, vorderen und hinteren Buffer einer Grafikkarten unabhängig voneinander zu schreiben. In der Stereoskopie zeigen dabei die vorderen linken und rechten Buffer das stereoskopische Bild an, während die hinteren stetig aktualisiert werden. Dadurch wird eine angenehme Anzeige stereoskopischer Darstellungen ermöglicht.
Querdisparation	Beschreibt die Bildlage leicht versetzter Abbildungen auf der Netzhaut.
Retroreflexion	Als Retroreflexion bezeichnet man die Reflexion, die weitgehend unabhängig von der Ausrichtung des Reflektors großteils in Richtung zurück zur Strahlungsquelle erfolgt.
RID	Abkürzung für Retinal Image Display.
Rig	Zu deutsch Vorrichtung.
Route	Definiert den Weg, den eine Umleitung von einem Feld zu einem anderen nehmen muss. Findet in der X3D-Programmierung statt, um externe und interne Inhalte miteinander verbinden zu können.
ShearedStereoViewModifier	Beschreibt einen Modifikanten innerhalb einer Viewarea, welcher für die stereoskopischen Parameter einer Darstellung zuständig ist.
Sweet Spot	Ist der Bereich, in dem ein Betrachter eines autostereoskopischen Display stehen darf um noch einen stereoskopischen Effekt zu sehen. Außerhalb dieses Bereichs geht dieser Effekt verloren.
SXGA	Ist die Abkürzung für Super Extended Graphics Array. Es bezeichnet in der Computergrafik üblicherweise eine Bildauflösung von 1280x1024 Pixel.
Tile	Zu deutsch Kacheln. Sind (Computer-)Grafiken, die mosaikartig ein vielfach größeres Gesamtbild zusammensetzen.
Tracking	Umfasst alle Bearbeitungsschritte, die der Verfolgung von bewegten Objekten dienen.
Viewarea	Repräsentiert eine rechteckige Ansicht im Viewer InstantPlayer.
Viewpoint	Repräsentiert den zur Viewarea zugehörigen Blickwinkel einer Ansicht.
Virtual Reality	Beschreibt die virtuelle Realität. Im Gegensatz zur Augmented Reality entsteht die virtuelle Realität ausschließlich am Computer. Dabei werden dort alle Wahrnehmung der Wirklichkeit und ihre physikalischen Eigenschaften simuliert.
VRML	steht für Virtual Reality Modeling Language und ist eine Beschreibungssprache für 3D-Szenen inklusive Geometrien, Ausleuchtungen, Animationen und Interaktionsmöglichkeiten.
W3C	World Wide Web Consortium. Ist das Gremium zur Standardisierung der das World Wide Web betreffenden Techniken.

---

Wellenlängenmultiplex	Ist eine Methode der Signalübertragung, bei der mehrere Signale mit verschiedenen Frequenzbereichen getrennt übertragen werden.
X3D	eXensible 3D. Ist eine auf XML basierende Beschreibungssprache für 3D-Szenen und zugleich der Nachfolger von VRML.
XML	steht für eXtensible Markup Language und ist eine Auszeichnungssprache zur Darstellung hierarchisch strukturierter Daten in Form von Textdaten.
Zeitmultiplex	Mehrere Signale werden zeitversetzt mittels eines Mediums übertragen.



# Literaturverzeichnis

- [Ace09] Acer. *H5360 Brochure*, 2009.
- [Ali] Alioscopy. *Alioscopy 3D HD 47 Datasheet*.
- [Aut] Autodesk. *Maya stereoscopic white paper*.
- [Aut08] Autodesk. *Stereoscopic filmmaking whitepaper*, 2008.
- [Bla08] Prof. Dr. Karlheinz Blankenbach. *Mes multimedia displays 3d*. Technical report, Hochschule Pforzheim, 2008.
- [Boc09] Dr.-Ing. Ulrich Bockholt. *Presseinformation 04/2009*. Technical report, Fraunhofer IGD, 2009.
- [Bro08] Brother. *Brother Develops Spectacle-type Wearable Retinal Imaging Display*, 2008.
- [DB07] Leonard Daly Don Brutzman. *X3D: Extensible 3D Graphics for Web Authors*. Denise E. M. Penrose, 2007.
- [dig] digital-IMAGE. *Cube3D Produktbeschreibung*.
- [Fra] Fraunhofer Institut für Nachrichtentechnik. *Free2C 3D Display*.
- [Fri07] Oliver Friedrich. *Physiologie - GK 1*. Springer Lehrbuch, 2007.
- [Fuj10] Fujifilm. *Faszination 3D-Fotografie*, 2010.
- [ICI] ICIDO. *IMSYS NEWS HMI 2010*.
- [IMS10] IMSYS. *Datenblatt solid - Highend Gehäuse für Tageslichtsituationen*, 2010.
- [Jia10] Prof. Dr. Xiaoyi Jiang. *Kapitel 22, 3d video*. Technical report, Westfälische Wilhelms-Universität Münster, 2010.
- [Klo10] Jörg H. Kloss. *X3D Programmierung interaktiver 3D-Anwendungen für das Internet*. Addison Wesley, 2010.
- [Kuh99] Gerhard Kuhn. *Stereofotografie und Raumbildprojektion*. Verlag für Foto, Film und Video, 1999.

- [Leh05] Thomans M. Lehmann. *Handbuch der medizinischen Informatik*. Carl Hanser Verlag, Pauwelsstraße 30, 52057 Aachen, 2005.
- [Mag] Magnetic 3D. *Magnetic M3D-ALL570DX 57 LCD Monitor*.
- [Mah05] Gerhard Mahler. *Die Grundlagen der Fernsehtechnik*. Springer Verlag, 2005.
- [Men09] Bernard Mendiburu. *3D Movie Making: Stereoscopic Digital Cinema from Script to Screen*. Focal Press, Burlington, MA 01803, USA, 2009.
- [Mic] Microvision. *Color Eyewear Program Brief*.
- [MRW05] Chris Kazmier Mark R. Wilkins. *MEL Scripting for Maya Animators*. Morgan Kaufmann, 2005.
- [NVI] NVIS. *nVisor SX111 Head Mounted Display*.
- [Sch07] Roland Schmitz. *Kompendium Medieninformatik*. Springer Verlag, Nobelstr. 10, 70569 Stuttgart, 2007.
- [Sch10] Dipl.-Ing. Ellen Schwalbe. Vorlesung photogrammetrie. Technical report, Technische Universität Dresden, 2010.
- [See] SeeReal Technologies. *Discover the third dimension*.
- [Son10] Sony. *KDL-60LX905 Bedienungsanleitung*, 2010.
- [vH52] Hermann von Helmholtz. Ueber die accomodation des auges. *Graefe's Archive for Clinical and Experimental Ophthalmology*, 1(2), 1852.
- [Vir] Virtual Research. *VR1280 Specifications*.