

Virtual Reality zur Untersuchung neuronaler Prozesse:
Entwicklung und Validierung einer adaptiven
Hirn-Computer-Schnittstelle zur dynamischen Interaktion
mit Virtual Reality Paradigmen

Dissertation

zur Erlangung des akademischen Grades

Doktoringenieur (Dr.-Ing.)

angenommen durch die Fakultät für Informatik
der Otto-von-Guericke-Universität Magdeburg

von Dipl.-Inform. Charles Müller
geb. am 21.10.1979 in Lutherstadt Eisleben

Gutachterinnen/Gutachter

Prof. Dr. Dr. Johannes Bernarding

Prof. Dr. Heinz Handels

Prof. Dr. Alexander Horsch

Magdeburg, den 21. August 2013

Ehrenerklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; verwendete fremde und eigene Quellen sind als solche kenntlich gemacht. Insbesondere habe ich nicht die Hilfe eines kommerziellen Promotionsberaters in Anspruch genommen. Dritte haben von mir weder unmittelbar noch mittelbar geldwerte Leistungen für Arbeiten erhalten, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen. Ich habe insbesondere nicht wissentlich:

- Ergebnisse erfunden oder widersprüchliche Ergebnisse verschwiegen,
- statistische Verfahren absichtlich missbraucht, um Daten in ungerechtfertigter Weise zu interpretieren,
- fremde Ergebnisse oder Veröffentlichungen plagiiert,
- fremde Forschungsergebnisse verzerrt wiedergegeben.

Mir ist bekannt, dass Verstöße gegen das Urheberrecht Unterlassungs- und Schadensersatzansprüche des Urhebers sowie eine strafrechtliche Ahndung durch die Strafverfolgungsbehörden begründen kann. Die Arbeit wurde bisher weder im Inland noch im Ausland in gleicher oder ähnlicher Form als Dissertation eingereicht und ist als Ganzes auch noch nicht veröffentlicht.

Magdeburg, den 19. März 2013

Charles Müller

Danksagung

Ich danke Prof. Dr. Dr. Johannes Bernarding für die Unterstützung bei der Erstellung dieser Dissertation.

Des Weiteren bedanke ich mich bei Prof. Dr. Heinz Handels und Prof. Dr. Alexander Horsch für die Begutachtung dieser Arbeit und die wertvollen Hinweise im Vorfeld der Dissertation.

Mein Dank gilt auch den Mitarbeiterinnen und Mitarbeitern des Instituts für Biometrie und Medizinische Informatik, des Tumorregisters sowie der MRT-Teams am 3T und 7T, die durch ihre fachkundige, technische und freundschaftliche Unterstützung zur Erstellung dieser Arbeit beigetragen haben.

Im Speziellen danke ich Daniela Adolf, Ramona Grzeschik, Snezhana Weston, Sebastian Baecke, Ralf Luetzkendorf, Dr. Markus Plaumann, Frank Hertel, Thomas Trantschel und Dr. Maurice Hollmann für die konstante Unterstützung sowie für die vielen hilfreichen Anmerkungen zur Umsetzung der Arbeit. Weiterhin bedanke ich mich bei Andrea Thoms, Nance Kämmerer und Michael Lührs, die mit ihren Diplom- und Bachelorarbeiten zum Erfolg dieser Arbeit beigetragen haben.

Mein besonderer Dank gilt meiner Familie, meinen Eltern, meiner Tante und meinem Onkel, die mich stets förderten und mir einen Grund gaben, ehrgeizige Ziele zu verfolgen und dabei niemals aufzugeben.

Zusammenfassung

Die Stimulation mit visuellen Reizen ist seit langem ein Kernaspekt bei der Erforschung perzeptueller und kognitiver Prozesse (Savoy 1998, Huettel et al. 2008). Visuelle Stimuli werden hier zumeist über eine schlichte, zweidimensionale Stimuluspräsentation angezeigt. Der Grund hierfür ist die Reduktion der Stimuluspräsentation auf die zu untersuchenden Reizkomponenten. Dies birgt die Gefahr, dass wichtige Komponenten der real ablaufenden Prozesse nur unzureichend abgebildet werden.

Der Einsatz hochrealistischer 3D-Umgebungen, sogenannter Virtueller Realitäten, verspricht hier eine Lösung. Verschiedene neurokognitive Studien haben gezeigt, dass Virtuelle Umgebungen alltägliche Prozesse authentischer und mit einer größeren Effektivität simulieren können (Aguirre 1998, Maguire 1998). Des Weiteren zeigten speziell die Ergebnisse aus Studien mit echtzeitbasierter Datenauswertung, wie z.B. Echtzeit-fMRT-Neurofeedback-Studien, dass der Einsatz von Virtuellen Realitäten zu einer erhöhten Interaktion mit dem Stimulus und darauf aufbauend zu einer effizienteren Konditionierung im Rahmen des Neurofeedback-Trainings führt (deCharms et al. 2005, Thoms et al. 2011).

Die größten Hindernisse bei der Nutzung dieser neuen Form der Stimuluspräsentation sind die Einbindung in ein medizinisches Datenakquirierungssystem, die individuelle Anpassung und Kombination der Virtuellen Umgebung an die komplexen Regeln neurokognitiver Experimentabläufe sowie die mangelnde Verbreitung geeigneter VR-Technologien im neurowissenschaftlichen Forschungsbereich (Mueller et al. 2012a;b). Im Rahmen eines vom Land Sachsen-Anhalt geförderten Projektes zur Entwicklung von Hirn-Computer-Schnittstellen wurde hier ein Beitrag für ein neuartiges und allgemeingültiges VR-Stimuluspräsentationskonzept entwickelt und implementiert. Es ermöglicht anhand von speziell entwickelten Prozessabläufen die Interaktion mit hochrealistischen 3D-Geometrien in Echtzeit-fMRT-Studien. Dieses Konzept ermöglicht weiterhin die Modellierung neurokognitiver VR-Experimente als Block- bzw. als ereigniskorrelierten Experimentablauf und stellt frei definierbare 3D-Szenarien mit Interaktionsmöglichkeiten bereit.

Die VR-Stimulusapplikation gestattet zudem eine unmittelbare Adaption von dynamisch-interaktiven VR-Stimuli anhand der neuronalen Hirnaktivierungen in Echtzeit. Nach der aktuellen Kenntnislage wurde ein solches Konzept zur dynamischen Echtzeit-VR-Adaption in Neurofeedback-Experimenten bisher nicht realisiert. Das entwickelte System wurde in einer neurokognitiven VR-fMRT-Studie zur Untersuchung von Immersion und topographischer Orientierung validiert. Die Ergebnisse anderer Studien bestätigen die neuronalen Aktivierungen, welche durch die Online- und Offline-

Datenauswertung in den Parahippokampalen Arealen, im Cuneus und in den Okzipitalen Regionen gefunden wurden.

Abstract

The presentation of visual stimuli is one major aspect in the research of perceptual and cognitive processes (Savoy 1998, Huettel et al. 2008). Visual stimuli were often used for a plain and two-dimensional stimulus presentation. In order to investigate underlying neuronal mechanisms of certain stimulus components, stimulus presentation is reduced to a minimum of potentially important stimuli. In return this research strategy involves the risk of an inadequate stimulus presentation which prevent the authentic simulation of relevant perceptual processes.

The use of high realistic three-dimensional environments so-called virtual realities (VR) could be a highly promising solution. In several neuroscientific studies the virtual environment stimulus presentation even showed a more authentic and effective simulation of perceptual processes (Aguirre 1998, Maguire 1998).

In addition, various VR studies based on real-time data analysis techniques, e.g. neurofeedback real-time fMRI studies, indicated a higher interaction with the subject and mentioned a more effective operant conditioning during neurofeedback training (deCharms et al. 2005, Leins et al. 2007, Thoms et al. 2011).

The main obstacles encountered in using virtual realities are: the integration in medical data acquisition systems, the individual adaptation of virtual environments according to the rules of experimental modelling, and the limited distribution of virtual reality technologies usable to neuroscientific research studies (Mueller et al. 2012a;b).

According to this research project funded by the state of Saxony-Anhalt, Germany, a new virtual reality stimulus concept was developed and implemented. The stimulus concept provides the interaction with user-defined, highly realistic virtual environments in fMRI and real-time fMRI studies and supports the experimental modelling of neurocognitive VR paradigms either in block or event-related designs. In addition, the implemented VR stimulus application allows an immediate adaptation of dynamic VR stimuli based on real-time data analysis of neuronal activations. According to current scientific knowledge, this concept for the real-time adaptation of dynamic virtual reality stimuli in neurofeedback studies has been unique in neuroscientific experiments so far. The developed application was validated in a virtual reality fMRI study investigating the underlying cognitive mechanisms of immersion and spatial navigation. Results from other VR spatial memory studies confirmed the neuronal activation that was detected in parahippocampal areas, cuneus, and occipital regions.

Inhaltsverzeichnis

Abbildungsverzeichnis	XII
Tabellenverzeichnis	XIII
Verzeichnis der Abkürzungen	XV
1 Einführung	1
1.1 Erkenntnistheoretische Ansätze als Ursprung der Neurokognition . . .	2
1.2 Interaktionen mit immersiven virtuellen Umgebungen	3
1.3 Integration virtueller Realitäten in neurokognitive Fragestellungen . .	4
1.4 Zielstellung	6
1.5 Aufbau der Arbeit	7
2 Grundlagen der kernmagnetischen Resonanz und der fMRT	9
2.1 Allgemeine Grundlagen der Magnetresonanztomographie	9
2.1.1 Kernmagnetische Resonanz	10
2.1.2 Relaxationseigenschaften	12
2.1.3 Ortskodierung	15
2.1.4 Bildkontraste und Pulssequenzen	18
2.2 Funktionelle Magnetresonanztomographie	20
2.2.1 <i>BOLD</i> -fMRT	20
2.2.2 Modellierung von funktionellen MRT-Experimenten	22
2.2.3 Vorverarbeitung funktioneller MRT-Daten	27
2.2.4 Statistische Auswertung funktioneller MRT-Daten	31
2.3 Echtzeit-Auswertung funktioneller Bilddaten	36
2.3.1 Optimierte Echtzeit-Datenauswertung auf Basis der Korrelationsanalyse	36
2.3.2 Das Allgemeine Lineare Modell zur Echtzeit-Datenanalyse . .	39
2.3.3 Echtzeit-Datenauswertung durch datengetriebene Auswerteverfahren	40
3 Neurophysiologische Grundlagen und Hirn-Computer-Schnittstellen	43
3.1 Topographisches Gedächtnis und räumliche Orientierung	43
3.1.1 Topographisches Lernen	44
3.2 Arbeitsgedächtnis	45

3.2.1	Räumlich-visueller Speicher	46
3.3	Langzeitkonsolidierung	47
3.4	Neokortikale Repräsentation räumlicher Informationen	48
3.5	Immersionseffekt und Vigilanzzustände in Virtuellen Realitäten	49
3.5.1	Immersion	49
3.5.2	Vigilanz und saliente Reize	51
3.5.3	Neurokognitive Anwendungen Virtueller Realitäten	51
3.6	Hirn-Computer-Schnittstellen - Geschichte und Motivation	52
3.7	Neurofeedback und Hirn-Computer-Schnittstellen	54
3.7.1	Invasive Hirn-Computer-Schnittstellen	55
3.7.2	Nichtinvasive Hirn-Computer-Schnittstellen	57
3.8	Anwendungsformen von BCI-Systemen	60
4	Informationstheoretische Grundlagen	65
4.1	Grundlagen Virtueller Umgebungen	65
4.1.1	Definition und Geschichte	65
4.1.2	Modellierung und Klassifikation Virtueller Umgebungen	66
4.2	Mensch-Computer Interaktionen	71
4.2.1	Arch-Modell	71
4.2.2	Transitionssysteme	72
4.3	Kollisionsdetektion von Polygonobjekten	73
4.3.1	Methoden zur Kollisionsverarbeitung	74
4.4	Physikalisch-basiertes Rendern	76
4.4.1	Texturierung und Repräsentation künstlicher Materialien	77
4.4.2	Methodik des physikalisch-basierten Renderns	78
4.4.3	Echtzeit-Rendern	79
5	Problemanalyse	81
5.1	Allgemeine Anforderungen an neurowissenschaftliche Stimulusappli- kationen	81
5.2	Aufbau virtueller Umgebungen zur Stimuluspräsentation	86
5.3	Modellierung eines neurowissenschaftlichen Experiments am Beispiel	88
5.4	Existierende Softwaresysteme	90
5.4.1	<i>Presentation</i> - Neurobehavioral Systems Inc.	90
5.4.2	<i>VRWorld 2</i> - Psychology Software Tools Inc.	90
5.4.3	<i>rtPresent</i> - IBMI, Universität Magdeburg	91
5.4.4	<i>Turbo BrainVoyager</i> - Brain Innovation B.V.	92
6	Entwurf einer multimodalen VR-Stimulusapplikation	93
6.1	Einbindung der VR-Stimulusapplikation in eine MRT-Infrastruktur	93
6.2	Modellierung eines VR-Stimulusframeworks auf Basis einer skalier- baren Softwarearchitektur	95
6.3	Integration virtueller Szenen mit dem Vision Engine Framework	100
6.4	Interne Datenhaltung von Experimentinformationen	101
6.5	Konzept zur Echtzeit-Stimulusadaption von VR-Paradigmen	103

6.6	Verbreitung und Speicherung von Experimentinformationen	106
6.7	Verwendete Softwaretechnologien	109
6.7.1	Havok Vision Engine Framework	109
6.7.2	Nvidia PhysX	110
6.7.3	Nokia Qt - User Interface Framework	110
6.7.4	Autodesk 3ds Max	111
7	Realisierung und technische Evaluierung der Applikation	113
7.1	Realisierung der Datenzugriffsschicht	113
7.1.1	Ein TCP-basiertes Kommunikationsprotokoll zur Informations- übermittlung	116
7.2	Realisierung der Applikationsschicht	117
7.2.1	Experimentsteuerung und Einbindung des Vision Engine Frameworks	117
7.2.2	Echtzeit-Stimulusadaption von Neurofeedback-Experimenten	120
7.2.3	Datenmanagementframework	123
7.2.4	Einbindung benutzerdefinierter Funktionalitäten	125
7.3	Realisierung der Präsentationsschicht	127
7.3.1	Strukturierung der Oberflächenkomponenten	127
7.4	Evaluation der Softwareanwendung	132
7.4.1	Material und Methodik	132
7.4.2	Benutzerakzeptanztest	134
7.4.3	Ressourcennutzung und Performanztest	135
8	Evaluierung des Stimuluskonzepts im Rahmen einer Pilotstudie	141
8.1	Problembeschreibung	142
8.2	Methodik und Materialien	143
8.2.1	Technischer Aufbau und Datenakquirierung	143
8.2.2	Probanden und Paradigma	145
8.2.3	Integration des Turbo BrainVoyager	148
8.2.4	Echtzeit-Adaptionsframework	149
8.3	Ergebnisse	150
8.3.1	VR-fMRT-Experiment - Echtzeit-Datenauswertung	150
8.3.2	VR-fMRT-Experiment - Post-experimentelle Datenanalyse . .	151
8.3.3	Echtzeit-VR-Stimulusadaption - Datenanalyse	153
8.4	Diskussion der Experimente	158
9	Zusammenfassung und Ausblick	165
9.1	Zusammenfassung und Diskussion	165
9.2	Ausblick	171
A	Probandenfragebogen	191
B	Benutzerakzeptanztest	195

Abbildungsverzeichnis

1.1	VR-fMRT-Studien - Beispiele	4
2.1	Klassische Darstellung des Kernspins	11
2.2	Darstellung der longitudinalen und transversalen Relaxation	13
2.3	Zerfall der Transversalmagnetisierung	14
2.4	Schichtselektion	15
2.5	Kodierung/Auslesen der Signale in x-Richtung	17
2.6	Kodierung/Auslesen der Signale in y-Richtung	18
2.7	Kontrastunterschiede in der MRT	19
2.8	Lokale Sauerstoffanreicherung bei neuronaler Aktivität	21
2.9	Theoretischer Verlauf der hämodynamischen Antwortfunktion	22
2.10	Paradigmenarten neurowissenschaftlicher Experimente	25
2.11	Prozesskette zur Auswertung von fMRT-Daten	28
2.12	Das allgemeine Lineare Modell - Modellfaktoren	33
2.13	Kumulative Auswertung vs. Sliding-Window-Verfahren	38
3.1	Räumliches Gedächtnis - Hirnareale	44
3.2	Das <i>three-presence components</i> -Modell immersiven Verhaltens	50
3.3	Selbstregulierung neuronaler Aktivität	54
3.4	Platzierung von ECoG-Elektroden	56
3.5	Akquirierungsmodalitäten für Hirn-Computer-Schnittstellen	58
3.6	Aufbau des <i>Thought Translation Device</i> (EEG-BCI)	59
3.7	Aufbau eines BOLD-fMRT-BCI	61
3.8	Veröffentlichte Neurofeedback-Studien von 1996 bis 2011	62
3.9	Neurofeedback-Studien im Bereich der fMRT	63
4.1	Modellbildung Virtueller Umgebungen	67
4.2	Klassifikation von Virtuellen Umgebungen	68
4.3	AEIO-Paradigma	70
4.4	Das Arch-Modell	72
4.5	Simulationsprozess mit Kollisionsmanagement	74
4.6	Hierarchische Kollisionsdetektion	76
4.7	Licht-Materie-Interaktion eines Photons	77
4.8	Übersicht - Materialmodelle	78
4.9	Virtuelle Umgebungen in der 3D-Grafik Engine <i>Vision</i>	80

5.1	Prozessabläufe vorhandener Echtzeit-fMRT-Systeme	82
5.2	Struktur des V-Modells	84
5.3	Aufbau einer Drei-Schichten-Kommunikationsstruktur	85
5.4	Beispielexperiment im Block-Design	88
6.1	Aufbau einer Echtzeit-fMRT-Infrastruktur	94
6.2	Prozessabläufe - VR-Stimulusframework	96
6.3	Drei-Schichten-Architekturmodell der VR-Stimulusanwendung	98
6.4	Elemente und Struktur des Informationsprotokolls	102
6.5	Konzept zur Echtzeit-Stimulusanpassung	104
6.6	Anwendungsdiagramm der Echtzeit-Anpassung von Paradigmen	106
6.7	Konzept zur Verbreitung von Experimentinformationen	107
6.8	Module des <i>Qt</i> -Frameworks	110
7.1	Klassenaufteilung der Datenzugriffsschicht	114
7.2	Sequenzdiagramm - Neurofeedback-Adaption mit TCP-Framework	115
7.3	Sequenzdiagramm - Stimuluspräsentation	118
7.4	Prozessablauf - Anpassungsalgorithmus	122
7.5	Datenhaltung - <i>experimentProtocol</i>	124
7.6	Plug-in-Einbindung und -Typisierung	126
7.7	Registerkartenmodell der Benutzeroberfläche	128
7.8	Stimulus-Registerkarte zur Erstellung von Paradigmen	129
7.9	<i>Condition</i> -Registerkarte: Bedingungsverknüpfung mit Mediendateien	130
7.10	Benutzerschnittstelle - Syntaktische Fehlerüberprüfung	131
7.11	Performanztest - Anzahl Experimentparameter	136
7.12	Performanztest - Verarbeitungszeit	137
7.13	Performanztest - Prozessorauslastung	139
8.1	VR-Szenario des VR-fMRT-Experiments	146
8.2	Paradigmendesign des VR-fMRT-Experiments	147
8.3	Ergebnisse - TBV-Echtzeit-Datenauswertung	151
8.4	Ergebnisse - Neuronale Aktivität Fixed-Effect Gruppenanalyse	152
8.5	Ergebnisse - VR-Stimulusadaption - Kalibrierungsmessung	154
8.6	Ergebnisse - VR-Stimulusadaption - Hauptmessung	156
8.7	Diskussion - VR-Stimulusadaption	164

Tabellenverzeichnis

2.1	Softwareanwendungen zur Analyse von fMRT-Daten	36
7.1	Technische Spezifikationen Testumgebung	133
7.2	Performanzergebnisse der VR-Stimulusanwendung	138
7.3	Benutzerakzeptanztest-Eingangsbefragung	140
8.1	MRT-Parameter für den Experimentablauf	144
8.2	Ergebnisse Fixed-Effect Gruppenanalyse	153
8.3	Gesamtergebnisse VR-Adaptionsexperiment	157

Abkürzungsverzeichnis

ALM	Allgemeine Lineare Modell
API	Application Programming Interface
BCI	Brain-Computer-Interface
BOLD	Blood Oxygen Level Dependent
CBF	Cerebral Blood Flow
CBV	Cerebral Blood Volume
CPU	Central Processing Unit
CT	Computertomographie
EDL	Experiment Description Language
EEG	Elektroenzephalographie
EPI	echo planar imaging
FDR	False Discovery Rate
FWE	Family Wise Error
fMRT	funktionelle Magnetresonanztomographie
GLM	General Linear Model
GPU	Graphical Processing Unit
GUI	Graphical User Interface
HRF	Hämodynamische Responsefunktion
ICA	Independent Component Analysis
LAN	Local Area Network
MEG	Magnetoenzephalographie
MFC	Microsoft Foundation Classes
MRT	Magnetresonanztomographie / Magnetresonanztomograph
NGP	Next Generation Portable
NIRS	Nahinfrarotspektroskopie
NMR	Nuclear Magnetic Resonance
NURBS	nicht-uniforme rationale B-Splines
PCA	Principal Component Analysis
PRT	Protokoll-Dateiformat der BrainVoyager-Anwendung
RAM	Random Access Memory (Arbeitsspeicher)
ROI	Region of Interest
SPM	Statistical Parametric Mapping
TBV	Turbo Brain Voyager
TCP	Transmission Control Protocol
VR	Virtuelle Realität

VU	Virtuelle Umgebung
VSCENE	Szenen-Dateiformat der Vision Engine
XML	Extensible Markup Language

Kapitel 1

Einführung

Ein Kernthema der Neurowissenschaften stellt die Erforschung wahrnehmungsorientierter und kognitiver Prozesse dar. Häufige Fragestellungen beschäftigen sich vor allem mit dem Thema, wie anhand der visuellen, auditorischen, motorischen sowie sensorischen Wahrnehmung (Perzeption) Informationen aus unserer Umwelt verarbeitet werden. Hier stellen das Abspeichern, die Weiterverarbeitung und das erneute Abrufen von Informationen in den verschiedenen Regionen des Gehirns wichtigen Aufgaben dar. Nur durch das Kombinieren der verschiedenen Reizinformationen kann eine sichere Interaktion des Individuums mit der Projektion der Umgebung entstehen (Birbaumer und Schmidt 2006).

Ein weiteres Problem ist die Erkenntnis, dass jeder Mensch durch sein subjektives Abbild der Umgebung sowohl geprägt als auch eingeschränkt ist. Alle sensorischen Informationen werden durch psychische Prozesse interpretiert, bewertet und manipuliert (Stroebe et al. 2002). Ein Beispiel für eine solche psychische Informationsinterpretation ist die sogenannte *amodale Ergänzung*, die im Bereich der Wahrnehmungspsychologie, z.B. bei *Kanizsa-Figuren*, eingesetzt wird (Pollmann 2008).

Um ein Nutzen aus dieser vermeintlichen Einschränkung zu ziehen, wurden sensorische Stimuli zur Erforschung der neuronalen Verarbeitung von Reizinformationen eingesetzt. Durch die technischen Fortschritte der vergangenen Jahre ist mit den sogenannten virtuellen Realitäten eine Technologie entstanden, welche mehrere sensorische Stimulustypen kombiniert und zum Beispiel mit Tiefenwahrnehmungseffekten die Illusion einer künstlichen Umgebung aufbaut.

Diese Technologie verspricht zwar großes Potential, ist aber im Hinblick auf neurowissenschaftliche Einsätze kaum erforscht. So stellt sich die Frage, ob sich künstliche Realitäten überhaupt zur Erforschung von kognitiven Prozessen eignen und ob klassische Fragestellungen mit dieser Technologie nicht neu bewertet werden müssen.

Zusätzlich zu den neurowissenschaftlichen Forschungsschwerpunkten rücken hier zunehmend die klassischen Fragestellungen des erkenntnistheoretischen *Leib-Seele-Problems* in den Mittelpunkt, aus denen sich die ursprüngliche Intention der neurowissenschaftlichen Forschung ableitet (Beckermann 2008). Wie interagieren Körper und Geist? Wie werden im Gehirn Abbilder bzw. Karten der realen, umgebenden oder einer künstlich präsentierten Umwelt erschaffen? Wie laufen, abhängig vom jeweiligen Individuum, die unterschiedlichen Interaktions-, Interpretations- und Repräsentations-

prozesse mit dieser Umwelt ab?

Um diese Fragestellungen weiter zu verdeutlichen, soll zunächst die Grundproblematik des *Leib-Seele-Problems* erläutert werden.

1.1 Erkenntnistheoretische Ansätze als Ursprung der Neurokognition

Die Unterscheidungsmerkmale zwischen mentalen und physischen Prozessen stehen von jeher im Zentrum der klassischen Neurowissenschaften. Philosophische und erkenntnistheoretische Fragen standen hier vor allem im Vordergrund. Wie interagieren Körper und Geist miteinander? Sind unsere Gedanken und unser Handeln vorbestimmt oder, aktuell betrachtet, kann eine künstlich generierte Welt an die Stelle unserer eigenen, realen Welt treten. Im Zentrum all dieser Fragen steht die *Philosophie des Geistes* mit ihrem Kern, dem *Leib-Seele-Problem* (Beckermann 2008). Die klassische Formulierung dieses Problems geht dabei auf den Philosophen und Stoiker René Descartes und seinen *Interaktionistischen Dualismus* zurück. Dieser vertrat die These, dass Geist und Materie zwei voneinander unterschiedliche „Substanzen“ seien, welche aber durch eine Interaktionsbeziehung aufeinander einwirken.

Werden einem Menschen Reize dargeboten, so wandert beispielsweise ein visueller Stimulus vom Auge an die zuständige Position im Gehirn, wo letztendlich das Gehirn auf den immateriellen Geist einwirkt. Nach Descartes Theorie wirkt dieses Prinzip auch bidirektional. Als Beispiel hierfür wird oftmals das Schmerzempfinden genannt. Dabei wirkt der Schmerz zunächst mittels des immateriellen Geistes auf eine bestimmte Region im Gehirn, von wo aus das Schmerz-Signal an die richtige Stelle des Körpers gesendet wird (Descartes 1641).

Aufgrund der fortschreitenden technischen Entwicklung wurden Descartes Theorien über den Substanzdualismus schon bald verworfen. Der von Baruch de Spinoza beschriebene *Monismus* trat in den Vordergrund und setzte die Existenz einer einzigen Substanz voraus (Beckermann 2008). Spinoza vertrat die Theorie, dass eine vorhandene „Substanz“ weder geistigen noch physischen Ursprungs sei und dass Psyche als auch Physis zu gleichen Teilen eine Eigenschaft dieser Materie darstellen. Die eigentlichen Ursprünge dieser Theorie, abgeleitet von fernöstlichen Glaubensphilosophien, sind heute immer noch ein aktuelles Thema. Durch den Einsatz Virtueller Realitäten können Teilaspekte dieser Fragen neu untersucht und anhand der Ergebnisse gegebenenfalls neu bewertet werden.

Aufgrund des in verschiedenen Aspekten zugrunde liegenden philosophischen Charakters, führt dieses Modell von Spinoza jedoch zu Problemen bei der Erforschung kognitiver Prozesse und mentaler Zustände. Viele kognitive Prozesse laufen introspektiv ab. Das heißt, ein Mensch reflektiert seine Handlungen nach innen gerichtet, was eine Überprüfung der damit verbundenen psychologischen Prozesse unmöglich macht (Stroebe et al. 2002). Eine Lösung dieses klassischen Problems aus der Psychologie wurde mit der sogenannten *Identitätstheorie* beschrieben (Birbaumer und Schmidt 2006). Im Zentrum dieser Theorie steht die Erkenntnis, dass ein bestimmter mentaler

Zustand identisch mit einem Zustand des Gehirns ist. Zusammen mit dem *Funktionalismus*, der unterschiedlichen Gehirnzuständen den gleichen funktionalen Zustand zuordnet, wurde ein Modell entwickelt, welches anhand heutiger neurowissenschaftlicher Möglichkeiten die Überprüfung der klassischen Ansätze bedingt möglich macht. So lassen sich durch die Stimulation eines Probanden mit bestimmten Reizen die Funktionalitäten den jeweiligen Hirnregionen zuordnen (Jäncke 2005).

Aufgrund der zunehmenden Erkenntnisse im Bereich der visuellen Wahrnehmung können so auch komplexere visuelle Stimuli, sogenannte computergenerierte Umgebungen, zur Reizpräsentation eingesetzt werden (Maguire 1998). Hier spielen vor allem die Erforschung von kognitiven Mechanismen zur räumlichen Orientierung sowie zum episodischen und topographischen Gedächtnis eine zentrale Rolle (Aguirre 1998, Burgess et al. 2002, Pine et al. 2002, Maguire et al. 2006).

Diverse kognitive Studien lieferten bereits wichtige Erkenntnisse zur Konkretisierung von *Leib-Seele-Fragestellungen* und geben so Zuversicht für zukünftige Antworten (Aguirre 1998, Birbaumer et al. 1999, Doeller et al. 2010). Mit der vorliegenden Arbeit sollen Voraussetzungen geschaffen werden, welche eine gleichzeitige Untersuchung von Immersion und Benutzerinteraktionen mit der Virtuellen Umgebung durch die Analyse der neuronalen Aktivierung erlauben. Dieser Ansatz liefert somit eine vielversprechende Methodik, interne und externe Prozesse gleichzeitig in gewissem Maße messbar zu machen und sich damit der Lösung des Leib-Seele Problems für einzelne Fragestellungen zu nähern.

1.2 Interaktionen mit immersiven virtuellen Umgebungen

Ein wesentlicher Punkt, welcher virtuelle Realitäten als Ersatz für reale Stimuli interessant macht, ist die hohe Realitätsnähe des Stimulus, d.h., die Möglichkeit mit der künstlichen Welt zu interagieren sowie künstliche Charaktere zu steuern.

Wie die Verbreitung von Computerspielen zeigt, können die interagierenden Personen im Idealfall mental so tief in diese Umgebung eintauchen, dass sie sich anschließend wie in ihrer realen Welt verhalten (Brill 2009).

Realisiert wird dies alles durch sogenannte Wahrnehmungstäuschungen. Der geschickte Einsatz von Perspektivansichten, Schatten, Beleuchtungen, Texturen, etc. erlaubt den Einsatz von optischen Täuschungen sowie Effekten aus dem Bereich der Tiefenwahrnehmung und des dreidimensionalen Sehens (Pollmann 2008).

Schnell aufeinanderfolgende Bilder simulieren eine sich vermischende visuelle Wahrnehmung, die so nur im Alltagsleben auftritt. Zusätzlich dazu können auditive Reize die Qualität der Simulation erhöhen. So werden oftmals begleitend zum visuellen Reiz Geräusche und Töne eingespielt, die eine Erhöhung der Realitätsnähe nach sich ziehen (Craig et al. 2009).

Im Kontext der virtuellen Realitäten wird Realitätsnähe mit einem weiteren Begriff neu definiert: *Immersion*. Dieser Begriff wird zu einem Indikator für realitätsnahes Verhalten und beschreibt zugleich den sogenannten Eintauch-Prozess, der bei der In-

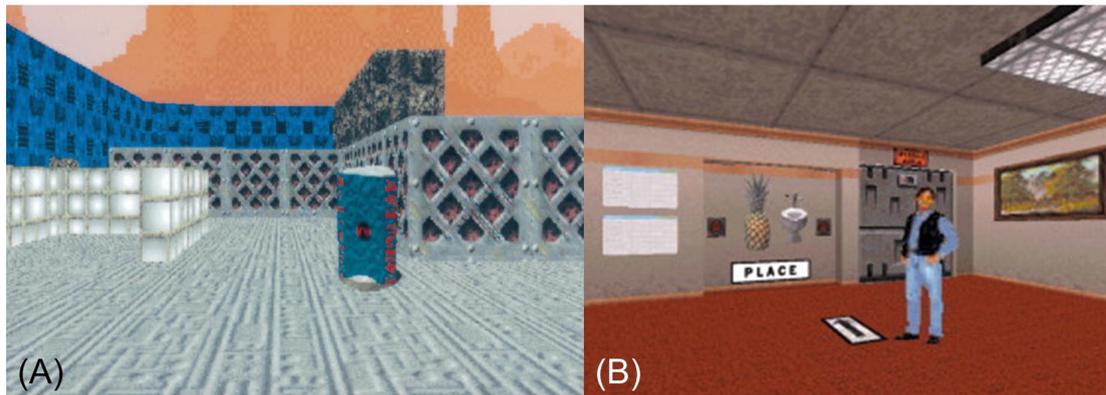


Abbildung 1.1: Viele neurokognitive Studien nutzen unveränderbare virtuelle Umgebungen, welche das Versuchsparadigma einschränken: Marathon2, Bungie Software, Chicago, Illinois, USA (Aguirre 1998) (A); Duke Nukem 3D, 3D Realms Entertainment, Dallas, Texas, USA (Burgess et al. 2002) (B).

teraktion mit der künstlichen Welt eintritt. Abhängig von der Qualität der Stimuli beschreiben Anwender einen stark auf die virtuelle Welt ausgerichteten Bezug, welcher temporär das reale Selbst vergessen lässt. Ausgehend von diesen technischen Möglichkeiten können zu den klassischen Ansätzen der Psychologie, allen voran das *Leib-Seele-Problem*, neue Diskussionspunkte abgeleitet werden. Üblicherweise erfolgt die Steuerung dieser Virtuellen Charaktere, der sogenannten Avatare, über externe Eingabehilfen wie Joystickss, Tastatur oder Datenhandschuh. Der Einsatz sogenannter Hirn-Computerschnittstellen erlaubt als neuartige Strategie, dass Probanden allein durch Selbstregulation ihrer Gehirnfunktionen einen virtuellen Charakter (Avatar) direkt und berührungslos alleine durch ihre Hirnaktivität innerhalb einer künstlichen Umgebung steuern. Das diese scheinbar extremitätenlose Fortbewegung, gemessen an den Gesichtspunkten des *Leib-Seele-Problems*, einer klassischen Trennung von Körper und Geist entspricht, wird allerdings angezweifelt. Dessen ungeachtet muss der zunehmende Bedarf an virtuellen Realitäten in den Neurowissenschaften weiterhin gedeckt werden. Hier müssen möglichst zeitnah weitere Modelle, Konzepte und schließlich auch Anwendungen bereitgestellt werden, die das Potential dieser Technologie vollends an die Bedürfnisse neurowissenschaftlicher Forschungen anpassen (Mueller et al. 2011a; 2012b).

1.3 Integration virtueller Realitäten in neurokognitive Fragestellungen

Das große Problem bei der Integration von Virtuellen Umgebungen in neurokognitive Fragestellungen bzw. Versuchsaufbauten ist die technische Realisierung. Die Implementierung und Einbindung virtueller Stimulussysteme erfordert ein hohes Maß an informationstechnischen Expertenwissen sowie finanziellen und personellen Ressourcen, welche bei vielen Forschungsgruppen nicht zur Verfügung stehen (Craig et al.

2009, Kim 2005).

Um dennoch diese Anwendung zu nutzen, wurden in einzelnen Studien bereits diverse unveränderbare Umgebungen aus Computerspielen für neurowissenschaftliche Studien herangezogen (Abb. 1.1). Die so entstandenen neurowissenschaftlichen Stimuli beinhalten eine unveränderbare virtuelle Umgebung, deren Restriktionen das Paradigma des Experimentes nachhaltig bestimmen und damit genau auf die in der jeweiligen Studie zu untersuchende wissenschaftliche Fragestellung einschränken.

Da in diesem speziellen Forschungskontext nur wenige Forschungsgruppen überhaupt Zugang zu virtuellen Umgebungen haben, ergaben sich hieraus weitere Probleme. Vor allem die Reproduzierbarkeit und die Transparenz der neurowissenschaftlichen Ergebnisse wurden oftmals kritisch betrachtet (Beck et al. 2007).

Ein großer Kritikpunkt ist zudem die oftmals mangelhafte Dokumentation der technischen Umsetzung und der konkreten Einbindung von virtuellen Umgebungen. Meist sind die Struktur der virtuellen Umgebung und deren technische Details so spezifisch, dass die Reproduktion der wissenschaftlichen Ergebnisse ohne einen ähnlichen Versuchsaufbau mit derselben virtuellen Umgebung schlichtweg unmöglich ist.

Der dynamisch-interaktive Charakter erweiterter VR-Umgebungen erweist sich hier als großer Vorteil, birgt aber gerade für das experimentelle Design und die post-experimentelle statistische Auswertung von neurokognitiven Experimenten neue Gefahren.

Hier werden virtuelle Realitäten als eine völlig neue Stimulusgeneration angesehen. Im neurowissenschaftlichen Bereich orientierte sich die Anzeige von visuellen Stimuli stets an einer einfachen Präsentation alltagsnaher Prozesse (Savoy 1998).

Die authentischere und realere Stimuluspräsentation durch VR-Umgebungen widerspricht zwar dieser Regelung, verringert aber gleichzeitig die Gefahr, dass einfache Stimuli alltagsnahe Prozesse nur unzureichend abbilden.

Aufgrund dieser Komplexität von VR-Stimuli müssen für eine unkomplizierte, standardisierte Einbindung von virtuellen Umgebungen die Prozesse zur Experimentmodellierung überdacht sowie an die statistische Auswertung von neurokognitiven VR-Experimenten angepasst werden.

Hier stellen sogenannte Mensch-Computer-Schnittstellen (*Brain-Computer-Interface*), die eine Echtzeit-Stimulusanpassung anhand der Ergebnisse einer statistischen Echtzeit-Datenauswertung vornehmen könnten, eine besondere Herausforderung dar.

Durch sogenannte Neurofeedback-Techniken, bei denen Probanden ihre lokale Hirnaktivität mit Hilfe gezielter operanter Konditionierung selbstregulieren, können computergenerierte Charaktere (Avatare) allein durch die Ableitung der neuronalen Hirnaktivierungen gesteuert werden.

Dies verspricht eine erhöhte Motivation und stärkere Interaktion des Probanden mit dem Versuchparadigma, was in diversen Studien zur Erforschung des räumlichen Gedächtnisses oder der topographischen Navigation bereits genutzt wurde (Maguire 1998, Burgess et al. 2002, Pine et al. 2002, Mueller et al. 2012b).

Nach unserem besten Kenntnisstand ist ein derartiges System bisher weder kommerziell zu erwerben noch in entsprechenden Studien dokumentiert wurden. Die Realisierung eines frei nutzbaren, skalierbaren und multimodalen VR-Stimuluskonzepts würde basierend auf den genannten Anforderungen einen entscheidenden Beitrag zur Verbreitung von Virtuellen Umgebungen in neurokognitiven Studien darstellen.

Dieses Konzept sollte in einer gut strukturierten Anwendung durch eine weitverbreitete Programmiersprache realisiert werden und zudem mehrere virtuelle Umgebungen bereitstellen, die vom Anwender sofort nach Erhalt der Anwendung durch ein komfortables *Drag & Drop*-Konzept genutzt werden können.

1.4 Zielstellung

Ausgehend von diesen Überlegungen soll als Ziel eine VR-Stimulusapplikation entworfen und entwickelt werden, welche die Präsentation und die Interaktion von hochrealistischen, dreidimensionalen Umgebungen erlaubt. Das immersive Verhalten von Probanden soll dabei in besonderem Maße durch entsprechende Implementierungsaspekte unterstützt und ermöglicht werden.

Die Anwendung soll weiterhin die Modellierung neurokognitiver VR-Experimente auf Basis der Block- bzw. der ereigniskorrelierten Experimentstruktur erlauben sowie mehrere vordefinierte 3D-Szenarien bereitstellen, welche ohne weitere programmier-technische Anpassungen in eine neurowissenschaftliche Experimentstruktur integriert werden können.

Für die Präsentation von Stimuli soll das *Vision Engine Framework* der Firma Havok, Inc. (Havok Inc. 2012) in die Anwendung eingebettet werden. Die Erstellung einer anwenderorientierten und -akzeptierten Benutzerschnittstelle ist dabei ein oft unterschätzter Aspekt, welcher im Rahmen dieses medizintechnischen Projektes in besonderem Maße beachtet werden soll. Die erfolgreiche Bearbeitung der Arbeit beinhaltet zwei Hauptziele:

- Planung, Durchführung und Auswertung einer Proof-of-Principle Studie mit der zu entwickelnden VR-Stimulusapplikation, bei der die neuronalen Mechanismen der räumlichen Navigation sowie des topographisch-episodischen Gedächtnisses in Kombination mit Immersionseffekten untersucht werden
- Entwicklung und Evaluierung eines bis dato neuen Echtzeit-VR-Adaptionskonzeptes zur Anwendung in einer Echtzeit-fMRT Hirn-Computer-Schnittstelle

Zur erfolgreichen Realisierung müssen zunächst folgende technischen Konzepte erarbeitet werden:

- Modellierung und Entwurf einer multimodalen VR-Stimulusapplikation, welche durch ein entsprechendes Architektur- und Kommunikationskonzept zukünftige Erweiterungen und die Integration in verschiedene Akquirierungsmodalitäten, z.B. EEG, fMRT und Echtzeit-fMRT, unterstützt
- Unter Berücksichtigung der post-experimentellen statistischen Auswertung, sollen VR-Stimuli hinsichtlich ihrer neuartigen Verwendung in neurowissenschaftlichen Experimentabläufen analysiert und durch angepasste Block- und ereigniskorrelierte Experimentstrukturen in das zu entwickelnde Konzept der VR-Stimulusapplikation aufgenommen werden

- Entwicklung und Implementierung einer Benutzerschnittstelle, welche sich insbesondere an den anwenderorientierten Bedürfnissen sowie gängigen Strukturen neurowissenschaftlicher Applikationen orientiert und neue Funktionalitäten aspektorientiert und ortsnahe zu relationalen Funktionen anordnet; entsprechende Sicherheitsroutinen sollen Experimentparameter syntaktisch und semantisch auf mögliche Inkonsistenzen überprüfen
- Entwicklung, Realisierung und Evaluierung eines neuartigen Echtzeit-Adaptionskonzeptes zur unmittelbaren Anpassung von 3D-Szenarien anhand neuronaler Aktivitäten aus multiplen Hirnregionen; dies setzt die Entwicklung eines entsprechenden Kommunikationsframeworks sowie die Nutzung der VR-Stimulusapplikation in Neurofeedback-Experimenten voraus.
- Validierung und Evaluierung der VR-Stimulusapplikation mit Hilfe einer fMRT-Studie, in welcher ein virtuelles Stadt-Szenario erforscht und enkodiert werden soll. Die VR-Szene soll frei verfügbar und austauschbar sein. Die Daten dieser Studie sollen in einer post-experimentellen Simulation weiterverwendet und zur Validierung der adaptiven Hirn-Computer-Schnittstelle genutzt werden. Die Evaluierung des Gesamtkonzeptes und seiner Einbindung in die MR-Infrastruktur soll außerdem die Durchführung von Performanz-, Stress- und Anwendertests sowie eine prä-experimentelle Modellierung des Experiments durch die zu entwickelnde VR-Stimulusapplikation einschließen.

Die informationstechnische Modellierung und Realisierung der Applikation sowie die Durchführung der *proof-of-principle* Grundsatzstudie mit neurowissenschaftlichen Inhaltsschwerpunkten sollen im Weiteren den interdisziplinären Charakter dieser Arbeit hervorheben.

Hier müssen für die Evaluierung des neuartigen Echtzeit-Adaptionskonzeptes spezielle Sicherheitsmaßnahmen hinsichtlich der Probanden getroffen werden. Ein Vergleich der Studienergebnisse mit den Ergebnissen aus vergleichbaren Studien soll zudem die Funktionsfähigkeit und die zuverlässige Reproduktion neurowissenschaftlicher Ergebnisse durch die VR-Stimulusapplikation sicherstellen.

1.5 Aufbau der Arbeit

Während Kapitel 1 einen kurzen Überblick über Motivation und Zielstellung gibt, liefert Kapitel 2 eine Einführung zur kernmagnetischen Resonanz. Hier stehen die allgemeinen physikalischen aber auch die technischen Grundlagen der Magnetresonanztomographie im Mittelpunkt. Des Weiteren werden die zum Verständnis der Arbeit wichtigen Grundlagen der funktionellen Magnetresonanztomographie präsentiert und im weiteren Verlauf bis hin zur Echtzeit-fMRT vorgestellt.

Anschließend werden in Kapitel 3 die Besonderheiten sogenannter Hirn-Computer-Schnittstellen beschrieben. Im Zentrum dieses Kapitels stehen die Motivation, die Formen sowie die verschiedenen Anwendungsgebiete dieser Technologie.

Die informationstechnischen Anforderungen virtueller Realitäten für Hirn-Computer-Schnittstellen werden in Kapitel 4 eingeführt.

Dazu wird im besonderen auf die Definition und Geschichte virtueller Umgebungen, auf die Grundlagen der Mensch-Computer-Interaktion, auf die Kollisionsdetektion in virtuellen Umgebungen und auf das physikalisch-basierte Rendern von 3D-Szenarien eingegangen.

Im Hinblick auf die geplante Proof-of-Principle Studie, bei der die neuronalen Mechanismen der räumlichen Navigation sowie des topographisch-episodischen Gedächtnisses zusammen mit Immersion untersucht werden, beschäftigt sich Kapitel 5 mit den psychologischen und neurophysiologischen Grundlagen von Memorisierungsvorgängen. Ausgehend von der Zielstellung der Arbeit beschäftigt sich dieses Kapitel zudem mit der neurokognitiven Anwendung von VR-Umgebungen und der Analyse des Immersionseffektes, der im weiteren Verlauf grundlegend erörtert wird.

Aufbauend auf den Prinzipien des Softwareentwicklungsprozesses wird in Kapitel 6 die Problemstellung der Arbeit analysiert und die Anforderungen an die Software abgeleitet. Zusätzlich dazu werden vergleichbare Softwareanwendungen vorgestellt sowie deren Vorzüge und Besonderheiten evaluiert.

Anhand der abgeleiteten Anforderungen wird in Kapitel 7 der Entwurf der Arbeit spezifiziert. In Abhängigkeit der informationstechnischen Grundlagen wird hier die strukturelle Planung der Gesamtanwendung sowie die konzeptuelle Planung der verschiedenen Kernfunktionalitäten vorgestellt.

Mit Kapitel 8 dieser Arbeit wird die Implementierung dieser VR-Stimulusanwendung beschrieben. Ausgehend von der strukturellen Planung in Kapitel 7 werden die konkreten Funktionalitäten der Anwendung sowie deren Aufteilung im ausgewählten Architektur-Modell präsentiert.

Die Verwendung einer VR-Stimulusanwendung bietet weitreichende Möglichkeiten zur Durchführung neurowissenschaftlicher Experimente. Zwei praktische Anwendungsmöglichkeiten dieser VR-Stimulusapplikation, welche zudem die Funktionalitäten der Applikation unter Experimentbedingungen untersuchen, werden durch die Präsentation der Pilotstudie in Kapitel 9 vorgestellt. In dieser ersten VR-fMRT-Studie werden Netzwerke zur räumlichen Orientierung sowie deren Zusammenhänge mit dem topographischen und dem episodischen Gedächtnis analysiert und diskutiert. Im Vordergrund steht dabei die Frage, ob die realisierte Anwendung den Anforderungen einer VR-Stimulusanwendung gerecht wird und daher für neurowissenschaftliche Studien eingesetzt werden kann.

Die zusammenfassende Beschreibung der Arbeit sowie einen Ausblick auf zukünftige Erweiterungen schließen diese Arbeit ab.

Kapitel 2

Grundlagen der kernmagnetischen Resonanz und der fMRT

In diesem Kapitel sollen die zum Verständnis notwendigen Grundlagen der Magnetresonanztomographie näher vorgestellt werden. Im ersten Teil wird kurz auf die physikalischen Grundlagen der kernmagnetischen Resonanz eingegangen. Anschließend folgen die Grundzüge der funktionellen Magnetresonanztomographie (fMRT), welche in der vorliegenden Arbeit durch die sogenannte BOLD-fMRT, der experimentellen Modellierung von fMRT-Experimenten und deren statistische Auswertung repräsentiert werden. Zum Abschluss dieses Kapitels werden die Besonderheiten der sogenannten Echtzeit-fMRT erläutert und die wichtigsten statistischen Verfahren zur Echtzeit-Auswertung funktioneller MRT-Daten vorgestellt.

2.1 Allgemeine Grundlagen der Magnetresonanztomographie

Die Anfänge der medizinischen Bildgebung mittels Magnetresonanztomographie gehen im Wesentlichen auf die 70er Jahre des letzten Jahrhunderts zurück. Vor allem Paul C. Lauterbur und Peter Mansfield lieferten wichtige technische Beiträge zur humanmedizinischen Nutzung der MR-Bildgebung (Horowitz 1995, Haacke 1999, Huettel et al. 2008). Paul C. Lauterbur gelang dabei das Legen zweier wesentlicher Grundsteine. Der erste Grundstein war die Einführung sogenannter magnetischer Gradientenfelder, um eine räumliche Zuordnung der Signale innerhalb einer Probe zu erreichen. Der zweite Grundstein war die Präsentation einer Methode zur Rotation des ortskodierenden Magnetfeldgradienten, wodurch letztendlich unterschiedliche Teilabbildungen eines Objektes erzielt und am Ende zu einem Gesamtbild rekonstruiert werden konnten (Projektions-Rekonstruktions-Verfahren) (Lauterbur 1973). Peter Mansfields entscheidender Anteil zur Nutzung der heutigen Magnetresonanztomographie bestand in der Entwicklung von Techniken zur schichtselektiven Anregung und der Entwicklung eines mathematischen Verfahrens, welches die Umwandlung der Signale in Bildinformationen erheblich beschleunigt (Mansfield 1977). Des Weiteren führte er die Verwendung schneller Gradienten für sogenannte Hochgeschwindigkeitsmessungen (Echopla-

nare Bildgebung) ein. Dies lieferte die Grundlage für heutige kognitive Forschungen im Bereich der funktionellen MRT.

Die Stärken der Magnetresonanztomographie liegen vor allem in der hohen räumlichen Auflösung der Bilder sowie im sehr guten Weichteilkontrast. Besonders bei der funktionellen MRT können Aktivierungen in den Hirnarealen nichtinvasiv auf einige Millimeter genau ermittelt werden. Dies unterscheidet die Magnetresonanztomographie von anderen Messmethoden, wie zum Beispiel der Elektroenzephalographie. Im Gegenzug kann die Magnetresonanztomographie nur durch größeren Aufwand die neuronalen Aktivierungen in Echtzeit ermitteln. Echtzeit-fMRT ist gerade deshalb problematisch, weil die Verarbeitung der Bildinformationen und das Erstellen der Bilder einen hohen zeitlichen Aufwand erfordert und Anwendungen nur bedingt oder nur mit einer gewissen Zeitverzögerung ablaufen können (Cox et al. 1995, Weiskopf et al. 2007). Dieses Kapitel soll hauptsächlich die zum Verständnis notwendigen Grundlagen vermitteln und erhebt nicht das Prädikat der Vollständigkeit. Für weiterführende Informationen zu diesem Thema sei auf (Horowitz 1995, Haacke 1999, Huettel et al. 2008, Schneider und Fink 2007) verwiesen.

2.1.1 Kernmagnetische Resonanz

Eine grundlegende physikalische Eigenschaft von Protonen, Elektronen und Neutronen ist der für sie typische Drehimpuls, der oft auch als Spin I bezeichnet wird. Zur medizinischen Bildgebung mittels Magnetresonanztomographie nutzt man hier im Speziellen die Grundlagen der kernmagnetischen Resonanz (NMR) bzw. die Spineigenschaften des Atomkerns. Nur Kerne mit einem Spin I ungleich Null sind für NMR-Untersuchungen nutzbar (Haacke 1999). Neben zahlreichen anderen Elementen, finden in der NMR-Spektroskopie vor allem das Wasserstoff-Isotop (1H), das Kohlenstoff-Isotop (^{13}C), das Fluor-Isotop (^{19}F) und das Natrium-Isotop (^{23}Na) ihre Anwendung. Für die medizinische MR-Bildgebung sind Messungen des 1H -Nukleus mit $I = \frac{1}{2}$ von besonderem Interesse. Wasserstoff als wesentlicher Bestandteil von Wasser und Fett, ist aufgrund seiner hohen Konzentration im menschlichen Gewebe besonders gut zu detektieren (Haacke 1999, Jezzard et al. 2001). Zur besseren Veranschaulichung beziehen sich alle nachfolgenden Erläuterungen auf das Wasserstoff-Isotop 1H . In klassischen physikalischen Betrachtungen wird der Spin als Eigendrehimpuls \vec{J} eines Teilchens betrachtet. Die dadurch hervorgerufene Drehung des Protons erzeugt dadurch einen Kreisstrom, aus dem wiederum ein magnetisches Feld entsteht (Abb. 2.1). Jeder (positive) Atomkern mit Spin besitzt dadurch eine vektorielle Größe, das magnetische Moment $\vec{\mu}$, welches mit anderen Magnetfeldern in Wechselwirkung treten kann (Horowitz 1995, Haacke 1999). Es ist durch folgende Gleichung definiert:

$$\vec{\mu} = \gamma \cdot \vec{J}. \quad (2.1)$$

Das gyromagnetische Verhältnis γ repräsentiert hier die Proportionalitätskonstante zwischen dem Eigendrehimpuls und dem magnetischen Moment. Bei der makroskopischen Betrachtung eines Spin-Ensembles innerhalb einer Probe zeigt sich, dass die Summe aller magnetischen Momente Null ergibt. Die Ausrichtung der Spins ist da-

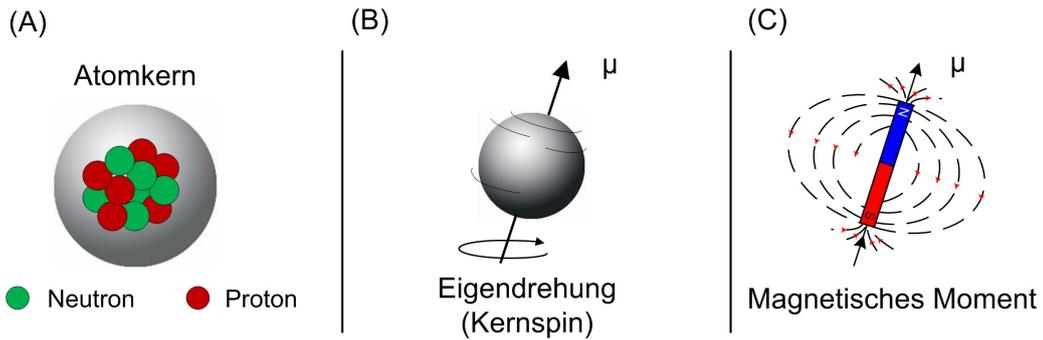


Abbildung 2.1: In einer vereinfachten Darstellung setzt sich ein Atomkern aus Neutronen und Protonen zusammen (A). Dabei besitzt beispielsweise der Wasserstoff-Nukleus einen als Kernspin bezeichneten Eigendrehimpuls (B). Der durch die Bewegung des Teilchens hervorgerufene Kreisstrom erzeugt wiederum ein magnetisches Feld, welches in Grundzügen dem eines Stabmagneten ähnelt (C). Eigener Entwurf nach (Horowitz 1995).

bei zufällig gleichverteilt. Durch das Anlegen eines äußeren Magnetfeldes \vec{B}_0 , entlang der z-Achse, wird diese Ausrichtung gebrochen (Brown und Semelka 2010). Aus der Wechselwirkung zwischen magnetischem Moment $\vec{\mu}$ und dem Magnetfeld \vec{B}_0 entsteht eine Energiebeziehung:

$$E = -\vec{\mu} \cdot \vec{B}_0 = -\gamma \cdot \vec{J}_z \cdot \vec{B}_0. \quad (2.2)$$

Die dadurch entstehende Wechselwirkung zwischen magnetischem Moment $\vec{\mu}$ und dem Magnetfeld \vec{B}_0 führt zu einer Energiedifferenz ΔE . Diese als *Zeeman-Aufspaltung* bekannte Gesetzmäßigkeit, beschreibt dabei das Aufspalten der Energieniveaus abhängig von der Kernspinzahl des zu untersuchenden Isotops (Horowitz 1995, Haacke 1999). Die Wirkung des äußeren Magnetfeldes auf die Spinausrichtung kann dabei mit Hilfe einer *Pauli-Spinmatrix* dargestellt werden. Bezogen auf das Wasserstoff-Isotop 1H mit einem Spinraum von $H_S = \frac{1}{2}$ können die Spins zwei präparierte Zustände annehmen. Die Differenz der beiden Zustände ist durch folgende Gleichung definiert:

$$\Delta E = \gamma \cdot \left(\frac{h}{2\pi} \right) \vec{B}_0. \quad (2.3)$$

Die Variable h repräsentiert dabei das *Plancksche Wirkungsquantum*, welches das Verhältnis zwischen Energie und Frequenz eines Teilchens definiert (Haacke 1999). Das genaue Verhältnis sowie die Anzahl der Kernspins die jeweils einen der beiden Zustände besetzen, wird dabei durch die *Gibbs-Boltzmann-Verteilung* der Thermodynamik beschrieben (Horowitz 1995, Haacke 1999):

$$\frac{n_{hoch}}{n_{niedrig}} = e^{-\frac{\Delta E}{k \cdot T}}. \quad (2.4)$$

Ausgehend vom Wasserstoff-Kern ergibt sich durch die ungerade Anzahl der Kernteilchen ein leichter Überschuss von parallel ausgerichteten Spins mit energetisch niedrigerem Zustand. Ein kleinerer Anteil von Spins nimmt den energetisch höheren Zustand

an und richtet sich antiparallel zum Magnetfeld aus. In diesem Zusammenhang definiert die Variable k die Boltzmann-Konstante und die Variable T die absolute Temperatur. Bei Raumtemperatur entsteht so der sehr kleine Überschuss von parallel zum Magnetfeld ausgerichteten Spins, welcher durch Anlegen stärkerer Magnetfelder erhöht werden kann. Die Mittelung aller magnetischen Momente ergibt nun nach dem Anlegen des Magnetfeldes einen Wert ungleich Null und damit die für die Magnetresonanztomographie wichtige makroskopische Magnetisierung M_0 (Horowitz 1995, Schneider und Fink 2007). Ähnlich wie bei Elektronenübergängen im Atom können Übergänge zwischen Zuständen nur durch Energieabsorption oder durch externe Strahlungsemissionen erfolgen. Ausgehend davon wird zur Nutzung dieser quantenmechanischen Betrachtung ein zeitlich veränderliches, elektromagnetisches Feld eingestrahlt, welches die Energiedifferenz ΔE zum Wechsel der Energiezustände bereitstellt (Haacke 1999). Die zur Energiedifferenz ΔE korrespondierende Frequenz ergibt sich aus der folgenden Gleichung und ist auch als Larmorfrequenz ω bekannt:

$$\omega = \gamma \cdot \left| \vec{B}_0 \right|. \quad (2.5)$$

Durch das Anlegen eines äußeren Magnetfeldes \vec{B}_0 ergibt sich ein Drehmoment auf das magnetische Moment $\vec{\mu}$. Dieses bewirkt eine Präzessionsbewegung der Spins um die, durch Feldlinien beschriebene, Richtung des Magnetfeldes \vec{B}_0 . Diese Kreisbewegung erfolgt mit der sogenannten Larmorfrequenz ω und richtet sich nach der Stärke des Magnetfeldes sowie dem gyromagnetischen Verhältnis γ . Bei einer Magnetfeldstärke von 1,5 Tesla präzedieren die Spins von Wasserstoffprotonen mit einer gleichverteilten Phasenlage und einer Larmorfrequenz von ca. 64 MHz.

Zum Verständnis ist wichtig, dass die Spins asynchron zueinander, um die Richtung des Magnetfeldes kreiseln und dass in diesem Zustand noch kein für die Magnetresonanztomographie messbares Signal entsteht. Die Spins halten vielmehr ein energetisches Gleichgewicht aufrecht.

2.1.2 Relaxationseigenschaften

Um ein messbares Signal für die medizinische Bildgebung zu erhalten, muss die makroskopische Magnetisierung mit Hilfe eines hochfrequenten elektromagnetischen Pulses aus ihrer Gleichgewichtslage gelenkt werden. Die Frequenz des Pulses entspricht dabei der Larmorfrequenz. Über die Einstrahldauer und die Leistung wird die exakte Auslenkung der makroskopischen Magnetisierung M_0 angegeben, welche abhängig von der Ausrichtung des Feldes auch als Längsmagnetisierung M_z bezeichnet wird (Abb. 2.2). Die magnetische Komponente, die aus der Einstrahlung des Pulses entsteht, wird als \vec{B}_1 -Feld beschrieben. Der Winkel dieser Auslenkung wird allgemein als Flip-Winkel definiert (Haacke 1999, Jezzard et al. 2001). Ausgehend davon wird zur Auslenkung der makroskopischen Magnetisierung (M_0) in die xy-Ebene ein 90° -Puls benötigt. Die nach der Auslenkung der Spins in die xy-Ebene resultierende Magnetisierung wird als Transversalmagnetisierung M_{xy} bezeichnet. In der Empfangsspule wird durch diese Transversalmagnetisierung eine zeitlich veränderliche Spannung induziert,

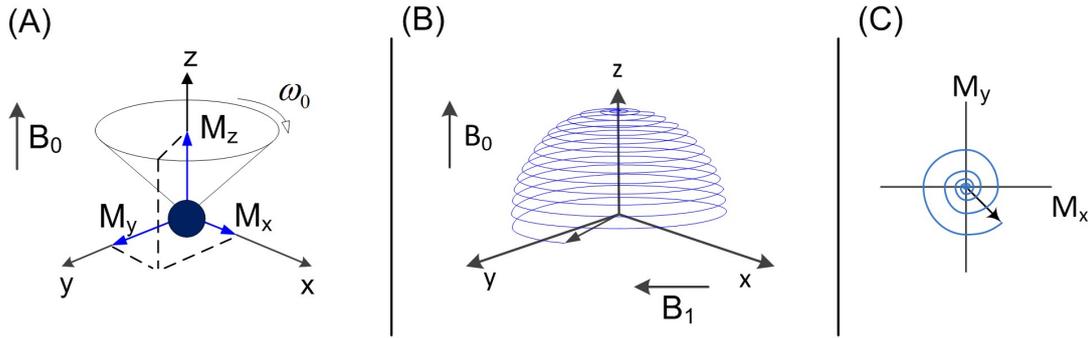


Abbildung 2.2: Ohne \vec{B}_1 -Feld präzedieren die Spins parallel um die Richtung des Magnetfeldes \vec{B}_0 (A). Durch Einstrahlen der Quermagnetisierung \vec{B}_1 wird die makroskopische Magnetisierung aus ihrer Ruhelage gelenkt und es entsteht eine Phasenkohärenz. Nach Abschalten des Hochfrequenz-Pulses kehren die Spins durch eine spiralförmige Bewegung in ihren Ausgangszustand zurück (B,C). Die Bewegungskurve wird durch die Bloch-Gleichungen beschrieben. Eigener Entwurf nach (Brown und Semelka 2010).

welche über das *Faradaysche Induktionsgesetz* als Signal gemessen werden kann (Jeppard et al. 2001). Ausgehend von den Gesetzmäßigkeiten des *thermischen Gleichgewichts*, kehren die Spins nach dem Abschalten des eingestrahlten Hochfrequenz-Pulses aus ihrer Auslenkung zurück in den Ausgangszustand und bilden so wieder eine parallel zum Hauptfeld ausgerichtete Magnetisierung. Dieser Prozess erfolgt exponentiell wobei die Relaxation der Transversalmagnetisierung T_2 in der Regel deutlich schneller erfolgt als die longitudinale Relaxation T_1 (Haacke 1999, Jeppard et al. 2001, Huettel et al. 2008). Die durch die Anregung aufgenommene Energie wird während des Relaxationsprozesses an die Umgebung abgegeben.

Eine theoretische Beschreibung dieses paramagnetischen Resonanzeffekts über alle drei Raumrichtungen liefern die Bloch-Bewegungsgleichungen (Bloch et al. 1946):

$$\frac{dM_x(t)}{dt} = \gamma \cdot (M(t) \times B(t))_x - \frac{M_x(t)}{T_2} \quad (2.6)$$

$$\frac{dM_y(t)}{dt} = \gamma \cdot (M(t) \times B(t))_y - \frac{M_y(t)}{T_2} \quad (2.7)$$

$$\frac{dM_z(t)}{dt} = \gamma \cdot (M(t) \times B(t))_z - \frac{M_z(t) - M_0}{T_1}. \quad (2.8)$$

Abhängig von den beschriebenen Wechselwirkungen der Spins mit ihrer Umwelt, werden die Relaxationsprozesse in zwei Arten unterteilt. Die Spin-Gitter Relaxation beschreibt die Energieabgabe der Spins an die Nachbarmoleküle und gibt damit die Zeit des Wiederanstiegs der Längsmagnetisierung M_z an.

Wichtig für die Längsrelaxation ist die sogenannte Repetitionszeit T_R , welche die zeitliche Abfolge der hochfrequenten Pulse beschreibt. Wird die Repetitionszeit im Verhältnis zum T_1 -Wert zu klein gewählt, so bildet sich nur ein Teil der Längsmagnetisierung zurück und das MR-Signal schwächt ab. Diese als T_1 -Sättigung beschriebene Bildwichtung wird gezielt zum Abbilden von Gewebe mit entsprechend gewichteten

Pulssequenzen verwendet (Wuestenberg et al. 2005).

Die zeitliche Abhängigkeit der Längsmagnetisierung nach dem Anregen mit dem hochfrequenten Puls ergibt sich aus Gleichung (2.8):

$$M_z(t) = M_0 \cdot \cos \alpha + M_0 \cdot (1 - \cos \alpha) \cdot (1 - e^{-\frac{t}{T_1}}). \quad (2.9)$$

Die Spin-Spin Relaxation beschreibt die energetischen Einflüsse der Spins untereinander und damit den Zerfall der Transversalmagnetisierung M_{xy} (Abb. 2.3). Der Verlust der Phasenkohärenz bzw. das Dephasieren der Spins geschieht um so schneller, je stärker benachbarte Spins miteinander wechselwirken. Genau wie T_1 ist T_2 molekül-, material- bzw. gewebespezifisch. T_2 ist als die Zeit definiert, nach der 63% der transversalen Magnetisierung M_{xy} wieder zerfallen sind (Haacke 1999, Dössel 2000, Jezard et al. 2001). Neben den vorgestellten Einflüssen sind weitere Faktoren für den

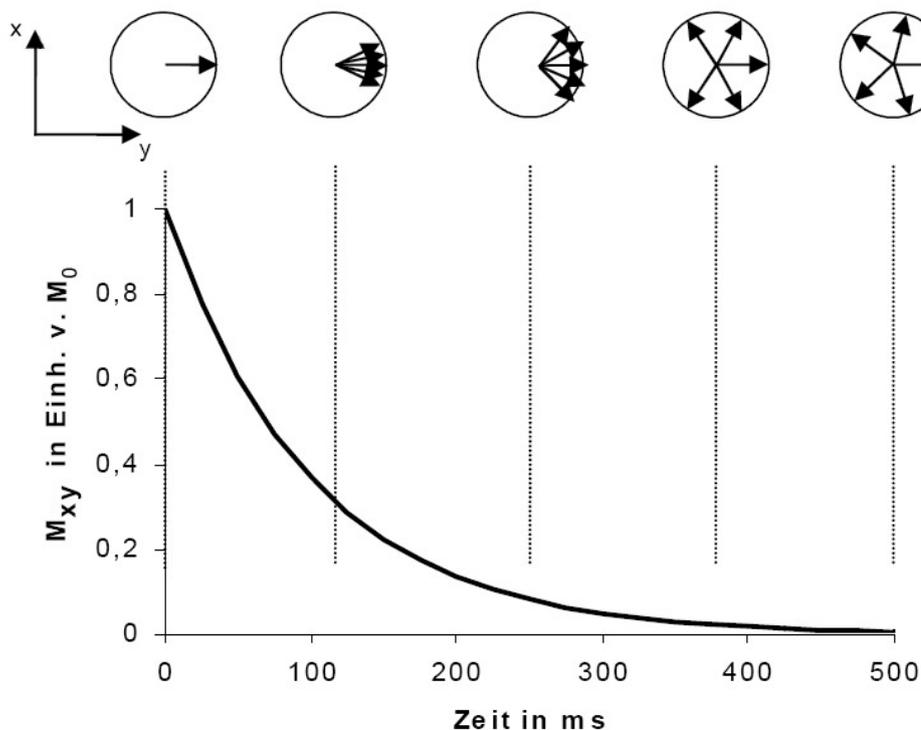


Abbildung 2.3: Die Transversalmagnetisierung zerfällt aufgrund des Verlustes der Phasenkohärenz durch Spin-Spin Wechselwirkungen. Nach (Jezard et al. 2001).

Zerfall der Phasenkohärenz verantwortlich. Lokale Magnetfeldänderungen, Feldinhomogenitäten oder Suszeptibilitätsunterschiede beschleunigen den Zerfall der Phasenkohärenz und sorgen für ein schnelleres Dephasieren der Spins. Die genannten Einflüsse werden allgemein als T_2' zusammengefasst (Dössel 2000, Jezard et al. 2001). Die Zeit der effektiven transversalen Relaxation wird mit T_2^* angegeben und ist wie folgt definiert:

$$\frac{1}{T_2^*} = \frac{1}{T_2} + \frac{1}{T_2'}. \quad (2.10)$$

Allgemein gilt:

$$T_2^* < T_2 < T_1. \quad (2.11)$$

2.1.3 Ortskodierung

In den letzten Kapiteln wurde die Magnetresonanz als Grundlage der medizinischen Bildgebung am Beispiel der MRT vorgestellt. Die Erzeugung des eigentlichen MR-Bildes wurde dabei noch weitestgehend außer Acht gelassen. In diesem Unterkapitel werden die Techniken zur MR-Bilderzeugung mit einer 2D-Bildgebungssequenz beschrieben und die Signalgewinnung aus kleinen, vorher festgelegten Volumina einer Probe erläutert. Diese Volumina, auch als Voxel bezeichnet, stehen für das dreidimensionale Pendant zum Pixel. Die Kodierung von Rauminformationen mittels magnetischer Gradientenfelder wird dabei in drei, zeitlich aufeinander folgende, Schritte unterteilt.

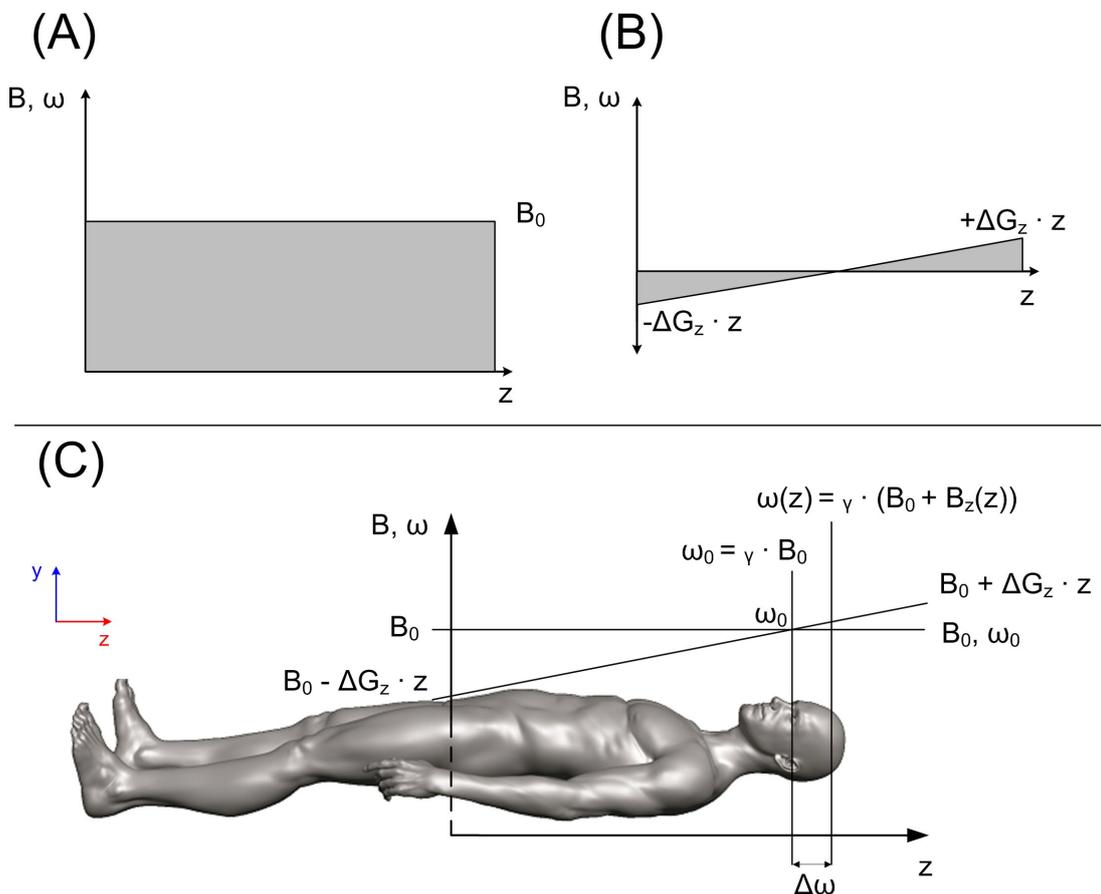


Abbildung 2.4: Die ursprüngliche Verteilung des Magnetfeldes B_0 (A) wird durch Zuschalten des Schichtselektionsgradienten G_z modifiziert (B). Dabei beeinflusst der Gradient die Magnetfeldstärke B_z sowie die lokale Resonanzfrequenz ω_z in z -Richtung (C). Eigener Entwurf nach (Haacke 1999).

Schichtselektion

Der erste Schritt zur Erzeugung von Schichtbildern in der MR-Bildgebung ist die schichtselektive Resonanzanregung. Dabei werden durch das Einstrahlen von schichtselektiven hochfrequenten Pulsen festgelegter Bandbreite (Radiofrequenz-Pulse) lediglich die Spins innerhalb einer definierten Schicht aus ihrer Gleichgewichtslage ausgelenkt. Als Anregungspulse werden sogenannte *sinc-Pulse* verwendet (Dössel 2000, Jezzard et al. 2001). Diese werden durch die Faltung einer monochromatischen Radiofrequenz mit einem rechteckigen Zeitfenster erzeugt (Brown und Semelka 2010). Durch das synchrone Zuschalten des Schichtselektionsgradienten G_z kommt es zu einer ortsabhängigen, linearen Veränderung des Magnetfeldes B_z :

$$\vec{B}(z) = \vec{B}_0 + \Delta G_z \cdot z. \quad (2.12)$$

Ausgehend von der Berechnung der Larmorfrequenz in Gleichung (2.5) wurde bereits festgestellt, dass die Präzessions- oder Larmorfrequenz ω der Spins in direkter Proportionalität zur Stärke des Magnetfeldes steht. In anderen Worten, die Larmor-Gleichung erlaubt durch Multiplikation des gyromagnetischen Verhältnis γ mit der Magnetfeldstärke B_0 die ortsabhängige Bestimmung der lokalen Resonanzfrequenz $\omega(z)$. Zusammen mit dem Schichtselektionsgradienten G_z ergibt sich folgende Beziehung:

$$\omega(z) = \gamma \cdot (\vec{B}_0 + B_z(z)). \quad (2.13)$$

Die außerhalb der definierten Schicht vorherrschenden Frequenzunterschiede variieren so stark von der eingestrahlten Frequenz ω_0 des hochfrequenten Pulses, dass die Resonanzbedingung nicht länger erfüllt ist. Ausgehend von den verschiedenen Präzessionsfrequenzen erlaubt dieser Umstand direkte Rückschlüsse auf die Lage der Spins entlang der z-Achse.

Der hochfrequente Puls nutzt hierfür keine einzelne Frequenz, sondern strahlt eine festdefinierte Bandbreite verschiedener Frequenzen $\Delta\omega$ ein (Haacke 1999, Jezzard et al. 2001). Ausgehend von dieser Bandbreite erhält die angeregte Schicht eine endliche Breite, welche durch das Frequenzspektrum des Pulses und der Stärke des Gradienten definiert wird (Abb. 2.4). Die Richtung und Stärke des Gradienten bestimmen dabei die Ausrichtung der Schicht. Des Weiteren können Breite und Position direkt über die Frequenzjustage sowie über die Form des hochfrequenten Pulses definiert werden.

Die Grenzen für hochauflösende Messungen werden vor allem durch die technischen Beschränkungen der Gradienten diktiert. Hier steht die Möglichkeit zur Messung von dünnen bzw. hochauflösenden Schichten in direkter Proportionalität zur Leistungsfähigkeit der Gradienten. Im Unterschied zu 2D-Sequenzen verwenden 3D-Sequenzen den hochfrequenten Puls, um das gesamte Volumen anzuregen. Die räumliche Zuordnung in z-Richtung wird dabei nicht durch eine schichtselektive Resonanzanregung übernommen, sondern verwendet eine zusätzliche Phasenkodierung (Schneider und Fink 2007).

Frequenz- und Phasenkodierung

Nach der Schichtselektion muss nun die Raumlage der einzelnen Volumeneinheiten in xy-Richtung innerhalb der ausgewählten Schicht bestimmt werden. Legt man während

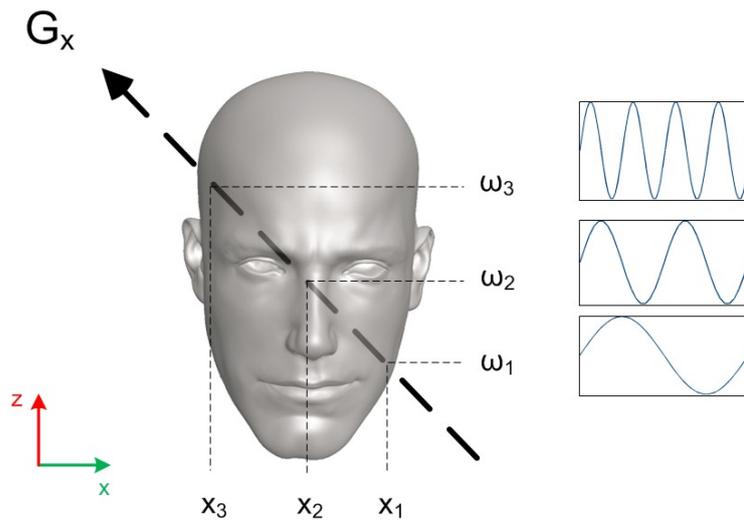


Abbildung 2.5: Mit Hilfe des Frequenzkodiergradienten wird die Position der Spins in x-Richtung bestimmt. Der Pfeil beschreibt die Wirkung des Frequenzkodiergradienten. Eigener Entwurf nach (Huettel et al. 2008).

des Auslesens des Signals einen zeitlich konstanten Gradienten an, der orthogonal zum Schichtselektionsgradienten, z.B. in x-Richtung, ausgerichtet ist, präzedieren die Spins entlang dieses Frequenzkodiergradienten mit unterschiedlichen Frequenzen (Abb. 2.5). Das gemessene komplexe Signal aus mehreren Frequenzen kann anschließend durch eine Fourieranalyse in seine Anteile zerlegt und damit räumlich in x-Richtung zugeordnet werden (Haacke 1999, Jezzard et al. 2001). Für die Kodierung entlang der y-Richtung wird nach dem Schichtselektionsgradienten aber vor dem Frequenzkodiergradienten der sogenannte Phasenkodiergradient geschaltet, welcher orthogonal zu diesen beiden Gradienten ausgerichtet ist. Dieser wird lediglich für eine bestimmte Zeit zugeschaltet, lässt aber dadurch die Spins entlang der y-Richtung mit unterschiedlicher Frequenz präzedieren. In Abhängigkeit zur Position in y-Richtung ergeben sich nach dem Abschalten des Phasenkodiergradienten unterschiedliche Phasenlagen (Abb. 2.6). Um ein zweidimensionales Bild mit einer Auflösung von 256×256 aufzunehmen, muss die Phasenkodierung mit unterschiedlicher Amplitude des Phasenkodiergradienten 256 mal wiederholt werden. Auch hier erfolgt die räumliche Zuordnung mit Hilfe einer Fouriertransformation. Der konjugierte Bildraum dieser Raumfrequenzen, auch als k-Raum benannt, wird durch die vorgestellten Verfahren Zeile für Zeile gefüllt. Dabei ist jedes Element des k-Raumes eindeutig durch die Phasenausrichtung und die Frequenz kodiert. Durch die anschließende zweidimensionale Fouriertransformation können die gemessenen Signale den Bildpunkten (Pixel) einer Schicht eindeutig zugeordnet werden. Es entsteht ein zweidimensionales MR-Bild dessen Grauwerte einen Rückschluss auf das gemessene MR-Signal zulassen. Für vertiefende Informationen zu den einzelnen Frequenz- und Phasenkodierschritten wird auf (Horowitz 1995, Haacke 1999, Dössel 2000, Buxton 2003, Huettel et al. 2008) verwiesen.

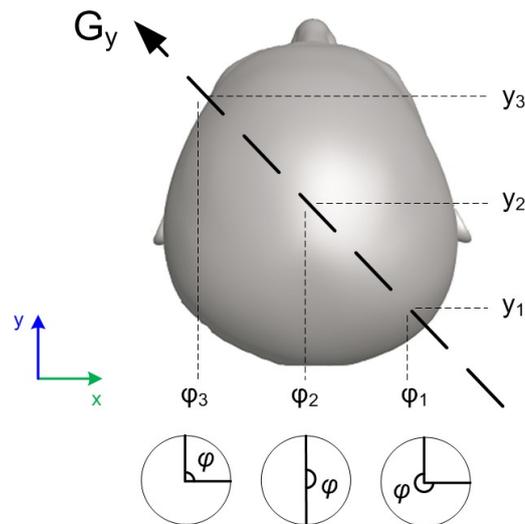


Abbildung 2.6: Die Kodierung der Spins in y-Richtung erfolgt über den Phasenkodiergradienten. Der Pfeil verdeutlicht die Richtung des Gradienten. Eigener Entwurf nach (Horowitz 1995)

2.1.4 Bildkontraste und Pulssequenzen

Ein wichtiger Aspekt bei der MR-Bildgebung ist der Kontrast der Bilder. Verschiedene Faktoren spielen dabei eine Rolle: die Protonendichte ρ , die bereits vorgestellte Längsrelaxationszeit T_1 sowie die Querrelaxationszeiten T_2 bzw. T_2^* . Zur Wichtung der jeweiligen Faktoren werden bei MRT-Messungen verschiedene Akquisitionsparameter genutzt. Die wichtigsten sind der bereits genannte Flipwinkel, die Echozeit T_E und die Repetitionszeit T_R . Dabei bestimmt T_R das zeitliche Interval in dem der hochfrequente Anregungspuls eingestreut wird. Der Parameter T_E hingegen definiert die Zeit zwischen dem hochfrequenten Anregungspuls und der Aufzeichnung des Signals. Durch die Abhängigkeiten der Relaxationszeiten mit den unterschiedlichen Gewebetypen lassen sich nun unterschiedliche Bildkontraste erzielen. Einen T_1 -gewichteten Bildkontrast, der fettreiches Gewebe hell und Flüssigkeiten dunkel erscheinen lässt, erhält man durch die Wahl eines kurzen T_E -Wertes (ca. 15 ms) und eines kurzen T_R -Wertes (ca. 500 ms). Dementsprechend erzeugt ein langer T_E -Wert (ca. 90 ms) und ein langer T_R -Wert (ca. 3000 ms) einen T_2 -gewichteten Bildkontrast, der Flüssigkeiten hell und fettreiches Gewebe intermediär darstellt (Abb. 2.7).

Zwei wichtige Verfahren der MR-Bildgebung sind das *Spin-Echo-Verfahren* und die von Mansfield entwickelte *echoplanare Bildgebung*, kurz EPI (Mansfield 1977). Wie bereits erwähnt, zerfällt das MR-Signal nach der Anregung durch den hochfrequenten Puls exponentiell, in Abhängigkeit von T_2^* . Bei regulären MR-Echo-Verfahren möchte man das Signalmaximum nicht nach dem hochfrequenten Puls, sondern nach der Schichtselektion und der Phasenkodierung, aber noch während der Frequenzkodierung, aufzeichnen. Das *Spin-Echo-Verfahren* ermöglicht es, die T_2^* -Dephasierung der Spins rückgängig zu machen. Dabei wird nach dem 90° -Puls, mit einer Zeitdifferenz von $T_E/2$, ein zusätzlicher, refokussierender 180° -Puls eingestrahlt. Dieser bewirkt,

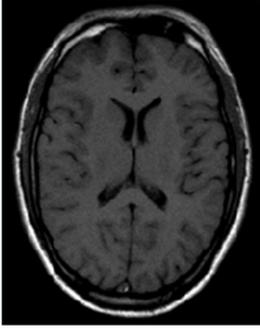
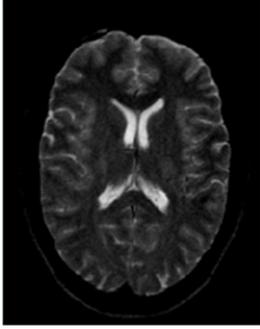
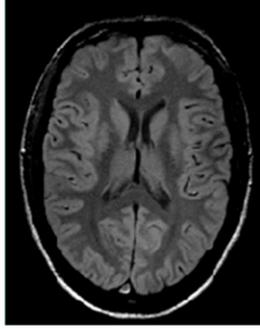
Wichtung	T1	T2	Protonendichte
Echozeit T_E	kurz (z.B. 15 ms)	lang (z.B. 90 ms)	kurz (z.B. 15 ms)
Repetitionszeit T_R	kurz (z.B. 500 ms)	lang (z.B. 3000 ms)	lang (z.B. 3000 ms)
Flüssigkeiten Fett	hypo-intens hyper-intens	hyper-intens hypo-intens	Gemischter Kontrast
Aufnahme			

Abbildung 2.7: Kontrastunterschiede bei unterschiedlicher Wichtung der Gewebeparameter T_1 (Längsrelaxation), T_2 (Querrelaxation) und ρ (Protonendichte, PD) durch Variation von T_E und T_R . Eigener Entwurf nach (Buxton 2003).

dass die Spins quasi ihre Orientierung tauschen. Spins mit einer kleineren Frequenz, die vorher noch einen Phasenrückstand auf die *vorausgehenden* Spins mit höherer Frequenz hatten, erhalten nun einen *Vorsprung*. Wichtig für das Verständnis ist, dass durch den 180° -Puls die Spins nicht ihre Bewegungsorientierung umkehren, vielmehr wird die Orientierung der Spins bzw. ihre Phasenreihenfolge umgekehrt. Die schnelleren Spins liegen jetzt hinter den langsameren und haben diese nach der Echozeit T_E wieder eingeholt. Zu diesem Zeitpunkt nehmen die Spins eine phasenkohärente Lage ein, was zu einem maximalen MR-Signal führt.

Besonders geeignet für die funktionelle Bildgebung sind sogenannte echoplanare Bildgebungstechniken. Sie erlauben Bilder von funktionellen Messungen mit hoher zeitlicher Auflösung. Anders als bei der Spin-Echo-Sequenz wird nicht für jede Phasenkodierung ein hochfrequenter Puls erzeugt. Stattdessen werden nach einer einmaligen Anregung durch den Anregungspuls gleich mehrere Bildzeilen aufgenommen. In diesem Zusammenhang spricht man auch von einem *single-shot* Verfahren. Durch das gleichmäßige wechselnde Schalten des Auslesegradienten (Frequenzgradient) wird ein kompletter Echozug erzeugt, welcher regulär aus maximal 128 Echos bestehen kann (Horowitz 1995, Haacke 1999). Größere Matrizen sind durch den schnellen Signalabfall und der dadurch verbleibenden geringen Zeit für Phasen- und Frequenzkodierungen nur selten möglich. Die Anzahl der Echos bestimmt den EPI-Faktor und damit die Größe der EPI-Matrix.

2.2 Funktionelle Magnetresonanztomographie

Der primäre Beweggrund hinter funktionellen Messungen mittels MRT ist die Aufzeichnung von Hirnaktivitäten als Funktion von Ort und Zeit. Prinzipiell unterscheidet man zwischen elektrophysiologischen Methoden, wie z.B. Elektroenzephalographie (EEG) und Magnetoenzephalographie (MEG) sowie Methoden, die sich auf metabolische Veränderungen bei Hirnaktivität stützen. Zu den Letzteren zählt neben der Positronen-Emissions-Tomographie (PET) auch die funktionelle Magnetresonanztomographie. Hier sei vollständigkeithalber erwähnt, dass alle elektrophysiologischen Methoden eine sehr gute zeitliche, jedoch eine sehr schlechte räumliche Auflösung aufweisen. Umgekehrt besitzt zum Beispiel die funktionelle Magnetresonanztomographie eine sehr hohe räumliche aber eine sehr schlechte zeitliche Auflösung.

Im Allgemeinen gilt, funktionelle Anwendungen der MRT müssen Kontraste zwischen ruhenden und aktiven Hirnregionen abbilden können. Hier existieren derzeit drei Verfahren. Die invasiven fMRT-Verfahren (1), bei denen intravenös applizierte Kontrastmittel zur Bilderzeugung des zerebralen Blutflusses genutzt werden. Die nichtinvasiven, perfusionsbasierten fMRT-Verfahren (2), welche die Eigenschaften der MRT nutzen, um den zerebralen Blutfluss mit Spinmarkierungen zu detektieren sowie die am häufigsten verwendete Methode, die sogenannte BOLD-fMRT (3). Im Folgenden sollen die physiologischen Grundlagen der BOLD-fMRT, die Modellierung funktioneller MRT-Experimente und die statistischen Verfahren zur Auswertung von fMRT-Daten vorgestellt werden.

2.2.1 BOLD-fMRT

Der BOLD-Effekt, *blood oxygen level dependency*, ist die Grundlage messbarer Signaländerungen in der BOLD-fMRT. Der Effekt macht sich vor allem die unterschiedlichen magnetischen Eigenschaften von oxygeniertem und desoxygeniertem Blut zu nutze. Die Ursache für diese verschiedenen, magnetischen Eigenschaften ist die Oxidationsstufe des am Hämoglobin gebundenen Eisen (Jezzard et al. 2001). Transportiert das Hämoglobin Sauerstoff, so liegt das Eisen in der Oxidationsstufe III vor. Das resultierende Oxyhämoglobin besitzt aufgrunddessen diamagnetische Eigenschaften, was zu einer proportionalen Abschwächung des inneren Magnetfeldes des Moleküls zum angelegten Magnetfeld führt (Buxton 2003, Frackowiak 2004).

Im Gegensatz dazu liegt Desoxyhämoglobin ohne gebundenem Sauerstoff in der Oxidationsstufe II vor und besitzt paramagnetische Eigenschaften (Jezzard et al. 2001). Es induziert eigene Magnetfelder, welche die Homogenität des äußeren Magnetfeldes stören. Es entstehen lokale Magnetfeldinhomogenitäten, die eine Signalaufzeichnung mit T_2^* -gewichteten Pulssequenzen negativ beeinflussen (Ogawa 1990, Huettel et al. 2008). Im Vergleich zu sogenannten „Ruhe-Phasen“ (Abb. 2.8 - A) kommt es bei zunehmender neuronaler Aktivität zu einer Steigerung der kortikalen Metabolismusrate ($CMRO_2$) (Abb. 2.8 - B).

Begleitend dazu steigt der lokale Verbrauch von Sauerstoff und es entwickelt sich ein kurzzeitiger, lokal erhöhter Anteil an Desoxyhämoglobin, was sich durch den in Abbildung 2.9 erkennbaren Signalabfall, dem „Initial Dip“, bemerkbar macht (Buxton

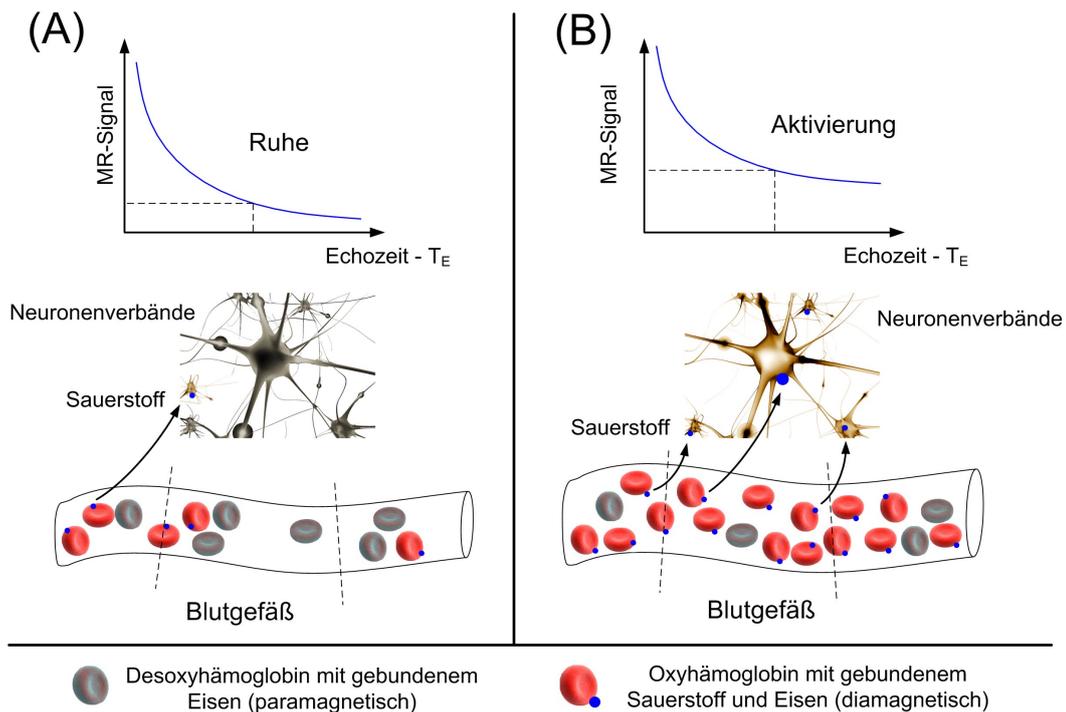


Abbildung 2.8: Die im Vergleich zum Ruhezustand (A) erhöhte neuronale Aktivierung (B) bewirkt eine Zunahme der lokalen Sauerstoffanreicherung des Blutes. Die lokale Abnahme desoxygeniertem Hämoglobins bewirkt einen Signalanstieg in T_2^* -gewichteten Pulssequenzen. Eigener Entwurf nach (Buxton 2003).

2003, Frackowiak 2004). Im weiteren Verlauf kommt es zu einer Überkompensation des regionalen Sauerstoffverbrauchs und damit zu einer Erhöhung des Anteils von Oxyhämoglobin (Abb. 2.8 - B).

Einhergehend steigen der lokale zerebrale Blutfluss (CBF - Cerebral Blood Flow) sowie das lokale zerebrale Blutvolumen (CBV - Cerebral Blood Volume) (Friston 2007, Huettel et al. 2008).

Dieser dem *BOLD*-Effekt zugrundeliegende neurophysiologische Prozess wird als *Neurovaskuläre Kopplung* beschrieben (Ogawa 1990, Buxton 2003, Huettel et al. 2008). Durch das Ansteigen der Oxyhämoglobinkonzentration während der Aktivierungsphase werden auch die Magnetfeldinhomogenitäten reduziert. Die angeregten Spins dephasieren langsamer, was zu einem Signalanstieg in T_2^* -gewichteten Pulssequenzen führt. Nach dieser Aktivierung fällt das MR-Signal recht schnell unter das Ausgangsniveau ab und es beginnt die Phase der Deaktivierung (Frackowiak 2004). Dieser Abfall unter das Ausgangsniveau wird häufig auch als „Undershoot“ bezeichnet (Abb. 2.9).

Anschließend kommt es zu einem langsamen Ausgleich zwischen Oxy- und Desoxyhämoglobin, was schließlich zum Zurückkehren des MR-Signals auf das Ausgangsniveau führt. Die Zeit zwischen Stimulusbeginn und einem signifikanten Signalanstieg, erkennbar durch den bereits erwähnten „Initial Dip“, beträgt je nach Physiologie zwischen vier und sechs Sekunden.

Dies ist auch der Grund für die erwähnte schlechte zeitliche Auflösung der fMRT. Die-

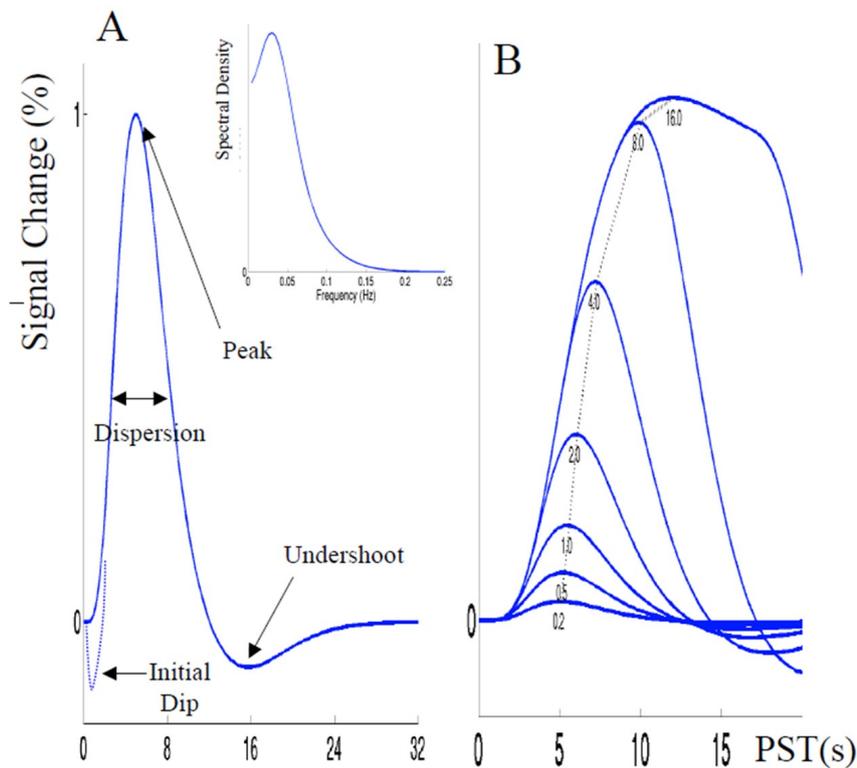


Abbildung 2.9: Theoretischer Verlauf der (kanonischen) hämodynamischen Antwortfunktion, bestehend aus einer ersten Unterschwingung, *Initial Dip*, einem Signalanstieg und anschließendem Abfall mit erneuter Unterschwingung, dem *Undershoot* sowie das zugehörige „Powerspektrum“ der Funktion (A). Das gefaltete BOLD-Signal in Abhängigkeit der Dauer der neuronalen Aktivität findet sich rechts daneben (B) (Frackowiak 2004).

ser Prozess wird als die „Hämodynamische Antwortfunktion“, oder häufig abgekürzt als HRF bezeichnet.

2.2.2 Modellierung von funktionellen MRT-Experimenten

Durch den Einsatz optimierter MR-Sequenzen gestaltet sich die Aufnahme von fMRT-Bilddaten inzwischen weniger problematisch als noch vor ein paar Jahren. Dafür gewinnt eine strukturierte Reizpräsentation mehr und mehr an Bedeutung (Huettel et al. 2008). Zu Beginn eines jeden Forschungsexperiments steht eine Reihe von informellen Forschungsfragen. Im weiteren Verlauf des Prozesses wird hieraus die Ausformulierung einer sogenannten Forschungshypothese vorangetrieben. Diese kann entweder eine Proposition natürlicher Abläufe oder eine Vorhersage von Ergebnissen eines Experiments darstellen (Buxton 2003, Savoy 2005). Ausgehend von dieser noch unbestätigten Hypothese soll die Durchführung eines modellierten Experimentablaufs die Basis für eine grundlegende Beweisführung liefern. Dabei wird durch Manipulation einer Menge von gegebenen unabhängigen Variablen eine Menge von abhängigen Va-

riablen erzeugt (Huettel et al. 2008). Alle unabhängigen Variablen bzw. Faktoren, welche den Experimentablauf bestimmen, werden vom Versuchsleiter im experimentellen Design des Versuchs definiert. Im weiteren Verlauf wird durch die Anwendung statistischer Auswertemethoden, meist hypothesengetriebenen Verfahren, die Signifikanz der gemessenen Werte evaluiert und somit die Nachhaltigkeit der Hypothese überprüft (Frackowiak 2004). Entscheidend für diese Evaluierung ist vor allem die Organisation des Versuchs und damit das effiziente Testen der Forschungshypothese. Eine erweiterte und gut strukturierte Versuchsplanung ist für fMRT-Experimente in besonderem Maße essentiell und gewährleistet aufgrund der limitierten Ressourcen, wie Betriebskosten des MRT-Scanners, fest definierte Versuchszeiten sowie Auswertezeiten von Messdaten, eine effiziente Durchführung von Experimenten (Friston 2007).

Die wichtigsten Elemente eines Experiments stellen die verschiedenen Variablen dar. Dabei wird zwischen *unabhängigen Variablen* und *abhängigen Variablen* unterschieden (Huettel et al. 2008). Unabhängige Variablen repräsentieren spezifische Aspekte eines experimentellen Designs, welche durch den Experimentleiter erstellt und verändert werden. Die Wahl der unabhängigen Variablen richtet sich nach der zu testenden Hypothese. Untersucht man beispielsweise die Aufmerksamkeit eines Probanden hinsichtlich eines bestimmten Objekts, so könnte im einfachsten Fall das Experiment allein aus zwei Versuchsbedingungen bestehen. Basierend auf diesem konkreten Beispiel wären so die zwei Versuchsbedingungen „schenkt dem Objekt Beachtung“ oder „schenkt dem Objekt keine Beachtung“ die einfachsten Möglichkeiten, um mit unabhängigen Variablen die Versuchshypothese zu testen (Huettel et al. 2008).

Die verschiedenen Werte einer unabhängigen Variablen werden in diesem Zusammenhang als Bedingungen (conditions) oder Stufen (levels) bezeichnet (Huettel et al. 2008).

Abhängige Variablen sind durch die mit unabhängigen Variablen gemessenen Versuchsdaten repräsentiert und ermöglichen so die Beweisführung der Forschungshypothese. Die Überprüfung einer Hypothese ist also nicht auf eine einzelne abhängige Variable beschränkt. Obwohl in regulären fMRT-Experimenten meist nur die zeitlichen Variationen des BOLD-Signal betrachtet werden, existieren auch kombinierte fMRT-EEG-Versuche, bei denen das BOLD-Signal und das EEG-Signal als zwei abhängige Variablen in die Beweisführung einfließen (Coben und Evans 2010).

Zusätzlich dazu können unabhängige und abhängige Variablen entweder einen kategorialen oder kontinuierlichen Variablentyp annehmen (Buxton 2003, Huettel et al. 2008). Kategoriale Variablen entspringen dabei einer Menge diskreter Werte. Als Beispiel sei das Betätigen eines Tasters entweder mit der rechten oder der linken Hand genannt. Die unabhängige Variable ist in dem Fall *Hand*, welche nur durch einen der beiden Zustände *drücke Taster rechts* und *drücke Taster links* repräsentiert wird. Kontinuierliche Variablen können jeden Wert innerhalb eines kontinuierlichen Wertebereichs annehmen und erlauben deswegen oft eine statistisch mächtigere Beweisführung (Friston 2007). Als Beispiel sei ein Experiment zur Bestimmung der Reaktionszeit (in Millisekunden) auf ein spezifisches Ereignis genannt. Die Reaktionszeit repräsentiert hier eine typische kontinuierliche Variable (Huettel et al. 2008). Zur vereinfachten Datenauswertung von fMRT-Experimenten werden kontinuierliche Variablen oftmals in eine beschränkte Anzahl von Kategorien unterteilt und definieren so eine mögliche

Mischform beider Typen (Savoy 2005). Ausgehend von den genannten Variablentypen muss ein Experiment mindestens zwei Werte einer unabhängigen Variable beinhalten, um die Veränderungen der abhängigen Variablen zu messen (Savoy 2005, Friston 2007).

Die einfachste Möglichkeit, dies zu erreichen, ist die Modellierung eines Experiments mit nur zwei Bedingungen. Diese werden üblicherweise zu verschiedenen Zeitpunkten präsentiert und sind in eine Experimental- sowie in eine Kontroll- oder Null-Bedingung unterteilt (Huettel et al. 2008). Des Weiteren sollten beide Bedingungen sich allein durch den zu untersuchenden Effekt unterscheiden. Divergieren die Bedingungen in mehr als dem untersuchenden Effekt, ergeben sich multiple Erklärungen für die vorherrschenden experimentellen Effekte und eine Beweisführung der Hypothese wird erschwert. Jeder Effekt, der mit unabhängigen Variablen des Experiments kovariiert, wird dann als Störgröße (confounding factor) bezeichnet (Huettel et al. 2008).

Ziel eines jeden experimentellen Designs ist die optimale Wahl der Experiment- und Kontrollbedingungen zur Minimierung der Störgrößen (Buxton 2003, Friston 2007, Huettel et al. 2008).

Neben der mehrmaligen Stimuluswiederholung zur Steigerung der Messempfindlichkeit spielt die Randomisierung von Ereignissen zur Prävention von Störgrößen eine entscheidende Rolle. Die Randomisierung bezieht sich dabei auf die zufällige Präsentation von variierenden Faktoren innerhalb des Experiments (Savoy 2005). Ziel dieser Methode ist das Ausgleichen von variierenden Faktoren durch ihre zufällige Anordnung bzw. Streuung über die gesamte Dauer des Experiments. So kann durch die Mittelung des Signalverlaufs das Signal-Rausch-Verhältnis verbessert werden. Dies gewährleistet bei fMRT-Messungen wiederum eine bessere Detektion von minimalen aber signifikanten Signaländerungen des BOLD-Signals (Friston 2007). Bisher haben sich zwei Formen des experimentellen Designs durchgesetzt, wobei anzumerken bleibt, dass beide Strategien auch in Mischformen ihre Anwendung finden (Abb. 2.10).

- *Block-Design*: Das Block-Design-Paradigma oder auch *Boxcar-Paradigma* basiert auf einer Unterteilung von Experimentalbedingungen in distinkte Blöcke, in denen jede einzelne Versuchsbedingung für einen ausgedehnten Zeitraum präsentiert wird (Huettel et al. 2008). Diese diskreten Stimulusvariationen zwischen Experimentalbedingungen sind auch für die Namensgebung dieses Design-Typs verantwortlich und charakterisieren den experimentellen Ablauf eines Versuchs. Die effizienteste Möglichkeit um innerhalb eines Block-Design-Experiments die Effekte zwischen unabhängigen und abhängigen Variablen zu untersuchen, ist der Vergleich einer Experimental- mit einer Kontrollbedingung. Hier gilt es zu beachten, dass innerhalb eines Stimulusblocks die Versuchsbedingung einer unabhängigen Variablen konstant bleibt (Savoy 2005).

Die Vorteile des Block-Designs liegen hauptsächlich in der vereinfachten Versuchsmodellierung sowie in der hohen statistischen Aussagekraft. Große Nachteile des Designs beziehen sich vor allem auf die Erwartungs- bzw. Antizipationsbildung sowie auf die Schwierigkeiten der Randomisierung von Stimuli (Jezzard et al. 2001, Friston 2007). Ausgehend davon können Wiederholungseffekte nur bedingt kontrolliert werden. Des Weiteren müssen die Experimental-

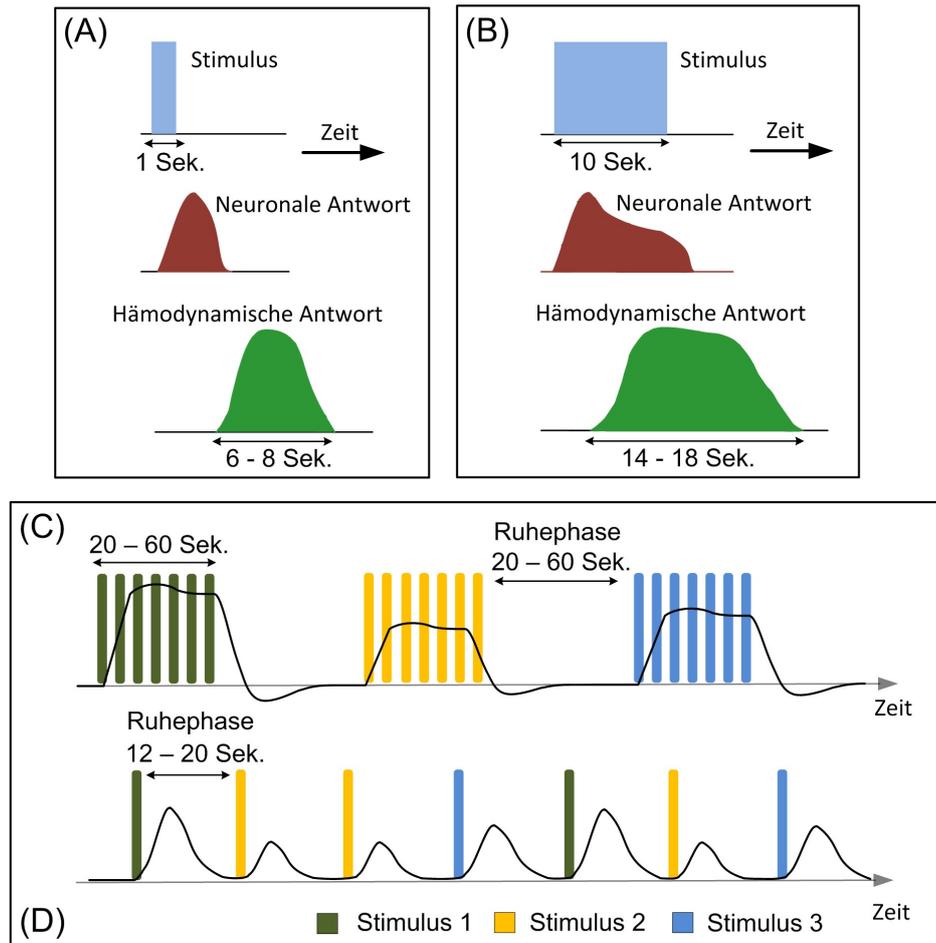


Abbildung 2.10: Die Abbildung verdeutlicht die zeitlichen Relationen und die Antwortfunktionen zwischen der neuronalen und der hämodynamischen Antwort auf kurze (A) und lange Stimulusreize (B). Vereinfacht ausgedrückt, ergeben sich die erwarteten zeitlichen Signalverläufe eines Block-Designs (C) und eines ereigniskorrelierten Designs (D) aus der Faltung der hämodynamischen Antwortfunktion sowie der zeitlichen Dauer der Bedingungen (Stimulus 1 bis Stimulus 3). Eigener Entwurf nach (Savoy 2009).

und Kontrollbedingungen eine ausreichende Länge aufweisen. Durch zu kurze Blocklängen kann per Definition die Rückkehr der hämodynamischen Antwort nicht innerhalb der Nullbedingung auf das Grundniveau (Baseline) zurückkehren, was wiederum die Gesamtvariabilität in den Daten und die statistische Aussagekraft des Experiments reduziert (Huettel et al. 2008).

- **Ereigniskorreliertes Design:** Beim ereigniskorrelierten Design (Single-Trial Design) handelt es sich um eine stark variierbare Ereignissequenz. Alle zu untersuchenden, neuronalen Aktivitäten sollen hier durch eine entsprechende Reizpräsentation innerhalb einer kurzen und diskreten Zeitperiode (*event*) hervorgerufen werden (Huettel et al. 2008). Die Reihenfolge dieser Ereignisse wird dabei zufallsbasiert festgelegt und anschließend dem Probanden präsentiert. Durch diese zufällige Anordnung der Stimuli soll für Probanden vor allem die Antizipation

und das Einstellen auf wiederkehrende Stimulustypen bzw. -ereignisse erschwert werden (Buxton 2003, Savoy 2005). Aufeinanderfolgende Stimuli werden dabei durch die Einführung sogenannter *Inter-Stimulusintervalle (Jitter)* zeitlich voneinander getrennt. Hier erlaubt die Einführung von zufälligen Längen des Inter-Stimulusintervalls oder von sukzessiven Stimulusereignissen eine weitere Optimierung des Experimentdesigns (Huettel et al. 2008). Im Vergleich zu klassischen Block-Design-Paradigmen bei dem das BOLD-Signal zumeist zum Vergleich der neuronalen Aktivierungsstärke zwischen einzelnen Versuchsbedingungen benutzt wird, können in ereigniskorrelierten Studien auch transiente Veränderungen in der neuronalen Aktivität bezogen auf diskrete Stimuli registriert und untersucht werden (Friston 2007, Huettel et al. 2008).

- *Mixed-Design*: Mixed-Design-Paradigmen vereinen die Grundzüge des Block- und ereigniskorrelierten Designs und gehören zu den komplexesten Paradigmen. Die Präsentation von Mixed-Design-Paradigmen setzt hier auf die Anzeige multipler Stimulusereignisse innerhalb von diskreten und regulären Stimulusblöcken (Savoy 2005). Ein fundamentaler Unterschied zu den beiden vorhergehenden Design-Typen ist hier die Einbindung mehrerer Stimulusereignisse innerhalb eines einzigen Stimulusblocks. Die Reihenfolge der Ereignisse wird dabei zufällig festgelegt (Huettel et al. 2008). Der größte Vorteil von Mixed-Design-Studien ist die effiziente Analyse von unabhängigen Variablen, welche aufgrund ihrer zeitlichen Charakteristiken sonst nur in Block-Design- oder ereigniskorrelierten Studien untersucht werden. Dies erlaubt innerhalb eines einzigen Paradigmas die zuverlässige und gemeinsame Evaluierung von zustandsbezogenen Prozessen (*state-related processes*) und objektbezogenen Prozessen (*item-related processes*), welche sonst entweder nur in Block-Design-Studien oder in ereigniskorrelierten Studien eingesetzt werden können (Buxton 2003).

Designklassifikation

Unabhängig vom Designtyp des Experimentes spielen die Anzahl der Bedingungen sowie die Art der zu unterscheidenden Faktoren die größte Rolle. Generell werden dabei einfaktorielle und mehrfaktorielle Designs unterschieden. Hinzu kommt die Unterteilung in kategorial und parametrisch (Savoy 2005, Huettel et al. 2008).

Ein *kategoriales Design* setzt voraus, dass neuronale Prozesse sich additiv miteinander verknüpfen lassen. Die einfachste Methode besteht in der Verknüpfung von der sogenannten Baseline-Phase bzw. Nullbedingung mit einer bestimmten Aufgabe des Versuchs. Auf diese Weise können über eine Subtraktion die einzelnen Komponenten einer Bedingung isoliert und deren spezifische Aktivierungsmuster extrahiert werden (Savoy 2009). Wichtig für den Einsatz *kategorialer Designs* ist, dass sämtliche übrigen Prozesse davon nicht beeinflusst werden. Lässt sich eine einzelne Fragestellung nicht durch die Subtraktionslogik lösen, so kann durch die Ausweitung der Auswertung und das gleichzeitige Testen auf mehrere Fragestellungen ein weiterer Lösungsansatz generiert werden. Dieser Ansatz wird in der Literatur auch als *Conjunction-Analyse* beschrieben (Buxton 2003). Ein großer Vorteil dieser Methode ist vor allem das Aufdecken von in-

varianten Komponenten und somit das Beantworten von komplexeren Fragestellungen zwischen den einzelnen Versuchsbedingungen. So lässt sich genau bestimmen, welche neuronalen Mechanismen sowohl in Aufgabe A, in Aufgabe B als auch in Aufgabe C signifikant involviert sind (Friston 2007).

Parametrische Designs bieten sich vor allem dann an, wenn regionale, neuronale Aktivierungen systematisch mit endogenen oder exogenen Parametern variieren (Buxton 2003). Um dennoch Vergleiche ziehen zu können, greift man auf eine quantitative oder blockweise Anpassung von Stimulusparametern während der verschiedenen Versuchsbedingungen, zurück (z.B. die schrittweise Erhöhung motorischer Aktivierung durch stärkere Fingerbewegung in den einzelnen Versuchsbedingungen) (Savoy 2005). Die große Stärke des parametrischen Designs zeigt sich bei der Erfassung neuronaler Korrelate zur Diagnose klinischer Symptome. Hierzu werden Regressionstechniken angewendet, welche Zusammenhänge zwischen der neuronalen Aktivität und verschiedenen Abstufungen klinischer Krankheitsbilder erfassen (Schneider und Fink 2007).

Im Unterschied zu allen anderen Designs, besitzt das (*mehr*)*faktorielle Design* die Möglichkeit der Interaktionsanalyse (Friston 2007). Für das Design heißt dies, dass sich die Wirkung einer Kombination aus zwei Faktoren nicht allein durch die additive Verknüpfung der Einzelfaktoren erklären lässt. Vielmehr treten sogenannte differenzielle Kombinationseffekte auf, die eine genaue Interaktionsbetrachtung notwendig machen (Buxton 2003). So werden bei der Betrachtung meist zwei oder mehr Faktoren untersucht und miteinander kombiniert. Das faktorielle Design wird vor allem im klinischen Bereich, zur Untersuchung gesunder Kontrollprobanden bezüglich gestörter neuronaler Prozesse und zur Effektivitätsbestimmung von Therapieformen verwendet (Huettel et al. 2008, Buxton 2003).

$$\left. \begin{array}{l} \text{Aufgabe A} - \text{Baseline} \\ \text{Aufgabe A} - \text{Aufgabe B} \end{array} \right\} \quad \textbf{Kategoriales Design} \quad (2.14)$$

$$A > B > C \quad \textbf{Parametrisches Design} \quad (2.15)$$

$$\left. \begin{array}{l} (A - B) - (C - D) \\ (A - C) - (B - D) \end{array} \right\} \quad \textbf{Faktorielles Design} \quad (2.16)$$

2.2.3 Vorverarbeitung funktioneller MRT-Daten

Bei der fMRT liegt der Signalunterschied zwischen aktivierten und nicht aktivierten Hirnregionen im Bereich weniger Prozente. Hinzu kommt, dass das Signal hoch- und niederfrequenten Störungen unterliegt. Die Ursache dafür sind die bereits erwähnten Bewegungsartefakte sowie das physiologische und das Messtechnik-bedingte Rauschen (Friston 2007). Die Einflüsse dieser Störgrößen auf die statistische Auswertung können durch verschiedene Vorverarbeitungsschritte reduziert werden (Abb. 2.11). Weitere Informationen finden sich in der hier zitierten Fachliteratur (Buxton 2003, Frackowiak 2004, Friston 2007, Huettel et al. 2008).

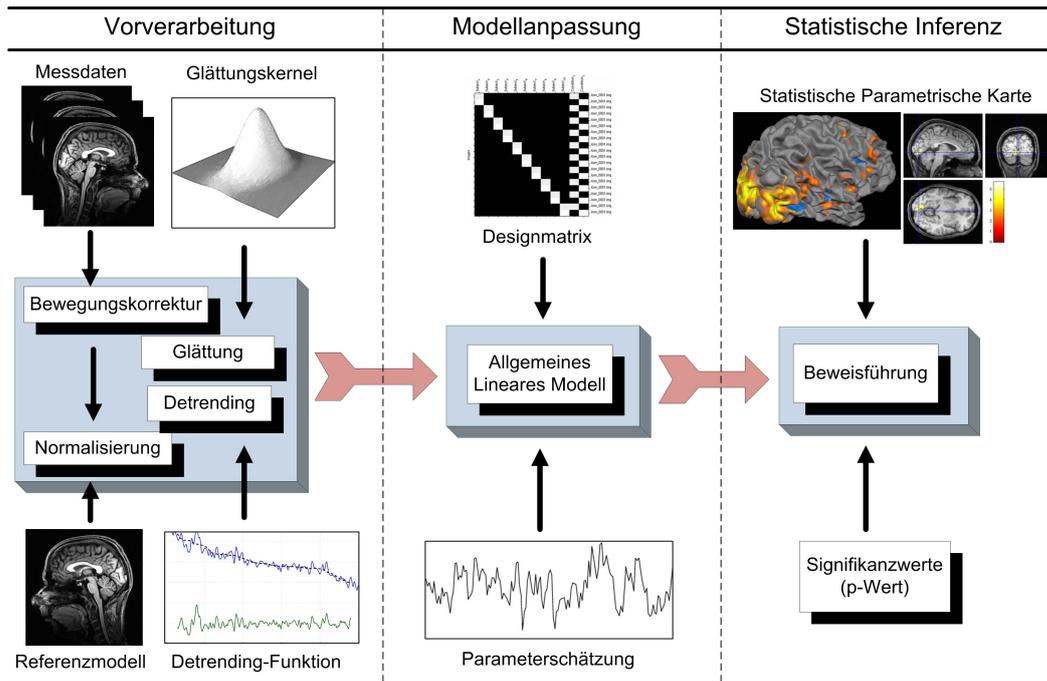


Abbildung 2.11: Die Prozesskette zur Auswertung von fMRT-Daten startet mit den sogenannten Vorverarbeitungsschritten. Anschließend folgen der Modellierungsprozess zur Anpassung des zugrundeliegenden statistischen Modells und die statistische Beweisführung bzw. Schlussfolgerung. Eigener Entwurf nach (Frackowiak 2004)

Bewegungskorrektur

Die *Bewegungskorrektur* oder auch *Realignment* ist in der Regel der erste Schritt in der Vorverarbeitungskette. Bei der statistischen Auswertung von funktionellen Messungen wird davon ausgegangen, dass während der aktuellen Messung jedes Volumenelement (Voxel) immer die gleiche Region des Gehirns abbildet. Durch Bewegungsartefakte verursachte Inhomogenitäten des Magnetfeldes führen neben den Aktivierungen des BOLD-Signals zu Scheinaktivierungen, welche letztendlich die eigentlichen Messresultate verfälschen (Buxton 2003, Frackowiak 2004).

Um diesen Problemen entgegenzuwirken, unterscheidet man zwischen zwei Arten von Bewegungskorrekturen. Zum einen die prospektive Bewegungskorrektur und zum anderen die retrospektive Bewegungskorrektur (Schneider und Fink 2007). Prospektiv bedeutet hier, dass bereits während der Messung und in Echtzeit die Positionsänderungen eines spezifischen Volumenelements oder einer ganzheitlichen anatomischen Struktur registriert und durch geeignete Anpassungen der Gradientenfelder sowie durch Modulation der Anregungs- und Empfangsspule ausgeglichen werden (Buxton 2003). Spezifische Algorithmen zur echtzeitfähigen Bestimmung von Bewegungsparametern stellen hier eine besondere Herausforderung dar und können nur durch die Auswahl eines geeigneten Bewegungsmodells realisiert werden (Gedat et al. 2004).

Die retrospektive Bewegungskorrektur bezieht sich auf alle Korrekturverfahren, welche nach der eigentlichen Messung auf die Daten angewendet werden. Das meist ver-

wendete Prinzip in der funktionellen MRT bleibt hier die Koregistrierung (Frackowiak 2004). Hierbei werden zunächst die sukzessiven Bildvolumen der Zeitserie auf ein einzelnes Referenzbild, meist der erste Volumendatensatz, koregistriert. Da hier immer das gleiche Objekt aufgenommen wird, muss zudem eine nicht formverändernde Transformation (*rigid-body transformation*) auf alle Volumendaten angewandt werden (Buxton 2003). Um eine präzise Überlagerung der Objekte zu erreichen, werden zunächst kombinierte Translationsbewegungen, Verschiebungen des Bildvolumens entlang der drei Raumrichtungen, und anschließend kombinierte Rotationsbewegungen, Rotationen des Bildvolumens entlang der drei Raumrichtungen, durchgeführt (Frackowiak 2004, Huettel et al. 2008).

Die Bewegungskorrektur stößt an ihre Grenzen, wenn zu schnelle Bewegungen ausgeglichen werden müssen. In diesem Fall müssen häufige Interaktionen mit Peripheriegeräten, wie z.B. das Betätigen von Tastern, als gesonderte Störgröße in das verwendete statistische Modell einfließen (Buxton 2003, Friston 2007, Huettel et al. 2008).

Slice Time Correction

Bei vielen Aufnahmeverfahren, die für die funktionelle Bildgebung eingesetzt werden, wird das Signal zur Präsentation der einzelnen Schichten zu verschiedenen Zeitpunkten innerhalb der Repetitionszeit aufgenommen. Die Schichten werden dabei entweder nacheinander (aufsteigend oder absteigend) oder überlappend (interleaved) aufgezeichnet, so dass zwischen der Aufnahme verschiedener Schichten im Extremfall mehrere Sekunden liegen können (Dössel 2000, Schneider und Fink 2007).

Um diese Zeitdifferenz auszugleichen und den negativen Einfluss auf die Datenauswertung zu minimieren, wird eine zeitliche Interpolation der Messdaten durchgeführt. Sie passt die zeitlichen Verschiebungen der Signalamplituden durch eine Interpolation der Intensitätswerte an einen definierten Zeitpunkt innerhalb der Repetitionszeit an und erlaubt dadurch eine einheitliche Betrachtung von Messdaten während des Datenanalyseprozesses (Buxton 2003, Huettel et al. 2008).

Stereotaktische Normalisierung

Ein weiteres Verfahren zur statistischen Vorverarbeitung ist die Normalisierung. Ein Grundproblem bei funktionellen Messungen sind individuelle, anatomische Unterschiede zwischen den einzelnen Probanden, welche einen direkten Vergleich zwischen Messergebnissen innerhalb einer Versuchsgruppe und zwischen mehreren Versuchsgruppen sehr schwierig gestalten. Die räumliche Normalisierung erlaubt es, die individuellen anatomischen Besonderheiten in ein standardisiertes anatomisches Koordinatensystem zu überführen und individuelle Gehirne aufeinander abzubilden (Huettel et al. 2008). Dies ermöglicht eine ungefähre räumliche Zuordnung innerhalb eines dreidimensionalen Koordinatensystems. Als Grundlage für die räumliche Normalisierung kommen verschiedene standardisierte Referenzmodelle zum Einsatz. Das Atlssystem von Talairach und Tournoux ist eines der bekanntesten Modelle, welches ursprünglich auf einer reinen linearen Reskalierung der Größenverhältnisse beruhte (Buxton 2003, Schneider und Fink 2007). Grundlage dieses Systems ist die Gestaltung eines soge-

nannten *Normgehirns*, in welches jedes beliebige Gehirn transformiert werden kann. Zur Definition dieses Normgehirns im dreidimensionalen Raum wurden zunächst primäre Landmarken verwendet. Die anteriore und posteriore Kommissur (AC/PC) werden durch eine vorgestellte Gerade miteinander verbunden. Das Gehirn wird anschließend so ausgerichtet, dass diese Gerade die beiden Punkte exakt verbindet. Das Gehirn wird dabei nicht verändert, sondern erhält damit eine waagerechte Ausrichtung. Im weiteren Verlauf werden außerdem sekundäre Landmarken (räumliche Extrempunkte) definiert (Friston 2007). Durch das Verbinden dieser Punkte entstehen die direkten Raumachsen, welche das räumliche Referenzmodell aufspannen (Huettel et al. 2008). Ein weiteres bekanntes Referenzmodell ist das sogenannte MNI-Referenzmodell des Montréal Neurological Institute (McGill University, Montreal, Canada). Es liefert eine weitere Möglichkeit zur interindividuellen Vergleichbarkeit von Ergebnissen und kann als Alternative zum Talairach und Tournoux Atlssystem aufgefasst werden (Buxton 2003). Will man Aktivierungsmuster verschiedener Probanden in einer Gruppenanalyse (Second-Level Analyse) miteinander vergleichen, dann ist die Normalisierung ein notwendiger Vorverarbeitungsschritt für die interindividuelle Vergleichbarkeit von Auswertergebnissen (Frackowiak 2004).

Räumliche und zeitliche Filterung

Die räumliche Filterung dient im Allgemeinen zur Verbesserung des Signal-Rausch-Verhältnisses, um eine bessere Abgrenzung echter, experimentell induzierter Aktivierungen von Störsignalen zu erhalten (Huettel et al. 2008). Zur Realisierung der räumlichen Filterung wird häufig eine *dreidimensionale Gauß-Verteilung* verwendet, welche auch als Kernel-Filtermaske bezeichnet wird. Dabei wird jeder einzelne Grauwert eines Voxels mit den benachbarten Bildpunkten verrechnet. Diese Neuberechnung der Voxelwerte richtet sich nach der *Euklidischen Distanz* zum Voxel im Zentrum. Durch Faltung mit der isotropen Gauß-Kernel-Filtermaske können so zufällige Effekte aus den Experimentdaten reduziert und eine höhere, statistische Vergleichbarkeit erreicht werden (Buxton 2003, Frackowiak 2004). Die Größen des isotropen Gauß-Kerns variieren im Allgemeinen zwischen 3 und 15 mm *volle Halbwertsbreite* der Gauß-Verteilung (FWHM - Full Width At Half Maximum). Neben der räumlichen Filterung oder Glättung kann auch eine sogenannte zeitliche Filterung zum Entfernen von konstanten Störsignalen durchgeführt werden. Meist weisen diese Störsignale ein charakteristisches Frequenzmuster auf, welches hauptsächlich durch respiratorische oder kardiale Faktoren hervorgerufen wird (Friston 2007). Die zeitliche Filterung von Störsignalen wird wie die räumliche Filterung im Fourier-Raum durchgeführt (Schneider und Fink 2007, Grüningen 2008).

Detrending

Das Detrending stellt eine besondere Form der zeitlichen Filterung dar und wird speziell bei der Optimierung von Echtzeit-fMRT-Experimenten eingesetzt (Cox et al. 1995, Gembris et al. 2000). Infolge von unerwünschten Einflüssen sind fMRT-Messdaten häufig durch zusätzliche Störeinflüsse bzw. Drifts belastet. Ein Drift ist als eine lang-

sam veränderliche und kontinuierliche Störgröße definiert, die während des gesamten Experiments das Nutzsignal überlagert (Grüning 2008). Um ein lineares Störsignal aus dem Nutzsignal $x(n)$ herauszufiltern, wird ein Polynom 0-ter oder 1-ter Ordnung modelliert und anschließend vom Gesamtsignal subtrahiert:

$$x_{dt}(n) = x(n) - (a_0 + a_1 \cdot (n)) \quad (2.17)$$

Die Variablen a_0 und a_1 repräsentieren die Polynomkoeffizienten oder *Detrending-Koeffizienten*, welche im Allgemeinen durch das L_2 -Polynomapproximationsverfahren ermittelt werden (Grüning 2008). In fMRT-Experimenten werden vor allem respiratorische oder kardiale Faktoren sowie Drifts durch Geräteerwärmungen zum Detrending ausgewählt (Schneider und Fink 2007). Neben der Anwendung von Detrending-Funktionen bei der Vorverarbeitung wird das Detrending auch in der Echtzeit-fMRT beim Einsatz von Echtzeit-Korrelationsanalyseverfahren eingesetzt (Cox et al. 1995, Gembris et al. 2000).

2.2.4 Statistische Auswertung funktioneller MRT-Daten

Nach der Anwendung der Vorverarbeitungsschritte, erfolgt die eigentliche statistische Auswertung funktioneller fMRT-Daten. Dabei werden vor allem signalspezifische Analysen der einzelnen Bildvolumina oder Voxel durchgeführt. Hierzu muss bei einer hypothesengestützten, statistischen Auswertung ein Modell spezifiziert werden, welches möglichst exakt die Überprüfung der zugrundeliegenden Hypothesen und des zeitlichen Signalverlaufs ermöglicht (Friston 2007). Ausgehend von den definierten Kontroll- oder Nullhypothesen werden so meist Verfahren wie der t-Test oder die F-Statistik zur Bestimmung von neuronalen Aktivierungen verwendet. Für ausführliche Beschreibungen zu statistischen Auswerteverfahren sowie zu musterbasierten Klassifikationsmethoden sei auf weiterführende Literatur verwiesen (Friston et al. 1995, Bishop 2006, Friston 2007, Huettel et al. 2008).

t-Test-Verfahren

Die *t-Test*-Verfahren als einfachste Realisierung des Allgemeinen Linearen Modells beruhen auf dem Prinzip, aus Beobachtungen einer Stichprobe Schlussfolgerungen auf Eigenschaften der Gesamtmenge abzuleiten. Abhängig von der Anzahl der Stichprobenelemente analysiert der t-Test, ob die Mittelwerte zweier beobachteter Stichproben übereinstimmen (Fahrmeir et al. 2004). Auf die Auswertung von fMRT-Daten bezogen, vergleicht der t-Test den gemittelten zeitlichen Signalverlauf eines Voxels zwischen Experimental- und Kontrollbedingung und überprüft Signalunterschiede auf zufällige Signalschwankungen. Die Differenzen zwischen den gemittelten Signalverläufen der Versuchsbedingungen werden anschließend mit der Standardabweichung gewichtet (Friston 2007, Huettel et al. 2008). Der resultierende t-Wert beschreibt die standardisierten Differenzen der Stichprobenmittelwerte und lässt sich für den in der

fMRT häufig eingesetzten *Welch-Test* wie folgt berechnen:

$$t = \frac{\bar{y}_1 - \bar{y}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}. \quad (2.18)$$

Die Indizes 1 bzw. 2 bezeichnen in dem Zusammenhang die Werte der verschiedenen Ereignisblöcke. Die Terme \bar{y}_1 und \bar{y}_2 repräsentieren die Mittelwerte der Stichproben n_1 bzw. n_2 und s_1 bzw. s_2 beschreiben die zugehörigen Standardabweichungen. Die dem t-Wert zugrundeliegenden Freiheitsgrade f_W sind über folgende Gleichung definiert:

$$f_W = \frac{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)^2}{\frac{\left(\frac{s_1^2}{n_1}\right)^2}{n_1-1} + \frac{\left(\frac{s_2^2}{n_2}\right)^2}{n_2-1}}. \quad (2.19)$$

Ein Nachteil des *Welch-Tests* bzw. aller t-Test-Verfahren ist die hohe Anfälligkeit gegenüber Artefakten. Da t-Test-Verfahren per Definition alle systematischen Differenzen zwischen den Experimentbedingungen aufzeigen, beinhalten diese Signalunterschiede nicht nur das erwünschte BOLD-Signal, sondern u.a. Bewegungs- und Flussartefakte (Friston 2007). Des Weiteren kann aufgrund der Mittelung aller Zeitpunkte einer Bedingung keine signifikante Aussagen über die Zeitspanne einer Aktivierung getroffen werden (Fahrmeir et al. 2004). Weitere Zusammenhänge werden im Abschnitt zum Allgemeinen Linearen Modell näher beschrieben.

Korrelationsanalyse

Ein weiteres statistisches Verfahren, welches häufig bei der Auswertung von fMRT-Daten eingesetzt wird, ist die *Korrelationsanalyse* (Frackowiak 2004). Anders als bei den t-Test-Verfahren vergleicht die Korrelationsanalyse nicht allein die Signalunterschiede zwischen den einzelnen Bedingungen. Vielmehr evaluiert sie die Stärke der Beziehung zwischen zwei Variablen. Als Basis dient ein Referenzmodell, welches in Ansätzen die Eigenschaften und den zeitlichen Verlauf der erwarteten Funktion modelliert (Fahrmeir et al. 2004, Huettel et al. 2008). Für die statistische Auswertung funktioneller MRT-Experimente wird hierfür die hämodynamische Antwortfunktion verwendet. Die Stärke der Korrelation zwischen zwei Variablen ist durch den Korrelationskoeffizienten ρ definiert. Durch die Schätzung von ρ , ergibt sich $\hat{\rho} = r$ mit:

$$r = \frac{\sum_{i=1}^n (y_i - \bar{y}) \cdot (F_i - \bar{F})}{\sqrt{\sum_{i=1}^n (y_i - \bar{y})^2 \cdot \sum_{i=1}^n (F_i - \bar{F})^2}}. \quad (2.20)$$

Der Wertebereich des Korrelationskoeffizienten ist mit dem oberen Grenzwert +1 (exakte Korrelation) und dem unteren Grenzwert -1 (exakte Antikorrelation) gegeben. Die Terme y_i und \bar{y} definieren in diesem Zusammenhang die Zeitreihe der vorhandenen Messwerte bzw. den empirischen Mittelwert der Zeitreihe. Die Terme F_i und \bar{F} repräsentieren die Referenzfunktion bzw. den entsprechenden empirischen Mittelwert. Die Korrelationsanalyse zeigt im Vergleich zum beschriebenen Welch-Test eine höhere Variabilität gegenüber Bewegungsartefakten.

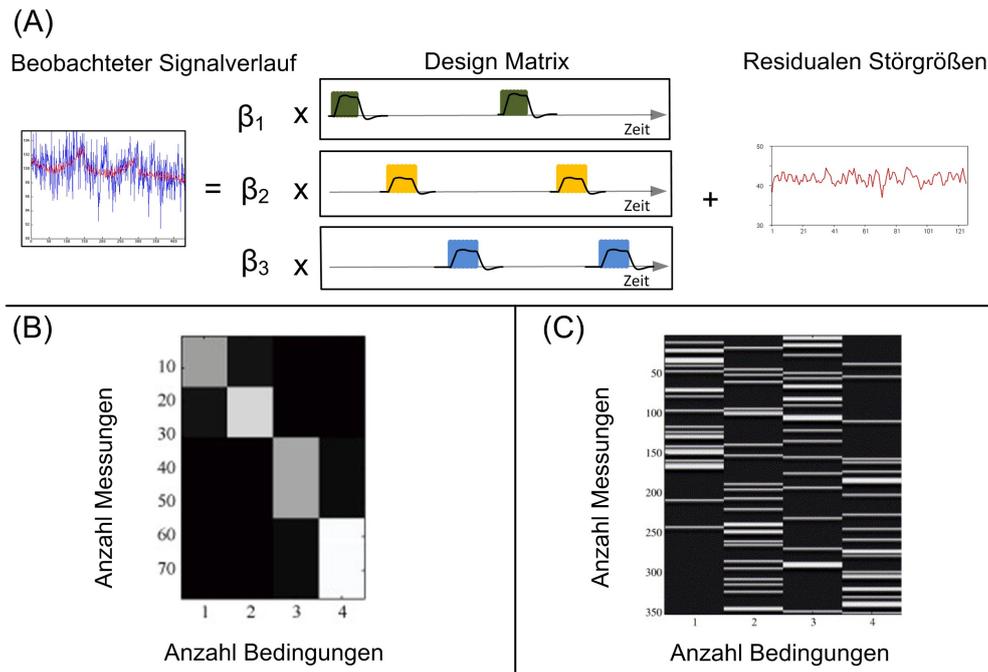


Abbildung 2.12: Das ALM beruht auf der Annahme, dass sich Ergebnisdaten aus einer gewichteten Kombination von verschiedenen Modellfaktoren (Design-Matrix) und einem additiven Fehlerterm zusammensetzen (A). Die Anordnung der Modellfaktoren bestimmt das Aussehen der Design-Matrix sowie die Typisierung des Experiments als Block-Design (B) oder als ereigniskorreliertes Design (C). Eigener Entwurf nach (Friston et al. 1995).

Allgemeine Lineare Modell

Das Grundprinzip des Allgemeinen Linearen Modells (ALM) oder General Linear Model (GLM) beruht auf der Annahme, dass sich die Ergebnisdaten aus einer gewichteten Kombination von verschiedenen Modellfaktoren und einem additiven Fehlerterm zusammensetzen (Abb. 2.12 - A). Die Modellfaktoren repräsentieren in fMRT-Experimenten meist eine Menge von hypothetischen Änderungen des BOLD-Signals, die sich entweder durch diverse Manipulationen der angenommenen unabhängigen Variablen oder anderer Variabilitätsgrößen zusammensetzen (Friston et al. 1995, Buxton 2003, Frackowiak 2004, Huettel et al. 2008). Bei der postexperimentellen statistischen Auswertung wird so die modellierte Zeitreihe mit dem tatsächlichen Signalverlauf eines jeden Voxels verglichen. Die resultierende Abweichung des Modells vom tatsächlichen Signalverlauf wird berechnet und durch das Regressionsgewicht als Qualitätsindikator repräsentiert. Im Fall von fMRT-Studien möchte man aus einem starken Hintergrundrauschen die relevanten und schwachen BOLD-Signaländerungen zuverlässig herausfiltern. Dementsprechend wird der regionale zerebrale Blutfluss innerhalb eines bestimmten Voxels als abhängige Variable mit y_j repräsentiert. Der Index j ($j = 1, \dots, N$) beschreibt hier die verschiedenen Beobachtungszeitpunkte. Des Weiteren existiert für jeden Beobachtungszeitpunkt eine Menge K ($K < N$) von experimentellen Faktoren x_{jl} , welche zusammengefasst die *Design-Matrix* eines Versuchs repräsentieren (Friston et al. 1995, Friston 2007) (Abb. 2.12 - B bis C). Der Index l

($l = 1, \dots, K$) steht in diesem Zusammenhang für die in der Design-Matrix verwendete Anzahl der experimentellen Faktoren. Ausgehend von diesen Betrachtungen ist die abhängige Variable y_j als Linearkombination in folgender Form beschrieben:

$$y_j = x_{j1}\beta_1 + x_{j2}\beta_2 + \dots + x_{jK}\beta_K + \epsilon_j, \quad (j = 1, \dots, N). \quad (2.21)$$

Die unbekannt Parameter β_l korrespondieren zu jedem der Anzahl K gegebenen experimentellen Faktoren x_{jl} und repräsentieren die verschiedenen Wichtungen der Versuchsbedingungen. Der Fehlerterm ϵ_j ist eine unabhängige und gleichverteilte Zufallsgröße, welche durch Berechnung der einzelnen Wichtungen möglichst gering sein sollte. Das resultierende Minimum des Fehlerterms nach Auflösen des linearen Modells wird als Residuum bezeichnet (Frackowiak 2004, Huettel et al. 2008). Schaut man sich die verschiedenen Beobachtungszeitpunkte der abhängigen Variablen y_j an, so ergibt sich folgende Matrix-Form:

$$\begin{pmatrix} y_1 \\ \vdots \\ y_j \\ \vdots \\ y_N \end{pmatrix} = \begin{pmatrix} x_{11} & \cdots & x_{1l} & \cdots & x_{1K} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{j1} & \cdots & x_{jl} & \cdots & x_{jK} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{N1} & \cdots & x_{Nl} & \cdots & x_{NK} \end{pmatrix} \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_l \\ \vdots \\ \beta_K \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \vdots \\ \epsilon_j \\ \vdots \\ \epsilon_N \end{pmatrix}, \quad (2.22)$$

mit der Matrix-Notation:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}. \quad (2.23)$$

Der Vektor \mathbf{y} steht hier für den Beobachtungsvektor oder den Spaltenvektor der abhängigen Variablen. Der Vektor $\boldsymbol{\epsilon}$ definiert den Spaltenvektor der Fehlerterme und der Vektor $\boldsymbol{\beta}$ repräsentiert den Spaltenvektor der Regressionskoeffizienten. Die $N \times K$ -dimensionale Design-Matrix \mathbf{X} beinhaltet alle unabhängigen Variablen oder Regressoren.

Nach der Durchführung eines Experiments stehen so eine Vielzahl von Experimentbeobachtungen bzw. abhängige Variablen y_j zur Verfügung. Zur Lösung des linearen Gleichungssystems bedarf es einer Methode zur Parameterschätzung, welche einen möglichst genauen Wert der Parametermenge $\boldsymbol{\beta}$ liefert (Frackowiak 2004). Dies wird durch die Parameterschätzung der *Methode der kleinsten Quadrate* erreicht und ist wie folgt definiert:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1} \mathbf{X}'\mathbf{y}. \quad (2.24)$$

Hier gilt es zu beachten, dass in dieser Form die Parameterschätzung nur postexperimentell durchgeführt werden kann, da für die Berechnung die gesamte Zeitreihe der abhängigen Variablen y_j benötigt wird (Friston 2007).

Für die Erstellung einer statistisch-parametrischen Karte müssen die im Vektor $\boldsymbol{\beta}$ enthaltenen Parameter hinsichtlich der Versuchshypothesen getestet werden. Dabei wird für weniger komplexe Versuchshypothesen mit nur einem Kontrastvektor \mathbf{c} der Dimension $s \times 1$ auf die t-Teststatistik in Gleichung (2.18) zurückgegriffen. Es erfolgt

ein Testen der Nullhypothesen der Form $H_0 : \mathbf{c}'\boldsymbol{\beta} = 0$ mittels der verallgemeinerten Form der t-Teststatistik, welche wie folgt bestimmt ist:

$$t = \frac{\mathbf{c}'\hat{\boldsymbol{\beta}}}{\sqrt{\hat{\sigma}^2 \mathbf{c}'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{c}}}. \quad (2.25)$$

Die zur Berechnung notwendige Schätzung der Fehlervarianzen $\hat{\sigma}^2$ wird aus den Residuen des Versuchsmodells gewonnen und ist ebenfalls von den $n - s$ Freiheitsgraden abhängig:

$$\hat{\sigma}^2 = \frac{[(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})]}{(n - s)}. \quad (2.26)$$

Komplexere Versuchshypothesen ab zwei Kontrastvektoren sind in einer Kontrastgewichtsmatrix \mathbf{C} mit der Dimension $S \times M$ definiert und werden mit Hilfe der F-Statistik getestet:

$$F = \frac{n - s}{m} \frac{h}{g}. \quad (2.27)$$

Dabei ist g mit:

$$g = (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})'(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}), \quad (2.28)$$

und h mit:

$$h = \hat{\boldsymbol{\beta}}'\mathbf{C}(\mathbf{C}'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{C})^{-1}\mathbf{C}'\hat{\boldsymbol{\beta}}, \quad (2.29)$$

definiert. Die Freiheitsgrade der F-Verteilung sind durch $F_{G1} = m$ und $F_{G2} = n - s$ gegeben. Durch die Kombination der berechneten statistischen Ergebnisse aller Voxel können die Hirnaktivierungen entsprechend der Irrtumswahrscheinlichkeit unterschieden werden. Die Ergebnisse dieser statistischen Berechnungen werden in der fMRT häufig als Parameterkarten dargestellt und durch eine spezifische Farbkodierung abgebildet.

Ein mögliches Problem ist das Auftreten von falsch positiven Testergebnissen, was auch durch die separate Berechnung des Parameterschätzers $\hat{\boldsymbol{\beta}}$ für jedes einzelne Voxel nicht ausgeschlossen werden kann. Der Grund für dieses Problem liegt im multiplen Testen sehr vieler Voxel hinsichtlich der Signifikanz einer Nullhypothese. Eine Möglichkeit zum Beheben des genannten Problems kann die Anpassung der Signifikanzgrenze mittels Korrekturverfahren, wie z.B. durch die Berechnung der *Anteile falscher Ablehnungen* (False Discovery Rate) oder durch die *Kontrolle des multiplen Niveaus* (Family-Wise Error Rate), sein (Buxton 2003, Frackowiak 2004). Das ALM wurde durch das MATLAB Softwarepaket SPM erstmalig zur Auswertung von fMRT-Studien eingeführt (SPM 2008). Derzeit existieren neben SPM noch weitere Applikationen, welche die statistische Auswertung von fMRT-Daten mittels ALM unterstützen (Huetzel et al. 2008) (Tabelle 2.1). Neben univariaten statistischen Auswerteverfahren finden auch multivariate Auswerteverfahren vereinzelt ihre Anwendung. Hervorzuheben

Softwareanwendung	Verfügbarkeit	Webseite
BrainVoyager	kommerzielles Produkt	www.brainvoyager.com
SPM	frei verfügbar / MATLAB	www.fil.ion.ucl.ac.uk/spm/
AFNI	frei verfügbar	afni.nimh.nih.gov/afni/
MEDx	kommerzielles Produkt	medx.sensor.com

Tabelle 2.1: Einige der häufigsten Softwareanwendungen zur Analyse funktioneller MRT-Daten, nach (Huettel et al. 2008). Frei verfügbare Softwarepakete, wie z.B. SPM, können nicht ohne die kommerzielle Software MATLAB betrieben werden.

sind hier die Hauptkomponentenanalyse (*principal component analysis* - PCA), die Unabhängigkeitsanalyse (*independent component analysis* - ICA) und diverse Musterklassifikationsverfahren, wie die *Support Vector Machine* sowie die probabilistische *Relevance Vector Machine* (Duda et al. 2004, Bishop 2006, Huettel et al. 2008), auf die hier nicht weiter eingegangen werden soll.

2.3 Echtzeit-Auswertung funktioneller Bilddaten

Dieser Abschnitt soll sich mit den statistischen Methoden der Echtzeit-Datenauswertung von funktionellen MRT-Experimenten befassen. Der Begriff Echtzeit-fMRT wurde Mitte der 1990er Jahre erstmalig eingeführt (Cox et al. 1995, Coben und Evans 2010). Ursprünglich wurde die Bezeichnung Echtzeit auch für Auswertungen verwendet, die wenige Minuten nach dem Abschluss der Messung erste Ergebnisse liefern. Heute bezieht sich der Begriff Echtzeit-fMRT auf Auswertungen, die nach jeder Volumenakquisition oder sogar noch innerhalb der Repetitionszeit, eine Auswertung durchführen und so statistisch parametrische Aktivierungskarten berechnen (Weiskopf et al. 2004a; 2007). Verschiedene Echtzeit-Algorithmen zur Auswertung funktioneller MRT-Experimente wurden seit der Prägung des Begriffs entwickelt. Dabei bleibt anzumerken, dass seitdem ein Ablöseprozess stattgefunden hat. Wurden zu Anfang noch verschiedene Methoden zur Echtzeit-Berechnung von Korrelationskoeffizienten (Cox et al. 1995, Gembris et al. 2000) und Techniken zur multiplen linearen Regression (Smyser et al. 2001) angewendet, so sind seit den letzten Jahren optimierte Berechnungsmethoden des hypothesengetriebenen Allgemeinen Linearen Modells (Pollock 1999, Bagarinao et al. 2003, Sorger et al. 2012) und datengetriebene Auswertungsverfahren, wie z.B. die *Independent Component Analysis* (Esposito et al. 2003) im Einsatz. Diese drei Hauptmethoden sollen im Folgenden näher erläutert werden.

2.3.1 Optimierte Echtzeit-Datenauswertung auf Basis der Korrelationsanalyse

Ein Ziel der Echtzeit-Datenanalyse auf Basis der Korrelationsanalyse ist die Optimierung des Verfahrens, damit dessen Rechenzeit nicht mit der Anzahl der Messzeitpunk-

te skaliert, d.h. damit die Daten eines laufenden Experiments auch ohne Kenntnis der nachfolgenden abhängigen Variablen in Echtzeit statistisch ausgewertet werden können. Um dieses Ziel zu erreichen, wird mit jedem neu gemessenen Volumen kein komplett neues anatomisches Bild präsentiert. Stattdessen werden funktionell aktive Bereiche immer wieder aktualisiert und auf das anatomische Bild überlagert. Zur Umsetzung dieser Datenaktualisierung wurde beim Cox-Korrelationsalgorithmus eine rekursive Berechnung des Korrelationskoeffizienten mit dem Detrending-Verfahren verknüpft (Cox et al. 1995). Die zugehörige Detrending-Matrix \mathbf{S} wurde dazu in die Berechnung des Korrelationskoeffizienten integriert. Als Grundlage dient die Berechnung des Korrelationskoeffizientenschätzers aus Gleichung (2.20), welche wie folgt angepasst wurde:

$$r = \frac{\mathbf{y}_S \cdot \mathbf{F}_S}{|\mathbf{y}_S| \cdot |\mathbf{F}_S|}. \quad (2.30)$$

Hier fällt auf, dass sowohl der Vektor der Messwerte \mathbf{y} als auch der Vektor der Referenzfunktion \mathbf{F} in das Detrending einbezogen wurden und so als \mathbf{y}_S und \mathbf{F}_S bezeichnet sind. Um den linearen Trend, entstanden durch Bewegungsartefakte und technisches Equipment, innerhalb der beobachteten Werte zu beseitigen, wird hier die Menge der zu entfernenden Datenvektoren s_k mit $k = 1, \dots, L$ festgelegt. Daraus resultierend ergibt sich die Dimension der Detrending-Matrix \mathbf{S} mit $(L + 2) \times (L + 2)$, welche somit sämtliche für die Berechnung von r notwendigen Elemente beinhaltet. Durch Einbeziehung der sogenannten *bordering-Technik* leitet Cox die Inverse Δ der rechten, unteren 2×2 ($L = 2$) Untermatrix ab, deren Elemente wiederum eine Charakterisierung des Korrelationskoeffizientenschätzers r zulassen:

$$r = \frac{\Delta_{12}}{(\Delta_{11} \cdot \Delta_{22})^{\frac{1}{2}}}. \quad (2.31)$$

Die Reduktion der ansteigenden Rechenzeit durch fortwährende Akquirierung von Datensätzen gelingt Cox durch die Einführung der *Cholesky-Zerlegung*. Dadurch können die Elemente der originalen Detrending-Matrix \mathbf{S} innerhalb der Cholesky-Matrizenform dargestellt werden. Per Definition entsprechen diese Cholesky-Elemente $C_{i,j}$ der Bedingung $C_{i,j} = 0$ für $j > i$ (Cox et al. 1995). Es ergibt sich die finale Gleichung:

$$r = \frac{C_{L+2,L+1}}{(C_{L+2,L+2}^2 \cdot C_{L+2,L+1}^2)^{\frac{1}{2}}}. \quad (2.32)$$

Ein weiterer Algorithmus, welcher das Detrending-Verfahren von Cox et al. aufgreift und dieses durch das sogenannte *Sliding-Window*-Verfahren aus der Netzwerktechnik aufwertet, wurde für die Echtzeit-Datenauswertesoftware *FIRE* konstruiert (Gembris et al. 2000). Die Idee dieser Methode beruht auf einer Minimierung des Berechnungsaufwands durch eine selektive Korrelationsberechnung der n letzten MR-Datensätze. Da der selektive Rahmen (Sliding-Window) durch eine konstante Größe definiert ist, behalten zwischen den Stimulationszyklen alle berechneten Korrelationskoeffizienten eine gleichbleibende Empfindlichkeit gegenüber Änderungen der Signalform bzw. Signalamplitude (Gembris et al. 2000) (Abb. 2.13). Ein nicht zu vernachlässigender

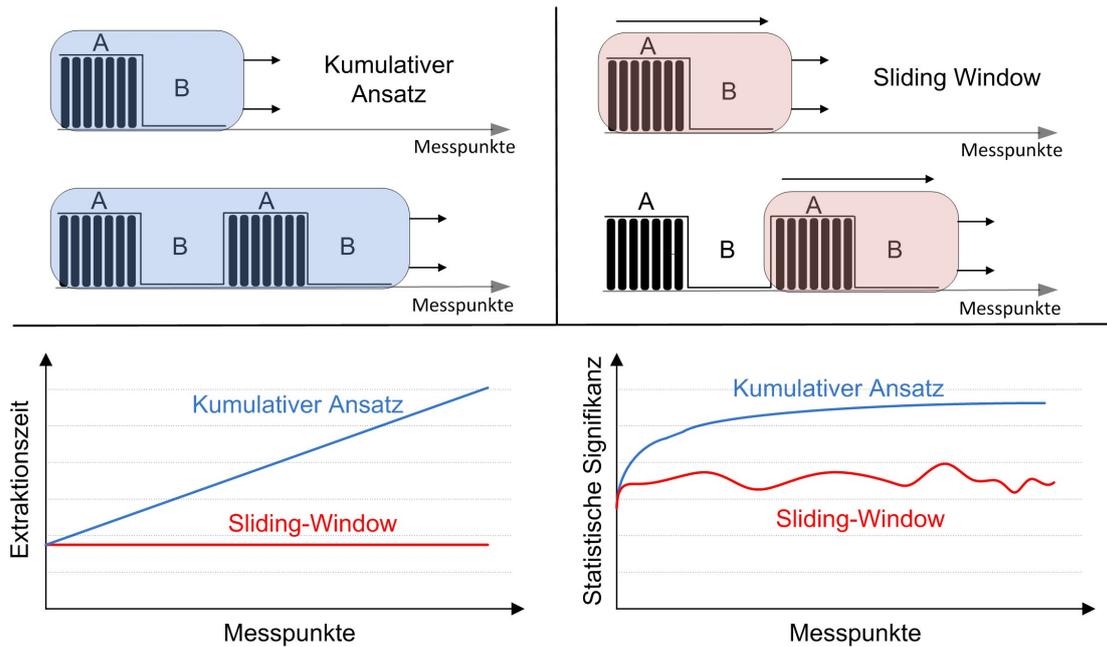


Abbildung 2.13: Während beim herkömmlichen kumulativen Auswerteansatz die Anzahl der Extraktionsschritte linear ansteigt, bleibt diese beim selektiven Rahmenkonzept des Sliding-Window-Verfahrens konstant. Im Gegenzug bleibt die statistische Signifikanz beim regulären Sliding-Window-Verfahren nahezu unverändert, was sich gegenüber dem kumulativen Ansatz als nachteilig auswirkt. Eigener Entwurf nach (Esposito et al. 2003).

Nachteil gegenüber des kumulativen Ansatzes von Cox et al. ist die Tatsache, dass mit zunehmender Dauer die statistische Signifikanz im gesamten Experiment nicht ansteigt sondern auf einer konstanten Stufe bleibt. Ohne Einbeziehung des Detrending-Verfahrens von Cox et al. ist die Berechnung des Korrelationskoeffizienten mit Sliding-Window-Verfahren folgendermaßen definiert:

$$r = \frac{\sum_{i=1}^n y_i r_i - \bar{r} n \bar{y}}{\sqrt{\sum_{i=1}^n r_i^2 - \bar{r}^2 n} \sqrt{\sum_{i=1}^n y_i^2 - \bar{y}^2 n}}. \quad (2.33)$$

Die Terme \bar{r} und \bar{y} repräsentieren jeweils das arithmetische Mittel aller Werte y_i und r_i ($i = 1, \dots, n$), welche durch den selektiven Rahmen des Sliding-Windows gegeben sind. Die Korrelationsberechnung durch das Sliding-Window-Verfahren lässt sich in zwei Abschnitte unterteilen. In der Akkumulationsphase, die sich über die ersten n Messzeitpunkte erstreckt, werden alle aufgenommenen Bilddaten sukzessiv in einen Datenspeicher geladen. In der stationären Phase, wird der älteste Datensatz durch den jeweils neu aufgenommenen Datensatz ersetzt (Gembris et al. 2000). Sliding-Window-Verfahren werden im Allgemeinen als robust gegenüber Bewegungsartefakten eingestuft. Aufgrund des unregelmäßigen Auftretens der Artefakte und der konstanten Aktualisierung innerhalb der stationären Phase wird eine gute Kompensierung von Störfaktoren angenommen (Gembris et al. 2000, Coben und Evans 2010).

2.3.2 Das Allgemeine Lineare Modell zur Echtzeit-Datenanalyse

Waren zu Beginn die verschiedenen Korrelationsanalysemethoden noch weit verbreitet, gewinnen inzwischen die verschiedene Echtzeit-Datenauswerteverfahren auf Basis des Allgemeinen Linearen Modells (ALM) immer mehr an Einfluss. Dies liegt zum einen an der hohen Reputation sowie an der weitverbreiteten Nutzung des regulären ALM bei der statistischen Auswertung von fMRT-Studien. Ein weiterer Punkt ist die Nutzung der Echtzeit-Datenanalysesoftware *Turbo BrainVoyager* (Brain Innovation 2002-2006), welche häufig zur Durchführung von Echtzeit-fMRT verwendet wird und zur statistischen Auswertung u.a. ein Echtzeit-ALM anbietet. Dieses baut auf einer rekursiven Berechnung des Parameterschätzers $\hat{\beta}$ auf (Pollock 1999) und soll im Folgenden näher betrachtet werden.

In der ursprünglichen Form des ALM wird der Vektor des Parameterschätzers $\hat{\beta}$ über die *Methode der kleinsten Quadrate* berechnet, welche die gesamte Zeitserie der fMRT-Daten nutzt, siehe Gleichung (2.24). Diese Methode wurde beim vorliegenden Algorithmus durch einen rekursiven Ansatz der kleinsten Quadrate-Methode ersetzt und erlaubt dadurch die Schätzung der aktuellen β -Werte ohne Kenntnis der gesamten Zeitreihe (Pollock 1999). Der Algorithmus wurde zur Anwendung in der Echtzeit-Datenauswerteapplikation *Turbo BrainVoyager* folgendermaßen angepasst (Brain Innovation 2002-2006):

$$\hat{\beta}_{t+1} = \hat{\beta}_t + (\mathbf{X}'_t \mathbf{X}_t)^{-1} \mathbf{x}_{t+1} \frac{(y_{t+1} - \mathbf{x}_{t+1} \hat{\beta}_t)}{f_t}. \quad (2.34)$$

In der rekursiven Form aktualisiert der Algorithmus die aktuellen β -Werte ($\hat{\beta}_{t+1}$) für jedes Voxel ausgehend von der letzten Schätzung $\hat{\beta}_t$ und unter Einbeziehung der zu diesem Zeitpunkt neu hinzugekommenen Design-Matrixwerte (\mathbf{x}_{t+1}) (Brain Innovation 2002-2006). Zusätzlich dazu muss die inverse Matrix $(\mathbf{X}'_t \mathbf{X}_t)^{-1}$ ebenfalls für jeden neuen Zeitpunkt t rekursiv berechnet werden. Dies passiert durch das zeilenweise Einlesen der neu hinzugekommenen Design-Matrixwerte:

$$(\mathbf{X}'_{t+1} \mathbf{X}_{t+1})^{-1} = (\mathbf{X}'_t \mathbf{X}_t)^{-1} - \frac{(\mathbf{X}'_{t+1} \mathbf{X}_{t+1})^{-1} \mathbf{x}_{t+1} \mathbf{x}'_{t+1} (\mathbf{X}'_{t+1} \mathbf{X}_{t+1})^{-1}}{f_{t+1}} \quad (2.35)$$

Wie die Berechnung der aktualisierten β -Werte basiert auch die Kalkulation der aktualisierten inversen Matrix auf einem rekursiven Ansatz, welcher ausschließlich die neuen Design-Matrixwerte einbezieht. Hier fällt auf, dass die Berechnung (2.35) komplett ohne die gemessenen abhängigen Variablen \mathbf{y} erfolgt, und dass allein die unabhängigen Variablen \mathbf{x}_t aus der Design-Matrix zur Berechnung verwendet werden (Brain Innovation 2002-2006). Übrig bleibt der sogenannte f -Term, welcher wie folgt definiert ist:

$$f_{t+1} = 1 + \mathbf{x}'_{t+1} (\mathbf{X}'_t \mathbf{X}_t)^{-1} \mathbf{x}_{t+1}. \quad (2.36)$$

Bei der Betrachtung der Berechnungsgrundlage wird deutlich, dass aufgrund der festen Anzahl an Berechnungen die Performanz des Algorithmus immer auf einer konstanten Stufe bleibt und die Berechnungszeiten auch mit zunehmender Dauer der Messung

nicht größer werden.

Der zweite bekannte Algorithmus, der die Berechnungsgrundlagen des ALM zur Echtzeit-Datenauswertung optimiert, beruht auf einer Orthogonalisierungstechnik zur Parameterschätzung (Bagarinao et al. 2003). Die Grundidee des Algorithmus ist die Konvertierung der Basisfunktion bzw. der unabhängigen Variablen in Orthogonalfunktionen mittels des *Gram-Schmidt-Algorithmus*. Hierfür wird zunächst die bekannte Gleichung (2.21) durch Orthogonalfunktionen dargestellt:

$$y_j = a_{j,1}\phi_{t,1} + a_{j,2}\phi_{t,2} + \dots + a_{j,K}\phi_{t,K} + \epsilon_j. \quad (2.37)$$

Die Variablen $a_{j,K}$ und $\phi_{t,K}$ mit $t = 1, \dots, N$ repräsentieren hier die gesuchten Hilfskoeffizienten (auxiliary coefficients) bzw. die Orthogonalfunktion, welche der Orthogonalbedingung $E[\phi_{i,t}\phi_{j,t}] = \delta_{i,j}$ entspricht (Bagarinao et al. 2003). Der Term $\delta_{i,j}$ ist dabei als Erwartungsoperator definiert. Ausgehend von der Gram-Schmidt-Orthogonalisierung ergibt sich folgendes Gleichungssystem mit der Normalisierungsmatrix \mathbf{N} ($\mathbf{N}_m^2 = E[\mathbf{z}_{m,t}\mathbf{z}_{m,t}]$ und $\alpha_{mr} = E[\mathbf{f}_{m,t}\phi_{r,t}]$):

$$\mathbf{z}_{m,t} = \mathbf{f}_{m,t} - \sum_{r=1}^{m-1} \alpha_{mr} \phi_{r,t}. \quad (2.38)$$

$$\phi_{m,t} = \frac{\mathbf{z}_{m,t}}{\mathbf{N}_m}. \quad (2.39)$$

Die Funktion $\mathbf{f}_{m,t}$ (explanatory function) definiert in diesem Zusammenhang die Zeitserie des Experimentmodells bzw. die unabhängigen Variablen. Nach dem Erhalt der Orthogonalfunktion $\phi_{m,t}$ können die Koeffizienten $a_{j,l}$ über die Orthogonalbedingung berechnet werden:

$$a_{j,l} = E[y_j \phi_{t,l}]. \quad (2.40)$$

Obwohl der Orthogonalisierungsalgorithmus insgesamt mit weniger Rechenoperationen als das rekursive ALM auskommt (Bagarinao et al. 2003), konnte sich die Prozedur bislang nicht durchsetzen. Der Hauptgrund dafür ist die Tatsache, dass die oft zum Detrending benutzten *diskreten Kosinus Transformationsfunktionen* (DCT) durch Orthogonalisierungsverfahren verfälscht werden (Friston 2007). Die resultierenden DCT-Störgrößen sind leider bisher noch nicht ausreichend charakterisiert, so dass eine nachträgliche Filterung nur bedingt möglich ist.

2.3.3 Echtzeit-Datenauswertung durch datengetriebene Auswerteverfahren

Neben der hypothesengetriebenen Auswertung, bei der zuvor unabhängige Variablen zu einem Experimentmodell spezifiziert werden müssen, existieren auch noch sogenannte datengetriebene bzw. explorative Analyseverfahren (Buxton 2003, Schneider

und Fink 2007). Am populärsten sind hier die faktorenanalytischen Verfahren, wie z.B. die Hauptkomponentenanalyse (*principal component analysis* - PCA) und die Unabhängigkeitsanalyse (*independent component analysis* - ICA). Diese Verfahren bieten sich immer dann an, wenn keine aussagekräftigen Modellspezifikationen über die erwarteten abhängigen Variablen erstellt werden können. Auch können mit Hilfe von explorativen Verfahren neue und nicht erwartete Ergebnisse aufgezeichnet und evaluiert werden.

Die Grundidee hinter diesen Verfahren beruht auf der Berechnung von Korrelationsmatrizen und einer Varianzzerlegung mittels der zuvor kalkulierten Datenvarianz. Es werden zunächst sogenannte Cluster extrahiert, welche anhand ihrer Charakteristik eine bestmögliche Repräsentation der Datenmengen ermöglichen. Als exemplarisches Beispiel soll hier das ICA-Verfahren zur Echtzeit-Datenverarbeitung von Esposito et al. vorgestellt werden (Esposito et al. 2003). Die Grundgleichung der ICA lässt sich folgendermaßen darstellen:

$$\mathbf{y}_i = \mathbf{A} \cdot \mathbf{x}_i. \quad (2.41)$$

Auf die fMRT-Datenanalyse bezogen, repräsentiert der Vektor \mathbf{x}_i die Menge der unabhängigen statistischen Zufallsvariablen für jedes Voxel i ($i = 1, \dots, p$), welche durch Multiplikation mit der $N \times K$ dimensionalen Matrix \mathbf{A} die beobachtete Zeitreihe \mathbf{y}_i eines Voxels i widerspiegelt. Per Definition gilt: die ICA kann nur eingesetzt werden, wenn die Anzahl der verwendeten, gaussverteilten Zufallsvariablen größer eins ist.

Der erste Schritt generischer ICA-Verfahren ist das sogenannte „Whitening“, dass Entfernen sämtlicher Korrelationen aus den Daten. Dazu wird die vorhergehende Gleichung aufgegriffen und die Mischmatrix \mathbf{A} durch ihre Pseudoinverse \mathbf{W} , mit $\mathbf{W} = \mathbf{A}^{-1}$ ersetzt. Die Dimensionen der Vektoren sowie der Matrix bleiben dabei unverändert:

$$\mathbf{y}_i^* = \mathbf{W} \cdot \mathbf{x}_i. \quad (2.42)$$

Der Vektor \mathbf{y}_i^* repräsentiert hier das unkorrelierte beobachtete Signal für alle Voxel i mit ($i = 1, \dots, P$). Der ICA-Algorithmus von Esposito et al. wurde mit dem bereits bekannten Sliding-Window-Verfahren zur Echtzeitverarbeitung von fMRT-Daten kombiniert und greift zur Darstellung der funktionellen Echtzeit-fMRT Ergebnisse auf eine sogenannte *Kontrast-Funktion* G zurück. Diese objektive Kontrastfunktion wird im weiteren Verlauf der Berechnung durch die Maximierung der Negentropie-Approximation berechnet (Esposito et al. 2003):

$$J_G(\mathbf{w}) = [E\{G(\mathbf{w}^T \mathbf{x})\} - E\{G(v)\}]^2. \quad (2.43)$$

Hierbei repräsentiert der Term w einen m -dimensionalen Gewichtsvektor, und die Funktion $J_G(\mathbf{w})$ definiert die sogenannte Negentropie-Funktion (negative Entropie). Zudem wird vorausgesetzt, dass der Erwartungswert mit $E\{(\hat{\mathbf{w}}^T \hat{\mathbf{x}})^2\} = 1$ fest definiert und v als normierte Gauß'sche Zufallsvariable integriert ist. Im Algorithmus von Esposito et al. erlaubt die Maximierung von J_G die Anwendung des Fixpunktiterationsverfahrens (*fixed-point-algorithm*) mittels Deflationstechnik. Das Ziel ist hier die schrittweise Annäherung an eine „independent component“ als Linearkombination der

Form: $\mathbf{w}^T \mathbf{x}$.

Durch Diagonalisierung und Invertierung der Jacobi-Matrix sowie durch Ersetzen von \mathbf{w}_0 durch \mathbf{w} ergibt sich folgende Gleichung für das Fixpunktiterationsverfahren, bei dem man sich durch die iterative Berechnung des geschätzten Gewichtsvektors $\hat{\mathbf{w}}^+$ schrittweise an eine exaktere Lösung $\hat{\mathbf{w}}^*$ annähert:

$$\hat{\mathbf{w}}^+ = E\{\hat{\mathbf{x}} \cdot G'(\hat{\mathbf{w}}^T \hat{\mathbf{x}})\} - E\{G''(\hat{\mathbf{w}}^T \hat{\mathbf{x}})\} \hat{\mathbf{w}}, \quad (2.44)$$

$$\hat{\mathbf{w}}^* = \frac{\hat{\mathbf{w}}^+}{\|\hat{\mathbf{w}}^+\|}. \quad (2.45)$$

Die Terme G' und G'' repräsentieren die erste bzw. die zweite Ableitung der Kontrastfunktion G . Zudem definiert die Notation $\|\cdot\|$ die Euklidische Vektornorm.

Für die Anwendung der Fixpunktiteration im Echtzeit-ICA-Verfahren nutzen Esposito et al. drei unabhängige Variablen, welche den Eingaberaum der Datenvektoren definieren. Diese drei Variablen sind L (Länge des Sliding-Window), D (Dezimierungsfaktor bzw. die Repetitionszeit der MR-Messung) sowie S (Zeit zur Aktualisierung des Versuchsszenarios). Die zeitliche Dimension T der dynamischen Analyse wurde von Esposito et al. wie folgt definiert:

$$T = \text{integer}\left\{\frac{L}{D}\right\}. \quad (2.46)$$

Auf Basis dieser Betrachtung und der Berechnung durch die Fixpunktiteration ergibt sich so nach S neuen Messzeitpunkten eine bestimmte Menge von „independent component“ Karten. Die Anzahl der Karten kann dabei zwischen 0 und der maximalen Anzahl T variieren. Ausgehend davon kann bei einer vereinfachten Annahme von $D = S = 1$ und $L = T$ folgende Voxelbetrachtung zu einem Zeitpunkt i mit den L letzten Messpunkten definiert werden:

$$\mathbf{x}_i = \begin{pmatrix} \mathbf{x}_{i-L+1} \\ \vdots \\ \mathbf{x}_i \end{pmatrix} \quad (2.47)$$

Alle so ermittelten räumlichen Aktivierungen beziehen sich immer nur auf die letzten zur Auswertung einbezogenen Messzeitpunkte und repräsentieren lediglich eines der Maxima der L detektierten Aktivierungszustände. Auf Grundlage der Basisvektorschätzung aus Gleichung (2.39) und (2.40) kann so eine Präsentation der aktuellen statistischen Parameterkarte in Echtzeit durchgeführt werden.

Obwohl datengetriebene Auswerteverfahren ohne a priori-Annahmen eine funktionelle Aktivität über das Versuchsdesign zielgerichtet identifizieren können, haben diese Auswerteverfahren noch nicht die Grundakzeptanz, um hypothesengetriebene Auswerteverfahren von ihrem Platz zu verdrängen. Die häufigsten Verfahren in der Echtzeit-fMRT bleiben die optimierte Korrelationsanalyse und rekursive Methoden des ALM, welche hauptsächlich beim Einsatz von sogenannten Hirn-Computer-Schnittstellen ihre Anwendung finden (Coben und Evans 2010).

Kapitel 3

Neurophysiologische Grundlagen und Hirn-Computer-Schnittstellen

Die Planung, Durchführung und Auswertung einer Pilotstudie zur Evaluierung der implementierten Teilaspekte dieser VR-Stimulusapplikation wurde als ein Hauptziel dieser Arbeit definiert. Im Rahmen dieser VR-Studie sollen u.a. die Mechanismen der räumlichen Navigation und des topographisch-episodischen Gedächtnisses untersucht werden. In diesem Kapitel sollen dazu die neurophysiologischen Grundlagen sowie wichtige Erkenntnisse aus Studien zum räumlichen Gedächtnis und zur topographischen Navigation vorgestellt werden. Des Weiteren werden die Definition des Immersionsbegriffes und die, für das Immersionsempfinden entscheidenden, medienpsychologischen Faktoren erörtert. Der zweite Teil dieses Kapitels beschäftigt sich zunächst mit der Entstehungsgeschichte und der Motivation von Hirn-Computer-Schnittstellen. Anschließend werden die zwei grundlegenden Formen von Hirn-Computer-Schnittstellen sowie deren Anwendungsgebieten detailliert vorgestellt. Besonderes Augenmerk wird dabei auf Anwendungen im Bereich der Echtzeit-fMRT gelegt.

3.1 Topographisches Gedächtnis und räumliche Orientierung

Um die Interpretation neuronaler Aktivierungen zu ermöglichen, werden zunächst die Besonderheiten des topographischen Gedächtnisses und der räumlichen Orientierung erläutert. An dieser Stelle muss zunächst der große Umfang und die Weitläufigkeit dieses Forschungsgebietes betont werden. Verschiedene Bereiche aus der Psychologie und der Neurowissenschaften sind dabei involviert.

Der genannte Themenschwerpunkt: *topographisches Gedächtnis und räumliche Orientierung* beschreibt dabei die Interaktion verschiedener Themenbereiche, wie z.B. der Repräsentation topografischer Umgebungen durch das Arbeits- und Langzeitgedächtnis (Pollmann 2008) sowie die Interaktion unterschiedlicher Hirnareale bei der Enkodierung und der Dekodierung von Landmarken (Aguirre 1998).

Für die topographische Navigation und das räumliche Gedächtnis sind hier vor allem der Hippocampus, die parahippokampalen Bereiche, der Cuneus, der Precuneus, das

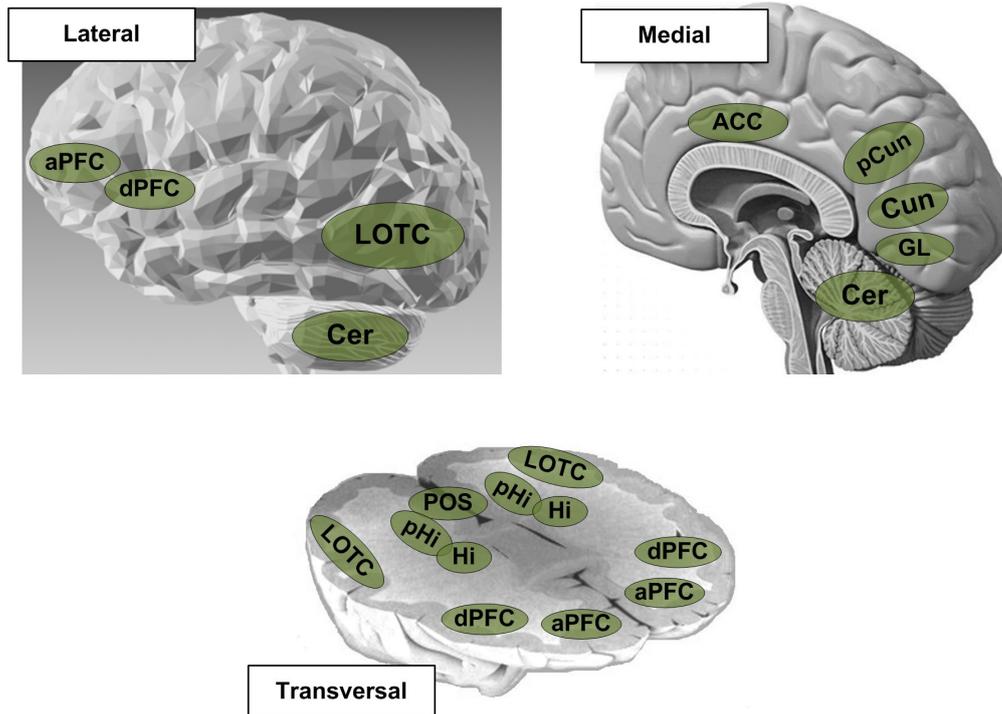


Abbildung 3.1: Die verschiedenen Hirnareale, die mit den Funktionen des topographischen Gedächtnis und räumlicher Orientierung in Verbindung gebracht werden. Eigener Entwurf nach (Aguirre et al. 1996, Maguire 1998, Burgess et al. 2002, Marsh et al. 2010, Doeller et al. 2010): **Laterale Ansicht:** Anteriorer Präfrontaler Kortex (aPFC), Dorsaler Präfrontaler Kortex (dPFC), Cerebellum (Cer), Lateraler Okzipito-Temporaler Kortex (LOTC). **Mediale Ansicht:** Anteriores Cingulum (ACC), Precuneus (pCun), Cuneus (Cun), Gyrus Lingualis (GL). **Transversale Ansicht:** Hippocampus (HI), Parahippokampaler Kortex (pHi), Parieto-Okzipitaler Sulcus (POS).

Cerebellum und vereinzelte Areale des Präfrontalen Kortex zu nennen (Abb 3.1).

3.1.1 Topographisches Lernen

In den letzten Jahren haben mehrere Studien zum räumlichen Gedächtnis gezeigt, dass kein einzelnes Hirnareal allein für die Speicherung und das Abrufen topographischer Repräsentationen verantwortlich ist und dass Funktionen des episodisch-zeitlichen Gedächtnisses eine wichtige Rolle spielen (Aguirre 1998, Pine et al. 2002). Aufgrund dieser Erkenntnisse wurden die Prozesse des topographischen Lernens in zwei Hauptaspekte untergliedert:

- **Enkodierung:** das Abspeichern von topographischen Repräsentationen wird vornehmlich mit der Verknüpfung von episodischen Informationen in Verbindung gebracht. Von zentraler Bedeutung sind hier Funktionen des *Arbeitsgedächtnisses* und der *räumlich-visuelle Speicher*. Die charakteristischen neuronalen Aktivierungen beim topographischen Lernen unterstützen diese These und zeigen zudem Aktivität im Cerebellum, was auf mögliche Priming-Effekte schließen

lässt (Abb. 3.1) (Maguire et al. 2006).

- Dekodierung: das Abrufen von gelernten Informationen ist bisher nur wenig erforscht und erlaubt deswegen auch nur wenig Rückschlüsse auf das Dekodieren von topographischen Informationen (Schneider und Fink 2007, Pollmann 2008). Viele Annahmen basieren auf bekannten Mustern des *Arbeits-* oder *Langzeitgedächtnisses* (vgl. Abschnitte 3.2 und 3.3). Da man im Vergleich zur Enkodierung noch zusätzliche Aktivierungsmuster beobachten kann, vermutet man hier noch weitere neuronale Netzwerke, die in den Prozess involviert sind (Aguirre 1998, Ekstrom et al. 2003).

3.2 Arbeitsgedächtnis

Das Arbeits- oder Kurzzeitgedächtnis ist in der Regel für die kurzfristige und unmittelbare Enkodierung von Informationen zuständig. Des Weiteren ist seine Aufgabe nicht ausschließlich auf die Speicherung von Informationen beschränkt, sondern es kann Informationen auch aktiv zur Steuerung nachfolgenden Verhaltens manipulieren. Eines der bekanntesten und meist verwendeten Arbeitsgedächtnismodelle ist das Vier-Komponentensystem (Baddeley 2000):

- *Zentrale Exekutive*: die zentrale Exekutive gilt als die wichtigste und gleichzeitig am wenigsten erforschte Komponente dieses Systems. Sie wird im Allgemeinen mit der Aufmerksamkeitssteuerung in Verbindung gebracht (Birbaumer und Schmidt 2006) und repräsentiert zudem die Schnittstelle zwischen Arbeits- und Langzeitgedächtnis (Baddeley 2000).
- *Phonologische Schleife*: aufgrund der Annahme, dass visuell-räumliche Informationen und sprachliche Informationen nicht in einer Komponente verarbeitet werden können (Baddeley 2000), entschied man sich für die Einführung der Phonologischen Schleife als Subkomponente dieses Systems. Die Phonologische Schleife gilt als ein sprachbasiertes Speichersystem, welches sprachliche Informationen in einer phonetischen Form (Lautform) enkodiert und verändert (Pollmann 2008). Die Möglichkeiten zur Speicherung von sprachbasierten Informationen sind begrenzt und werden allgemein mit ein bis zwei Sekunden angegeben (Baddeley 2000, Birbaumer und Schmidt 2006).
- *Räumlich-visueller Speicher*: der räumlich-visuelle Speicher ist das zweite begrenzte Speichersystem innerhalb des Vier-Komponentensystems. Er verarbeitet und manipuliert visuelle Informationen und übermittelt sie zur räumlichen Verarbeitung an beteiligte Hirnareale. Die begrenzte Kapazität dieses Systems wurde in verschiedenen Studien zur sogenannten *Veränderungsblindheit* bereits nachgewiesen (Birbaumer und Schmidt 2006). Eine genaue neuroanatomische Zuordnung dieser Arbeitsgedächtnisfunktionen ist problematisch. Anhand verschiedener fMRT-Studien konnte jedoch gezeigt werden, dass bei Funktionalitäten des Arbeitsgedächtnisses der dorsolaterale Präfrontale Kortex involviert ist

(Abb. 3.1). Darüber hinaus gibt es Anzeichen, dass der ventrolaterale Präfrontale Kortex bei der Enkodierung von Informationen eine entscheidende Rolle spielt (D'Esposito et al. 1998).

- *Episodischer Speicher*: verschiedene aktuelle Forschungserkenntnisse konnten mit älteren Arbeitsgedächtnismodellen nicht mehr vollständig erklärt werden. Aus diesem Grund wurde im Jahre 2000 eine vierte Komponente zum Arbeitsgedächtnismodell hinzugefügt (Baddeley 2000). Der episodische Speicher gilt als ein multimodales Speichersystem, welches visuelle, räumliche und phonologische Informationen mit spezifischen Informationen des episodischen Gedächtnisses verknüpft. Des Weiteren vermutet man Verbindungen zum Langzeitgedächtnis und zum semantischen Gedächtnis (Baddeley 2000, Maguire et al. 2006).

Aufgrund der geplanten Pilotstudie mit VR-Stimulusreizen gewinnt die Verarbeitung von visuellen Informationen durch den räumlich-visuellen Speicher eine besondere Bedeutung. Im folgenden Abschnitt sollen die Besonderheiten des räumlich-visuellen Speichers näher erläutert werden.

3.2.1 Räumlich-visueller Speicher

Das Verarbeiten räumlicher Informationen ist ein spezieller Aspekt des Arbeitsgedächtnisses. Ausgehend von dem vorgestellten Modell, ist der räumlich-visuelle Speicher gezielt für die temporäre Speicherung und Manipulation von räumlich-visuellen Informationen zuständig, was auch die Funktionalitäten zur räumlichen Orientierung einschließt (Pollmann 2008).

Ein wichtiger Aspekt dieses Modells ist die sogenannte *mentale Rotation*, deren Funktionen im räumlich-visuellen Speicher vermutet werden (Pollmann 2008).

Die mentale Rotation beschreibt hier vornehmlich die Funktion zur Wiedererkennung ein und desselben Objekts im Raum, auch wenn dieses von unterschiedlichen Positionen wahrgenommen wird (Schneider und Fink 2007).

Ausgehend davon unterscheidet man auf perzeptueller Ebene zwischen der Objektwahrnehmung und der Wahrnehmung der Lage eines Objektes. Da topographische Navigation vornehmlich anhand von Landmarken realisiert wird, fällt der mentalen Rotation hier eine besondere Bedeutung zu (Sanchez-Vives und Slater 2005, Marsh et al. 2010).

Ein weiterer wichtiger Aspekt, der die Objektwahrnehmung beeinflusst, ist die Veränderung der egozentrischen Perspektive durch gleichzeitige Bewegung. Neueste Forschungsergebnisse zeigen, dass während der simulierten Bewegung innerhalb einer virtuellen Umgebung die Aktualisierung der eigenen Position in Relation zu bekannten Landmarken höchste Priorität hat und das vor allem Areale im Cuneus und im dorsalen Präfrontalen Kortex involviert sind (Wolbers et al. 2008).

Des Weiteren werden sämtliche Perspektivänderungen mental abgespeichert und können auch noch nach einer längeren Zeitperiode wiedergegeben werden. So wird deutlich, dass durch die Einbindung von spezifischen Bezugspunkten bzw. Landmarken das

Enkodieren von topographischen Besonderheiten entscheidend erleichtert wird (Maguire et al. 2006).

Weitere fMRT-Studien zeigten, dass beim Speichern von räumlich-visuellen Informationen vorwiegend der ventrolaterale Präfrontale Kortex sowie Gebiete im Parietallappen aktiviert waren. Dagegen wird für die Manipulierung von räumlich-visuellen Informationen der dorsolaterale Präfrontale Kortex und der ventrolaterale Präfrontale Kortex verantwortlich gemacht (Abb. 3.1). Zudem vermutet man eine Dominanz dieser Prozesse in der rechten Hirnhemisphäre (Baddeley 2000, Schneider und Fink 2007).

3.3 Langzeitkonsolidierung

Bisher wurden die relevanten Funktionen des Kurzzeitgedächtnisses vorgestellt und in Relation zum topographischen Lernen gesetzt. Für eine langfristige Erkennung von topographischen Merkmalen ist jedoch die Konsolidierung der Kurzzeitinformatoren in das Langzeitgedächtnis erforderlich (Pollmann 2008).

Dies gewinnt besonders an Bedeutung, wenn häufiger auf bestimmte topographische Informationen zugegriffen wird oder die Informationen nach einer längeren Zeitperiode dekodiert werden (Sturm 2009).

Das Langzeitgedächtnis wird in der Literatur als Komponente mit langfristig verhaltensmodifizierenden Prozessen beschrieben (Pollmann 2008).

Aufgrund der Weitläufigkeit und der Komplexität dieser Definition müssen auch unbewusste Lernvorgänge zu den Funktionen hinzugerechnet werden. Prinzipiell erfolgt die Unterteilung des Langzeitgedächtnisses aufgrund der Inhalte und kann in zwei wesentliche Zweige aufgespalten werden. Dies ist zum einen das deklarative Gedächtnis (a) und zum anderen das nondeklarative Gedächtnis (b). Im deklarativen Teil des Langzeitgedächtnis befinden sich das episodische Gedächtnis, welches allgemeine Kontextinformationen und Detailinformationen bereitstellt, sowie das semantische Gedächtnis, welches Umgebungswissen sowie Faktenwissen speichert (Pollmann 2008).

Das nondeklarative Gedächtnis lässt sich wiederum in vier Teile klassifizieren. Das nicht-assoziative Lernen, die klassische Konditionierung, das Priming sowie die allgemeinen Fertigkeiten (Schneider und Fink 2007).

Für die hier vorliegende Studie mit Aspekten zum topographischen Lernen sind vor allem das episodische und das semantische Gedächtnis von besonderer Bedeutung. Die Verarbeitung von semantischen bzw. episodischen Informationen ist aus heutiger Sicht noch nicht vollständig erforscht. Es wird angenommen, dass Informationen zunächst kurzzeitig in kortikalen Assoziationsarealen abgelegt und später weiterverarbeitet werden. Der laterale parietale Kortex wird hier hauptsächlich, aber nicht ausschließlich, in der Literatur genannt (Sturm 2009).

Von dort werden Informationen in das limbische System übermittelt, wo schließlich die eigentliche Enkodierung sowie Konsolidierung ins Langzeitgedächtnis vermutet wird. Das *limbische System* wird daher oftmals als zentrales Areal zum Transport der Informationen zu neokortikalen Speicherorten genannt (Birbaumer und Schmidt 2006, Pollmann 2008).

Von besonderer Bedeutung sind hier die hippokampalen und parahippokampalen Struk-

turen sowie das Cerebellum, welche bei der Enkodierung und der Dekodierung von topographischen Informationen Aktivität zeigten (Aguirre et al. 1996, Aguirre 1998, Maguire et al. 2006, Ekstrom et al. 2003).

3.4 Neokortikale Repräsentation räumlicher Informationen

Wie bereits in den vorherigen Unterkapiteln gezeigt, stellen der sensorische Input sowie die räumlich-visuellen Informationen entscheidende Faktoren bei der Untersuchung der topographischen Navigation dar. Durch die Präsentation einer virtuellen Umgebung können verschiedene räumlich-visuelle Informationen realitätsnah jedoch gleichzeitig sehr kontrolliert abgebildet werden (Aguirre 1998, Maguire et al. 2006, Ekstrom et al. 2003).

Anhand diverser Studien weiß man, dass die Präsentation einer dreidimensionalen virtuellen Umgebung vor allem neuronale Aktivitäten in Bereichen der primären und der sekundären Sehrinde hervorruft. Insbesondere die neuronale Aktivierung für Tiefenwahrnehmungseffekte, größtenteils lokalisiert im Mittleren Okzipitalen Gyrus, wurde für ähnliche Versuchsbedingungen bereits nachgewiesen (Abb. 3.1) (Baecke et al. 2009).

Es wird vermutet, dass die Enkodierung topographischer Merkmale vordergründig durch die beschriebenen Prozesse des Arbeitsgedächtnisses geschieht. Sensorische Informationen werden zunächst über den räumlich-visuellen Speicher verarbeitet und im weiteren Verlauf über die hippocampalen Areale enkodiert. Episodische und semantische Informationen werden mit den visuellen Reizen verknüpft und anschließend abgespeichert. In verschiedenen fMRT-Studien wurde dieser Prozess oft mit neuronaler Aktivität im Cuneus und Precuneus in Verbindung gebracht (Abb. 3.1) (Aguirre et al. 1996, Burgess et al. 2002, Frings et al. 2006, Andreano et al. 2009, Mellet et al. 2010). Des Weiteren existieren Hinweise auf eine schrittweise Abspeicherung einer sogenannten Umgebungsrepräsentation (Maguire et al. 1997, Aguirre 1998). Durch verschiedene Experimente konnten Funktionen zur Erkennung von Gebäuden und Orten im parahippokampalen Kortex identifiziert werden (Aguirre et al. 1996, Burgess et al. 2002, Maguire et al. 2006, Wolbers et al. 2008).

Aufbauend auf diesen Ergebnissen entstanden verschiedene Theorien, welche die Interaktion zwischen räumlicher Orientierung und Wegfindungsprozessen mit den Aktivierungen in parahippokampalen Arealen erklären sollten. Aus diesem Grund wird der vordere Teil des Gyrus Parahippocampalis in der Literatur gern als *Parahippocampal Place Area* bezeichnet und soll für eben diese Umgebungsrepräsentation verantwortlich sein (Ekstrom et al. 2003).

Viele spezifische Fragen zu egozentrischen und exozentrischen Weltsichten oder Entfernungsabschätzungen durch metrische Distanzeinheiten sind bisher noch unbeantwortet (Wolbers et al. 2008). Für weitere Informationen zu diesem Thema sei auf die aktuellen Forschungen zu altersbedingten Störungen der räumlichen Orientierung hingewiesen (Pine et al. 2002).

3.5 Immersionseffekt und Vigilanzzustände in Virtuellen Realitäten

Ein großes Ziel der Neurowissenschaften ist die Erforschung kognitiver Prozesse des Menschen. In Experimenten mit einfachen Stimuli werden Teilaspekte aus unserem alltäglichen Leben abgebildet. Diese erlauben eine vereinfachte Darstellung komplexer Funktionen des menschlichen Gehirns und sollen Aufschluss über dessen Arbeitsweise geben (Savoy 1998, Huettel et al. 2008). Dieses Prinzip verspricht nicht immer Erfolg. Teilweise sind die Stimuli sehr weit von alltäglichen Prozessen entfernt oder zu einfach im Versuchsaufbau gestaltet. Hier tritt oft ein gegenteiliger Effekt ein. Bei Versuchspersonen verschiebt sich die Aufmerksamkeit weg vom eigentlichen Stimulus. Es treten Störquellen auf, die das Versuchsergebnis nachhaltig beeinflussen und somit auch manipulieren können (Savoy 1998).

Virtuelle Umgebungen versprechen hier eine Verbesserung. Abhängig vom Detailgrad und der Realitätsnähe können Virtuelle Umgebungen durch den sogenannten *Immersionseffekt* beim Probanden einen Zustand höherer Aufmerksamkeit hervorrufen (Leins et al. 2007). Hier wird vermutet, dass die Erhöhung der Aufmerksamkeit eine allgemeine Effektivitätssteigerung neurowissenschaftlicher Experimente nach sich zieht oder gar zu einer Verstärkung der gewünschten Hirnaktivität führen kann (Wiederhold und Wiederhold 2005). Die Ergebnisse von verschiedenen Studien bestätigen zu großen Teilen diese Vermutung (Rothbaum et al. 2000, Pine et al. 2002, Wiederhold und Wiederhold 2005, Maguire et al. 2006, Mueller et al. 2012b). Allerdings müssen für fundierte Erkenntnisse weitere Studien zu diesem spezifischen Forschungsthema durchgeführt werden. Im Folgenden sollen die Begriffe Immersion und Vigilanz näher vorgestellt und ihre Zusammenhänge mit Stimulusreizen in Virtuellen Umgebungen erläutert werden.

3.5.1 Immersion

Virtuelle Realitäten sind in erster Linie Benutzerschnittstellen. Sie stehen für Umgebungen, in denen die Nutzer innerhalb einer simulierten Realität handeln, eine Applikation steuern und sich im Idealfall so verhalten, wie sie es in ihrem realen, bekannten Umfeld tun würden (Sherman und Craig 2003, Brill 2009). Genau dieses immersive Verhalten eröffnet für neurowissenschaftliche Experimente neue Möglichkeiten und erlaubt eine authentischere Präsentation neurowissenschaftlicher Stimuli (Steuer 1992). Der Begriff Immersion, der in der psychologisch-neurowissenschaftlichen Literatur auch oft als „sense of presence“ bezeichnet wird (Slater et al. 1994, Schubert 2003), bezieht sich vornehmlich auf das subjektive Gefühl des Anwenders, welcher eine generierte Umgebung zu einem hohen Grad als real wahrnimmt. Dieser Effekt und seine zugrunde liegenden kognitiven Mechanismen sind bisher nur eingeschränkt erforscht worden (Sanchez-Vives und Slater 2005). Dennoch wird dieser Effekt bereits genutzt und steigert insbesondere bei Therapien mit Kindern und Jugendlichen das Gesamtergebnis (Wiederhold und Wiederhold 2005). Dabei wird die Effektivitätssteigerung vor allem auf die höhere Motivation der Probanden und die hohe Akzeptanz

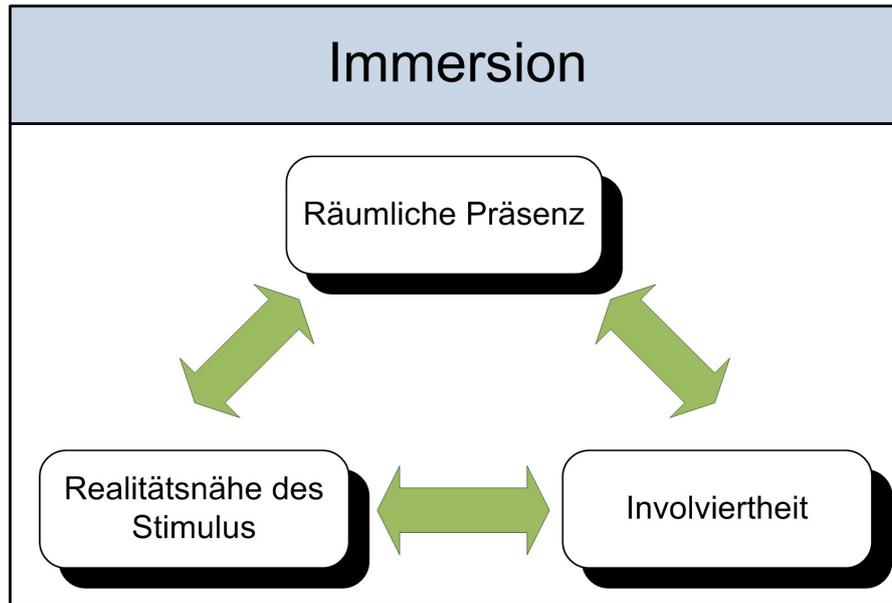


Abbildung 3.2: Das *three-presence components*-Modell immersiven Verhaltens aus der Medienpsychologie definiert die Grundvoraussetzungen für Immersion. Eigener Entwurf nach (Schubert 2003).

der VR-Stimuli zurückgeführt. Aus medienpsychologischer Sicht ist die Immersion in Virtuellen Umgebungen ein Effekt, der von drei Teilaspekten abhängig ist (Schubert 2003, Sanchez-Vives und Slater 2005):

1. Räumliche Präsenz
2. Involviertheit
3. Realitätsnähe der Virtuellen Umgebung (Stimulus)

Die räumliche Präsenz beschreibt dabei die Frage, ob sich der Proband in der virtuellen Welt präsent gefühlt hat (Abb. 3.2). Hatte er das Gefühl, selbst in der Welt zu agieren statt nur etwas von „außerhalb“ zu steuern? Die Einbindung in die virtuelle Welt oder die Involviertheit ist der zweite Aspekt. Hier ist die Aufmerksamkeit des Probanden in Bezug auf die Virtuelle Welt gemeint. War die Aufmerksamkeit des Probanden dauerhaft auf die Virtuelle Umgebung gerichtet oder wechselte seine Aufmerksamkeit häufig zur realen Welt? Wie stark war sein Bezug zur realen Welt während des VR-Stimulus? Der letzte Aspekt ist die Realitätsnähe der VR-Umgebung. Hier findet ein Vergleich des Probanden zwischen der ihm präsentierten, künstlichen Welt und der realen Welt statt. Potentielle Fragen, die auf diesen Aspekt abzielen, sind: waren Erfahrungen in der künstlichen Welt konsistent zu Erfahrungen aus der realen Welt oder wie real wirkte die künstliche Welt auf dich (Schubert 2003). Insgesamt bilden diese drei Komponenten (Three-presence components) einen wirksamen Indikator zur Ermittlung des Immersionseffektes und indizieren damit erst die Effizienz von VR-Stimuli.

3.5.2 Vigilanz und saliente Reize

Jede Reaktion auf eine neue Erfahrung oder Reiz erfordert zumeist einen Vergleich mit bereits gelernten Reizen. Kommen wir mit neuen oder unbekanntem Reizen in Berührung, kommt es zu neurophysiologischen Reaktionen und damit zu einer vermehrten Abgabe von Neurotransmittern. Einige dieser Neurotransmitter sind: Acetylcholin, Dopamin, Serotonin, Noradrenalin, und Histamin (Ehlers und Holder 2007). Prozesse, die mit einer neuen Reizverarbeitung zu tun haben, erhalten in Folge dessen eine gesteigerte Priorität. Prozesse, die vorher verarbeitet wurden, rücken dadurch in den Hintergrund. Der Körper stellt sich auf eine mögliche Aktivität ein und fördert gleichzeitig die Verarbeitung ablaufender Prozesse. Dieser Zustand der erhöhten Wachheit wird als *Vigilanz* bezeichnet (Birbaumer und Schmidt 2006).

Vigilanz ist im Vergleich zum Immersionseffekt recht gut untersucht worden und wird prinzipiell anhand seiner Intensität eingeteilt. Bei unbekanntem Reizen oder bei Alltagsaufgaben, wie z.B. Autofahren, kann eine selektive Wahrnehmung beobachtet werden, die letztendlich mit einem einfachen Vigilanzzustand einhergeht. Dieser einfache Vigilanzzustand wird als „Erregung“ definiert und verursacht eine gesteigerte Aktivierung im Hirnstamm und im Hypothalamus (Birbaumer und Schmidt 2006). Kommt es zu erhöhten Bedrohungs Zuständen, kann eine Überforderung des Organismus beobachtet werden, hier spricht man von Belastungsstress (Ehlers und Holder 2007). Auch wenn dieser Zustand als ein Indikator für ein hohes Realitätsempfinden gilt, so stellt er für die nachfolgende experimentelle Evaluierung ein Störfaktor dar und muss als Kontraindikation angesehen werden (Huettel et al. 2008).

Prinzipiell wird bei VR-Stimuli oft ein Zustand der einfachen Vigilanz beobachtet, der vornehmlich auf die Neuheit des VR-Stimulus zurückzuführen ist. Wird derselbe VR-Stimulus einem Probanden häufiger präsentiert, schwächt sich der Vigilanzzustand ab und es tritt schnell eine Verschiebung der Aufmerksamkeit ein (Mueller et al. 2011a). Insgesamt betrachtet gelten so auch die sozialpsychologischen Charakteristika von salienten Reizen. Bei VR-Welten kommt es zu einer selektiven Wahrnehmung mit allem was die künstliche Welt betrifft. Solange der VR-Reiz als intensiv und neu wahrgenommen wird, bleibt der Zustand der erhöhten Wahrnehmung bestehen (Schubert 2003).

3.5.3 Neurokognitive Anwendungen Virtueller Realitäten

Zunächst muss eine grundlegende Unterscheidung zwischen regulären Virtual-Reality Systemen und neurowissenschaftlich motivierten Anwendungen zur Stimuluspräsentation vorgenommen werden. Der Begriff „Virtual Reality“ wurde im neurowissenschaftlichen Kontext Ende der 1980er Jahre geprägt. Zu diesem Zeitpunkt war der Begriff ein überaus interessantes Schlagwort, welches sehr gerne mit dem Einsatz künstlicher Umgebungen aus regulären Computerspielen in Verbindung gebracht wurde (Rothbaum et al. 1999, Sherman und Craig 2003, Wiederhold und Wiederhold 2005, Brill 2009).

In den letzten Jahren gab es zunehmend neurowissenschaftliche Studien, die ihren Stimulus in einer VR präsentierten und so die speziellen Vorteile dieser Technologie genutzt haben. So kamen Virtuelle Umgebungen bei verschiedenen Studien zur Erfor-

schung von kognitiven Mechanismen zum Einsatz. Als Beispiel seien hier fMRT und PET Studien mit Virtuellen Umgebungen zur Erforschung des Hippocampus und seiner Rolle beim topographischen Lernen bzw. beim räumlichen und episodischen Gedächtnis genannt (Aguirre 1998, Burgess et al. 2002, Pine et al. 2002, Maguire et al. 2006, Hassabis et al. 2009, Mellet et al. 2010). Der große Vorteil solcher Studien liegt in der scheinbaren Aufhebung der Immobilität innerhalb medizintechnischer Akquirierungsgeräte. So kann sich ein Proband mittels VR durch real wirkende Umgebungen bewegen und dadurch echte Erfahrungen sammeln (Brill 2009).

Des Weiteren sollte durch Virtuelle Umgebungen der Nachweis von sogenannten *place cells* beim Menschen erbracht werden (Aguirre et al. 1996, Ekstrom et al. 2003, Doeller et al. 2010). Ein weiteres Beispiel für den Einsatz von Virtuellen Umgebungen im neurowissenschaftlichen Kontext sind sogenannte Neurofeedback-Studien oder Hirn-Computer Schnittstellen. Die Visualisierung des Neurofeedback-Signals erfolgt dabei nicht über standardmäßige Anzeigen, sondern wird anhand von Visualisierungstechniken direkt in die Virtuelle Umgebung eingebettet, z.B. durch Änderung von Wetterverhältnissen anhand der steigenden oder sinkenden Aktivierung bestimmter Neuronenverbände in vordefinierten Hirnarealen (Mueller et al. 2012a) (vgl. Abschnitt 3.8).

Trotz der vielen technischen Herausforderungen existieren bereits Studien, die sich den Effekt der Immersion gezielt zu Nutze machen. So wurden VR-Umgebungen beispielsweise begleitend zu Brennschmerz-Therapien (Hoffman et al. 2000), zur Reduzierung von klaustrophobischen Symptomen (Garcia-Palacios et al. 2007) oder als begleitende Therapiemaßnahme bei posttraumatischen Belastungsstörungen eingesetzt (Rothbaum et al. 1999). Die Basis dieser psychologischen Studien bildet oftmals die sogenannte Virtual Reality Konfrontationstherapie (virtual reality exposure therapy) (Rothbaum et al. 2000). Der zentrale Aspekt dieser Studien ist die Annahme, dass sich durch die gesteigerte Realitätsnähe und das höhere Interaktionsverhalten mit dem Paradigma bessere Therapieergebnisse erzielen lassen (Wiederhold et al. 2002, Andrea-no et al. 2009). Trotz der steigenden Anzahl von neurowissenschaftlichen Studien mit VR-Umgebungen, wurde in den letzten Jahren nur wenig, für Nicht-Experten brauchbare, Stimulussoftware entwickelt (Mueller et al. 2012b). Der Hauptgrund hierfür ist die zeitaufwändige Anpassung der VR-Umgebung an die neurokognitiven Experimentanforderungen und die damit verbundene Gefahr, dass experimentelle Design und das Forschungsziel nachhaltig zu beeinflussen.

3.6 Hirn-Computer-Schnittstellen - Geschichte und Motivation

Mit Hirn-Computer-Schnittstellen oder auch *Brain-Computer-Interfaces* (BCI) werden Verfahren und Versuchsanordnungen bezeichnet, welche interne Hirnaktivierungen in externe Signale umwandeln und so zwischen dem menschlichen Gehirn und einem Computer eine Verbindung herstellen können (Rief und Birbaumer 2006). Geprägt wurde der Begriff durch Jaques Vidal in den Jahren 1973-1977 (Vidal 1977). Sein Hauptinteresse lag vor allem in der Verwendung von Hirnsignalen zum Bau einer

mentalen Prothese (Sanei und Chambers 2007). Vidal definierte zunächst den Begriff *Hirn-Computer-Schnittstelle* als ein einfaches computerbasiertes System, welches Informationen über Hirnaktivitäten aufzeichnet (Vidal 1977).

Heute steht BCI für ein computerbasiertes System, welches vor allem zur Kommunikation zwischen Mensch und Maschine eingesetzt wird. Der Mensch verfügt, genau wie ein moderner Computer, über zahlreiche Schnittstellen, über welche er mit der Außenwelt in Verbindung treten kann: Die Augen gewährleisten die visuelle Wahrnehmung, die Gesichtsmuskulatur dient der Mimik, der Sprechapparat und die Ohren ermöglichen akustische Kommunikation (Rief und Birbaumer 2006). Bei der Definition eines BCI wird deutlich, dass diese Kommunikationswege bewusst außen vor gelassen werden. Vielmehr sollen neue Kommunikationswege erschlossen werden, die eine synthetische Kommunikation über technisch realisierte Pfade erlauben (Wolpaw et al. 2002). Ausgehend von dieser Definition wird ein wichtiger Anwendungsbereich eines BCI in der Therapie neuromuskulärer Erkrankungen bzw. Schädigungen wie Amyotropher Lateralsklerose, Schlaganfall des Hirnstamms, Hirn- oder Rückenmarksverletzungen, muskulären Dystrophien oder Multiple Sklerose deutlich (Birbaumer und Schmidt 2006). Als Prothese kann hier eine Hirn-Computer-Schnittstelle dienen. Sie soll die Schäden des zentralen oder peripheren Nervensystems kompensieren und eine Kontrolle über verlorenen Funktionen ermöglichen. Dieser Ansatz soll die Mobilität von Personen mit motorischen Behinderungen steigern oder gar wiederherstellen. Dahingehend bezog sich Vidal mit seinen Aussagen auf sogenannte EEG-BCI, bei denen ereigniskorrelierte Potentiale direkt von der Kopfoberfläche abgeleitet werden (Coben und Evans 2010).

Durch die Einführung der Echtzeit-fMRT und die erstmalige Präsentation einer Apparatur zur Echtzeit-fMRT Datenanalyse trat eine weitere Methodik zur BCI Implementation in den Vordergrund (Cox et al. 1995). Aufgrund der schlechten zeitlichen Auflösung bildgebender Verfahren, stellt die Nutzung der BOLD-fMRT zur Prothesensteuerung jedoch ein weniger praktikablen Ansatz dar. Der große Vorteil der BOLD-fMRT, das hochauflösende Aufzeichnen von Hirnaktivität aus tiefer liegenden Hirnarealen, erschließt jedoch eine Reihe von weiteren Forschungsoptionen, wie z.B. Neurofeedback-Studien auf Basis der Echtzeit-fMRT (Weiskopf et al. 2004b).

Ein weiterer Meilenstein in der Entwicklung von BCI war die Einbeziehung von Methoden aus dem Bereich des Maschinellen Lernens und des Data-Mining. Gerade im medizinisch-neurowissenschaftlichen Bereich fallen pro Experiment große Datenmengen an. Probleme treten gerade beim Datenmanagement und bei der Analyse dieser Datenmengen auf. Um schließlich die statistische Auswertung zu unterstützen und ein Optimum an Informationen aus den Datenmengen herauszufiltern, kamen Algorithmen wie die „Support Vector Machine“ oder Algorithmen aus der Graphentheorie (dynamische Graphen) zum Einsatz (Schneider und Fink 2007, Coben und Evans 2010). Auf diese Weise konnten in kürzerer Zeit bessere Aussagen über die Ergebnisse getroffen und die Verarbeitung neuronaler Signale durch herkömmlichen BCI beschleunigt werden (Blanchard und Blankertz 2004, Tan und Nijholt 2010).

BCI werden aber nicht nur bei der Therapie motorischer Erkrankungen eingesetzt. Ein weiteres Anwendungsgebiet ist die Therapie von neuronaler Dysregulation, z.B. im Bereich der Schmerztherapie. Auch hier spielt das Prinzip des Neurofeedback eine

wesentliche Rolle und bildet die Grundlage für die Selbstregulation neuronaler Aktivität mittels BCI (deCharms et al. 2005).

3.7 Neurofeedback und Hirn-Computer-Schnittstellen

Als zentrale Anwendungsform von Hirn-Computer-Schnittstellen gilt das sogenannte Neurofeedback-Konzept. Dieses Konzept wird im Allgemeinen als eine psychotherapeutische Behandlungsmethode verstanden, welche versucht, durch Konditionierungsmethoden die Auswirkungen verschiedener Krankheiten, wie z.B. ADHS oder Epilepsie, schrittweise zu senken (Coben und Evans 2010). Hier können sogar ähnliche positive Effekte wie bei der therapeutischen Behandlung mit Medikamenten, wie z.B. Ritalin, beobachtet werden. Diese Ergebnisse lassen Neurofeedback als eine mögliche Alternative zur medikamentösen Behandlung erscheinen (Rief und Birbaumer 2006). Das Grundmodell hinter Neurofeedback ist das sogenannte Prinzip der Selbstregulierung neuronaler Aktivität durch gezielte operante Konditionierung (Weiskopf et al. 2007). Dieses Prinzip baut insgesamt auf vier Grundbausteinen auf, welche während

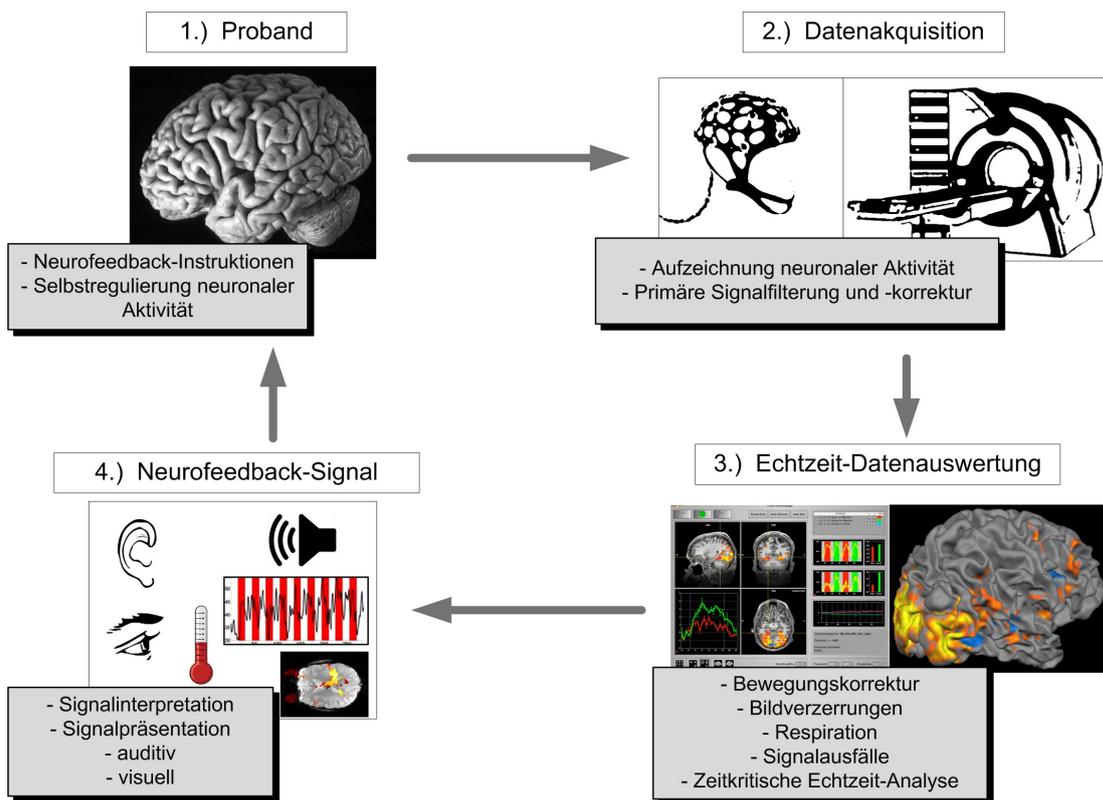


Abbildung 3.3: Das Prinzip der Selbstregulierung lokaler Hirnaktivität durch gezielte operante Konditionierung baut zunächst auf einer Instruierung des Probanden (1) mit einer Neurofeedback-Aufgabe zur Selbstregulierung auf. Anschließend wird die neuronale Aktivität registriert (2) sowie zeitkritisch, statistisch ausgewertet (3). Resultierende Informationen werden dem Probanden über ein Neurofeedback-Signal (4) rückübermittelt. Eigener Entwurf nach (Weiskopf et al. 2007).

des Neurofeedback-Trainings verschiedene Informationen untereinander austauschen (Abb. 3.3).

Der erste Baustein in diesem Konzept ist der Patient oder Proband. Dieser erhält zunächst Instruktionen und eine spezifische Neurofeedback-Aufgabe zur Regulierung seiner neuronalen Aktivität in einem vordefinierten Hirnareal. Gleichzeitig wird durch den zweiten Baustein, die Datenakquisition, die neuronale Aktivität des Probanden registriert und durch Algorithmen zur Signalfilterung und -korrektur vorverarbeitet. Die so akquirierten Daten werden anschließend durch eine Applikation zur Echtzeit-Datenauswertung noch einmal intensiver von spezifischen Störsignalen und Bewegungsartefakten befreit und schließlich in einem zeitkritischen, statistischen Auswerteprozess analysiert. Die resultierenden Informationen werden dann zum vierten Baustein des Konzepts weitergeleitet, bei dem das Neurofeedback-Signal zunächst interpretiert und anschließend in geeigneter Form, meist visuell oder auditiv, an den Probanden rückübermittelt wird. Dieser erhält so ein konkretes Feedback und kann im nächsten Schritt weitere Versuche zur Regulierung seiner neuronalen Aktivität unternehmen (Rief und Birbaumer 2006, Weiskopf et al. 2007, Coben und Evans 2010).

Basierend auf diesem Konzept des Neurofeedback haben sich abhängig von der Modalität der Datenakquisition zwei Hauptkategorien von Hirn-Computer-Schnittstellen entwickelt (Clausen 2006). Der generelle Aufbau dieser Systeme bleibt dabei weitgehend einheitlich. Ein BCI benötigt zunächst eine Input-Komponente (a), mit der geeignete neuronale Signale aufgezeichnet werden können. Im zweiten Schritt müssen diese Signale mit Hilfe von Algorithmen (b) analysiert und interpretiert werden. Die so gewonnenen Daten werden anschließend, z.B. zur Steuerung von Prothesen, an eine sogenannte Output-Komponente (c) übertragen (Clausen 2006). Dabei ist zu beachten, dass BCI-Systeme nicht ohne weiteres von einem Probanden bedient werden können. Ähnlich wie beim Neurofeedback, erfordert die flüssige Bedienung ein komplexes sowie zeitaufwendiges Training mit mehreren Übungssitzungen, welches durch einen Operanten gezielt angeleitet werden (Pfurtscheller et al. 2006, Rief und Birbaumer 2006).

Die verschiedenen Formen von Hirn-Computer-Schnittstellen ebenso deren Aufbau, deren Vor- und Nachteile sowie deren Anwendungsbereich und Risiken im neurowissenschaftlichen Forschungsbereich sollen im Folgenden näher vorgestellt werden.

3.7.1 Invasive Hirn-Computer-Schnittstellen

Als invasive Hirn-Computer-Schnittstellen werden Methoden bezeichnet, bei denen Geräte oder medizin-technische Schnittstellen direkt in den Körper eingepflanzt werden (Coben und Evans 2010). Für invasive Hirn-Computer-Schnittstellen werden als Input-Komponente hauptsächlich zwei verschiedene intrakraniale Signalleitungsarten verwendet und unterschieden. Die erste Methodik stellt die intrazerebrale Tiefenhirnableitung dar, bei der einzelne Mikroelektroden direkt in tiefer liegende Schichten des Gehirns eingepflanzt werden (Birbaumer und Schmidt 2006).

Die zweite Methodik ist das sogenannte Elektrokortikogramm (ECoG). Dort werden sogenannte Grid-Elektroden subdural, unterhalb der Hirnhaut, und über Regionen der Hirnoberfläche platziert (Wang et al. 2009) (Abb. 3.4). Grid-Elektroden bestehen oft

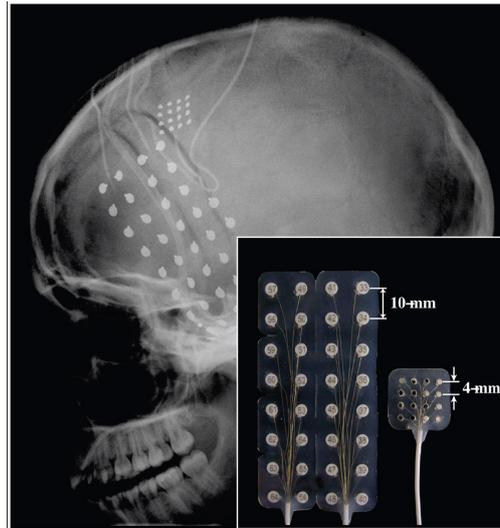


Abbildung 3.4: Beim Elektrokortikogramm werden Mikro-ECoG-Elektroden auf der Hirnoberfläche platziert. Elektrische Signale werden anschließend direkt von der Kopfhaut abgeleitet und über Kabelverbindungen zur Datenauswertung transportiert (Wang et al. 2009).

aus in Kunststoff eingeschweißte Platin-Iridium-Elektrodenkontakten. Die Anzahl kann dabei mit 4 bis 64 Kontakten variieren. Signale die von invasiven Input-Komponenten aufgenommen werden, gelangen meist über Kabelverbindungen vom Gehirn direkt zu einer Verarbeitungskomponente. Hier werden die Signale ausgewertet und zu einer Output-Komponente weitergeleitet. Dies kann ein Computer-Monitor oder eine externe Prothese sein (Chapin et al. 1999). Beim Elektrokortikogramm werden ebenso wie auch beim Elektroenzephalogramm die summierten elektrischen Potentiale größerer Neuronenverbände erfasst und analysiert. Dass sich derartige Signale zur Steuerung eines BCI eignen könnten, deutete sich erstmalig Mitte der 1970er Jahre an (Fetz und Finocchio 1975). Hier konnte in Tierversuchen gezeigt werden, dass Affen durch längeres Training (operante Konditionierung) die Entladungsrate einzelner kortikaler Zellen willentlich beeinflussen konnten. Von diesem Prinzip ausgehend, gelang in den späten 1990er Jahren eine erweiterte Anwendungsmöglichkeit dieses Ansatzes (Chapin et al. 1999). In Experimenten mit Ratten wurde gezeigt, dass es möglich ist, über die Entladungsraten kortikaler Neurone in Echtzeit einen Roboterarm zu steuern. Dazu wurden in sechs Versuchstieren Mikroelektroden im primären motorischen Kortex und im ventralen Thalamus implantiert. Die Ratten wurden durch operante Konditionierung dazu gebracht, mit ihrer Pfote einen Hebel zu betätigen. Der Hebeldruck löste eine zur Stärke des Pfootendrucks proportionale Bewegung eines Roboterarms aus, an der wiederum eine Trinkapparatur befestigt war. Durch die Bewegung des Roboterarms gelangte die Apparatur in die Reichweite der Tiere, so dass diese trinken konnten. Die Vorteile invasiver BCI-Techniken beruhen hauptsächlich auf einem besseren Signal-Rausch-Verhältnis. Aufgrund der subduralen Lage der ableitenden Elektroden sind vor allem die Abstände zu den aktivierten Neuronenverbänden wesentlich geringer. Des Weiteren wirkt sich der größere Abstand zu den externen Artefaktquellen, wie z.B.

Muskelbewegungen, positiv auf das Signal aus. Insgesamt sind die Amplituden im Durchschnitt 3 bis 10 Mal höher als im EEG (Birbaumer und Schmidt 2006). Zusätzlich dazu sind Tiefenhirnelektroden besonders sensitiv und erlauben in tiefer liegenden Hirnregionen eine sehr gute Signalerfassung (Baumgartner et al. 1998). Trotz dieser Vorteile besitzen invasive Hirn-Computer-Schnittstellen aber auch einige hervorsteckende Risiken und Nachteile. Neben der ethischen Vertretbarkeit solcher Methoden, ist die Implantation einer invasiven Input-Komponente auch immer mit einer Operation und den damit verbundenen Risiken verknüpft. Die Gefahr von Infektionen oder Hirnblutungen bei der Operation ist immer präsent und kann unweigerlich zu Komplikationen führen (Birbaumer und Schmidt 2006). Ein weiterer großer Nachteil ist der zeitlich begrenzte Einsatz von invasiven Input-Komponenten. Künstliche Implantate sind einem Abstoßungsvorgang unterworfen. Das heißt, dass der Körper auf die implantierten Elektroden reagiert und dementsprechend Immunreaktionen auslöst (Wang et al. 2009). Ausgehend davon müssen invasive Input-Komponenten nach einiger Zeit aus dem Körper entfernt werden. Die typische Betriebsdauer von ECoG-Implantaten ist abhängig von den Immunreaktionen des Organismus und beträgt durchschnittlich sechs bis zwölf Monate. Aus diesen Gründen werden invasive BCI meist nur in besonderen krankheitsbedingten Fällen beim Menschen, wie z.B. bei Epilepsie, eingesetzt (Coben und Evans 2010).

3.7.2 Nichtinvasive Hirn-Computer-Schnittstellen

Im Gegensatz zu invasiven Hirn-Computer-Schnittstellen werden bei nichtinvasiven Methoden keine medizin-technischen Geräte in den Körper implantiert. Alle Input-Komponenten arbeiten extern und nehmen entweder Signale direkt von der Kopfhaut auf oder benutzen andere nichtinvasive bildgebende Verfahren, wie z.B. die Magnetresonanztomographie (Weiskopf et al. 2007). Ausgehend davon existiert eine Vielzahl von Modalitäten mit denen der Aufbau nichtinvasiver-BCI möglich ist. Als Beispiel seien die Elektroenzephalographie (EEG) (Pfurtscheller et al. 2006), die funktionelle Magnetresonanztomographie (fMRT) (Sitaram et al. 2007a), die Magnetoenzephalographie (MEG) (Mellinger et al. 2007), die Positronen-Emissions-Tomographie (PET) (Pfurtscheller und Neuper 2001) oder die Nahinfrarotspektroskopie (NIRS) (Sitaram et al. 2007b) genannt. Eine Übersicht zu den verschiedenen Modalitäten für nichtinvasive und invasive BCI bietet Abbildung 3.5. Im Aufbau unterscheiden sich diese BCI wenig. Nichtinvasive BCI besitzen ebenfalls eine Input-Komponente, diverse Auswertelgorithmen sowie eine Output-Komponente. Die Verwendung von bildgebenden Verfahren, wie der fMRT, verlangt jedoch völlig andere Akquirierung- und Auswertemethoden als beispielsweise bei der Elektroenzephalographie (Birbaumer und Schmidt 2006).

Dennoch sind EEG-BCI und fMRT-BCI wohl die populärsten Typen von Hirn-Computer-Schnittstellen. Hier müssen BCI vorwiegend anhand ihrer räumlichen und zeitlichen Auflösung unterschieden werden. Modalitäten, wie z.B. EEG, haben eine hohe zeitliche Auflösung und erlauben eine schnelle und echtzeitnahe Kommunikation mit Prothesen. Modalitäten, wie z.B. fMRT, PET sowie diverse optische Verfahren, sind teuer in der Anschaffung aber haben ihre Stärke in der räumlichen Auflösung (Co-

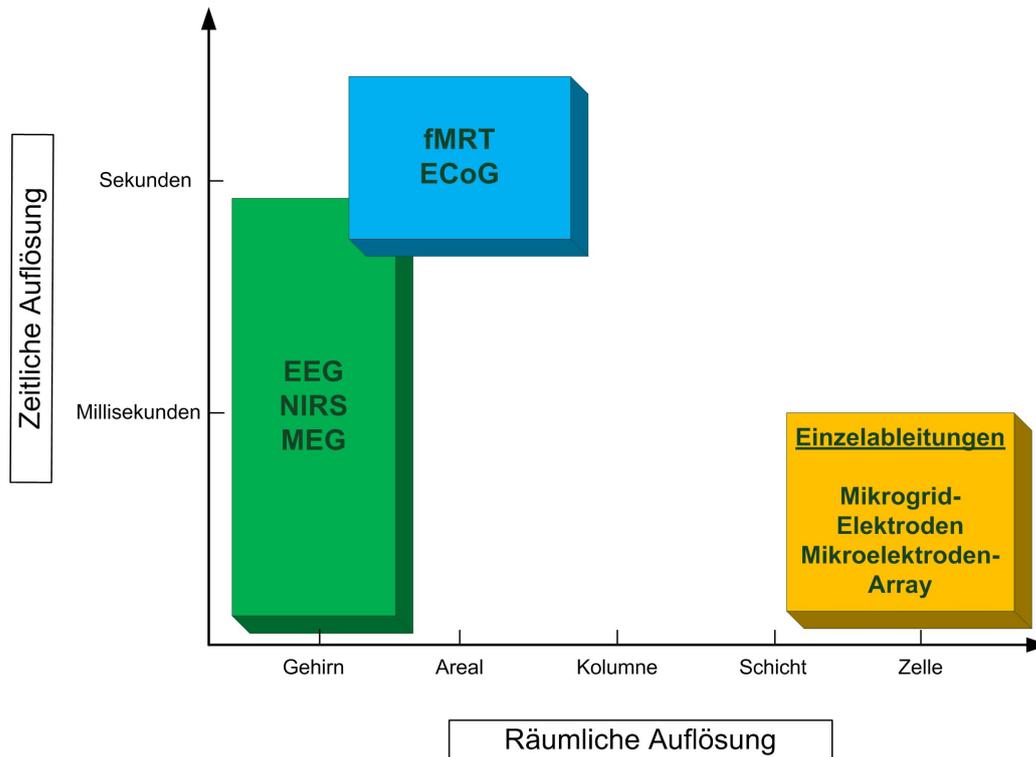


Abbildung 3.5: Verschiedenen Modalitäten können für nichtinvasive und invasive Hirn-Computer-Schnittstellen eingesetzt werden. Entscheidende Besonderheiten finden sich in Bezug auf die räumliche- und zeitliche Auflösung der Akquirierungstechniken. Eigener Entwurf nach (Birbaumer und Schmidt 2006).

ben und Evans 2010). Aufgrund ihrer schlechten zeitlichen Auflösung werden bildgebende Verfahren eher weniger zur Steuerung von Prothesen eingesetzt (Wolpaw et al. 2002). In der Zielstellung dieser Arbeit wurde langfristig die Realisierung einer VR-Stimulusumgebung für EEG- und fMRT-Studien festgelegt. Davon ausgehend werden im Folgenden die Besonderheiten beim Aufbau von EEG- und Echtzeit-fMRT-BCI näher erläutert.

EEG-BCI

Die Elektroenzephalographie (EEG) ist eine Modalität, welche anhand von Elektroden die Gehirnströme eines Individuums von der Kopfhaut ableitet. EEG gehört im Allgemeinen zu den klassischen Methoden der Biologischen Psychologie (Birbaumer und Schmidt 2006). Durch sogenannte Elektroden-Hauben, die eine variierende Anzahl von Elektroden besitzen, können elektrische Potentiale oder Spannungsschwankungen außerhalb des Kortex gemessen werden. Der Nachteil diese Methode besteht darin, dass eine örtliche Ungenauigkeit über den Ursprung einer elektrischen Aktivität im Gehirn existiert (Birbaumer und Schmidt 2006). EEG-Signale sind komplexe Signale und können anhand ihrer Frequenzanteile unterschieden werden. Insgesamt sind sechs verschiedene Frequenzbänder oder Rhythmen in der Literatur erwähnt. Für ein-

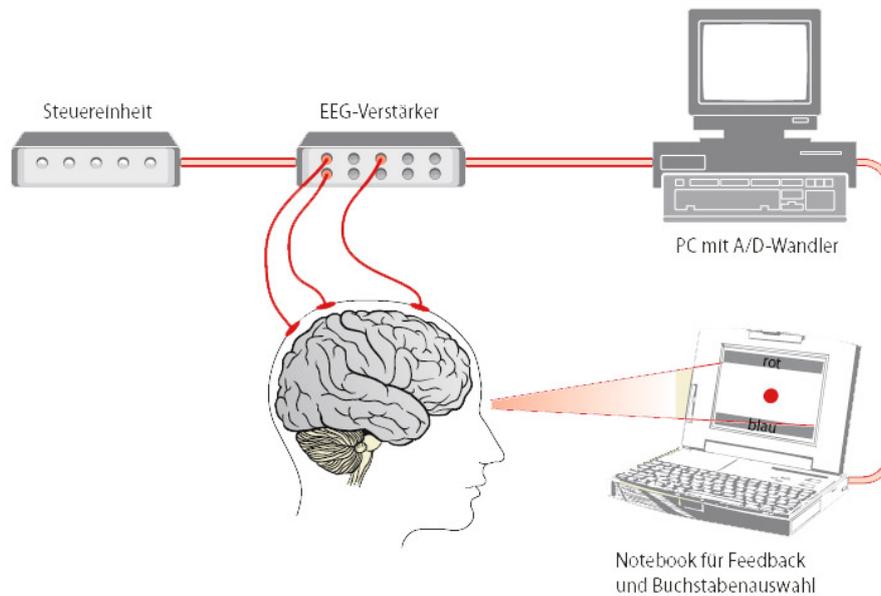


Abbildung 3.6: Der schematische Aufbau des *Thought Translation Device* (Hinterberger et al. 2004). EEG-Signale werden zunächst von der Kopfhaut abgeleitet und aufgrund der geringen Stromstärke verstärkt. Anschließend werden die analogen Signale in digitale umgewandelt und zur Datenauswertung sowie zur Ergebnispräsentation weitergeleitet.

gehende Informationen zu den einzelnen Frequenzbändern des EEG sei hier auf die allgemeine Fachliteratur verwiesen (Birbaumer und Schmidt 2006, Ehlers und Holder 2007). Speziell für EEG-BCI werden vorwiegend das Alpha-Band, Frequenzen von 8 bis 13 Hertz, sowie das Beta-Band, Frequenzen von 13 bis 30 Hertz, verwendet (Rief und Birbaumer 2006, Sanei und Chambers 2007). Dabei steht der sinusförmige Alpha-Rhythmus für einen Wachzustand mit geringer visueller Aufmerksamkeit. Der höherfrequente Beta-Rhythmus steht hingegen für Phasen visueller Konzentration oder Aufmerksamkeit und kann, abhängig vom Schwierigkeitsgrad der Aufgabe, im EEG-Signal beobachtet werden (Sanei und Chambers 2007).

EEG-BCI beruhen weitestgehend auf der Akquirierung von sogenannten langsamen, kortikalen Potentialen (engl. slow cortical potentials) (Hinterberger et al. 2003). *Slow cortical potentials* (SCP) sind sehr langsam generierte EEG-Signale, welche in etwa 0,5 bis 10 Sekunden andauern. Dabei sind negative SCP meist mit einer Bewegungsabsicht, positive SCP vorwiegend mit reduzierter kortikaler Aktivierung verknüpft (Elbert und Rockstroh 1987, Sanei und Chambers 2007). Die Detektion von SCP-Signalen ist die Grundlage einer einfachen Form der Informationsübertragung und wird oftmals im Rahmen von EEG-BCI zur Steuerung von externen Peripheriegeräten benutzt. Als Beispiel kann hier das EEG-BCI „Thought Translation Device“ genannt werden (Hinterberger et al. 2004), welches u.a. zur Kommunikation mit ALS-Patienten (Amyotropher Lateralsklerose) genutzt wird (Abb. 3.6). Neben der Akquirierung der Signale findet auch eine Signalverstärkung und eine Signalverarbeitung statt. Da EEG-Signale sich in der Größenordnung von 5 bis 100 Mikrovolt (μV) bewegen, muss ein empfindlicher Messverstärker verwendet werden. Zusätzlich dazu wandelt ein *A/D Wandler*

das analoge EEG-Signal in digitale Dateninformationen um. Erst durch diese Signaltransformation wird die spätere Analyse des EEG-Signals sowie die Verarbeitung der Messdaten durch spezielle Algorithmen ermöglicht (Sanei und Chambers 2007). Zur Verarbeitung des EEG-Signals werden zum einen Filteralgorithmen zur Reduktion von Artefakten und zum anderen Auswertalgorithmen zur Zerlegung des Signals in seine Frequenzanteile (Fouriertransformation) verwendet. Anschließend wird das verarbeitete Signal an eine Output-Komponente, z.B. ein Computerbildschirm oder eine künstliche Prothese, weitergeleitet und zur Interpretation von spezifischen Steuerbefehlen verwendet (Sanei und Chambers 2007, Coben und Evans 2010). Ein BCI-Training zur Steuerung einer Output-Komponente ist gerade für Neurofeedback-Studien unerlässlich. Im Fall des „Thought Translation Device“ konnte so nach mehrmaligem Training eine Trefferquote von bis zu 75 Prozent erreicht werden. Dabei erfolgte die Übermittlung und Transformation der EEG-Signale durchschnittlich in circa vier Sekunden (Birbaumer et al. 1999). Durch den Einsatz von optimierten Auswerteverfahren und leistungsstärkerer Rechentechnik können aktuelle EEG-BCI-Anwendungen diese Signale bereits in Echtzeit auswerten (BCI Competition 2011)

BOLD-fMRT-BCI

Mit der Entwicklung schneller Datenauswertetechniken und der Einführung der Echtzeit-fMRT (Cox et al. 1995) ermöglichte das Neurofeedback den Aufbau und die Anwendung von sogenannten Echtzeit-fMRT-BCI-Systemen, welche Signale aus tiefer liegenden Hirnarealen verarbeiten konnten. Aufgrund der physiologischen Restriktionen des BOLD-Effektes mussten jedoch stärkere Verzögerungen von sechs bis acht Sekunden bei der Auswertung von fMRT Bilddaten berücksichtigt werden (Weiskopf et al. 2007).

Trotzdem war die Möglichkeit, tiefer liegende Areale nichtinvasiv zu trainieren, ein Umstand, der gegenüber EEG-BCI-Systemen einen entscheidenden Vorteil versprach. Als Input-Komponente wird beim BOLD-fMRT-BCI ein Magnetresonanztomograph verwendet (Abb. 3.7). Dieser akquiriert durch die Anwendung einer T_2^* -gewichteten EPI-BOLD Messsequenz fMRT-Bilddaten, die auf die Erfassung der physiologischen Parameter des BOLD-Effektes ausgerichtet sind. Noch während der Messzeit werden diese fMRT-Bilddaten über Netzwerkkommunikationsprotokolle an die Echtzeit-Datenauswertung versendet. Hier erfolgt die eigentliche Anwendung der Auswertalgorithmen (Weiskopf et al. 2007, Coben und Evans 2010) (vgl. Abschnitt 2.3). Alle notwendigen Auswertinformationen werden dann zu einer Präsentationapplikation übermittelt, welche vom ursprünglichen Aufbau ausgehend, die eigentliche Output-Komponente darstellt. Das Neurofeedback-Signal zur Selbstregulation wird im Normalfall via Beamer an den Probanden im MRT übertragen.

3.8 Anwendungsformen von BCI-Systemen

Aufgrund der frühen Entwicklung von echtzeitnahen Signalauswertestrategien zählen EEG-BCI-Systeme zu den häufigsten und bekanntesten Neurofeedback-Anwendungen

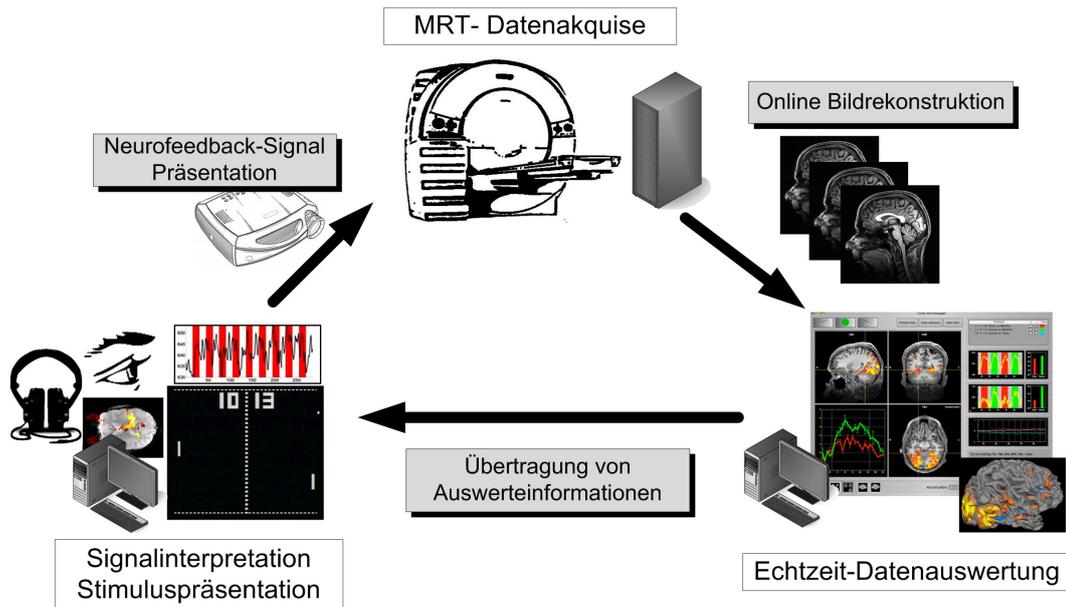


Abbildung 3.7: Der Aufbau eines BOLD-fMRT-BCI orientiert sich am Aufbau anderer BCI-Typen. Die akquirierten fMRT-Bilddaten werden zur Echtzeit-Datenauswertungssoftware weitergeleitet. Diverse Kommunikationsprotokolle übertragen anschließend die Auswertinformationen zur Anzeige des Neurofeedback-Signals an den Probanden. Eigener Entwurf nach (Coben und Evans 2010).

(Vidal 1977). Besondere Popularität erlangte diese Methodik im Jahre 1999 durch die Vorstellung des „Thought Translation Device“ (Birbaumer et al. 1999). Die Ergebnisse aus dieser Studie führten zu einer stetig ansteigenden Zahl von Veröffentlichungen auf dem Gebiet des Neurofeedbacks (Abb. 3.8).

Des Weiteren zeigte sich relativ früh, dass eine Einbeziehung von informationstechnischen und mathematischen Methoden zur Mustererkennung und Klassifikation einen entscheidenden Vorteil für Hirn-Computer-Schnittstellen birgt (Masic und Pfurtscheller 1993, Guger et al. 2000). Auch die Anwendung von Virtuellen Realitäten in EEG-BCI-Studien wurde frühzeitig untersucht (Othmer und Kaiser 2000, Bischof und Boulangier 2003). Jedoch wurde der Begriff „Virtual Reality“ mehr als „Modewort“ für sämtliche Formen von einfachen computerbasierten Präsentationsmitteln verwendet. So auch bei einfachen zweidimensionalen Computerspielen, wo eine immersive Erfahrung aus heutiger medienpsychologischer Sicht nicht gegeben war (vgl. Abschnitt 3.5). Dennoch konnten durch die kombinierte Anwendung von BCI-Anwendungen, virtuellen Umgebungen und Methoden aus dem Bereich des maschinellen Lernens erste Erkenntnisse auf diesem Gebiet gewonnen werden. Dieser Forschungszweig gewann weiter an Popularität durch die jährliche Austragung einer sogenannten „BCI Competition“ im Bereich der EEG- und ECoG-Modalitäten, bei der die neuesten Methoden zur Analyse und Interpretation von neuronalen Aktivitäten vorgestellt wurden (Wolpaw et al. 2002, Blanchard und Blankertz 2004, Blankertz et al. 2004, BCI Competition 2011).

Neben den reinen EEG-BCI-Anwendungen wurden auch Neurofeedback-Methoden

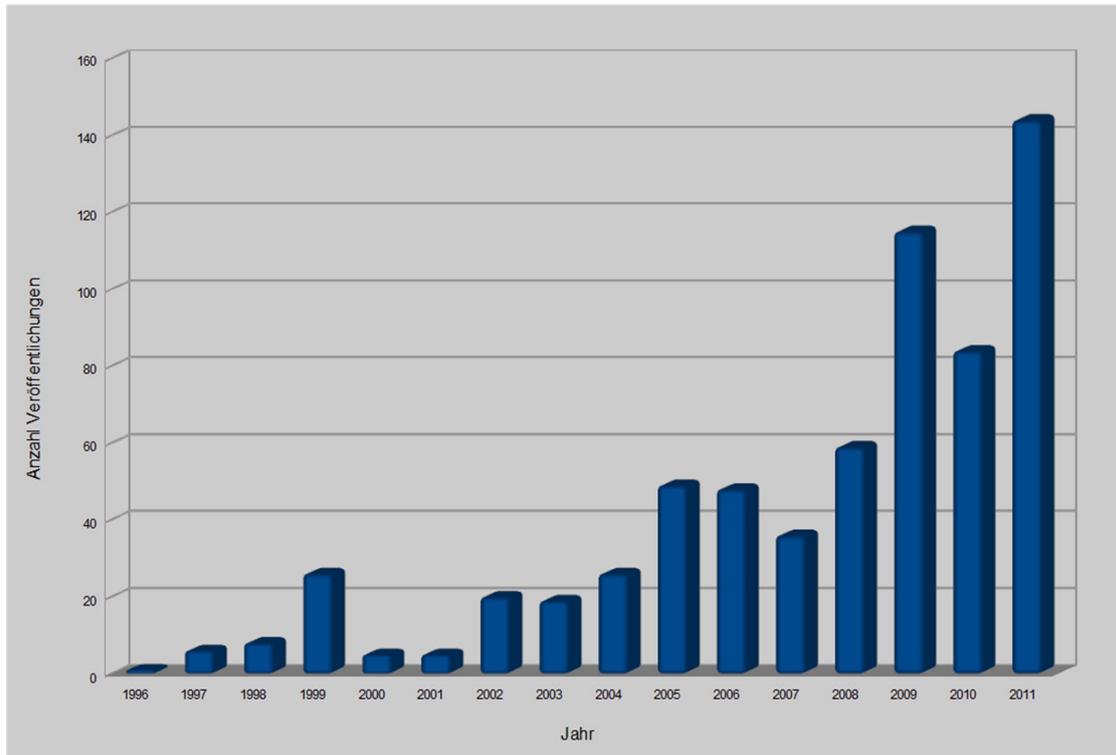


Abbildung 3.8: Das Diagramm verdeutlicht die Aktualität des Forschungsthemas *Neurofeedback* und den Anstieg der Neurofeedback-Veröffentlichungen in den Jahren von 1999 bis 2011. Quelle: www.sciencedirect.com

entwickelt, welche die Nachteile EEG-basierter Hirn-Computer-Schnittstellen ausgleichen sollten. Im Mittelpunkt stand hier die Kombination von EEG-Methoden und bildgebenden Verfahren (Hinterberger et al. 2003, Coben und Evans 2010). Diese neue Methodik verspricht großes Potential, ist aber aufgrund der komplexen Kombination beider Modalitäten, sowie deren begleitender Verstärkung von Artefakten, nicht leicht zu realisieren (Hinterberger et al. 2004).

Auch wenn die Verwendung von EEG-BCI-Systemen eine bessere zeitliche Auflösung der Messdaten ermöglicht, so birgt die Verwendung von BOLD-fMRT-BCI große Vorteile bei der Analyse von Hirnaktivierungen aus tiefer liegenden Arealen des Gehirns (Schneider und Fink 2007, deCharms 2008).

Darüber hinaus eröffnet die Verwendung von Echtzeit-fMRT-BCI-Anwendungen weitere Anwendungsgebiete. Mit der Abschätzung und Analyse der Hirnaktivität von Probanden, noch vor der eigentlichen fMRT-Messung, konnte die Effektivität von Probanden und Paradigmen bestimmt, experimentelle Fehler frühzeitig ausgeschlossen sowie etwaige Bewegungsartefakte reduziert werden (Voyvodic 1999). Dies verspricht eine effektive Durchführung von Experimenten und nachfolgenden Auswerteprozessen. Das Hauptanwendungsgebiet bleibt aber weiterhin die Analyse, Interpretation und Präsentation des Neurofeedback-Signals (Weiskopf et al. 2007). Um den Zugang zu dieser Technik nachhaltig zu vereinfachen und die Methoden weiter zu optimieren, wurden diverse Echtzeit-fMRT-Datenauswertealgorithmen und -applikationen entworfen (Cox

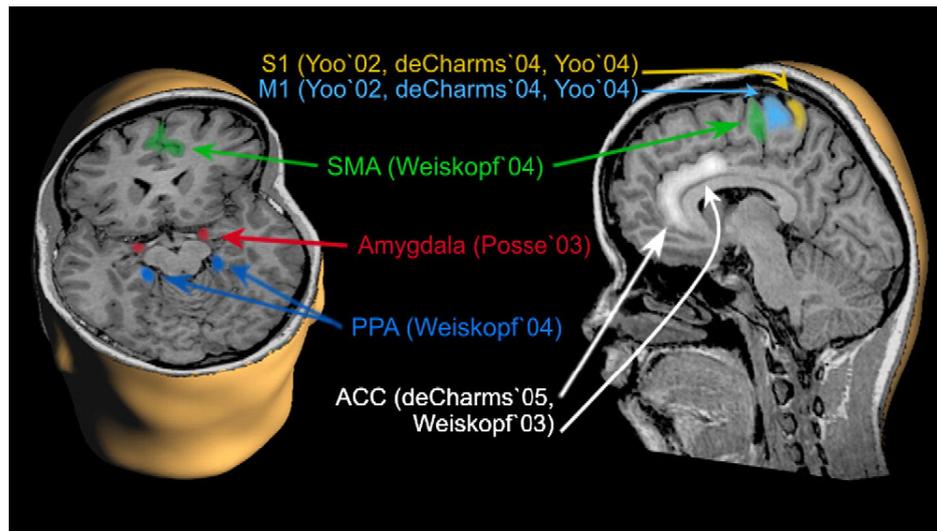


Abbildung 3.9: Verschiedene Neurofeedback-Studien wurden im Bereich der Echtzeit-fMRT durchgeführt. Die häufigsten Studien beziehen sich auf Hirnareale, wie z.B. den primären motorischen Kortex, den somatosensorischen Kortex, Regionen im parahippocampalen Gyrus sowie Areale im anterioren Cingulum (Weiskopf et al. 2004b).

et al. 1995, Gembris et al. 2000, Brain Innovation 2002-2006, Hollmann et al. 2008). Diese Auswerteapplikationen wurden vor allem in Neurofeedback-Experimenten zur Selbstregulierung von Hirnaktivitäten aus tiefer liegende Hirnareale verwendet (Abb. 3.9). So konnte zum Beispiel die Hirnaktivierung aus der Amygdala während des Zeigens von traurigen und neutralen Gesichtern in Echtzeit analysiert und zum Training von verhaltenspsychologischen Aspekten der teilnehmenden Probanden verwendet werden (Posse et al. 2003). Eine weitere Anwendung zielte auf die Verringerung von Schmerzintensitäten ab. So wurde die gemittelte, neuronale Aktivierung aus dem rechten anterioren Cingulären Cortex in Form einer virtuellen Flamme an die Probanden rückübermittelt (deCharms et al. 2005). Die Flamme diente dabei als direkter Indikator für das neuronale Signal und damit für das probandenspezifische Schmerzempfinden, welches möglichst gering gehalten werden sollte. Ein weiteres Beispiel für die Selbstregulation im anterioren Cingulären Cortex mittels Echtzeit-fMRT wurde bereits im Jahr 2004 veröffentlicht und zeigt die Bedeutung für die Erforschung verhaltenspsychologischer, kognitiver und emotionaler Zustände (Weiskopf et al. 2004b).

Auch die Erforschung emotionaler Zustände, ausgelöst durch soziale Interaktionen, wurde mittels Neurofeedback-Studien erforscht. So konnten durch die Einbeziehung eines neuroökonomischen Szenarios und der Echtzeit-fMRT-Analyse von Hirnaktivierungen eine gute Vorhersage zu probandenspezifischen Entscheidungen getroffen werden (Hollmann et al. 2009). In einer weiteren Studie konnten zwei Probanden in zwei verschiedenen MRT mittels ihrer Hirnaktivierungen und auf Basis eines Computerspiel gegeneinander und miteinander agieren. Die innerhalb dieses Kooperations- und Wettbewerbsprozesses aufgezeichneten Hirnaktivierungen konnten sicher registriert werden und lieferten wichtige Informationen zu sozialkognitiven Vorgängen im menschlichen Gehirn (Strobel et al. 2008). Besonders prosoziale Vorgänge, wie z.B.

Altruismus, oder negative Aspekte, wie z.B. soziale Manipulationen oder Täuschungen, spielten hier eine entscheidende Rolle.

Die Untersuchung von räumlichen Navigationsprozessen oder Prozessen zum räumlichen und episodischen Gedächtnis sind weitere Anwendungsformen im Bereich des Neurofeedbacks. So konnten beispielsweise mehrere Probanden anhand ihrer Hirnaktivierungen aus verschiedenen Arealen durch ein zweidimensionales, virtuelles Labyrinth navigieren (Yoo et al. 2004). Ausschlaggebend für das Gelingen dieses Versuchs war die Einbeziehung von Methoden aus dem Bereich des Maschinellen Lernens, welche die Aktivierungsunterschiede über die einzelnen Hirnareale richtig klassifizierten und so die Steuerung des Avatars ermöglichten. Aktuelle Studien bestätigen zudem, dass immer komplexere Virtuelle Umgebungen als Stimuli für neurokognitive Studien eingesetzt werden und dass der Bedarf an entsprechenden virtuellen Szenarios ungebrochen ist (Sanchez-Vives und Slater 2005, Maguire et al. 2006, Hassabis et al. 2009, Doeller et al. 2010, Mueller et al. 2012a).

Kapitel 4

Informationstheoretische Grundlagen

In diesem Kapitel soll zunächst der Begriff Virtuelle Realität (VR) im Kontext der informationstechnischen sowie der neurowissenschaftlichen Forschung vorgestellt und erörtert werden. Anschließend folgen die Grundlagen der Modellierung sowie der Klassifikation von Virtuellen Umgebungen (VU). Weitere Unterkapitel erörtern zudem die allgemeinen Anforderungen an die Mensch-Maschine Interaktion sowie die Methoden der Kollisionsdetektion und des physikalisch-basierten Renderns. Das Abstimmen von Echtzeit-Renderprozessen stellt zudem einen wichtigen Punkt bei der Realisierung des implementierten Echtzeit-VR-Adaptionsalgorithmus dar und wird zum Abschluss dieses Kapitels ausführlich vorgestellt.

4.1 Grundlagen Virtueller Umgebungen

Virtuelle Umgebungen können im weitesten Sinne als eine Form der Prozesssimulation aufgefasst werden. Sämtliche grundlegenden Entwicklungsprozesse beruhen auf einer sogenannten Modellbildung. In deren Rahmen wird zunächst ein reales bzw. theoretisches System abstrahiert und anschließend innerhalb einer generierten Umgebung simuliert (Kim 2005, Brill 2009). Um eine reale Umgebung durch Simulationsprozesse in einer Virtuellen Umgebung authentisch abzubilden, muss zunächst ein möglichst vollständiges Modell der zu lösenden Problemstellung gefunden sowie eine zuverlässige und benutzerorientierte Umsetzung mit geeigneten Mitteln realisiert werden (Burdea und Coiffet 2003, Kim 2005, Craig et al. 2009). Letzteres wurden in der Vergangenheit zumeist nur in eingeschränkter Form an die Bedürfnisse des Anwenders angepasst, was teilweise zu einer bis heute vorherrschenden Skepsis bzw. Ablehnung von Virtuellen Umgebungen im medizinisch-neurowissenschaftlichen Anwendungsbereich führte.

4.1.1 Definition und Geschichte

Virtuelle Realität (VR) hat sich im Laufe der Jahre zu einer interdisziplinären Technologie entwickelt, die mittlerweile in der Informatik, in der Medienpsychologie, in der Medizin und in den Neurowissenschaften sowie in der Prozess- und Maschinen-

entwicklung ihren Einsatz findet (Burdea und Coiffet 2003, Craig et al. 2009). Trotz dieser vielen unterschiedlichen Anwendungsbereiche gibt es doch überschneidende Charakteristiken, die zusammengefasst den Begriff der Virtuellen Realität allgemein beschreiben können.

- *Definition:* Eine Virtuelle Realität beschreibt eine computergenerierte, synthetische Umgebung, welche künstliche Erfahrungen an ihre Nutzer vermittelt. Sie ist ebenso eine Benutzer-Computer Schnittstelle, welche mit Hilfe von Echtzeit-Simulation und Interaktion visuelle, auditive, taktile, olfaktorische und haptische Reize an ihre Anwender übermittelt (Sherman und Craig 2003, Kim 2005, Brill 2009).

Andere Quellen beschreiben Virtuelle Realität als Simulation, welche echte Realitäten mit künstlichen Realitäten verschmelzen lässt (Codella et al. 1992).

Im Idealfall soll so die voraussetzungslose Nutzung einer Benutzerschnittstelle ermöglicht werden, welche Anwendern die Visualisierung, die Manipulation sowie die Interaktion mit komplexen Datenmengen gestattet (Brill 2009). In diesem Zusammenhang wird zudem zwischen Virtueller Realität und erweiterter Realität (*augmented reality*) unterschieden. Bezieht sich die Virtuelle Realität hauptsächlich auf das komplette Eintauchen eines Benutzers in die künstliche Realität, so setzt die erweiterte Realität auf die zusätzliche Einblendung von Metainformationen und eine mögliche Ausweitung bzw. Ergänzung von Sinneswahrnehmungen durch entsprechende Sensortechnik, z.B. Infrarotsensoren und Distanzbildeinblendungen (Kim 2005, Craig et al. 2009).

Über die Entstehung des Begriffs Virtuelle Realität existieren genauso viele Ausführungen wie Definitionsversuche. Der informationstechnische Begriff „Virtuelle Realität“ wird dem Informatiker und Autor Jaron Larnier zugeschrieben. Er prägte den Begriff in einem 1986 entstandenen Gespräch über die Arbeit von Scott Fisher (Carlson 2003), welcher erstmalig 1985 und 1986 den Begriff des „Virtual Environments“ (Virtuelle Umgebungen) prägte (Fisher 1985; 1986). Der eigentliche Ursprung des Begriffes geht aber auf den polnischen Schriftsteller Stanislaw Lem zurück. Er beschrieb schon in seinem 1964 erstmals veröffentlichten Grundlagenwerk der Futurologie „Summa technologiae“ den Begriff der „Phantomatik“, welcher gleichbedeutend mit der heutigen Virtuellen Realität verstanden wird (Lem 1981). Für weitere Präzisionen des Begriffes sorgte zum einen Myron Krueger mit seinem Werk „Artificial Reality“ (Krueger 1983) sowie zum anderen der oft zitierte William Gibson mit seinem Cyberpunk-Roman „Neuromancer“, welcher mit dem eingeführten Begriff „Cyberspace“ eine Art weltumspannende Mensch-Computer-Schnittstelle beschreibt und damit eine abstrakte Definitionsgrundlage für heutige Mensch-Maschine-Interaktionen legte. So ist die Modellbildung und die technische Realisierung einer VU auch heute noch die technische Grundlage für die Simulation und das Eintauchen eines Benutzers in eine künstliche Realität (Fisher 1986).

4.1.2 Modellierung und Klassifikation Virtueller Umgebungen

Die Grundvoraussetzung zur Generierung einer VU stellt die Modellbildung zur Abstraktion und Abbildung von realen Prozessen in einer computersimulierten Umge-

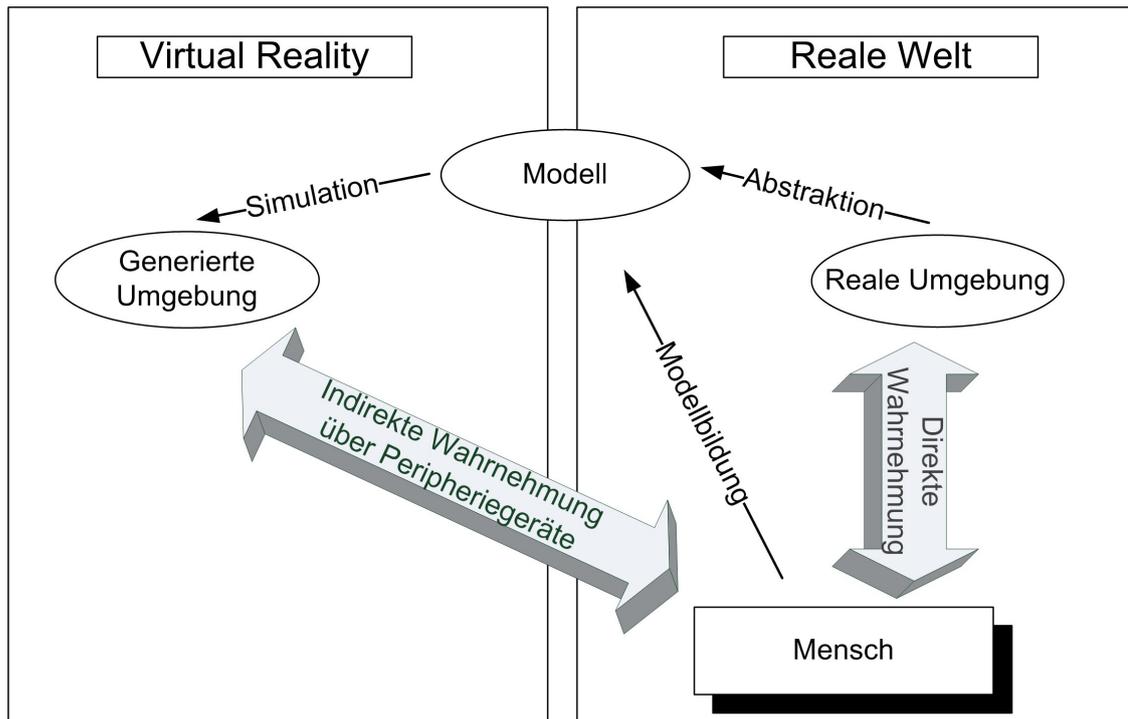


Abbildung 4.1: Die Grundlage jeder VU-Erstellung stellt die sogenannte Modellbildung und das damit verbundene Authoring dar. Es erlaubt die Abbildung und Abstraktion realer Prozesse innerhalb einer künstlich generierten Umgebung. Eigener Entwurf nach (Brill 2009).

bung dar (Abb. 4.1). Dazu werden im Rahmen des sogenannten *Authoring* sämtliche Anforderungen und Restriktionen für die Modellbildung zusammengetragen und durch ein Abbildungsmodell in den Prozess integriert (Zachmann 2000, Stanney 2002, Kim 2005). Hier müssen vor allem zwei wesentliche Dinge berücksichtigt und spezifiziert werden (Stanney 2002):

- die *Struktur* einer Virtuellen Umgebung, welche durch ihre dahinterliegende Geometrie, durch ihren Szenengraph und durch die verwendeten Materialien inklusive dem zugehörigen Beleuchtungsmodell repräsentiert wird,
- die Definition von Datenmodellen und Regeln für Objekte, künstliche Charaktere und die Benutzerschnittstelle.

Vorhandene VU lassen sich über verschiedenste Ansätze klassifizieren. Der am weitesten verbreitete Ansatz ist die Klassifizierung anhand des Realitätsgrades. Dabei werden VU zum einen in die Simulation eines realen Systems und zum anderen in die Simulation eines fiktiven oder theoretischen Systems eingeteilt (Craig et al. 2009).

Des Weiteren ergeben sich weitere Abstufungen. Simulationen von realen Systemen werden so in nicht mehr existente, existente und noch nicht existente Systeme unterteilt (Abb. 4.2). Wo hingegen Simulationen eines theoretischen Systems durch die Klassifizierung in unrealistische/fiktive Umgebungen und Informationsvisualisierungen definiert sind (Stanney 2002, Sherman und Craig 2003). Verschiedene Werkzeug-

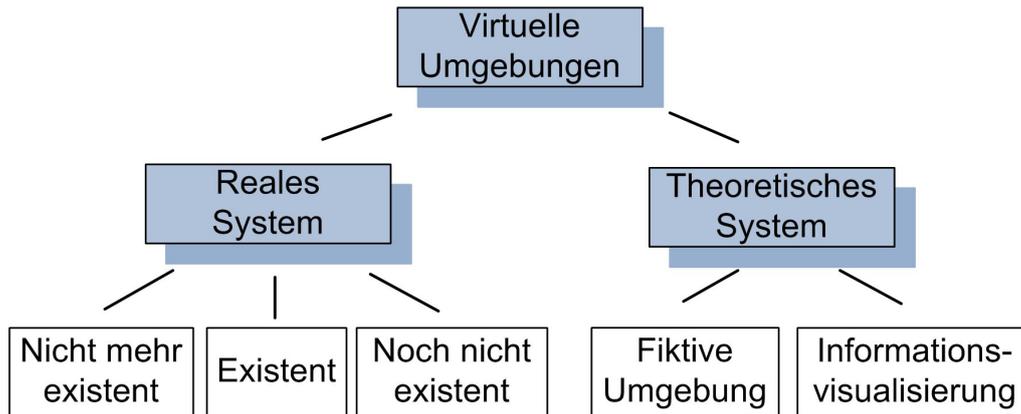


Abbildung 4.2: Der häufigste zur Klassifikation von Virtuellen Umgebungen angewendete Ansatz basiert auf der Einteilung anhand des Realitätsgrades. Eigener Entwurf nach (Zachmann 2000).

ge zur VU-Modellierung wurden bereits entwickelt, dennoch sind diese Applikationen aufgrund ihrer informationstechnischen Restriktionen für die Neurowissenschaften nur bedingt einsetzbar (Adamovich et al. 2009, Mellet et al. 2010). Die Ursache ist die bereits genannte, hohe Komplexität des Modellierungsprozesses, welche auch mit Einführung der Skript-Sprachen nur bedingt vereinfacht wurde. So existiert keine Applikation, welche eine komplette und vollautomatische Erstellung von VU unterstützt, den etwaigen immersiven Charakter vorhandener Realitäten kopiert und die damit verbundene menschlich-kreative Komponente der VR-Modellierung vollständig ersetzt (Mueller et al. 2012b).

Wohldefinierte generische Benutzer-Objekt- und Objekt-Objekt-Interaktionsprozesse stellen zumeist die Mehrzahl der Prozesse innerhalb einer VU dar und erlauben durch die Syntax ihrer Eingabeobjekte eine Charakterisierung (Stanney 2002). Aufgrund dessen werden VU prinzipiell in zwei teilweise komplementäre Ansätze unterteilt:

- Ereignisbasierte VU: die ereignisbasierte VU ist vornehmlich durch die vom Benutzer ausgelösten Ereignisse geprägt. Dies kann das Betreten vordefinierter Landmarken, das Auslösen bestimmter Aktionen oder die Betätigung von Peripheriegeräten innerhalb einer gesetzten Zeitspanne sein (Zachmann 2000).
- Verhaltensbasierte VU: die verhaltensbasierte VU orientiert sich am Verhalten des Benutzers. Dazu werden meist regelbasierte, autonome Objekte oder Entitäten erstellt, welche durch ihre Rezeptoren das Verhalten des Benutzers aufzeichnen und darauf reagieren (Blumberg und Galyean 1995).

Zukünftige Ansätze zielen darauf ab, diese beiden Präsentationstypen zu kombinieren, um so eine möglichst authentische VU zu simulieren (Brill 2009).

Objekte und Semantik

Entitäten oder Objekte sind ein Grundbaustein sämtlicher VU. Eine Entität wird dabei als ein eindeutig bestimmtes Objekt definiert, welches innerhalb der VU entweder

einen materiellen oder immateriellen Zustand annehmen kann (Sutcliffe 2003, Kim 2005, Craig et al. 2009). Zudem besitzt es eindeutige Zustandsinformationen, die diesen Funktionen innerhalb der VU widerspiegeln.

Der funktionelle Zustand einer Entität wird durch sogenannte semantische Attribute definiert. Diese Attribute beschreiben den aktuellen funktionellen Zustand bzw. die derzeitige Rolle eines Objekts innerhalb der VU mit einem zugehörigen Zustandswert (Sutcliffe 2003). Zustandsattribute werden meist in einer sogenannten Objekt-Liste zusammengefasst. Objekt-Zustände werden durch unterteilte Objekt-Listen oder Gruppierungen differenziert. Die Unterteilung eines Hauptattributs in mehrere Unterattribute ermöglicht die Implementierung von komplexeren aber auch effizienteren Algorithmen, wie z.B. den koordinierten Schreibzugriff mehrerer Threads auf ein Objekt durch das Mutex-Verfahren (Zachmann 2000). Hierarchische Objekt-Listen oder auch einzelne Objekte werden in VR-Szenarios innerhalb eines sogenannten *Szenengraph* zusammengefasst.

Zeitliche Steuerung

Viele Prozesse und Aktionen innerhalb einer VU sind zeitabhängig. Geräusche, Töne und Animationen sind entweder durch eine feste Länge definiert oder können von einem Zeitpunkt t_1 bis zum einem Zeitpunkt t_2 wiedergegeben werden (Kim 2005). Hierfür ist es notwendig, eine globale Simulationszeit und eine zeitliche Steuerung zu definieren. Die globale Simulationszeit wird in der Regel von der internen zeitlichen Repräsentation des entsprechenden Rechners bzw. der Systemzeit abgeleitet (Sutcliffe 2003).

Zur Darstellung von Zeitzuständen greift man auf sogenannte Zeitvariablen zurück. Zeitvariablen sind meist durch eine unbestimmte Menge von numerischen Werten definiert und leiten sich direkt von der regulären Systemzeit ab. Des Weiteren erlauben sie die koordinierte Ausführung von Ereignissen, Animationen und Aktionen innerhalb der generierten Umgebung (Craig et al. 2009). Für die spezielle Kombination einer VU mit der Präsentation von neurowissenschaftlichen Stimuli muss zudem ein Abgleich zwischen der Simulationszeit und der Messlaufzeit des Akquirierungssystems (MRT, EEG, etc.) erfolgen. Dies geschieht in der Regel durch ein Synchronitätssignal, welches die Startzeit des Akquirierungssystems mit der Simulationszeit der VU koordiniert (Mueller et al. 2012b).

Ereignissteuerung

Die Ereignissteuerung innerhalb einer VU wird üblicherweise von einem sogenannten Interaktionsmanager übernommen. Dieser erhält Benutzer- oder Systembefehle und setzt diese zur Steuerung von bestimmten Aktionen innerhalb der virtuellen Welt um. Ereignisse sind dabei genau zwischen den Eingaben des Benutzers und den ausgeführten Aktionen angesiedelt (Sutcliffe 2003, Kim 2005). Aktionen können in der Regel nur zwei Zustände annehmen: *Aktion ein* und *Aktion aus*. Ebenfalls denkbar ist die Ausführung der Aktion bei der Betätigung einer bestimmten Taste und das Beenden der Aktion beim Loslassen derselbigen (Zachmann 2000). Die Grundlagen dieser ereignis-

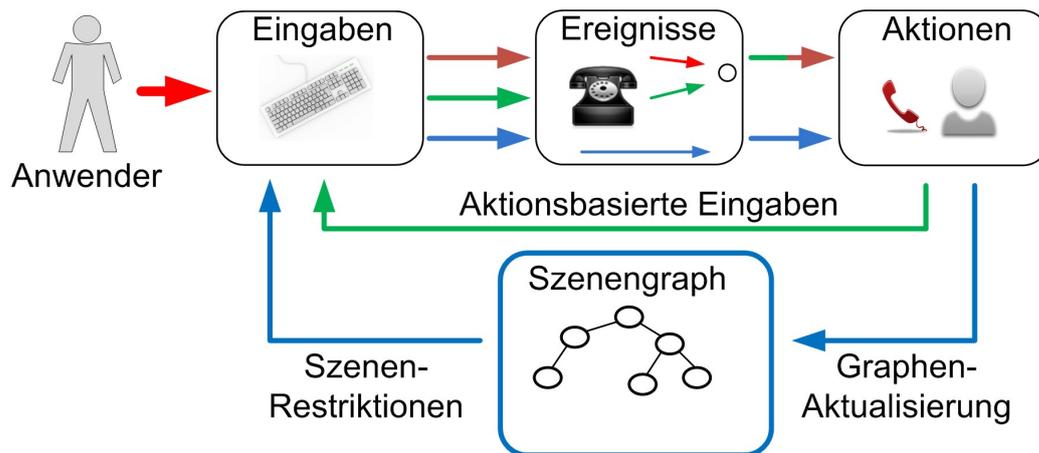


Abbildung 4.3: Das AEIO-Prinzip als Ereignis-Aktions-Paradigma definiert die Auswirkungen von Benutzereingaben auf simulierte Umgebungen. Die durch Eingaben ausgelösten Ereignisse starten wiederum Aktionen, die Änderungen an der Objektstruktur (Szenengraph) des Szenarios vornehmen. Eigener Entwurf nach (Kim 2005).

basierten Lösung werden durch das *AEIO-Paradigma* (actions, events, inputs, objects) beschrieben. Ein Anwender generiert zunächst Eingaben, die im Folgenden gezielt Ereignisse innerhalb der Virtuellen Umgebung auslösen. Diese wiederum rufen bestimmte Aktionen oder sogar benutzerspezifisches Verhalten hervor, welches zur Anpassung von Objekten innerhalb der dahinterliegenden Szenenstruktur (Szenengraph) verwendet wird (Kim 2005). Die Anpassung von einzelnen Objekten bzw. des gesamten Szenarios führt dementsprechend auch zu einer Anpassung der Szenenrestriktionen, welche anschließend in aktualisierter Form bei erneuten Benutzereingaben berücksichtigt werden (Abb. 4.3). Die Realisierung solcher Aktions-Ereignis-Paradigmen setzt die Umsetzung der folgenden Anforderungen voraus (Zachmann 2000):

- Ereignisse sind durch boolsche Ausdrücke kombinierbar.
- Verknüpfungen von unterschiedlichen Eingabeparametern soll ermöglicht werden.
- *Entity-Relationship-Modell - N:M-Beziehungstyp*: Mehrere Ereignisse können dieselbe Aktion auslösen; ein Ereignis kann mehrere gleichzeitige Aktionen auslösen; die unterschiedliche Kombination von mehreren Eingaben kann zum Auslösen des gleichen Ereignisses führen; eine Aktion kann sich auf mehrere Objekte auswirken.
- Aktionen können Benutzereingaben und nachfolgende Ereignisse beeinflussen.

Peripheriegeräte, welche Benutzereingaben registrieren und beim Einsatz von Virtuellen Realitäten verwendet werden, können oberflächlich in zwei Kategorien eingeteilt werden. Als erstes unterscheidet man hier Input-Peripheriegeräte, welche Signale vom Benutzer aufzeichnen und diese anhand vordefinierter Regeln in Aktionen umwandeln.

Die zweite Klasse umfasst Peripheriegeräte, welche als Output- oder Präsentationsgeräte eingesetzt werden. Sie geben dem Benutzer eine Rückmeldung und präsentieren die Virtuelle Welt über verschiedene Visualisierungsmodalitäten (Sherman und Craig 2003).

4.2 Mensch-Computer Interaktionen

Eine entscheidender Faktor für die allgemeine Akzeptanz einer Anwendung und das Immersionsverhalten innerhalb einer virtuellen Welt ist der „Dialog“ zwischen dem Benutzer und der Applikation (Preim und Dachsel 2010).

Für die Integration von einfachen Dialogkomponenten innerhalb einer VU existieren bereits diverse Middleware-Applikationen, sogenannte *Virtual Environment Management Systems* (VEMS) oder 3D-Grafik-Engines. Die grundlegenden Aspekte zur Modellierung von erweiterten Benutzerschnittstellen-Management-Systemen (User Interface Management System - UIMS) werden allerdings durch Modelle zur Modularisierung von UIMS bereitgestellt. Ausgehend davon wurden eine Vielzahl von Modellen zur Beschreibung von Mensch-Computer Interaktionen und 2D-Benutzerschnittstellen entworfen. Eines der bekanntesten Modelle zur Modularisierung von UIMS ist das sogenannte *Seeheim-Modell*, welches 1983 auf einem Workshop in Seeheim entwickelt wurde (Green 1984, Pfaff 1985). Es beschreibt Relationen, an denen sich nachfolgende Modelle orientiert haben und charakterisiert Benutzerschnittstellen anhand einer Präsentationskomponente, einer Dialogkontrolle und einer Applikationsschnittstelle.

4.2.1 Arch-Modell

Eine Erweiterung des Seeheim-Modells, welche in der vorliegenden Arbeit ihren Einsatz findet, wurde durch das sogenannte *Arch-Modell* eingeführt (Olsen 1992, Sutcliffe 2003, Preim und Dachsel 2010). Es kann als Erweiterung des Seeheim-Modells betrachtet werden und konzentriert sich vornehmlich auf eine abstrakte Einteilung der einzelnen Komponenten. Dabei werden die aus dem Seeheim-Modell bekannte Präsentationskomponente und das ebenfalls bekannte Applikationsinterface differenzierter betrachtet (Abb. 4.4). Die ursprüngliche Präsentationskomponente erhält dadurch eine sogenannte logische Komponente (Präsentationskomponente), welche die passenden Darstellungsmöglichkeiten auswählt, und eine physikalische Komponente (Interaktionstoolkit), welche die eigentliche Software bzw. den Quellcode der einzelnen Darstellungsobjekte beinhaltet (Tan und Nijholt 2010). Das Applikationsinterface wird in einen funktionalen Kern und einen Adapter für die Bereitstellung von zusätzlichen Programmbibliotheken zur Vereinfachung der Implementierung aufgeteilt.

- *Dialog-Komponente*: Analog zum Seeheim-Modell steuert sie die Interaktionen zwischen Anwender und UIMS. Sie kann Eingaben anfordern und Ausgaben an den Anwender senden.
- *Präsentationskomponente*: Die Präsentationskomponente ordnet den Ein- und Ausgaben die passenden Darstellungsobjekte zu.

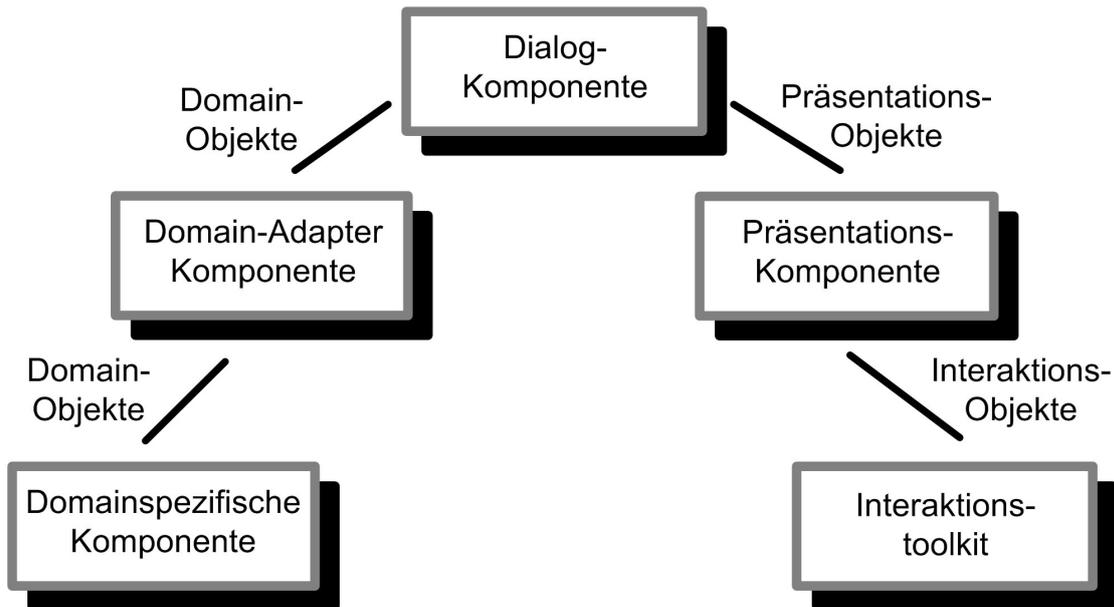


Abbildung 4.4: Das Arch-Modell für modularisierte Benutzerschnittstellen basiert in Grundzügen auf dem Seeheim-Modell und gestattet zum einen die Einbindung eines Interaktionstoolkits und zum anderen die weitere Spezifizierung von Domain- bzw. Applikationsspezifischen Komponenten. Eigener Entwurf nach (Tan und Nijholt 2010).

- *Interaktionstoolkit:* Das Interaktionstoolkit repräsentiert den Programmcode zur Darstellung von Interaktionsobjekten (Textfelder, Knöpfe, Schaltflächen, etc.).
- *Domain-Adapter-Komponente:* Sämtliche Domain-Objekte zur Kommunikation zwischen Applikationsklassen und Dialog-Komponente werden innerhalb dieser Komponente definiert.
- *Domainspezifische Komponente:* Sie symbolisiert die dahinterliegenden Applikationsklassen mit all ihren Kernfunktionen. Ein Austausch von Informationen zwischen Applikationsklassen und der Dialog-Komponente findet über die integrierten Domain-Objekte statt.

Für weitere ausführlichere Informationen zu diesem Thema sei auf die entsprechende Fachliteratur verwiesen (Sutcliffe 2003, Tan und Nijholt 2010).

4.2.2 Transitionssysteme

Die Modellierung von Benutzeroberflächenfunktionen durch Transitionssysteme ist eine weitere Methode, welche innerhalb der vorliegenden Arbeit eingesetzt wurde. Sie ermöglichte die Modellierung von Relationen durch eine festdefinierte endliche Menge von Zuständen und Zustandsübergängen (Transitionen) (Hopcroft et al. 2011). Das beschriftete Transitionssystem (Labeled Transition System) wird häufig zur Beschreibung von Zuständen innerhalb einer Benutzerschnittstelle eingesetzt und ist ein Tupel

der Form $T = (S, s_0, A, R, P, I)$ mit

S : eine endliche Menge von Zuständen

s_0 : ein Startzustand

A : eine Menge von Transitionsbeschriftungen

R : $R \subseteq S \times A \times S$ Transitionsrelation

P : eine endliche Menge von Prädikaten

I : $P \rightarrow P(S)$ die Interpretation der Prädikate an den Zuständen.

Ausgehend von dieser Definition können Transitionssysteme durch einen gerichteten Graph dargestellt werden. Die Knoten repräsentieren hier die einzelnen Zustände, und sämtliche Kanten des Transitionssystems werden mit einem Token, einer Bedingung und einer Aktion beschriftet (Hopcroft et al. 2011).

4.3 Kollisionsdetektion von Polygonobjekten

Ein wesentlicher Aspekt bei der Simulation von 3D-Szenarien in neurowissenschaftlichen Experimenten ist das Hervorrufen eines hohen Realitätsempfindens innerhalb der Virtuellen Umgebung. Um dies zu erreichen, wird ein authentisches und natürliches Objektverhalten erwartet (Sherman und Craig 2003, Craig et al. 2009).

Vor allem Kollisionsdetektionsprozesse, die im Rahmen von BCI- oder Neurofeedback-Experimenten arbeiten, müssen optimiert und auf die zeitkritischen Echtzeit-Datenanalyseprozesse abgestimmt werden (Mueller et al. 2012b). Die Grundlage hierfür bildet die Implementierung von zusätzlichen Programmfunktionen innerhalb der Kollisionsdetektionsschleife. Diese sollen die korrekte Abarbeitung der Kollisionsdetektionsprozesse überwachen und auf die zeitkritische Stimuluspräsentation in Echtzeit-fMRT-Experimenten abstimmen. Die zentrale Komponente ist das *Kollisionsmanagementsystem*, welches in den Simulationsprozess einer Anwendung eingebunden ist (Abb. 4.5) (Eckstein 1999, Zachmann 2000). Die erste Komponente dieses Systems ist die *Kollisionsprüfung*, welche für die sogenannte *ausgedehnte Phase* (Broad Phase) verantwortlich ist (Abb. 4.5). Sie ist für die Registrierung von Kollisionen verantwortlich und analysiert weitflächig, ob Objekte die Grenzen eines anderen Objekts verletzen. Werden keine Kollisionen erkannt, stoppt das Kollisionsmanagementsystem weitere Prüfungen und springt zum eigentlichen Simulationsprozess zurück (Eckstein 1999). Wurden überlappende Objekte entdeckt, erfolgt der Aufruf der *Markierungsphase*. Hier werden durch das Setzen von Statusvariablen die entsprechenden Objekte gekennzeichnet. Die letzte Komponente ist die *Kollisionsverarbeitung*. Hier erfolgt die Anwendung von rechenintensiven Approximationsverfahren, welche durch Überlappungstests die entsprechenden Kollisionspunkte bestimmen und die anschließende visuelle Anpassung der Objekte ermöglichen (Eckstein 1999).

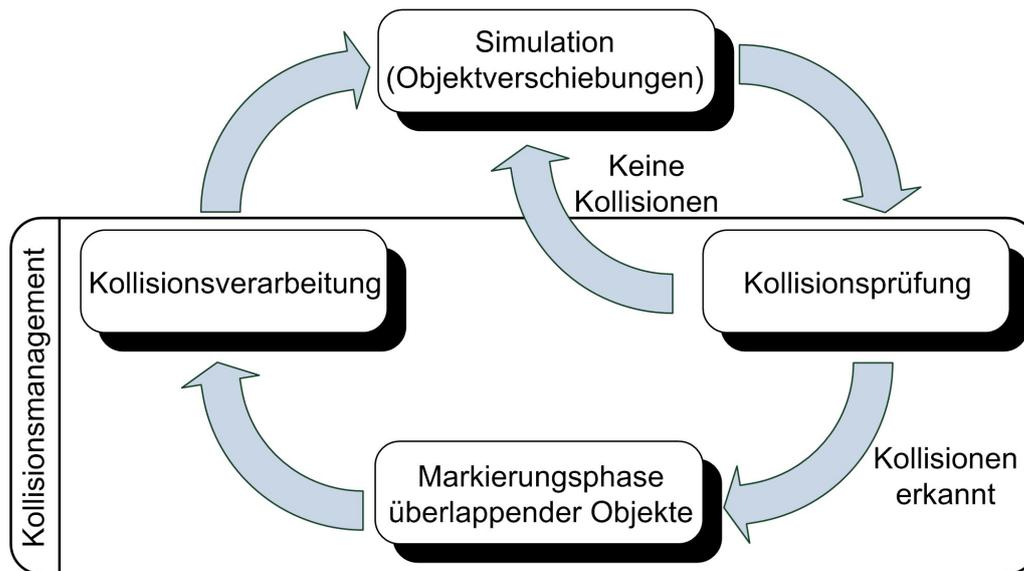


Abbildung 4.5: Die Algorithmen des Kollisionsmanagements sind regulär innerhalb des übergeordneten Simulationsprozesses eingebunden und werden sequenziell bei der Abarbeitung der einzelnen Routinen aufgerufen. Eigener Entwurf nach (Goodman und O'Rourke 2004).

4.3.1 Methoden zur Kollisionsverarbeitung

Da im Speziellen die Kollisionsverarbeitung einen sehr aufwendigen Prozess darstellt, empfiehlt sich die Verwendung von schnellen Algorithmen zur Erkennung von Objektkollisionen. Häufig eingesetzte Standardverfahren sind die sogenannten *minimal umgebenden Volumina* (MUV) (Eckstein 1999, Farin und Hansford 2003). Nach dem Prinzip der Vereinfachung wird zunächst das eigentliche komplexe dreidimensionale Objekt durch einen einfachen geometrischen Körper umschlossen und erlaubt so eine vereinfachte Registrierung möglicher Kollisionen. Im mathematischen Sinne stellen die MUV das kleinste d-dimensionale Rechteck oder den kleinsten Quader eines Objekts dar, in den das d-dimensionale Objekt hineinpasst (Mezger 2001).

Bounding-Box-Verfahren

Die einfachste Form der minimal umgebenden Volumina ist das *minimal umgebende Rechteck* oder *Bounding Box-Verfahren*. Das Grundprinzip basiert auf der Prüfung von Überschneidungen zwischen einem Polyeder R und einem Polyeder S und umgekehrt (Goodman und O'Rourke 2004, Preim und Bartz 2007). Ausgehend von dieser an dem Brute-Force-Algorithmus angelehnten Methodik definiert sich hieraus die Basis sämtlicher Kollisionsdetektionsalgorithmen (Zachmann 2000, Meseth 2006).

Für die allgemeinen Kollisionsbedingungen von MUV, welche sowohl für Bounding-Box-Verfahren (BBV) als auch für konvexe Polytopen zutreffen, gilt folgende Definition des Abstands d zweier Polyeder R und S (Ericson 2005):

$$\begin{aligned} d(MUV(R), MUV(S)) &\leq d(MUV(R), S) \\ &\leq d(R, S), \end{aligned} \tag{4.1}$$

und somit:

$$d(R, S) = 0 \quad \Rightarrow \quad d(MUV(R), MUV(S)) = 0. \quad (4.2)$$

Konvexe Polytope

Die Einführung von Verfahren für minimal umgebenden Volumina gestattete die Einbindung von effizienteren Methoden zur Registrierung von komplexen Objektkollisionen im dreidimensionalen Raum. Konvexe Polytope oder *Discrete Oriented Polytopes* (DOP) basieren auf dem Bounding Box-Verfahren sind aber nicht durch das Umhüllen mittels Rechteck oder Quader eingeschränkt (Goodman und O'Rourke 2004). Im zweidimensionalen Raum setzt sich die konvexe Hülle aus den Seiten eines konvexen Polygons zusammen. Im dreidimensionalen Raum werden dafür die Flächen eines konvexen Polyeders oder Polytops verwendet (Goodman und O'Rourke 2004, Ericson 2005, Preim und Bartz 2007).

Die k -DOP werden in der Literatur außerdem als eine durch die Anzahl der Hyperebenen k begrenzter, konvexer Polyeder beschrieben. Dies erlaubt eine zusätzlich visuelle Repräsentation durch eine Menge von Facettenstrukturen oder Polygonecken. Beispiele für k -DOP-Algorithmen sind der *gift-wrapping-Algorithmus* und der *Beneath-Beyond-Algorithmus* (Kallay 1984, Goodman und O'Rourke 2004).

Hierarchische Kollisionsdetektion

Die Idee der *hierarchischen Kollisionsdetektion* leitet sich von den *BB-Verfahren* bzw. von den *DOP-Verfahren* ab und wird vor allem für die Echtzeit-Kollisionsdetektion in virtuellen Szenarien genutzt (Kim 2005). Bounding-Box-Verfahren prüfen auf eine konservative Art und Weise, ob eine mögliche Schnittmenge zwischen zwei Objekten vorliegt. Bei der hierarchischen Kollisionsdetektion wird dieses Prinzip auf rekursive Polygonuntermengen beider Objekte ausgeweitet (Zachmann 2000). Dafür werden zu Beginn sämtliche Objekte in verschiedene hierarchische Untermengen unterteilt und in einer dahinterliegenden Baumstruktur angeordnet. Dieses Prinzip der Anordnung ist auch als *Bounding-Volume-Hierarchie* bekannt und leitet sich vom hierarchischen Szenengraph ab (Goodman und O'Rourke 2004, Ericson 2005).

So werden bei einer Kollisionsprüfung während der Laufzeit mögliche Überschneidungen von Polygonuntermengen durch eine Abfrage der Baumhierarchie aufgedeckt und anschließend durch eine Aktualisierung der Baumstruktur kenntlich gemacht. Beim Betrachten von Abbildung 4.6 wird deutlich, dass bei einer Überschneidung der Objekte K und M nur die Flächen und Kanten der überlappenden Boxen durch die hierarchische Kollisionsdetektion geprüft werden. Tritt nur eine Teilüberlappung auf, und die Polygonuntermengen k_{xy} und m_{yx} würden sich nicht schneiden, dann würden auch von k_{xy} eingeschlossene Polygonuntermengen nicht mit m_{yx} eingeschlossene Polygonuntermengen auf vorhandene Schnittpunkte getestet werden (Zachmann 2000, Mezger 2001, Meseth 2006). Da per Definition die Kinderknoten nicht überlappender Knotenpaare ebenfalls keine Schnittmenge bilden, werden diese Knoten bei der Prüfung ausgelassen. Auf diese Weise werden überflüssige Box-Box-Kollisionsprüfungen vermieden und der Zeitfaktor des Gesamtprozesses verringert sich (Ericson 2005).

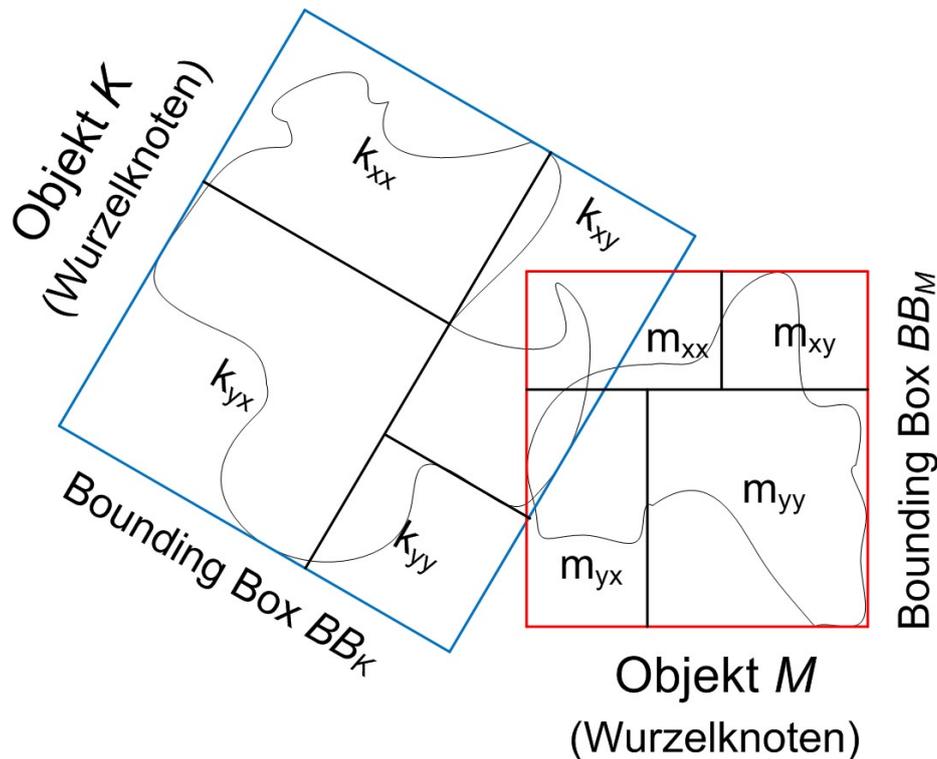


Abbildung 4.6: Die hierarchische Kollisionsdetektion ordnet zunächst die verschiedenen Strukturen eines Objektes in hierarchische Untermengen ein. Anschließend erfolgt die Kollisionsverarbeitung, bei der schrittweise die Polygonuntermengen auf Überlappungen geprüft werden. Eigener, vereinfachter Entwurf nach (Zachmann 2000).

4.4 Physikalisch-basiertes Rendern

Für den Einsatz von Virtuellen Umgebungen in neurowissenschaftlichen Experimenten, z.B. fMRT-Studien, spielt neben der Kollisionsdetektion vor allem die reale visuelle Präsentation bzw. das Rendern von Objekten eine große Rolle. Im Mittelpunkt steht hier die Steigerung des Realitätsempfindens bzw. des immersiven Charakters innerhalb der verwendeten VR-Szene (Mueller et al. 2012a). Die zuverlässigste Methode zur Evaluierung des Realitätsempfindens bleibt hierbei die Auswertung von post-experimentellen Befragungen (Schubert 2003) (Anhang A).

Ein essentieller Aspekt zur authentischen und immersiven Präsentation von virtuellen Szenen stellt das sogenannte *physikalisch-basierte Rendern* oder *predictive rendering* dar (Ulbricht et al. 2005). Im Zentrum stehen hier drei Charakteristika.

Dies sind die originalgetreue, physikalische Repräsentation von Licht-Materie-Interaktionen bei Materialien, die effiziente Anordnung virtueller Lichtquellen und die Simulation geometrischer Besonderheiten, welche durch die Erläuterungen zum Kollisionsmanagement bereits beschrieben wurden. Durch diese hoch angesetzten Maßstäbe gilt das *predictive rendering* als ein Idealmodell, was mit den heutigen technischen Möglichkeiten nur bedingt umzusetzen ist. Ausführliche Informationen zu diesem Thema finden sich in der hier zitierten Fachliteratur (Ashdown et al. 1995, Goesele 2004).

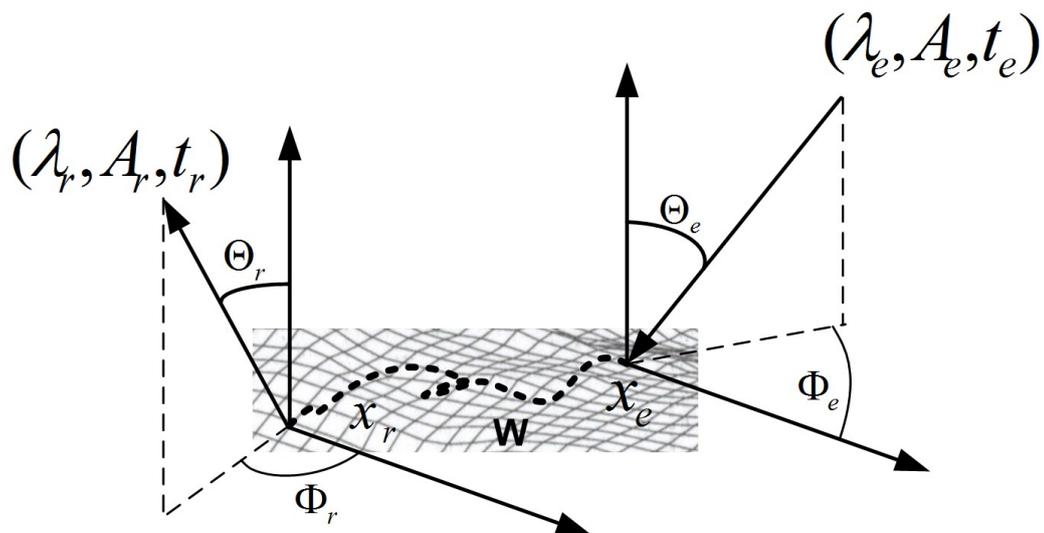


Abbildung 4.7: Das vereinfachte Verhalten eines Photons im Sinne der physikalischen Licht-Materie-Interaktion. Eigener Entwurf nach (Grehn 1998, Birn 2009).

4.4.1 Texturierung und Repräsentation künstlicher Materialien

Ausgehend vom Begriff des *predictive rendering* spielt bei der originalgetreuen Repräsentation von Materialien vor allem die Menge und die Verteilung des einstrahlenden Lichtes eine entscheidende Rolle. Verantwortlich für eine authentische Lichtreflexion ist das für die jeweilige Szene zugrundeliegende Licht-Materie-Interaktionsmodell (Meseth 2006, Birn 2009). Da solche physikalischen Modelle eine sehr hohe Komplexität aufweisen, muss für die schnelle Simulation von Lichtreflexionen auf weniger komplexe Modelle zurückgegriffen werden (Abb. 4.7). Ein eingestrahlt Photon der Wellenlänge λ_e und der Amplitude A_e erreicht zum Zeitpunkt t_e die Oberfläche eines Materials am Punkt x_e . Die Richtung des einstrahlenden Photons ist durch die beiden Einstrahlwinkel (Θ_e, Φ_e) gegeben. Nach dem Auftreffen des Photons bewegt sich dieses für eine Strecke W durch das Material und tritt am Punkt x_r zum Zeitpunkt t_r aus dem Material aus. Ausgehend von den Eigenschaften der Licht-Materie-Interaktion ergibt sich nach dem Austritt des Photons eine veränderte Wellenlänge λ_r und eine veränderte Amplitude A_r . Die Richtung des abstrahlenden Photons wird dementsprechend durch (Θ_r, Φ_r) definiert (Grehn 1998). Obwohl dieses Modell bereits sämtliche Interferenz-, Polarisations- und Diffraktionseigenschaften vernachlässigt, ist es für die praktische Anwendung von Render-Algorithmen immer noch zu komplex.

Infolgedessen wurde ein vereinfachtes, achtdimensionales Modell mit dem Namen *bidirectional scattering-surface reflection distribution*-Modell (BSSRDM) für die praktische Anwendung vorgestellt (Nicodemus et al. 1977).

Hierfür wurden einige Annahmen des komplexen, physikalischen Licht-Materie-Interaktionsmodells durch vereinfachte Annahmen ersetzt:

- Es werden ein sofortiger Lichttransport und zeitlich unveränderbare Materialien vorausgesetzt, welche das Vernachlässigen sämtlicher zeitlicher Abhängigkeiten erlauben.

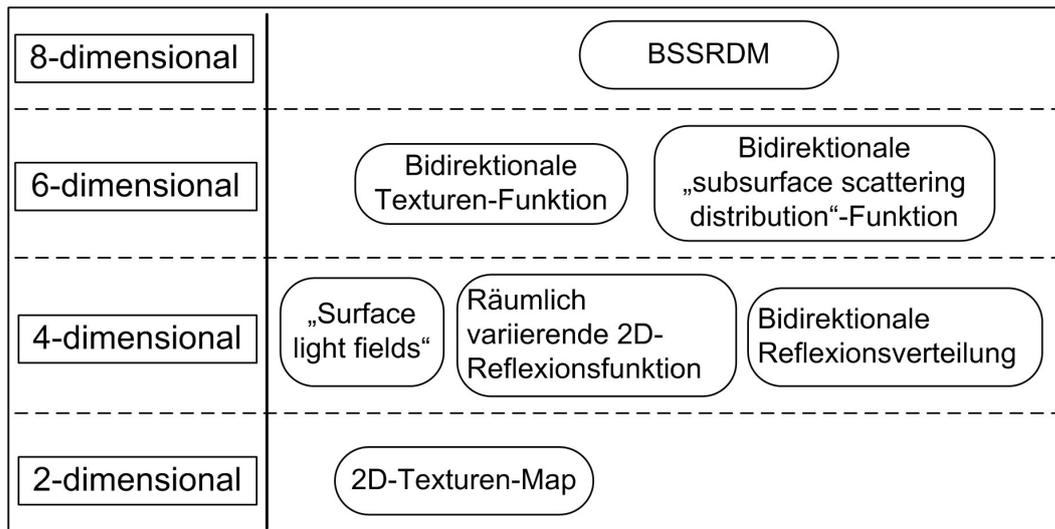


Abbildung 4.8: Die Komplexität vorhandener Materialmodelle lässt sich anhand ihrer Dimensionen beschreiben. Komplexe Modelle, wie z.B. das BSSRDM, nutzen mindestens acht Dimensionen. Modelle mit einer vereinfachten physikalischen Darstellung von Objekten kommen bereits mit vier Dimensionen aus. Eigener Entwurf nach (Birn 2009).

- Sämtliche Absorptionseffekte werden vernachlässigt.
- Für Licht-Materie-Interaktionen wird eine diskrete Menge von Wellenlängen eingeführt, die durch das RGB- oder das CMYK-Farbmodell repräsentiert werden können.

Aufgrund der allgemein gehaltenen Struktur gewährt das achtdimensionale BSSRDM-Modell die Einführung weiterer weniger komplexer Algorithmen. Eine Übersicht zu den einzelnen Licht-Materie-Interaktionsfunktionen, abhängig von ihrer Dimension, findet sich in Abbildung 4.8.

4.4.2 Methodik des physikalisch-basierten Renderns

Ausgehend von den zuvor beschriebenen Licht-Materie-Interaktionsmodellen sollen im Folgenden die Grundlagen des physikalisch-basierten Renderns vorgestellt und erläutert werden. Das Ziel physikalisch-basierten Renderns ist die Transformation einer sogenannten Szenenbeschreibung in ein korrespondierendes, zweidimensionales Bild, welches Erfahrungswerten bzw. abgespeicherten Perzepten des Betrachters entspricht (Birn 2009). Um dieses Ziel zu erreichen, muss ein Render-Algorithmus die notwendige Menge und die Verteilung des Lichtes innerhalb des Szenarios mittels der bereits vorgestellten physikalischen Licht-Materie-Interaktionsmodelle berechnen.

Als Basis sämtlicher Render-Algorithmen gilt heute immer noch die sogenannte formale Beschreibung des zum Rendern notwendigen Lichttransports, welche durch James Kajiya geprägt wurde (Kajiya 1986):

$$L(x', \vec{\omega}') = L_e(x', \vec{\omega}') + \int_S f_r(x) L(x, \vec{\omega}) G(x, x') V(x, x') dA. \quad (4.3)$$

Der Term $L(x', \vec{\omega}')$ definiert die vom Punkt x' in Richtung des Vektors $\vec{\omega}'$ ausgehende Strahldichte. Der additive Emissionsterm $L_e(x', \vec{\omega}')$ beschreibt die emittierte Strahlung die von x' in Richtung $\vec{\omega}'$ ausgestrahlt wird und entspricht dann einem Wert größer Null, wenn der Punkt x' selbst eine Lichtquelle repräsentiert. Der Streuungsterm $f_r(x)$ definiert die sogenannte *bidirektionale Reflexionsverteilungsfunktion* (BRDF) vom Punkt x , gemessen am Punkt x' . Sie beschreibt das Verhältnis zwischen emittiertem und absorbiertem Licht. Als weiterer Term dieser Gleichung repräsentiert $L(x, \vec{\omega})$ die ausgehende Strahlung vom Punkt x . $G(x, x')$ definiert in diesem Zusammenhang den sogenannten Geometrieterm zwischen den Flächen von x und x' . Der Term $V(x, x')$ beschreibt die Sichtbarkeit des Punktes x aus der Position x' . Mit Hilfe des letzten Terms dA wird das geometrische Verhältnis zweier Flächen in die Gleichung einbezogen. Kajiya nutzt hier die Fläche dA als äquivalente Beschreibung der Fläche $d\vec{\omega}'$, wobei $\vec{\omega}'$ den Richtungsvektor von x' nach x kennzeichnet. Der Vollständigkeit halber sei erwähnt, dass der Parameter S des Integrals die gesamte Menge aller sichtbaren Oberflächen beschreibt (Kajiya 1986).

Um den hohen Berechnungsaufwand der Render-Gleichung zu minimieren, wurden im Laufe der Zeit zwei verschiedene Berechnungsstrategien entwickelt (Meseth 2006, Birn 2009). Die erste Strategie geht von einer endlichen Menge von geometrischen Abschnitten aus und berechnet darüber die Verteilung von Wärme- oder Lichtstrahlung (*Radiositätsverfahren*) (Goral et al. 1984). Die zweite Strategie versucht den hohen Berechnungsaufwand der Render-Gleichung (Gleichung 4.3) zu umgehen und das Berechnungsprinzip durch Annahme von unendlich kleinen Flächenelementen direkt umzusetzen (*Strahlenverfolgungs- bzw. Raytracing-Verfahren*) (Pharr und Humphreys 2004).

4.4.3 Echtzeit-Rendern

Aufgrund des hohen Rechenaufwands, stieß das Echtzeit-Rendern komplexer Szenarien lange Zeit an seine Grenzen. Heute konzentrieren sich Grafikkarten mit eigenen Rechenregistern und Speicherkapazitäten allein auf die Abarbeitung von Beleuchtungsmodellen und entlasten dadurch die Kapazitäten des Zentralprozessors (Birn 2009). Aufgrund dieser Fortschritte bei Echtzeit-Renderprozessen gewinnt der Einsatz von Virtuellen Umgebungen in Echtzeit-fMRT- oder EEG-Studien zunehmend an Bedeutung (Mueller et al. 2011a; 2012b). In diesem Zusammenhang zeigen die Resultate von Neurofeedback-Studien zur Erforschung von ADHS bei Kindern (Heinrich et al. 2007, Leins et al. 2007) oder zur Erforschung des räumlichen Gedächtnisses bei Erwachsenen (Grzeschik 2007, Thoms et al. 2011) eine gesteigerte intrinsische Motivation und eine höhere Aufmerksamkeit bei den Probanden (Abb. 4.9). Durch Echtzeit-Rendern entstandene Bilder sind regulär durch eine geringere Qualität gekennzeichnet. Zur Reduzierung des Berechnungsaufwands und für eine erfolgreiche Umsetzung von Echtzeit-Render-Techniken konzentrieren sich Algorithmen vor allem auf das effiziente Zwischenspeichern und Wiederverwenden vorhandener Berechnungsergebnisse, auf die Verwendung von erweiterten Radiositäts- und Raytracing-Verfahren und auf die Echtzeit-Evaluierung und Anwendung von partiell vorberechneten Beleuchtungsdaten (Meseth 2006). Im Rahmen dieser Arbeit mussten zudem alle Methoden mit der

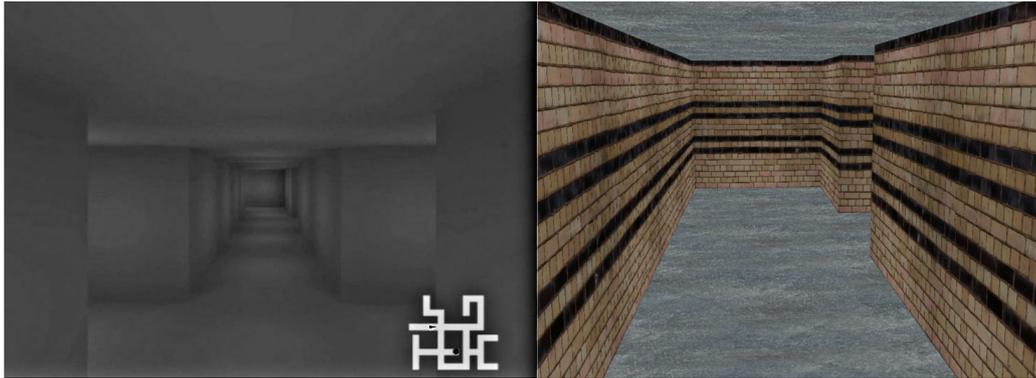


Abbildung 4.9: Zwei der im Institut für Biometrie und Medizinische Informatik modellierten Virtuellen Umgebungen, welche durch eine kommerzielle Software mit Raytracing-Algorithmen gerendert wurden (Grzeschik 2007, Thoms et al. 2011).

zeitkritischen Abarbeitung der Echtzeit-Datenauswertung abgestimmt werden. Neben den bereits vorgestellten Methoden des Radiositäts- und Raytracing-Verfahren sind hier vor allem Methoden zum effizienten Zwischenspeichern zu nennen.

Optimierung von Zwischenspeicherprozessen

Die Berechnung eines globalen Beleuchtungssettings für ein virtuelles Szenario ist ein zeitaufwendiger Prozess, der auch mit optimierten Berechnungsmethoden die Performanz einer Echtzeit-Applikation vermindert. Die Ursache dafür ist die oftmals hohe Anzahl an Pixel pro Einzelbild, welche durch den Render-Algorithmus berechnet werden. Je höher die Bildauflösung und die damit verbundene Anzahl der Pixel, umso höher ist auch der Berechnungsaufwand zum Rendern des Bildes (Craig et al. 2009). Des Weiteren sind hochauflösende Einzelbilder detailreicher und bewirken beim Betrachter ein gesteigertes Gefühl von Realismus. Hier gilt es ausgehend von der jeweiligen Zielsetzung des Anwenders ein sinnvolles Maß zwischen Berechnungsaufwand und gewünschtem Realismus festzulegen (Zachmann 2000).

Einige Render-Techniken unterstützen den Anwender bei dieser Entscheidung. Das *Ray-Caching*, zu dem bereits eine Vielzahl von Erweiterungen existieren, ermöglicht ein zeitsparendes Rendern auf niedriger Auflösung und gestattet anschließend die Verwendung der zwischengespeicherten Daten zur Interpolation des gerenderten Bildes auf eine höhere Auflösung (Meseth 2006).

Hier wird auch die beleuchtungsabhängige Zwischenspeicherung berücksichtigt (Teller et al. 1996, Bala et al. 1999). Dabei werden zunächst Orte mit konstant hohen Strahlungsemissionen anhand von Schattierungsfehlern (Shading-Errors) identifiziert. Ist der Wert des Shading-Fehlers kleiner als ein benutzerdefinierter Schwellenwert, wird die Zwischenspeicherung der Strahlenwerte in einer dafür vorgesehenen Datenstruktur veranlasst und für wiederkehrende Renderabfragen verwendet. Die Methode des *Ray-Caching* wurde innerhalb dieser Arbeit für das Rendern von VR-Stimuli verwendet und im Zuge des Entwicklungsprozesses auf die Kommunikations- und Auswerteprozesse der Echtzeit-Datenauswertung abgestimmt.

Kapitel 5

Problemanalyse

Dieses Kapitel beschäftigt sich mit der Problemanalyse und den speziellen Anforderungen an eine neuartige VR-Stimulussoftware, welche im Rahmen von Neurofeedback-Experimenten zusätzlich auf Echtzeit-Auswerteeinformationen zugreifen soll.

Im Zentrum dieser Problemanalyse stehen zunächst die Anforderungen an den geplanten Entwurf der Applikation und die Anforderungen für die Einbindung in ein medizintechnisches Akquirierungssystem. Des Weiteren werden die Abläufe des bevorstehenden Softwareentwicklungsprozesses sowie die wichtigsten Grundkomponenten zur Implementierung dieser VR-Stimulusapplikation analysiert und erörtert.

Um die Komplexität und die Zusammenhänge der verschiedenen Experimentabläufe zu verdeutlichen, soll anhand eines Beispiels der Modellierungsprozess eines neurokognitiven Experiments näher vorgestellt werden. Die Analyse verwandter Stimulus- und Datenanalyseapplikationen aus dem neurowissenschaftlichen Forschungsbereich schließt dieses Kapitel ab.

5.1 Allgemeine Anforderungen an neurowissenschaftliche Stimulusapplikationen

Für die Anforderungsanalyse zur Erstellung eines neuartigen VR-Stimulusframeworks mit Neurofeedback-Anbindung wurden zunächst vorhandene neurowissenschaftliche Stimulus- und Datenauswertesysteme hinsichtlich ihrer Kern- und Kommunikationsfunktionen analysiert (Mueller et al. 2009; 2011b; 2012b). Dabei zeigte sich, dass die Mehrzahl dieser Systeme eigenständig arbeiten und eine direkte Netzwerkkommunikation zum Austausch von relevanten Experimentinformationen mit allen beteiligten Applikationen, z.B. innerhalb einer MRT-Infrastruktur, bevorzugen (vgl. Abschnitt 5.4). Eine solche verteilte Lösung erlaubt vor allem für rechenintensive Prozesse eine höhere Performanz und könnte damit die Aufteilung der Visualisierungs- und Analyseressourcen auf eine VR-Stimulus- und eine Datenauswertekomponente ermöglichen.

Allerdings müssen bei dieser Lösung neben der Bereitstellung oder Implementierung auch die komplexe Einbindung der Einzelanwendungen in die bestehende Softwarearchitektur sowie die Interapplikationskommunikation innerhalb des medizinischen Akquirierungssystems realisiert werden (Abb. 5.1). Aufbauend auf diesen Arbeitsschrit-

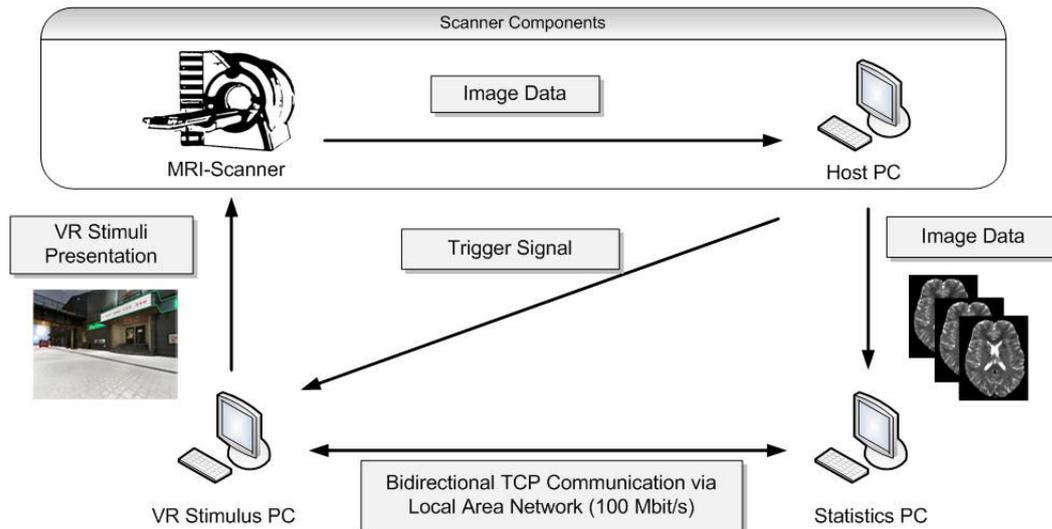


Abbildung 5.1: Die Realisierung von Echtzeit-fMRT-Systemen setzt verschiedene Prozessabläufe und technische Strukturen voraus. Eine mögliche Lösung präferiert die Verteilung sowie die Vernetzung aller beteiligten Komponenten (Mueller et al. 2012b).

ten müssen zunächst die Struktur des VR-Stimulusframeworks mit den globalen Prozessabläufe sowie die Interapplikationskommunikation modelliert werden. Anschließend kann im Rahmen eines Top-Down-Prozesses die konzeptuelle Planung sowie die Implementierung der VR-Stimulusapplikation erfolgen.

Ein weiterer, wichtiger Punkt ist der Entwurf eines erweiterbaren Stimulusframeworks, welches die Einbindung verschiedener Modalitäten mit Echtzeitcharakter, wie z.B. EEG oder Echtzeit-fMRT, unterstützt (Mueller et al. 2012a). Hier müssen zusätzliche Kommunikationsprotokolle zum schnellen und sicheren Datenaustausch geplant und integriert werden.

Ausgehend von diesen Überlegungen wurden zunächst mehrere Präsentationssysteme auf ihre Struktur und ihre Kernfunktionalitäten analysiert (Birbaumer et al. 1999, Weiskopf et al. 2004a, Hollmann et al. 2008). Präsentationssysteme lassen sich im Wesentlichen in folgende drei Hauptkomponenten einteilen:

- **Datenakquirierungssystem:** Diese Komponente beinhaltet sämtliche Hardware und Software zur Akquirierung neurowissenschaftlicher Daten. Als Beispiel sind hier die Magnetresonanztomographie, die Elektroenzephalographie oder die bereits genannte Nahinfrarotspektroskopie (NIRS) genannt. Speziell am MRT kommen Steuersequenzen zum Einsatz, welche die Kontrolle über die Gradientensysteme übernehmen und so die Aufnahme von medizinischen Bilddaten ermöglichen. Am Elektroenzephalographen werden für die Datenakquirierung zusätzlich ein Signalverstärker sowie ein Analog/Digital - Wandler eingebunden.
- **Datenverarbeitung:** Hier unterscheidet man zum einen die Vorverarbeitung und zum anderen die statistische Auswertung der neurowissenschaftlichen Daten. Abhängig von der jeweiligen Modalität werden in der Vorverarbeitung spezifi-

sche Algorithmen zur Filterung von Störsignalen eingesetzt. Des Weiteren werden definierte statistische Kenngrößen extrahiert, welche anschließend in der Datenanalyse die neuronale Aktivierung der Hirnareale beschreiben.

- **Stimuluspräsentation:** Die Stimuluspräsentation beinhaltet in allen Modalitäten überwiegend Funktionen zum Wiedergeben von visuellen, auditiven und taktilen Reizen. Die Unterstützung von olfaktorischen und kinästhetischen Reizen stellt eine Besonderheit dar und kann nur durch die Einbindung zusätzlicher Komponenten ermöglicht werden.

Alle Systeme, die im neurowissenschaftlichen Bereich eingesetzt werden, sind Mehrkomponentensysteme. Der Ausfall eines Systems sorgt unweigerlich für den Ausfall des Gesamtsystems (Mueller et al. 2012a).

Des Weiteren herrscht ein festdefinierter Informationsfluss mit implementierten Prozessabläufen. Gerade für Neurofeedback- oder BCI-Systeme besteht die Prozesskette aus einer Kombination von parallel und seriell ablaufenden Prozessen. Stimuluspräsentation und Datenakquirierung laufen parallel zueinander ab und sind eigenständige Prozesse. Für die statistische Auswertung eines neurowissenschaftlichen Experiments müssen diese Prozesse zueinander synchron ablaufen. Um Unterschied dazu findet die Datenanalyse immer zeitversetzt zur Datenakquirierung statt und ist zudem von deren Ergebnissen abhängig. Aus informationstechnischer Sicht muss eine Echtzeit-Datenauswertung immer innerhalb eines festdefinierten Zeitraums stattfinden und so das Kriterium von „harter Echtzeit-Verarbeitung“ erfüllen (Kopetz 2011).

Für Neurofeedback- oder BCI-Methoden wird außerdem ein schneller Übertragungsweg zwischen Datenauswertung und Stimuluspräsentation benötigt. In bisherigen Konzepten war dieser Übertragungsweg unidirektional. Für adaptive Stimulusparadigmen oder für einen erweiterten Datenaustausch empfiehlt sich ein beidseitig verknüpfter oder bidirektionaler Kommunikationsweg, welcher im Rahmen der Implementierung realisiert werden muss.

Aufgrund der allgemeinen Anforderungen an die VR-Stimulusapplikation wurde das *V-Modell* als Softwareentwicklungsmodell zum Entwurf und zur Realisierung der VR-Stimulusapplikation ausgewählt und durch das Tailoring an die spezifischen Gegebenheiten angepasst (Bröhl 1995, Schatten 2010). Die analytischen Phasen des Modells mit dem Systemdesign und den Systemspezifikationen sollten hier besondere Beachtung finden und damit zu einer hohen Benutzerakzeptanz der Anwendung beitragen (Mueller et al. 2009; 2010b; 2011b). Innerhalb der synthetischen Phasen sollten zudem verschiedene Anwender- und Performanztests sowie die Evaluierung der VR-Stimulusapplikation in einer neurowissenschaftlichen Studie die Qualität der implementierten Applikation validieren (Abb. 5.2).

Für ausführliche Informationen zum V-Modell sei auf (Bröhl 1995, Schatten 2010) oder auf einschlägige Literatur zur Softwareentwicklung verwiesen.

Skalierbares Architekturmodell zur Interapplikationskommunikation

Ein weiteres Problem ist der Entwurf einer möglichst flexiblen Stimulusapplikation, welche eine sehr große Menge von komplexen Informationen empfangen oder an ver-

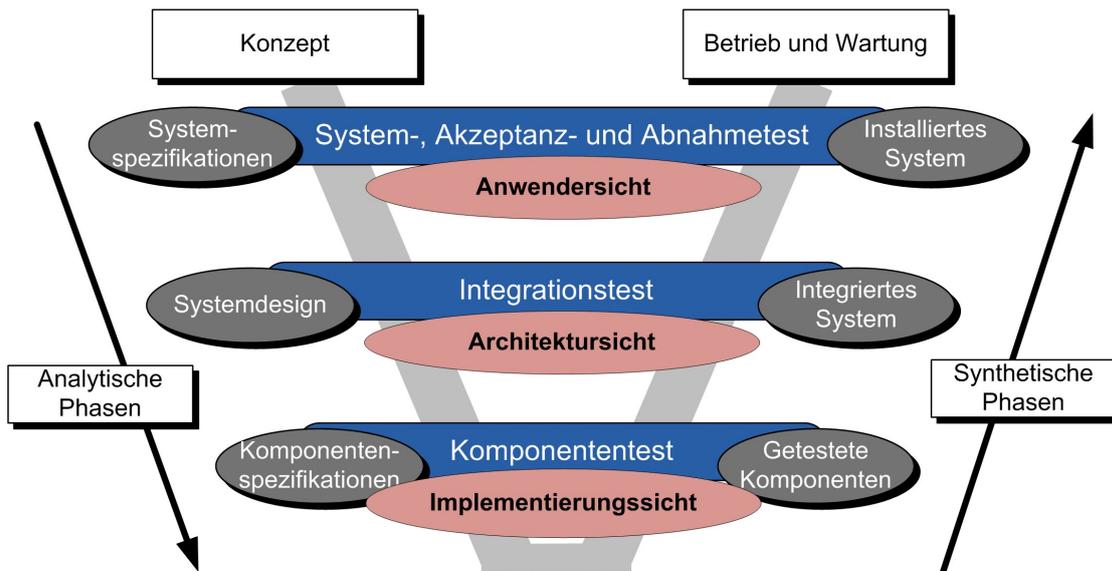


Abbildung 5.2: Auf der linken Seite der Abbildung befinden sich die analytischen Phasen des V-Modells, in denen das eigentliche Konzept und Teile der Implementierung stattfinden. Auf der rechten Seite sind die synthetischen Phasen angesiedelt, welche sämtliche Tests sowie Betriebs- und Wartungsroutinen beinhalten. Eigener Entwurf nach (Schatten 2010).

schiedene Objekte verteilen kann. Aufgrund der geplanten Nutzung als adaptive Hirn-Computer-Schnittstelle mussten spezifischen Prozesse zur Interapplikationskommunikation bereitgestellt werden. Obwohl alle Objekte auf die selben Informationen zugreifen, hat doch jedes Objekt unterschiedliche Anforderungen und benötigt meist nur bestimmte Aspekte einer großen Datenmenge. Ein Informationssystem muss diese Bedürfnisse der unterschiedlichen Objekte gewährleisten und herausfiltern können. Weiterhin muss das System so flexibel sein, dass es auch mit ständig neuen Informationen zurechtkommt und diese auch verarbeiten kann.

In diesem Zusammenhang spielt die Skalierbarkeit und Wartung dieser Stimulusapplikation eine große Rolle. Die meisten Systeme werden vordergründig von der Wartung und der zukünftigen Erweiterbarkeit dominiert. Gerade hier zeigt sich auch der Vorteil der objektorientierten Programmierung, die im Grunde nicht die Entwicklung von Systemen beschleunigt, sondern den zukünftigen Wartungsaufwand des Informationssystems reduzieren soll (Fowler 1997, Larman 2006).

Architekturmodelle greifen genau diese Punkte auf und garantieren durch ihre allgemeinen Spezifikationen eine geringe Kopplung zwischen den einzelnen Komponenten. Architekturmodelle werden prinzipiell anhand der Anzahl der verwendeten Schichten unterschieden und sind durch ihre mehrschichtige, hierarchische Struktur gekennzeichnet. Ausgehend davon kann immer nur von Schichten mit höherem Abstraktionsniveau auf Schichten mit niedrigerem Abstraktionsniveau zugegriffen werden.

Drei-Schichten-Architektur

Durch die geplante Nutzung der vorliegenden Anwendung als Hirn-Computer-Schnittstelle innerhalb einer verteilten Prozesstruktur gewinnt vor allem die Interapplikationskommunikation einen hohen Stellenwert (Mueller et al. 2010a; 2011a).

Um den verschiedenen Anforderungen an die Anwendung gerecht zu werden, plante man die Struktur der Anwendung ausgehend von einem angepassten Drei-Schichten-Architekturmodell. Programmeinheiten (Methoden, Klassen, Module) sollten damit logische Aufgaben wohldefiniert abbilden und somit zusätzlich eine hohe Kohäsion zwischen den Funktionalitäten gewährleisten. Aspektorientierte Programmierung soll in die Anwendung einfließen und so die unterschiedlichen Programmeinheiten gleicher Logik miteinander verknüpfen (Mueller et al. 2009). Zusätzlich dazu verlangt die 3D-Visualisierung der VR-Stimuli ein effizientes Ressourcenmanagement, welches durch die aufteilung der Funktionen im Drei-Schichten-Architekturmodell gewährleistet werden soll. Der Hauptunterschied zu weniger strukturierten Architekturmodellen, wie z.B. dem Zwei-Schichten-Modell, ist die Aufteilung der Kernprozesse in eine *Präsentationsschicht* und eine *Applikationsschicht* (Logikschicht) (Abb. 5.3). Dies ver-

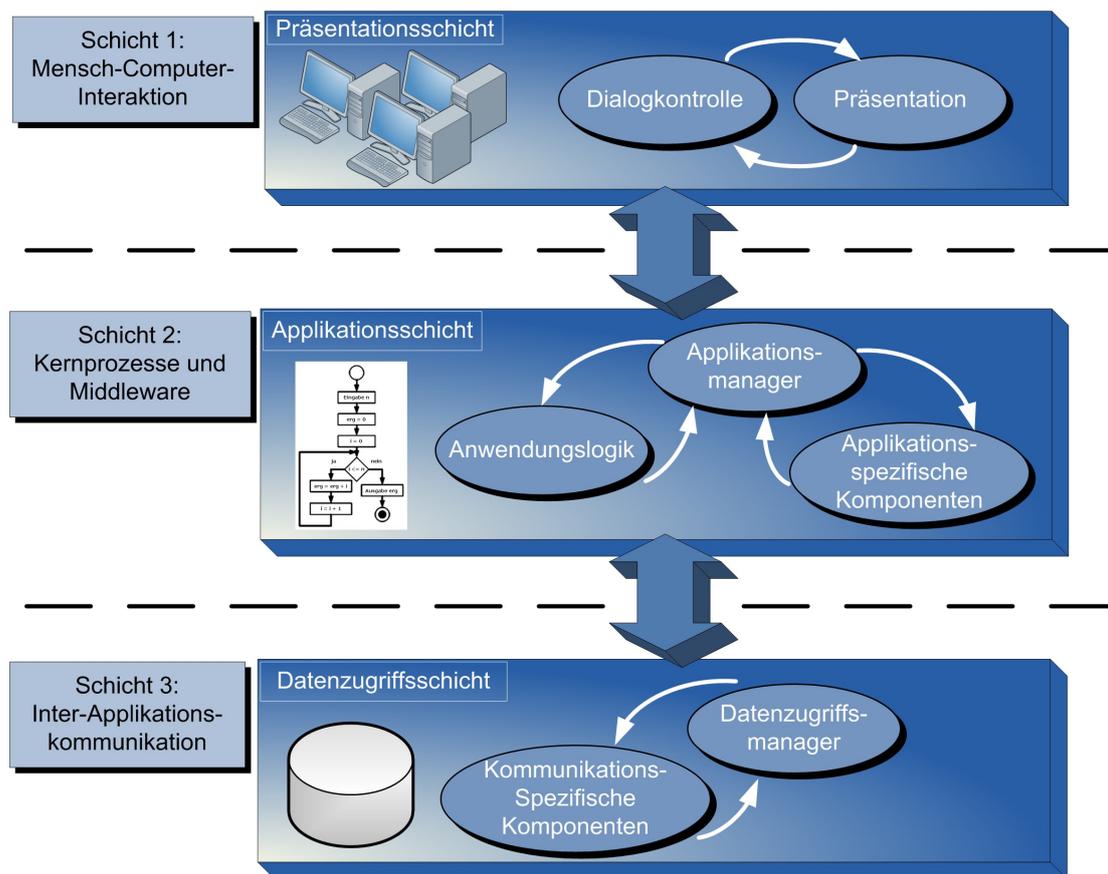


Abbildung 5.3: Im Vergleich zur zweischichtigen Architektur existiert bei der allgemeinen Drei-Schichten-Architektur eine zusätzliche Domänenschicht, welche die Aufgaben der Anwendungslogik und gegebenenfalls des Datenmanagement übernimmt. Eigener Entwurf nach (Coulouris et al. 2003).

spricht, solange bestehende Strukturen in den Erweiterungen konsequent eingehalten werden, eine effiziente und zuverlässige Stimuluspräsentation sowie eine Verbesserung von Wartung und Pflege der Anwendung. Des Weiteren erlaubt diese Strukturierung eine hohe Ordnung und eine gezielte Absicherung vorhandener Programmeinheiten gegenüber neuem oder variiertem Quellcode („protected variation“) (Larman 2006).

Die drei Schichten des Modells besitzen eine konkrete Aufgabenverteilung. Die *Präsentationsschicht* als hierarchisch höchste Schicht, oft auch als „Front-End“ bezeichnet, dient zur Präsentation der Daten und verarbeitet sämtliche Benutzereingaben.

Die *Applikationsschicht* ist in der Hierarchie die zweite Schicht des Modells. Sie beinhaltet die eigentliche Funktionalität sowie Logik der Anwendung und repräsentiert standardmäßig den Kern der Anwendung. Sämtliche Prozessergebnisse dieser Schicht werden zur *Präsentationsschicht* weitergeleitet und nur dort dargeboten.

Die dritte Schicht ist die Datenzugriffsschicht oder auch Kommunikationsschicht. Sämtliche Zugriffe auf Informationsspeicher bzw. der Inter-Applikationskommunikation zu und von entfernten Anwendungen wird in dieser Schicht realisiert und gesteuert.

5.2 Aufbau virtueller Umgebungen zur Stimuluspräsentation

Die Konzeption der Einzelsysteme wurde nach der Strukturplanung des neurowissenschaftlichen Gesamtsystems durchgeführt. Dazu wurden zunächst verschiedene neurowissenschaftliche Präsentationssysteme hinsichtlich ihrer Kernfunktionalitäten analysiert. Die untersuchten Präsentationssysteme waren: *Presentation* (Neurobehavioral Systems 2011), *VR Worlds 2* (Psychology Software Tools 2011) und *Cogent* (Laboratory of Neurobiology 2011). Zusätzlich dazu wurden die Datenverarbeitungsanwendungen *Turbo Brain Voyager* (Brain Innovation 2002-2006) und *rtExplorer* (Hollmann et al. 2008) zur Analyse von Neurofeedback-Funktionalitäten verwendet. Ausgehend von dieser Anwendungsanalyse wurden folgende Kernfunktionalitäten für neurowissenschaftliche Präsentationssysteme festgelegt:

- **Multimediale Präsentation:** Diese Komponente beinhaltet sämtliche Funktionalitäten zum Präsentieren multimedialer Stimuli. Die Stimulussoftware sollte das Abspielen von Audiodateien, die Präsentation von Einzelbildern und die Anzeige von Filmdateien unterstützen. Des Weiteren müssen zur Präsentation von dreidimensionalen, virtuellen Umgebungen und zur Interaktion mit der Geometrie spezielle Render- und Kollisionsdetektionsalgorithmen bereitgestellt werden.
- **Timing:** Die Stimulussoftware muss die zeitliche Steuerung der Stimuluspräsentation, abhängig vom vordefinierten Paradigmen-Design, übernehmen. Weiterhin müssen Informationen zum Paradigma gespeichert, verarbeitet und auf den Experimentenablauf angewendet werden. Eine exakte zeitliche Steuerung des Experimentes ist für die anschließende statistische Auswertung der Daten essenziell.

- **Hardware Interfacing:** Die Stimulussoftware muss Schnittstellen zur Steuerung von externen Peripheriegeräten bereitstellen. Dies ermöglicht die Verarbeitung von Tastersignalen zur Avatar-Steuerung oder die Registrierung von Augenbewegungen, die zur späteren Auswertung herangezogen werden können.
- **Event Logging:** Die Ereignisprotokollierung während eines Experiments ist gerade beim Einsatz virtueller Umgebungen unerlässlich. Sämtliche Bewegungs- und Interaktionsereignisse müssen in diesem Zusammenhang für die statistische Auswertung und die Verknüpfung mit aufgetretenen neuronalen Aktivierungen aufgezeichnet werden.
- **Externe Datenkommunikation:** Für Neurofeedback- und BCI-Modalitäten müssen Dateninformationen schnell und zuverlässig über Kommunikationswege übertragen werden. In der Regel werden hier standardisierte Netzwerkprotokolle an die jeweiligen Anforderungen angepasst sowie anschließend in das Stimulus-Framework integriert (Mueller et al. 2010b).

Neben den Anforderungen für klassische Stimulussoftware bedarf die Präsentation von interaktiven, virtuellen Umgebungen besonderer Aufmerksamkeit. Im Zentrum steht die Wahl einer geeigneten Software zur Simulation der virtuellen Welt und zum Rendern der visuellen Informationen. Eine zusätzliche eigenständige Entwicklung einer solchen Grafik-Engine war aufgrund der Komplexität der Einzelprozesse und des eng abgesteckten Entwicklungsplans nicht angedacht. Für die Entwicklung neurowissenschaftlicher Stimulussoftware, wo die Darstellung von virtuellen Umgebungen nicht das alleinige Ziel ist, stellt die Integration vorhandener Frameworks eine wirkungsvolle Alternative dar. Hier empfiehlt sich eine genaue Sondierung und Evaluation frei verfügbarer Grafik-Engines.

Es gilt zu beachten, dass die Darstellung professioneller, dreidimensionaler Umgebungen im Wesentlichen auf zwei Prozessen beruht: zum einen die Modellierung einzelner 3D-Modellen und das Zusammenfügen der Einzelmodelle zu einer virtuellen Welt oder 3D-Szene (a), und zum anderen das Einladen dieser 3D-Szene in eine dafür geeignete 3D-Grafik-Engine (b). Für neurowissenschaftliche Experimente muss zudem ein festdefiniertes „Timing“ zur Steuerung des Experimentablaufs auf die Szenenpräsentation realisiert und angewendet werden (Mueller et al. 2011a).

Ein weiterer wichtiger Aspekt bei der neurowissenschaftlichen Stimulation mit virtuellen Umgebungen ist die Bestimmung des Interaktionsmaßes zwischen Umgebung und Proband. Je höher die Interaktion, umso komplexer gestaltet sich die Szenenpräsentation. Hohe Interaktionsmöglichkeiten und Bewegungsfreiheiten innerhalb der virtuellen Welt können die Struktur des Paradigmen-design manipulieren und so die spätere statistische Auswertung des Experimentes erschweren. Im Gegenzug verspricht ein hoher Interaktionsfaktor auch ein immersives Verhalten des Probanden, was für neurokognitive Studien oft ein wünschenswerter Faktor darstellt (Bailenson et al. 2008, Mueller et al. 2012b). Um beide Punkte zu berücksichtigen, sollte der prä-experimentell entworfene Experimentenablauf gut gestaltet und hinsichtlich der statistischen Auswertbarkeit überprüft werden. Kurzfristige Änderungen der Stimuluspräsentation können

zudem die statistische Signifikanz von Einzelergebnissen bzw. das Gesamtergebnis sehr stark beeinflussen und sind aus diesem Grund zu vermeiden.

5.3 Erstellung eines neurowissenschaftlichen Experiments am Beispiel

Essenziell zur Durchführung von neurowissenschaftlichen Experimenten ist die Funktionalität zur Erstellung von geeigneten Paradigmen. Die grundlegenden Paradigmenarten wurden bereits in Kapitel 2.2.2 vorgestellt. Um die Komplexität parameterübergreifender Abhängigkeiten zu verdeutlichen, wird in diesem Unterkapitel ein einfaches fMRT-Beispielexperiment eingeführt. An dieser Stelle wird noch einmal darauf hingewiesen, dass die Erstellung eines Paradigmas sowie die statistische Auswertung funktioneller MRT-Daten eng an die hämodynamischen Veränderungen des Blutflusses gekoppelt ist. Des Weiteren sind die messtechnischen physikalischen Parameter einer

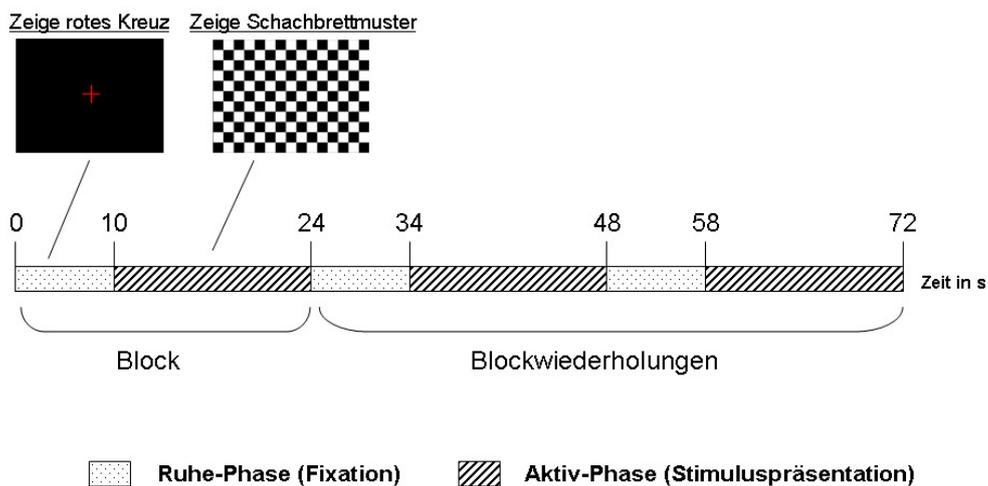


Abbildung 5.4: Das Beispielexperiment im Block-Design startet mit der Ruhe-Phase sowie der anschließenden Aktiv-Phase. Dieser erste Block dient hierbei als Vorlage für alle drei folgenden Blockwiederholungen.

MR-Messesequenz durch Abhängigkeiten eng mit den Modellierungsparametern eines Experiments verknüpft und für unerfahrene Nutzer nur schwer in Relation zu setzen. Inkonsistenzen bei der Erstellung eines neurowissenschaftlichen Experiments führen unweigerlich zur Unbrauchbarkeit des gesamten Experiments.

Der erste Schritt zur Erstellung eines Versuchs ist deshalb die Festlegung der Versuchsbedingungen, abhängig von den im Experiment zu beantwortenden Fragestellungen, und eine klar abgegrenzte Präsentationsstruktur mit den entsprechenden Stimuli.

Anhand des folgenden Beispiels werden zwei einfache Bedingungen definiert, welche die Frage klären sollen, welche neuronalen Aktivierungen während der visuel-

len Stimulation mit einem alternierenden Schachbrettmuster auftreten. Die erste Bedingung ist die Ruhebedingung (a). Die zweite Bedingung die Aktiv-Bedingung (b). Für beide Bedingungen muss ein der Bedingung entsprechender Stimulus festgelegt werden. Der Einfachheit halber soll in der Ruhe-Bedingung eine einfache Grafik mit einem roten Kreuz und schwarzem Hintergrund gezeigt werden (Abb. 5.4). Für die Aktiv-Bedingung wird ein komplexerer visueller Stimulus mit einem alternierenden Schachbrettmuster festgelegt. Beide Bedingungen sollen innerhalb eines modellierten Block-Design-Paradigmas präsentiert werden. Für die Repetitionszeit (T_R) bzw. das Intervall zur Erzeugung eines Datenvolumens wird ein Wert von zwei Sekunden definiert. Anschließend folgt die Festlegung des Parameters für die zeitliche Auflösung aus den beiden Möglichkeiten „Volumen“ und „Millisekunden“. Für dieses Experiment wird der Parameter „Volumen“ ausgewählt. Der nächste Schritt ist die Definition der Blocklänge. Diese wird mit insgesamt 12 Volumenakquirierungen angegeben, d.h. 12 Datenvolumen definieren einen Block und geben somit, ausgehend von unserem T_R , eine Akquirierungsdauer von 24 Sekunden pro Block an. Des Weiteren wird die Reihenfolge unserer Bedingungen spezifiziert. Vom ersten bis zum fünften Volumen wird die Ruhe-Phase angesetzt. Anschließend folgt die Aktiv-Phase vom sechsten bis zwölften Volumen (Abb. 5.4). Dieser erste Block soll nun ausgehend vom definierten Block-Design Paradigma dreimal wiederholt werden. Zusammengefasst ergibt das Experiment so eine Gesamtlänge von 36 Datenvolumen, die eine Akquirierungsdauer von 72 Sekunden benötigen.

An diesem einfach gehaltenen Beispiel wird deutlich, dass die klare Strukturierung eines Experiments in verschiedene Stimulusabschnitte die post-experimentelle Auswertung der neuronalen Aktivitäten und damit die erfolgreichen Beantwortung der gestellten Fragestellungen unterstützen soll. Darauf beziehend sollte die klare Abgrenzung der Stimuli auch zu einer klaren Abgrenzung der einzelnen Versuchsbedingungen führen und so eine statistische Vergleichbarkeit der verschiedenen Hirnaktivierungen ermöglichen.

Welche Hirnareale waren während der einen Versuchsbedingung aber nicht während der anderen Bedingung aktiv? Welche Areale waren in beiden Versuchsbedingungen aktiviert und wie unterscheiden sich die hämodynamischen Signalverläufe zu den unterschiedlichen Stimuluszeiten? Solche oder ähnliche Fragen könnten im Zentrum eines Versuchs stehen.

Leider liegt das Prüfen des korrekten Experimentablaufs zumeist allein in der Verantwortung des Versuchsleiters und wird nur von wenigen neurowissenschaftlichen Stimulusanwendungen automatisch überwacht. Die Integration von Modellierungs- und Sicherheitskontrollfunktionen zur korrekten Erstellung und Prüfung von Experimentverläufen stellt dementsprechend einen zentralen Punkt bei der Entwicklung dieser VR-Stimulusapplikation dar und muss für den Entwurf berücksichtigt werden.

Einige Applikationen haben bereits unterschiedliche Strategien zur Experimentmodellierung entwickelt. Im Folgenden werden diese Anwendungen kurz vorgestellt und ihre Besonderheiten näher erläutert.

5.4 Existierende Softwaresysteme

Ausgehend von der Analyse und der Festlegung der allgemeinen Anforderungen war die Evaluation existierender 3D-Stimuluskonzepte und Datenauswerteanwendungen ein wichtiger Teilaspekt bei der Entwicklung einer VR-Stimulusapplikation. Da die Verwendung des *Vision Engine Frameworks* bereits zu Beginn des Projektes feststand, sollten bestehende Konzepte hinsichtlich der Integration des *Vision Engine*-Konzepts analysiert und zur Weiterentwicklung adaptiert werden. Für die Evaluation verwandter Applikationen wurden drei kommerzielle und eine freie Applikation ausgesucht.

5.4.1 *Presentation* - Neurobehavioral Systems Inc.

Presentation ist eine weitverbreitete, kommerzielle Stimulussoftware, die speziell für neurowissenschaftliche Studien entworfen wurde (Neurobehavioral Systems 2011). Sie beinhaltet eine große Anzahl an Möglichkeiten zur Erstellung neurowissenschaftlicher Stimuli und liefert viele informationstechnische Konzepte zur Stimuluspräsentation. *Presentation* erlaubt dem Benutzer eine relativ einfache Erstellung und Durchführung von Experimenten. *Presentation* unterstützt zudem eine Vielzahl bildgebender Verfahren und kann auch für klassische Modalitäten, wie z.B. EEG, eingesetzt werden. Experimente werden in der Regel über die Skriptsprachen PCL2 und SDL3 erstellt.

Presentation besitzt ein eigenes Timing-Modul, welches an der Experimentsteuerung beteiligt ist und die zeitliche Kontrolle der einzelnen Bedingungen überwacht. *Presentation* ist ebenso in der Lage, Reize (visuell und auditiv) sowie „Response Events“ (benutzerspezifische Eingaben) über ein Logging-Modul zu protokollieren. Die Präsentation komplexer Virtueller Realitäten innerhalb neurokognitiver Experimente wird von *Presentation* nicht unterstützt. Ebenso sind keine Zusatzmodule zur Integration komplexer 3D-Umgebungen in *Presentation* bekannt.

Des Weiteren werden Funktionalitäten zur Protokollierung und zur zeitlichen Steuerung neurowissenschaftlicher Stimuli bereitgestellt. *Presentation* verwendet hier die oben genannten Skriptsprachen. In diesem Zusammenhang fehlt leider eine benutzerfreundliche Bedienoberfläche, welche eine schnelle und einfache Erstellung von Stimuli unterstützt.

5.4.2 *VRWorld 2* - Psychology Software Tools Inc.

VR Worlds 2 ist eine kommerzielle Stimulussoftware die speziell virtuelle Umgebungen für neurowissenschaftliche Studien bereitstellt (Psychology Software Tools 2011). Die Software baut auf einer Grafik-Engine auf, welche in den allgemeinen Produktinformationen der Firma leider nicht näher benannt wird. Im Rahmen einer veröffentlichten Studie präsentierte die Firma ein System, welches eine komplexe und begehbare 3D-Umgebung bereitstellt. Die Steuerung von Avataren wurde mit externen Peripheriegeräten realisiert. Ein Proband konnte diese Welt mittels Joystick erkunden und mit künstlichen Charakteren interagieren (Baumann et al. 2003).

Des Weiteren wird die Protokollierung von Ereignissen unterstützt. Eine professionelle Modellierung von neurowissenschaftlichen Experimenten nach einem Block- oder

Ereigniskorrelierten Design wird nicht näher beschrieben.

Dieses System stellt einen interessanten Ansatz zur Durchführung von VR-Studien dar, jedoch stehen dem hohe Erwerbskosten und nur wenige Produktinformationen gegenüber. Nach Firmenauskünften wird *VR Worlds 2* mit einer einzigen fest eingebundenen Szene geliefert, deren Kosten von der Größe der virtuellen Welt abhängig sind und die nach der Lieferung kaum veränderbar ist. Des Weiteren ist das Einladen von selbst erstellten oder extern erworbenen Virtuellen Umgebungen nicht möglich und verhindert damit eine breite Anwendbarkeit sowie die Mehrfachnutzung innerhalb unterschiedlicher Studien. *VR Worlds 2* vereint einige interessante Ansätze und beweist die Nutzbarkeit virtueller Umgebungen im Bereich der neurowissenschaftlichen Forschung. Allerdings erwies sich die Software aufgrund der Unveränderbarkeit als ungeeignet für das geplante Konzept der adaptiven Paradigmenanpassung. Der kommerzielle Charakter der Anwendung verbietet außerdem den uneingeschränkten Austausch von Virtuellen Szenarien zwischen verschiedenen Forschergruppen und schränkt damit indirekt die Reproduktion von Forschungsergebnissen sowie aufbauende Forschungen mit den entsprechenden Virtuellen Umgebungen ein.

5.4.3 *rtPresent* - IBMI, Universität Magdeburg

rtPresent ist ein freies Stimuluspräsentationswerkzeug zur Durchführung neurowissenschaftlicher Stimuli innerhalb der kommerziellen Software Matlab und ein Teil der Echtzeit-Datenauswerteapplikation *rtExplorer* (Hollmann et al. 2008). Es basiert auf der an XML angelehnten Skriptsprache *EDL* (experiment definition language) und wurde am Institut für Biometrie und Medizinischen Informatik, Medizinische Fakultät, Universität Magdeburg, entwickelt. Das Datenkonzept dieser Software beruht auf einer zentralen, textbasierten Speicherung von Informationen. Des Weiteren führt die Anwendung ein Konzept zur Wissensrepräsentation und Verbreitung von Paradigmeninformationen an alle beteiligten Komponenten ein.

Ein Nachteil stellt die Matlab-spezifische Implementierung und die komplexe Darstellung der Experimentinformationen in einer mehrfach gekapselten XML-Struktur dar. Einfache Adaptierungen sind dadurch nur bedingt möglich. Des Weiteren führt die Datenspeicherung und Informationsverbreitung über eine Textdatei zu Inkonsistenzen oder nicht nachweisbaren Schreib/Lese-Fehlern, welche durch die Einbeziehung von Netzwerk- und Kommunikationsprotokollen vermieden werden können (Coulouris et al. 2003, Tanenbaum et al. 2003, Schatten 2010).

Neben diesen Problemen steigt mit zunehmender Anzahl der Zugriffe auf die Datei die Fehleranfälligkeit der Gesamtprozesskette. Mögliche „Deadlocks“ könnten entstehen und dadurch den Abbruch von Funktionalitäten oder sogar des gesamten Experiments auslösen (Tanenbaum et al. 2003).

Des Weiteren ist *EDL* ein applikationsabhängiges Konstrukt, welches ausschließlich in Verbindung mit der Echtzeit-Datenauswerteanwendung *rtExplorer* genutzt werden kann. Dies bedeutet auch, dass die nachfolgende Offline-Auswertung der Daten nicht mit dem *rtExplorer* durchgeführt werden kann. Der modellierte Experimentablauf im *EDL*-Datenformat ist für die Offline-Auswertung nicht nutzbar und so muss eine weitere Portierung des ursprünglichen Experimentablaufs auf ein weiteres Datenformat

erfolgen.

rtPresent ist als einfaches Präsentationswerkzeug, welches vordergründig auf die Anzeige von statischen Bildern sowie auf akustische Reize spezialisiert ist. Es besitzt keine Schnittstelle zum Einladen von 3D-Formaten und kann komplexe Virtuelle Szenarien nicht darstellen.

Als großer Vorteil dieser Anwendung gilt die Verbreitung und der Austausch von Experimentinformationen innerhalb einer MRT-Infrastruktur. Ein verwandtes Grundkonzept wurde im Rahmen dieses Projektes konzipiert und wird u.a. zur Echtzeit-Adaption von VR-Stimuli verwendet.

5.4.4 *Turbo BrainVoyager* - Brain Innovation B.V.

Turbo BrainVoyager ist eine kommerzielle Anwendung zur Echtzeit-Datenauswertung von fMRT-Bilddaten (Brain Innovation 2002-2006). Die Software besitzt darüber hinaus ein Präsentationsinterface, welches anhand eines sogenannten virtuellen Thermometers die gemittelten Aktivierungswerte aus definierten Hirnregionen anzeigt (Weiskopf et al. 2004a). Diese Funktionalität unterstützt die operante Konditionierung eines Probanden und ermöglicht dadurch die Nutzung als Hirn-Computer-Schnittstelle. Der *Turbo BrainVoyager* ist eine Abspaltung der Hauptanwendung *BrainVoyager QX*, die zur Offline-Auswertung von medizinischen und neurowissenschaftlichen Daten verwendet wird.

Neben der primären Echtzeit-Datenauswertefunktion führt der *Turbo BrainVoyager* auch Vorverarbeitungsschritte in Echtzeit durch.

Dazu zählen eine 3D-Bewegungskorrektur, eine räumliche Glättung und Algorithmen zum Entfernen linearer Langzeittrends.

Für die Echtzeit-Auswertung nutzt der *Turbo BrainVoyager* einen rekursiven Ansatz des Allgemeinen Linearen Modells, welches eine zuvor definierte Designmatrix zur Datenanalyse nutzt (Pollock 1999). *BrainVoyager* ist ein weitverbreitetes Auswertewerkzeug und bietet darüber hinaus eine bekannte Nutzeroberfläche zur Erstellung und Strukturierung neurowissenschaftlicher Experimente (Savoy 2009). Das Konzept dieser Nutzeroberfläche wird von vielen Neurowissenschaftlern akzeptiert und genutzt. Für den Entwurf der VR-Stimulusapplikation wurden zunächst die verschiedenen Oberflächenfunktionen des *Turbo BrainVoyagers* zur Modellierung von Experimentabläufen analysiert und anschließend eigene Konzepte zur VR-basierten Stimuluspräsentation erstellt. Des Weiteren wurde das Konzept einer Plug-in-Schnittstelle zur benutzerdefinierten Erweiterung von Funktionalitäten aufgegriffen und an die Spezifikationen der VR-Stimulusapplikation angepasst.

Kapitel 6

Entwurf einer multimodalen VR-Stimulusapplikation

Dieses Kapitel befasst sich mit der detaillierten Planung und Entwicklung dieser multimodalen VR-Stimulusapplikation. Dabei soll ausgehend von der Problemanalyse der Entwurf dieser Arbeit näher beschrieben werden.

Im Zuge dieses Prozesses wurde zunächst mit der Modellierung des Gesamtkonzepts zur Einbindung der VR-Stimulusapplikation in eine MRT-Infrastruktur begonnen. Anschließend folgte der Entwurf der VR-Stimulusapplikation mit sämtlichen Konkretisierungsschritten, welche die verschiedenen Einzelschichten der Applikation exakt definieren (*Top-down Prinzip*).

Das Entwurfskapitel ist nach der Problemanalyse das zweite Kapitel, welches sich am klassischen Ablauf des Softwareentwicklungsprozesses auf Basis des *V-Modells* orientiert (Schatten 2010).

Wichtige Erkenntnisse des hier entworfenen Ansatzes wurden veröffentlicht (Mueller et al. 2009; 2010b; 2011a;b; 2012a;b).

6.1 Einbindung der VR-Stimulusapplikation in eine MRT-Infrastruktur

Als multimodale Stimulusanwendung konzipiert, sollte diese Software die Anforderungen an verschiedene Modalitäten unterstützen und zukünftige Erweiterungen durch eine flexible und strukturierte Architekturplanung ermöglichen. Der erste Schritt in diesem Modellierungsprozess stellt die Einbindung in eine fMRT- bzw. Echtzeit-fMRT-Infrastruktur und die Modellierung von Prozessabläufen zum Austausch von Informationen dar.

Die Problemanalyse zeigte, dass sich die Infrastruktur eines MR-Systems aus mehreren Komponenten zusammensetzt (Abschnitt 5.1).

Für die Einbindung einer VR-Stimulusanwendung mussten zunächst die globalen Prozessabläufe adaptiert und entsprechende Kommunikationswege zum Austausch der Experiment- und Datenauswertinformationen zwischen den Komponenten neu entworfen werden (Abb. 6.1).

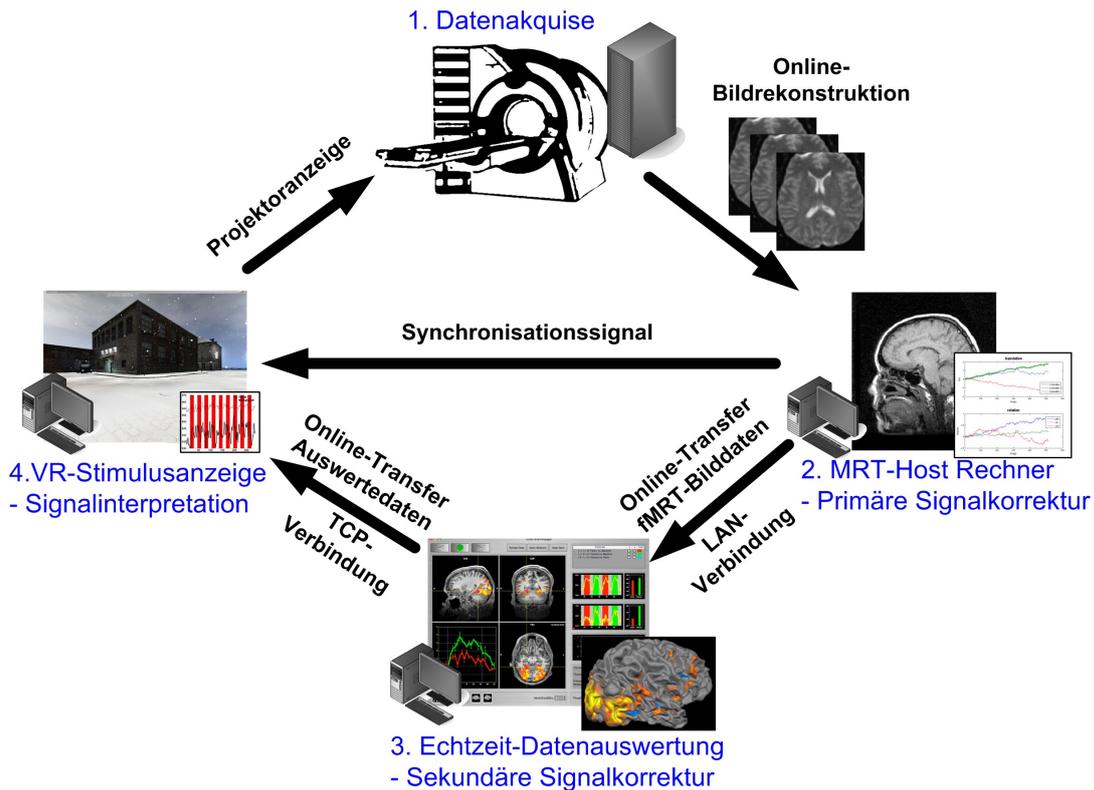


Abbildung 6.1: Die schematische Darstellung der Echtzeit-fMRT-Infrastruktur verdeutlicht die einzelnen Prozessabläufe sowie die zugehörigen Komponenten. Die VR-Stimulusanzeige tritt an die Stelle der regulären Stimuluspräsentation (Mueller et al. 2012a).

Aufgrund der Informationsgewinnung aus der Problembeschreibung und der Erörterung verwandter Arbeiten kann eine VR-Stimulusanwendung als eine komplexere Form der regulären Stimulusanwendung beschrieben werden.

Die *VR-Stimulusanzeige* (Nummer 4, Abb. 6.1.) übernimmt die Präsentation der VR-Stimuli, welche dem Probanden im MRT durch eine Projektoranzeige dargeboten werden. Gleichzeitig werden durch eine EPI-Messesequenz die MR-Bilddaten erzeugt. Dabei wird innerhalb eines definierten zeitlichen Intervalls jeweils ein Volumendatensatz des Kopfes aufgenommen und bereitgestellt. Während dieser *Datenakquise* (Nummer 1, Abb. 6.1.) werden die in Echtzeit gewonnenen MR-Bilddaten unverzüglich zum *MRT-Host-Rechner* (Nummer 2, Abb. 6.1.) transferiert und dort im internen, geschützten Netzwerk freigegeben (Nummer 3, Abb. 6.1.). Die Echtzeit-Datenauswertung greift auf die Bilddaten zu und analysiert diese in Echtzeit. Anschließend werden die Auswertinformationen an die VR-Stimulusanzeige übermittelt.

Um die verschiedenen Interaktionen eines Probanden mit der Virtuellen Umgebung zu ermöglichen, müssen außerdem entsprechende Prozesse zur Steuerung mit MR-konformen Peripheriegeräten entworfen und implementiert werden. Zudem müssen zur Durchführung und zur korrekten statistischen Auswertung von VR-fMRT-Studien der Start der MR-Messesequenz und die VR-Stimuluspräsentation durch ein Synchron-

nisationssignal zeitlich koordiniert werden. Dem gegenüber steht der Online-Transfer von Auswertedaten zwischen der *Echtzeit-Datenauswertung* und der *Stimulusanzeige*. Hier existieren verschiedene Kommunikations- und Implementierungsstrategien. Zum einen besteht die Möglichkeit der kombinierten Implementation, bei der alle Funktionen der Stimuluspräsentation und der Datenauswertung innerhalb einer Anwendung zusammengefasst sind. Als Beispiel für ein solches Konzept sei die Datenauswerteanwendung *Turbo BrainVoyager* genannt (vgl. Abschnitt 5.4.4).

Eine andere Strategie verfolgt der verteilte Ansatz ähnlich des vorgestellten *rtPresent* (Abschnitt 5.4.3). Stimuluspräsentation und Datenauswertung sind auf verschiedene Anwendungen verteilt und kommunizieren über Netzwerkprotokolle bzw. Textdateien. Aus Gründen der Performanz und aufgrund der bevorstehenden Entwicklung eines Konzeptes zur Echtzeit-Stimulusadaptation von VR-Paradigmen entschied man sich für die Implementierung einer eigenständigen VR-Stimulusapplikation auf Basis eines verteilten Konzeptes und mit einem TCP-Framework zur Übertragung von Echtzeit-Datenauswerteeinformationen (Abb. 6.1).

Ein Performanzverlust sowie eine mögliche Instabilität der Stimulusanwendung durch eine gleichzeitige Präsentation von rechenintensiven 3D-Simulationen und zeitlich kritischen Echtzeit-Datenauswertungen sollte auf diese Weise vermieden werden. Des Weiteren sollte durch diese Lösung die schnelle, fehlerfreie und direkte Übertragung von zeitkritischen Auswerteeergebnissen von der *Echtzeit-Datenauswertung* zur *VR-Stimulusanzeige* gewährleistet werden.

Für detaillierte Beschreibungen dieses Konzeptes sei auf das Entwurfskapitel zur Echtzeit-Stimulusadaptation von VR-Paradigmen verwiesen (Abschnitt 6.5).

6.2 Modellierung eines VR-Stimulusframeworks auf Basis einer skalierbaren Softwarearchitektur

Nach der Modellierung von Prozessabläufen zur globalen Einbindung der VR-Stimulusapplikation in eine MRT-Infrastruktur wurde als nächstes der eigentliche Entwurf der VR-Stimulusapplikation als Einzelanwendung durchgeführt. Im ersten Schritt wurden zunächst die groben Prozessabläufe für die VR-Stimuluspräsentation, für die Echtzeit-Datenauswertung (*Online-Auswertung*) und für die prä-experimentelle Datenauswertung (*Offline-Auswertung*) von VR-fMRT-Studien modelliert (Abb. 6.2).

Die Prozessabläufe der Online-Auswertung sind zunächst mit denen in Abbildung 6.1 identisch.

Unmittelbar nach der Bereitstellung des jeweils neuesten Volumendatensatzes durch den MRT registriert die Echtzeit-Datenauswertung die neuen Bilddaten und greift über einen gesicherten Lesezugriff auf diese zu (Punkt A, Abb. 6.2). Alle für die Datenanalyse notwendigen Informationen zum Experimentdesign werden dabei durch eine Protokolldatei (PRT-Datei) bereitgestellt. Diese wurde vor dem eigentlichen Experiment erstellt und kann sowohl durch die Echtzeit-Datenauswertung als auch durch die VR-Applikation eingelesen werden. Die Auswerteeinformationen der Echtzeit-Analyse werden anschließend innerhalb weniger Millisekunden über ein speziell entwickeltes

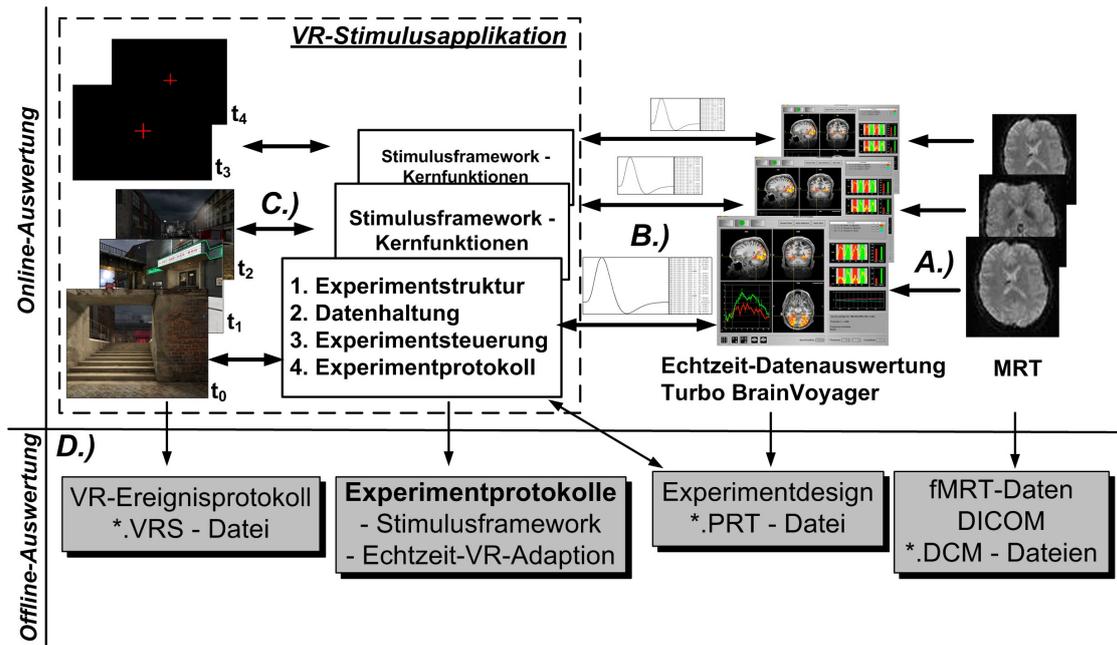


Abbildung 6.2: Die Prozessabläufe zur Online- und Offline-Auswertung verdeutlichen das Grundkonzept der VR-Stimulusapplikation. Die Grundlage bildet das zentrale Stimulusframework mit seinen Kernfunktionen, welche die Anzeige von VR-Stimuli basierend auf dem berechneten Experimentablauf steuern (Mueller et al. 2012b).

TCP-Kommunikationsprotokoll an die VR-Stimulusapplikation versendet (Punkt B, Abb. 6.2) (Mueller et al. 2011a).

Das Grundgerüst der zu entwickelnden VR-Stimulusapplikation bildet das sogenannte *Stimulusframework*, welches sämtliche Kernfunktionen enthält. Diese stellen Funktionen zur Experimentmodellierung, Routinen zur Speicherung von Experimentinformationen, Funktionen zur Berechnung und zeitlichen Steuerung von Experimentabläufen sowie Funktionen zur Ereignisprotokollierung bereit. In dieser übergeordneten Struktur soll ein zweites Framework eingebunden werden.

Das *Graphic Engine Framework* soll in diesem Zusammenhang die Anzeige von virtuellen Szenarios übernehmen. Im Mittelpunkt steht hier die korrekte Präsentation von Stimuli nach dem zuvor modellierten neurokognitiven Experimentablauf. Um dies zu gewährleisten, muss die Steuerung des Graphic Engine Frameworks durch die Kernfunktionen des übergeordneten Stimulusframeworks übernommen werden (Punkt C, Abb. 6.2).

Zur prä-experimentellen Datenanalyse (Offline-Auswertung) werden die aufgezeichneten DICOM-Daten, der Experimentablauf (PRT-Datei) sowie die Experimentprotokolle der VR-Stimulusapplikation, wie z.B. das VR-Ereignisprotokoll und das Protokoll zur Echtzeit-VR-Adaption, in die Auswertung mit einbezogen. Erst durch diese Protokolle ist es möglich, dass der Zeitpunkt und die Dauer spezifischer VR-Ereignisse, wie z.B. Interaktionen des Probanden mit VR-Objekten oder das Erreichen von Wegmarken, zeitlich genau bestimmt und mit neuronalen Aktivierungen in Relation gesetzt werden kann (Punkt D, Abb. 6.2).

Ausgehend von diesen Prozessabläufen sollte ein skalierbares Stimulusframework modelliert werden, welches den aktuellen Anforderungen objektorientierter und aspektorientierter Programmiermethoden genügt (Mueller et al. 2009). Aspektorientiert steht dabei für ein nach Anwendungslogiken aufgeteiltes Stimulusframework, welches im Einzelnen folgende Kernfunktionen bereitstellen soll:

- Die Präsentation multimedialer Stimuli soll mit Hilfe des *Vision Engine Frameworks* realisiert werden. Im Vordergrund steht die Durchführung von regulären fMRT-Experimenten sowie die Anzeige sogenannter Neurofeedback-Stimuli im Kontext der Echtzeit-fMRT (Mueller et al. 2012b).
- Das Stimulusframework soll die Datenhaltung von Experimentinformationen unterstützen. Sämtliche Informationen zu durchgeführten Experimenten sollen abgespeichert und für anschließende statistische Auswertungen zur Verfügung gestellt werden (Mueller et al. 2012a).
- Die zeitliche und funktionelle Steuerung der Stimuli im Rahmen einer festdefinierten Experimentstruktur soll ermöglicht werden. Hier sollen spezielle „Timer-Funktionalitäten“ den zeitlichen Ablauf eines Experimentes überwachen. Der eigentliche Präsentationsprozess sowie die zeitliche Überwachung des Experiments sollen simultan ablaufen (Mueller et al. 2009). Abhängig davon muss die gleichzeitige Abarbeitung mehrerer Prozessstränge innerhalb einer Applikation (Multithreading) unterstützt werden (Coulouris et al. 2003).
- Die Protokollierung von relevanten Ereignistypen außerhalb und während der Stimuluspräsentation soll gewährleistet werden. Dies schließt Ereignisse innerhalb der virtuellen Umgebung, benutzerspezifische Eingaben sowie Kommunikationsereignisse mit ein (Mueller et al. 2012a).
- Netzwerkspezifische Protokolle zur Kommunikation mit externen Anwendungen sollen eingebunden werden. Die Stimulusapplikation soll für Neurofeedback-Studien schnelle, sichere und direkte Kommunikationsmöglichkeiten bereitstellen (Mueller et al. 2010a). Des Weiteren soll eine Möglichkeit zur effizienten und schnellen Verbreitung der Informationen zum experimentellen Setup geschaffen werden. Im Rahmen von Neurofeedback-Studien soll eine flexible Prozessstruktur die Präsentation von Stimuli sowie die simultane Verwendung von Kommunikationsfunktionen ermöglichen. Eine Verwendung von TCP-basierten Kommunikationsprotokollen in Kombination mit Multithreading-Techniken bietet sich in diesem Kontext an (Mueller et al. 2010b).
- Die Anbindung und Verwendung von externen Peripheriegeräten soll für neurowissenschaftliche Studien ermöglicht werden. MRT-konforme Taster sollen benutzerspezifische Eingaben zur Steuerung von virtuellen Charakteren ermöglichen. Die Funktionalität soll über COM-Schnittstellen mit Hilfe des DirectX-Frameworks realisiert werden (Mueller et al. 2009).

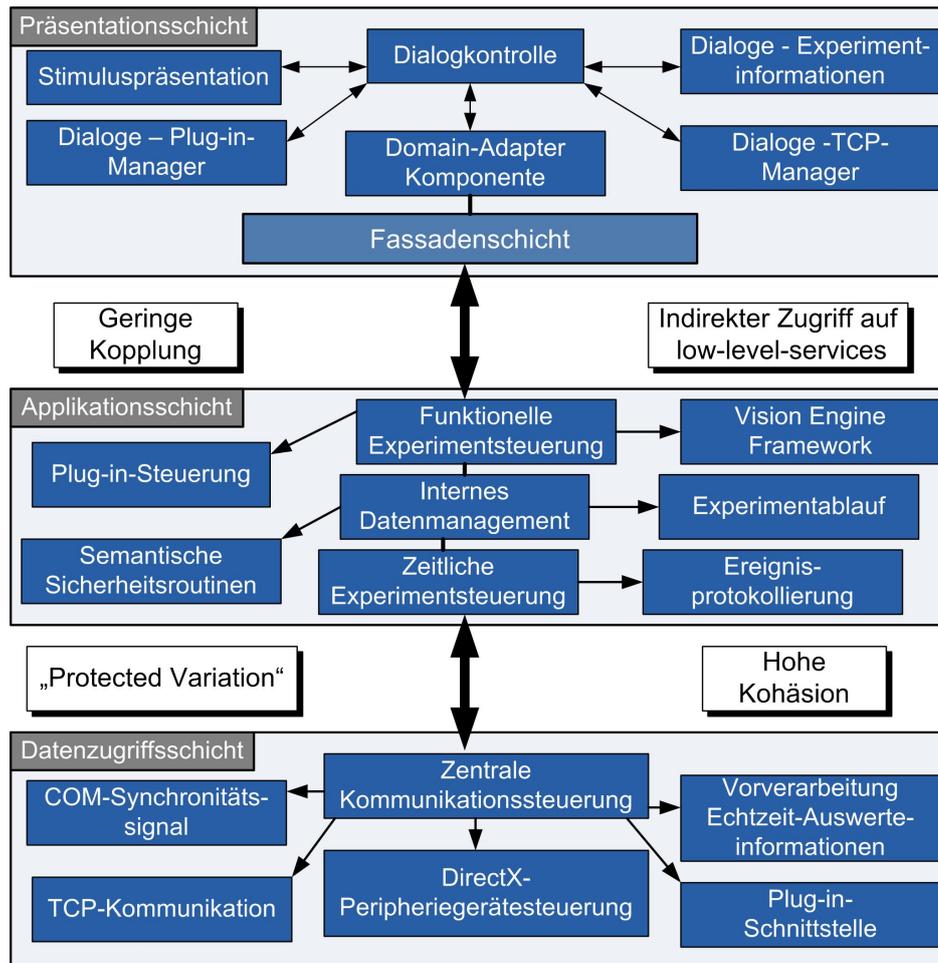


Abbildung 6.3: Ausgehend vom allgemeinen Konzept des Drei-Schichten-Architekturmodells wurde für das Stimulusframework eine erweiterte Form des Modells bestehend aus verschiedenen Entwurfsmustern entwickelt (Mueller et al. 2009). Vorherrschende Konzepte sind das Prinzip der „protected variation“ (Fowler 1997) sowie ein schichtenübergreifender, indirekter „low-level-Services“-Zugriff basierend auf einem strikten Ordnungsprinzip (Larman 2006).

Um schließlich den Anforderungen nach Strukturiertheit, Zuverlässigkeit, und Skalierbarkeit gerecht zu werden, wurden für die Erstellung des Stimulusframeworks verschiedene Modelle aus dem Bereich der Softwarearchitekturplanung herangezogen (Tsichritzis und Klug 1978, Fowler 1997, Coulouris et al. 2003, Larman 2006). Eine erweiterte *Drei-Schichten-Architektur* mit *striker Ordnung* wurde aufgrund der spezifischen Kommunikationsanforderungen als die effizienteste und praktikabelste Lösung für das Stimulusframework mit all seinen Kernfunktionen evaluiert (Abb. 6.3). Anschließend wurden die verschiedenen Aspekte und Aufgaben der einzelnen Schichten festgelegt, welche sich vor allem am Prinzip der „protected variation“ und dem indirekten Zugriff auf „low-level-services“ orientieren (Mueller et al. 2009). Aufgrund der Komplexität dieses Frameworks wurde auf die Einhaltung der festgelegten Architektur und auf die aspektorientierte Strukturierung der Kernfunktionen besonders großen

Wert gelegt (vgl. Abschnitt 5.1). Die Aspekte und Aufgaben der einzelnen Architekturschichten sind wie folgt definiert:

- Die Benutzeroberfläche der Anwendung und damit die hierarchisch am höchsten angesiedelte Schicht soll sämtliche Präsentationsoberflächen sowie Steuerungs- und Eingabemodalitäten beherbergen. Des Weiteren sollen hier sämtliche Experimentinformationen eingegeben und in einem geeigneten Format abgespeichert werden. Die Eingabeflächen zur Erstellung eines Experimentablaufs sowie Oberflächen zur Steuerung der Stimuluspräsentation, der festgelegten TCP-Kommunikationsfunktion und der Synchronitätssignale zwischen Präsentationsanwendung und Datenakquirierung sollen ebenfalls hier lokalisiert sein (Mueller et al. 2011a). Ausgehend vom *Arch-Modell* der Mensch-Maschine-Interaktion beinhaltet diese Schicht zudem eine Dialogkontrollkomponente mit Funktionen zur syntaktischen Fehlerüberprüfung von Benutzereingaben sowie eine Domain-Adapter-Komponente zur Weitergabe von Informationen an die Applikationslogik (Abb. 6.3). Ausgehend vom Prinzip der „protected variation“ (Fowler 1997, Larman 2006) wurde eine zusätzliche Fassadenschicht zwischen Präsentationsschicht und Applikationsschicht integriert. Diese erlaubt eine geschützte Anpassung vorhandener Klassen oder sogar eine Einbindung neuer Methoden ohne eine direkte, nachhaltige Beeinflussung vorhandener Funktionalitäten. Zudem wurden durch das Konzept der geringen Kopplung zwischen den Schichten und des Aspekts der hohen Kohäsion von Funktionalitäten weitere Entwurfsmuster für eine zukünftige, nachhaltige Erweiterbarkeit der Anwendung integriert (Larman 2006).
- Unterhalb der Benutzeroberfläche soll die Applikations- oder Datenverarbeitungsschicht implementiert werden. Diese stellt den Kern des Stimulusframeworks dar und beinhaltet zudem eine direkte Schnittstelle mit dem *Vision Engine Framework*. Des Weiteren sind hier die definierten Kernfunktionalitäten des internen Datenmanagement, der zeitlichen- und funktionellen Experimentsteuerung, der Ereignisprotokollierung, der Plug-in-Steuerung und der semantischen Fehlerüberprüfung angesiedelt.
- Die hierarchisch am tiefsten liegende Schicht stellt die Kommunikations- oder Datenzugriffsschicht dar. Sie ist eine spezielle Adaption der in der Literatur beschriebenen Datenschicht (Tsichritzis und Klug 1978, Fowler 1997, Tanenbaum et al. 2003). In der Datenzugriffsschicht sollen alle Kommunikationsfunktionen implementiert werden. Das schließt die TCP-basierenden Netzwerkprotokolle zur externen Kommunikation, eine Plug-in-Schnittstelle zur Einbindung benutzerdefinierter Funktionalitäten sowie COM-basierende Schnittstellen zur Kommunikation mit Peripheriegeräten ein.

Basierend auf diesem Architekturmodell konnte im nachfolgenden Entwurfsprozess die Einbettung des *Graphic Engine Frameworks* in die Applikationsschicht modelliert werden. Detaillierte Informationen zu den einzelnen Kernfunktionalitäten des Stimulusframeworks finden sich in Kapitel 7.

6.3 Integration virtueller Szenen mit dem Vision Engine Framework

Zur detailgetreuen Anzeige einer virtuellen Umgebung werden in der Regel zwei Software-Komponenten benötigt. Zum einen das 3D-Modell eines virtuellen Szenarios, welches die eigentliche Umgebung, z.B. geometrische Strukturen und verschiedene Objekte, repräsentiert. Zum anderen ein sogenanntes *Graphic Engine Framework*, welches sämtliche Interaktionen des Probanden mit der virtuellen Umgebung bzw. dem 3D-Modell berechnet. So werden beispielsweise die Charaktere innerhalb der VR-Umgebung und deren Bewegungen, das physikalische Verhalten sowie das Simulieren von visuellen Reizen und Geräuschen durch ein *Graphic Engine Framework* übernommen. Einen fundierten Einblick in diese Thematik lieferte bereits Kapitel 4.

Die Einbindung eines flexiblen und erweiterbaren *Graphic Engine Frameworks* und die Bereitstellung von mehreren 3D-Szenarien soll eine schnelle und einfache Modellierung von neurowissenschaftlichen Experimenten ermöglichen. Anders als in der Vergangenheit soll dadurch das eigentliche Forschungsziel von VR-fMRT-Studien wieder stärker in den Vordergrund rücken und der Modellierungsprozess des Experiments nicht mehr durch die Restriktionen eines statischen, unveränderbaren 3D-Modells aus einem Computerspiel geprägt sein.

Ohne die Schaffung eines übergeordneten Stimulusframeworks zur Koordination und zur Kontrolle von Experimentabläufe sind diese Präsentationsfunktionen eines *Graphic Engine Frameworks* für fMRT- und Echtzeit-fMRT-Studien nicht nutzbar (Muller et al. 2009; 2012a;b).

Das *Havok Inc.*, *Vision Engine Framework*, welches zur Erstellung von professionellen Computerspielen benutzt wird, soll im Rahmen dieses Projektes in das Stimulusframework integriert werden und die Anzeige und Simulation von 3D-Szenarien übernehmen (Havok Inc. 2012).

Es stellt als klassisches *Graphic Engine Framework* bereits diverse Grundfunktionalitäten und Schnittstellen zur visuellen Darstellung von 3D-Szenen zur Verfügung. Darüber hinaus besitzt die Engine bereits ein allgemeines Kollisionsmanagementframework, welches mit Hilfe von bereichsabhängigen Kollisionsprüfungen Überlappungen von Objekten registriert und diese anschließend mit Kollisionsverarbeitungsroutrinen in die Simulation des Szenarios einbindet.

Die Kollisionsmanagementprozesse müssen in diesem Zusammenhang auf die Repeitionszeit T_R der MR-Bilddatenakquirierung und auf die zeitlichen Abhängigkeiten der Echtzeit-Datenauswertung abgestimmt und erweitert werden. Das *Vision Engine Framework* besitzt ein einfaches Konzept zum Einladen einer 3D-Szene, welches aufgrund der Zielstellung durch ein erweitertes Framework zum Einladen von mehreren Szenen ergänzt wurde.

Die Simulation von realistischen, physikalischen Effekten wird von der Vision Engine nur in begrenztem Maße unterstützt. Aus diesem Grund wurden zusätzliche Routinen des *Nvidia Inc.*, *PhysX* extrahiert, an die Anforderungen von Echtzeit-fMRT-Studien angepasst und in die VR-Stimulusapplikation eingebunden.

Die zuverlässige und strukturierte Datenhaltung der zahlreichen Experimentinforma-

tionen gilt als Eckpfeiler für die Berechnung des Experimentablaufs sowie die sichere Präsentation der einzelnen VR-Stimuli und soll im Folgenden näher erläutert werden.

6.4 Interne Datenhaltung von Experimentinformationen

Ein Ziel bei der Realisierung der VR-Stimulusanwendung war die effiziente und anwenderorientierte Mensch-Maschine-Interaktion sowie eine nachfolgende, fehlerresistente und sichere Verteilung von Experimentinformationen.

Vor Beginn eines jeden neurowissenschaftlichen Experiments steht zunächst die zeitaufwendige Modellierung eines Experimentdesigns sowie die Eingabe zahlreicher Experimentinformationen im Vordergrund. Experimentinformationen können dabei verschiedene Aspekte des Experiments darstellen. Sie können technische Messparameter zur Beschreibung von Datenakquirierungsmethoden, Probandeninformationen, Informationen zum experimentellen Design oder allgemeine Experimenteinstellungen repräsentieren. Um diese Informationen für die nachfolgenden Berechnungen des Experimentablaufs strukturiert abzuspeichern, muss eine Datenstruktur entwickelt werden, welche durch Hinzunahme oder durch den Wegfall von Parametern trotzdem eine wohldefinierte Struktur behält. Des Weiteren sind Experimentinformationen nicht durch eine feste Anzahl von Parametern definiert, sondern erfordern ein flexibles, skalierbares Datenhaltungskonzept.

Um all diesen Anforderungen gerecht zu werden, wurde ein skalierbares Informationsprotokoll zur internen Repräsentation von Experimentinformationen mit verschiedenen Sektionen konzipiert, welches ebenfalls die Verbreitung von Informationen zwischen den Architekturschichten unterstützt.

Durch die Modellierung komplexerer Experimentdesigns oder durch die Integration neuer Funktionalitäten in ein Stimulus-Framework können Komplexität und die Anzahl der Parameter steigen oder sinken (rot markierte Sektion *Paradigmenstruktur*, Abb. 6.4).

Zusätzlich dazu müssen Relationen oder semantische Kopplungen zwischen einzelnen Experimentaspekten bzw. einzelnen Messparametern berücksichtigt werden. Viele Informationsmengen sind miteinander verknüpft und können nicht ohne weiteres getrennt betrachtet werden. Als Beispiel sei der MRT-spezifische Parameter „Repetitionszeit“ (T_R) genannt, welcher gleichzeitig als physikalischer Parameter und als Informationsträger für das Experimentdesign fungiert (Punkt 1, Abb. 6.4).

Ebenso können Informationen aus der Sektion *Neurofeedback* und der Sektion *Szenenanpassung* in Relation zueinander stehen und nur gemeinsam zur Anpassung von VR-Szenarien genutzt werden (Punkt 2, Abb. 6.4).

Nach der Eingabe der Experimentinformationen auf der Benutzeroberfläche werden diese zunächst abhängig von der Art der Information innerhalb der entsprechenden Sektion des *Informationsprotokolls* abgespeichert. Angedacht war hier eine listenabhängige Speicherung von Informationen. Dies bedeutet, dass alle Informationssektionen eine festdefinierte Position im Informationsprotokoll einnehmen und das Parame-

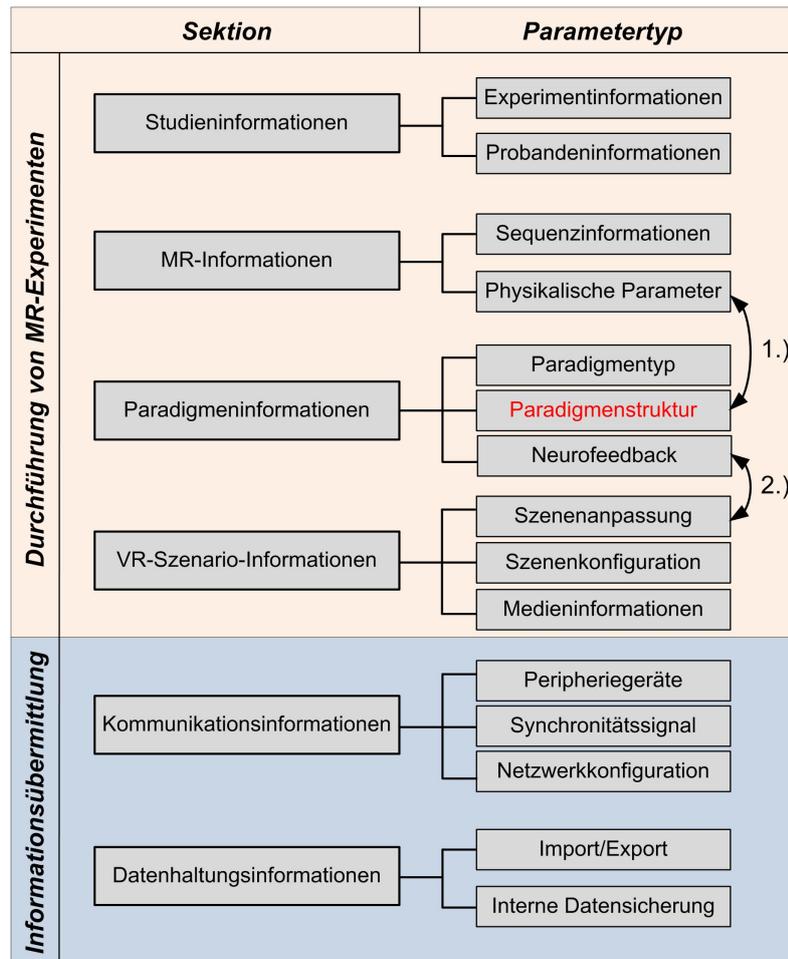


Abbildung 6.4: Das Konzept zur strukturierten Speicherung von Experimentinformationen dient als Basis für die Implementierung des Informationsprotokolls. Es besteht aus insgesamt sechs Sektionen, welche abhängig vom jeweiligen Aspekt die jeweiligen Parameterwerte des experimentellen Setups aufnehmen, z.B. T_R -Parameterwert in Sektion *Physikalische Parameter*. Zudem bilden Parameter unterschiedlicher Sektionen semantische Kopplungen (Punkt 1 u. 2), welche im Text näher erläutert werden.

terwerte innerhalb einer skalierbaren, objektorientierten Datenstruktur ablegt werden. Des Weiteren muss das Informationsprotokoll die Speicherung von numerischen Werten und Zeichenketten unterstützen und sollte zudem eine matrixbasierte Abspeicherung von Informationen ermöglichen.

Die Verarbeitung von Experimentinformationen ist nach streng aspektorientierten Gesichtspunkten ausschließlich für die darunterliegende Applikationsschicht angedacht und soll die eigentliche Präsentation sowie die funktionelle und zeitliche Steuerung des Experiments ermöglichen.

Eine Möglichkeit zur effizienten und variablen Speicherung von Parametern innerhalb des Informationsprotokolls bieten abstrakte Datentypen (ADT). Eine Möglichkeit wäre hier die Verwendung des klassischen *Variant-Datentyps*, der im Prinzip alle anderen Datentypen in sich aufnehmen kann (Blanchette und Summerfield 2009). Zusätzlich

dazu gehen sämtliche Informationen zum ursprünglichen Datentyp nicht verloren. So können im Hinblick auf die Datenhaltung sämtliche Parameternamen und ihre zugehörigen Werte in strukturierter und reproduzierbarer Form abgespeichert werden. Eine Kapselung von Informationen wird ebenfalls ermöglicht und gestattet eine flexible, übergeordnete Speicherung von abstrakten Datentypen in Listenform.

Da mit wachsender Komplexität der Experimentabläufe und der Experimentinformationen auch gleichzeitig die Fehleranfälligkeit der Datenzugriffe steigt, wurde für die fehlerresistente Verteilung von Informationen ein Konzept zur Absicherung von Informationen entworfen. Dieses Konzept setzt auf die sogenannte *Schnappschussreplikation* einer lokalen Replik (Blanchette und Summerfield 2009). Nur neue Informationen, die vom Anwender zeitnah eingegeben wurden, sollen innerhalb eines kumulativen Speicherprozesses der lokalen Replik hinzugefügt werden. Auf diese Weise werden unnötige Speicherzugriffe vermieden und die Performanz der Gesamtanwendung bleibt auf einer konstanten Stufe.

6.5 Konzept zur Echtzeit-Stimulusadaption von VR-Paradigmen

Ein weiterer Punkt der Zielstellung war die Erarbeitung eines neuen Ansatzes für eine adaptive VR-Stimuluspräsentation sowie die Einbindung von Neurofeedback-Techniken zur Anpassung von komplexen 3D-Umgebungen bzw. 3D-Geometrien. Dieser neue und zudem technisch anspruchsvolle Ansatz sollte erstmalig im Rahmen dieses Projektes entwickelt werden.

Das Ziel dieses Ansatzes ist die Optimierung von Neurofeedback-Trainingsmethoden durch eine fortgeschrittenere Stimuluspräsentation, die in erster Linie die intrinsische Motivation der Probanden steigert. Des Weiteren sind durch den Interaktivitätsfaktor und durch die Komplexität von VR-Stimulusreizen erweiterte Stimulusanpassungen möglich, welche ein gleichzeitiges Neurofeedback-Training über mehrere Hirnareale ermöglichen. Der Proband trainiert auf diese Weise nicht mehr nur seine neuronale Aktivität in einem Hirnareal, sondern bekommt die neuronalen Aktivierungen aus verschiedenen Arealen seines Gehirns angezeigt und lernt diese durch operante Konditionierung zu kontrollieren. Als Rückmeldung an den Probanden werden bei erfolgreicher Kontrolle der Hirnaktivität visuelle Reizadaptionen vorgenommen.

Um dieses Ziel zu erreichen, müssen zunächst reguläre Neurofeedback-Konzepte überdacht und neue Konzepte zur dynamischen Adaption von VR-Stimuli entwickelt werden (Mueller et al. 2012b).

Die neurophysiologischen Unterschiede zwischen den Probanden spielen dabei eine besondere Rolle. Im Mittelpunkt der Problematik steht hier die Varianz in der neurophysiologischen Antwort zwischen verschiedenen Probanden. Für das Neurofeedback bedeutet dies, dass sich die Lokalisation und die Charakteristiken der neuronalen Aktivierung, wie z.B. die Ausdehnung der neuronalen Aktivierung, die gemittelte Signalstärke der Aktivierung und die Signaländerungen von Ruhe-Phasen zu Aktiv-Phasen, für jede einzelne Versuchsperson unterscheiden. Hinzukommt, dass Versuchspersonen

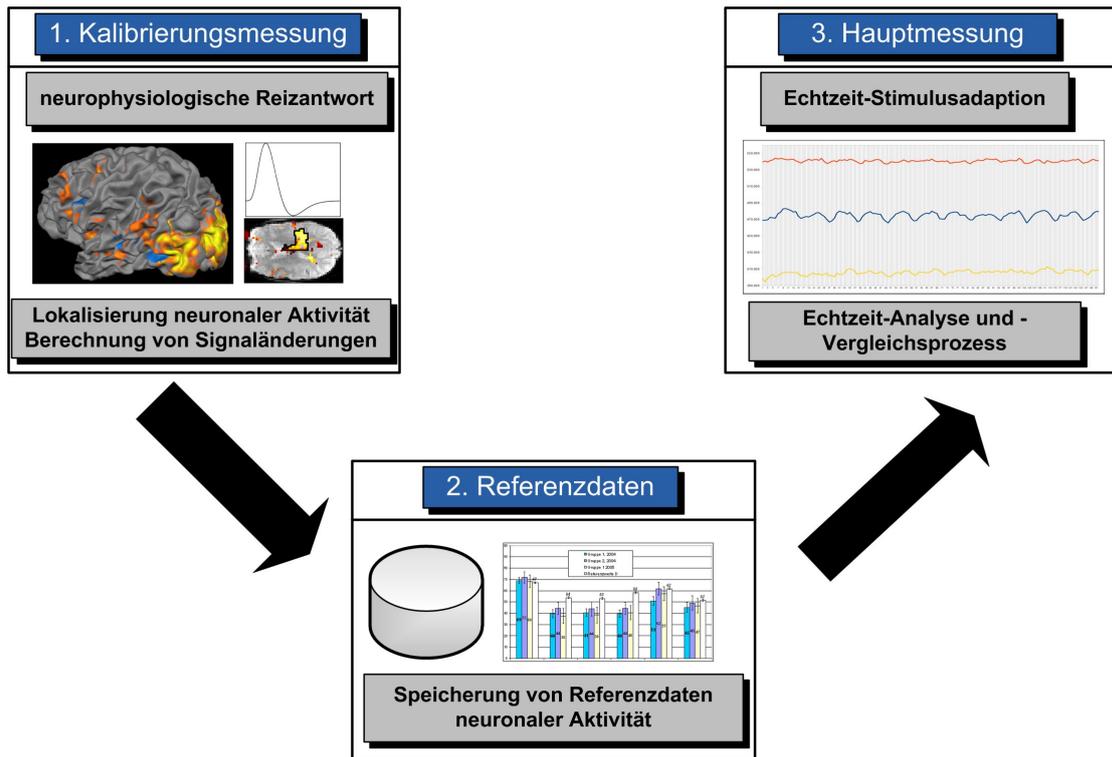


Abbildung 6.5: Das Konzept zur Echtzeit-Anpassung von VR-Stimuli baut zunächst auf einer Kalibrierungsmessung (1) auf. Aufgrund der Varianz in der neurophysiologischen Antwort verschiedener Probanden werden hier die neuronale Aktivierung registriert sowie repräsentative Signaländerungen berechnet. Diese Werte (Referenzdaten) werden abgespeichert (2) und in der anschließenden Hauptmessung (3) zur eigentlichen Echtzeit-Anpassung des VR-Stimulus verwendet (Mueller et al. 2012a).

meist über einen längeren Zeitraum und in mehreren Sitzungen ein Neurofeedback-Training durchführen. Treten im Training des Probanden Über- oder Unterforderungen auf, so ist auch eine zeitnahe Adaption der Stimuluspräsentation zur Optimierung des Trainings nur eingeschränkt möglich. Im Verlauf des Neurofeedback-Trainings kann dieser Effekt beispielsweise zu einer Stagnation der Trainingserfolge führen (Rief und Birbaumer 2006, Mueller et al. 2011a). Diesem Effekt könnte entgegengewirkt werden, wenn sich die VR-Umgebung so verändert, dass der Proband die volle Reizantwort in seinem kognitiven System entwickelt. Eine weitere potentielle Anwendung wäre die Beeinflussung der selektiven Aufmerksamkeit (Salienz) durch die zeitnahe und dynamische Einbindung von Objekten oder Strukturen. Die Adaption von VR-Stimuli erfordert, dass basierend auf dem Neurofeedback-Prinzip selektive Informationen aus dem Neurofeedback-Signal herausgefiltert und anschließend zur verbesserten Interaktion des Probanden mit der virtuellen Umgebung verwendet werden (Mueller et al. 2012a). In Abhängigkeit der genutzten Echtzeit-Datenauswerteanwendung können so Auswertinformationen aus einer bzw. aus mehreren Hirnregionen zeitnah über Netzwerkprotokolle versendet werden. Wichtig für diesen Teil des Konzeptes ist die Skalierbarkeit und die flexible Festlegung von Indikatoren der Aktivierung. Nutzer sollen

nicht auf die Verwendung von spezifischen Auswertinformationen beschränkt bleiben, sondern auch einfache oder komplexere Auswerteparameter als Indikator nutzen können (Mueller et al. 2012b).

Dieser Ansatz setzt ein Adaptionkonzept voraus, welches durch einen modellierten Algorithmus mit drei Prozessphasen repräsentiert wird (Abb. 6.5). Hier wird zunächst eine Kalibrierungsmessung durchgeführt, bei der sogenannte Referenzdaten der neuronalen Aktivierung für den jeweiligen Probanden ermittelt werden. Im Speziellen werden innerhalb dieser Prozessphase die neuronale Aktivität lokalisiert und repräsentative Signaländerungen (Referenzdaten) für alle Experimentphasen und alle definierten Hirnregionen berechnet. In der zweiten Prozessphase werden diese Referenzdaten innerhalb eines Datenkonstrukts strukturiert abgespeichert und schließlich in der dritten Phase für eine Echtzeit-Analyse sowie einen Echtzeit-Vergleichsprozess mit aktuellen Werten des Probanden verwendet. Dieser benutzerdefinierte Vergleichsprozess entscheidet dann automatisch über die zeitnahe Adaption des VR-Stimulus (Mueller et al. 2012a).

Eine Möglichkeit zur Ermittlung von repräsentativen Signaländerungen wäre beispielsweise die Berechnung von statistischen Auswerteparameter mittels eines rekursiven Allgemeinen Linearen Modells. Dies setzt vor allem Interaktivität und hohe Bewegungsfreiheiten innerhalb der Virtuellen Umgebung voraus. Das Ergebnis wäre aber, zumindest theoretisch, ein statistisches Auswertesystem, welches die Länge der einzelnen Phasen bzw. Bedingungen dynamisch und selbstständig festlegt und so die vorherige zeitliche Definition von Versuchsbedingungen überflüssig macht.

Eine weitere Lösung wäre die Berechnung der Signaländerungen, welche einen möglichen Indikator für die Stärke der neuronalen Aktivität liefern können. Obwohl dieses Prinzip mit der statistischen Signifikanz des Allgemeinen Linearen Modells nicht zu vergleichen ist, verspricht es doch eine effektive Berechnung von repräsentativen Signaländerungen und damit eine bessere Performanz bei der Durchführung zeitkritischer Echtzeit-Prozesse. Neben der statistischen Auswertung von neuronalen Aktivierungen spielt die Interpretation dieser Informationen für eine mögliche Neurofeedback-Präsentation eine weitere große Rolle. Um hier das Neurofeedback-Training über mehrere definierte regions of interest zu gewährleisten, bedarf es der Entwicklung eines speziellen Kommunikationsprotokolls, welches die Neurofeedback-Signalverläufe aller definierten Regionen gleichzeitig und in Echtzeit zum Adaptionalgorithmus übermitteln kann.

Alle bei der Stimulusapplikation eintreffenden Auswertinformationen müssen nach definierten Gesichtspunkten interpretiert, an zuständige Funktionalitäten weitergeleitet und auf regelbasierte Algorithmen angewendet werden. Langfristig sind hier komplexere mathematische Strukturen aus dem Bereich der Musterklassifikation und des maschinellen Lernens denkbar. Jedoch müssen hierfür erst Referenzdaten aufgezeichnet und extrahiert werden, welche dann zur Implementierung eines solchen Konzeptes genutzt werden können. Weniger aufwendige Ansätze, wie z.B. der Einsatz von mathematischen Fallunterscheidungen, stellen derzeit für die Evaluation bestehender Echtzeit-Adaptionkonzepte eine sinnvollere Alternative dar. Ein Entwurf zu den Prozessabläufen der Auswertinformationen und zur Informationsinterpretation findet sich in Abbildung 6.6.

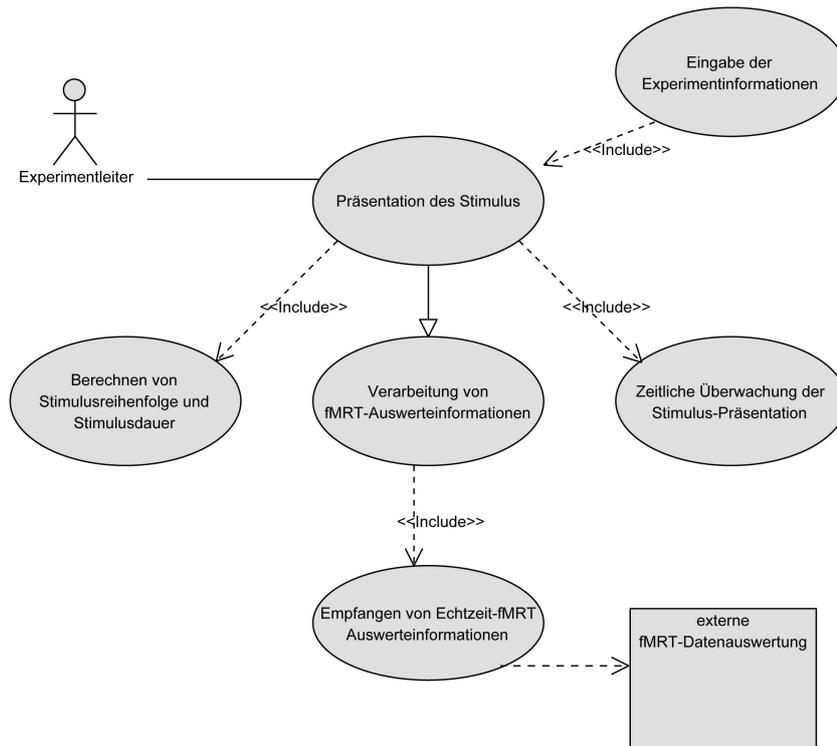


Abbildung 6.6: Das Anwendungsdiagramm zeigt den Entwurf zur Umsetzung der Echtzeit-Anpassung von VR-Stimuli durch Neurofeedback-Techniken auf der VR-Präsentationsseite. Auswertinformationen werden zunächst von einer externen Echtzeit-fMRT-Datenauswerteanwendung empfangen (unten rechts) und anschließend innerhalb der Signalanalyse und -interpretation weiterverarbeitet. Die Echtzeit-Anpassung von VR-Stimuli wird dabei abhängig von den Ergebnissen der Signalinterpretation ausgelöst (Mueller et al. 2012a).

Weiterhin können Informationen zu den probandenspezifischen neuronalen Aktivierungen protokolliert und zur Optimierung eines möglichen Neurofeedback-Trainings genutzt werden. Für dieses Gesamtkonzept und im Speziellen für die Übermittlung von Auswertinformationen ist die Integrierung eines eigens entwickelten, bidirektionalen Kommunikationsprotokolls vorgesehen, welches Informationen sowohl von der Datenauswerteseite als auch von der Stimulusseite flexibel und strukturiert übertragen kann (Luehrs et al. 2011, Mueller et al. 2011a; 2012b). Zusammen mit den hier beschriebenen Einzelkonzepten kann so ein Gesamtkonzept zur Echtzeit-Adaptierung durch klassische Neurofeedback-Techniken realisiert und evaluiert werden.

6.6 Verbreitung und Speicherung von Experimentinformationen

Um die geplanten Funktionen der syntaktischen und semantischen Überprüfung von Experimentparametern zu entlasten, wurde die Einführung eines Konzeptes zur siche-

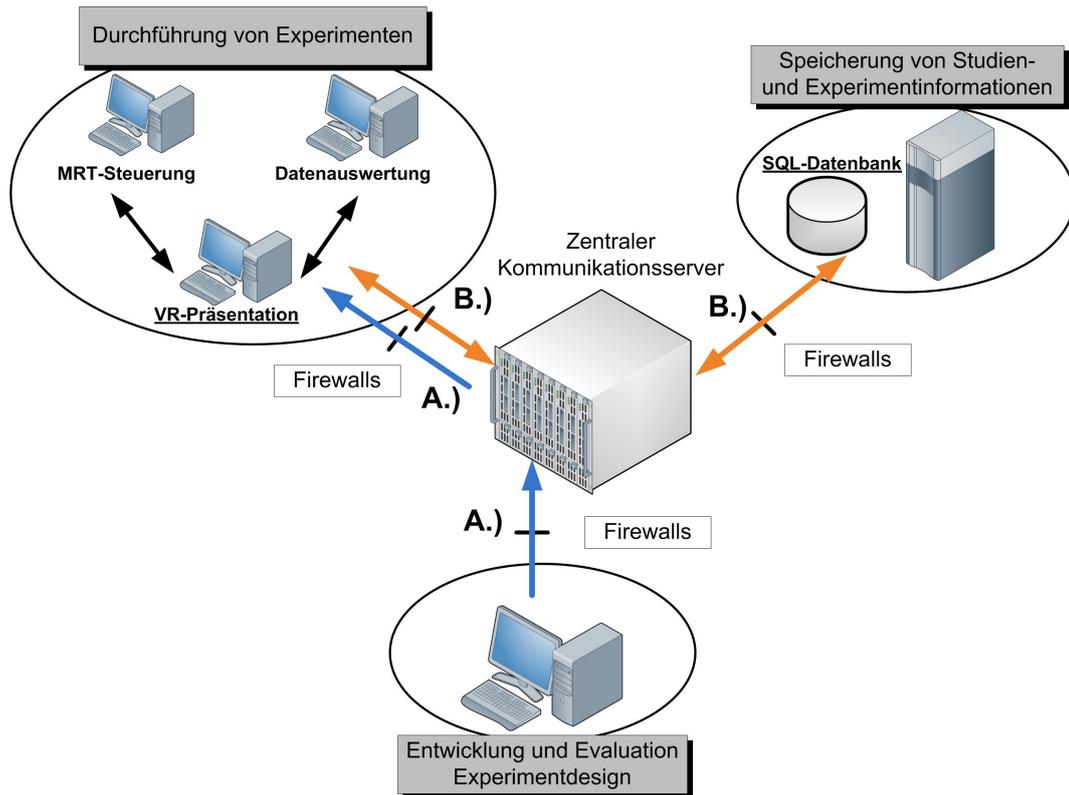


Abbildung 6.7: Zwei Aspekte stehen beim Konzept zur Verbreitung von Experimentinformationen im Vordergrund. Die zentrale Entwicklung und Evaluation eines Experimentdesigns mit anschließender unidirektionaler Übermittlung zur Infrastruktur eines MRT-Systems (blaue Pfeile) und die zentrale Speicherung sowie das Abrufen von probandenspezifischen Konditionierungsprofilen für Neurofeedback-Experimente (orange Pfeile) (Müller et al. 2012a).

ren prä-experimentellen Modellierung des VR-Experimentablaufs sowie zur Verteilung aller Experimentinformationen an sämtliche beteiligten Komponenten entworfen. Eine eingehende Beschreibung dieser Problematik lieferte bereits Kapitel 2.2.2.

Für den Entwurf eines solchen Konzepts steht zunächst die Planung entsprechender Kommunikationswege und -protokolle im Mittelpunkt. Aufgrund der durchgeführten Analyse des *rtExplorer-Frameworks* (vgl. Abschnitt 5.4.3) und aufgrund von Erfahrungen aus Vorexperimenten entschieden wir uns für die Modellierung und die Implementierung eines neuen Ansatzes auf Basis eines TCP-basierten Netzwerkprotokolls für fMRT-Studien (Luehrs et al. 2011). Dadurch ist es möglich, dass der Versuchsleiter zunächst das Experiment mittels des prä-experimentellen Modellierungsprozesses am eigenen Computer erstellt und es anschließend über TCP an die jeweiligen Komponenten des Akquirierungssystems übermittelt (blaue Pfeile, Abb. 6.7).

Ein Nachteil bei der Verwendung dieses TCP-Konzeptes ist die notwendige Verfügbarkeit einer Netzwerkverbindung. Da ein Ausfall der Netzwerkverbindung auch das Einladen des experimentellen Setups verhindert, wurde für dieses Konzept eine weitere Möglichkeit zur Verbreitung von Experimentinformationen angedacht.

Um die spätere Nutzerakzeptanz der Anwendung zu fördern und den Experimentablauf sowohl für die Durchführung als auch für die anschließende statistische Offline-Auswertung zu verwenden, sollte die Abspeicherung und das Einladen von Experimenteinstellungen über eine BrainVoyager kompatible Protokoll-Datei (PRT-Datei) ermöglicht werden (vgl. Abschnitt 5.4.4).

Dies ermöglicht eine sichere prä-experimentelle Planung und Simulation des Versuchs sowie den reibungslosen Ablauf einer Probandenmessung. Zusätzlich dazu bietet die weitverbreitete Offline-Datenauswerteanwendung *BrainVoyager QX* die Möglichkeit des Import/Export von PRT-Datenformaten und gewährleistet damit ein einmaliges Erstellen des Experimentablaufs innerhalb eines Datenformats (Mueller et al. 2010b). Das Konzept zur Speicherung mittels PRT-Datei fördert die Nutzerakzeptanz der VR-Stimulusanwendung und greift auf bereits bekannte Mechanismen der Experimentmodellierung zurück.

Neben der Verbreitung von Experimentinformationen wurden im Rahmen dieses Modellierungsprozesses auch Prozesse zur pseudonymisierten Nutzung und Sicherung von probandenspezifischen Konditionierungsprofilen in Neurofeedback-Experimenten entworfen (Mueller et al. 2012a).

Zur Optimierung des Neurofeedback-Trainings sollen probandenspezifische Konditionierungsprofile die spezifischen neurophysiologischen Besonderheiten und die Trainingserfolge des Probanden erfassen und zur schrittweisen Verbesserung im Konditionierungsprozess beitragen (Mueller et al. 2012a). Dies erfordert eine längerfristige, kumulative und sichere Speicherung von immer neuen Trainingsdaten innerhalb einer gesicherten Datenbank und ein Abrufen der letzten erfassten Konditionierungsdaten vor einer neuen Versuchsreihe (orange Pfeile, Abb. 6.7).

Für die Realisierung eines TCP-basierten Übertragungsweges mussten zunächst einige Aspekte berücksichtigt werden. TCP-Prozesse unterteilen den Stream in eine Folge von Datensegmenten. Jedem Segment wird dabei eine aufsteigende Nummer zugeordnet. Stimmen die vorhergehende Nummer und die Folgenummer nicht überein, können Informationen solange im Puffer zurückgehalten werden bis alle notwendigen Segmente eingetroffen sind (Tanenbaum et al. 2003). Erst dann erfolgt eine Platzierung der Daten im Eingabe-Stream des empfangenden Prozesses. Bei Verwendung des TCP-Protokolls als Übertragungsmedium für Echtzeit-Datenauswerteeinformationen besteht daher die Gefahr einer zeitlichen Verzögerung, die das Experiment potentiell gefährden würde. Um dies auszuschließen, wurde das lokale Netzwerk der MRT-Infrastruktur erfolgreich auf eine ausreichend hohe Übertragungsgeschwindigkeit überprüft (Mueller et al. 2009).

Zusätzlich dazu sollten bei der Implementierung des Übertragungsprotokolls die Prozesse zur Informationsextraktion aus dem Stream an die zeitkritischen Vorgaben der Echtzeit-Datenauswertung angepasst werden. Die Besonderheiten der VR-Stimuluspräsentation und der Protokollierung von VR-Ereignissen wurden ebenso berücksichtigt und für die Realisierung eingeplant.

6.7 Verwendete Softwaretechnologien

Hier sollen kurz die verwendeten Softwaretechnologien von Drittanbietern vorgestellt werden. Die Einbindung von Toolkits stellt einen wichtigen Punkt bei der Umsetzung von komplexen Softwareprojekten dar und dient vor allem zur Auslagerung von Funktionalitäten. Alle vorgestellten Frameworks wurden entweder zur Modellierung von Virtuellen Umgebungen verwendet, z.B. Autodesk 3ds Max, oder haben Teilfunktionalitäten in das vorliegende Softwareprojekt eingebracht. Diese wurden während des Implementierungsprozesses an die Anforderungen von fMRT- und Echtzeit-fMRT-Studien angepasst.

6.7.1 Havok Vision Engine Framework

Die kommerzielle *Vision 3D Engine* ist eine von der Firma *Havok Inc.* vertriebene 3D-Grafik-Engine für Windows, Xbox 360, PlayStation 3, Nintendo Wii, NGP und diverse Webbrowser. Die Engine basiert auf einem C++ Framework zum Rendern von komplexen und aufwendigen 3D-Szenen. Sie unterstützt die Einbettung in externe Applikationen über eine C++ Schnittstelle (API) und liefert damit eine Anzahl von integrierbaren Modulen (Havok Inc. 2012). Mit der Einbettung kann das Engine Framework komplett über die übergeordnete Applikation gesteuert werden. 3D-Szenen können durch vorherige Bearbeitung mit *Autodesk 3ds Max* in das firmeneigene Datenformat *VSCE-NE* konvertiert und in das *Vision Engine Framework* eingebunden werden. Objekte, Charaktere und Kameras können mit sogenannten „Entitäten“ verknüpft und darüber gezielt angesprochen oder manipuliert werden. Dies erlaubt die adaptive Änderung der virtuellen Umgebung in Echtzeit sowie die Einbindung von Programmerroutinen zur Simulation künstlichen Intelligenzverhaltens.

Die Engine unterstützt ebenso diverse Beleuchtungsmodelle, Partikelsysteme und Animationen. Weiterhin können über eine spezielle Schnittstelle eine Reihe von meist lizenzpflichtigen Drittanbieter Plug-ins eingebunden werden. Als Beispiel seien das *Nvidia PhysX Engine Plug-in* zum Einbinden physikalischer Effekte, das *Havok Physics Plug-in* ebenfalls zur Kollisionsdetektion und Einbinden physikalischer Effekte, das *IKinema Plug-in* zur Steuerung und Animation virtueller Charaktere, das *Kynapse AI Plug-in* zur Implementierung künstlicher Intelligenz oder *Scaleform Gfx* zur Einbettung von Video-Dateien, genannt (Havok Inc. 2012). Für die Verwendung des *Vision Engine Frameworks* im Rahmen des vorliegenden Projektes müssen zudem die Visualisierungsprozesskette sowie Funktionen des Echtzeit-Renderings an die Anforderungen von Echtzeit-fMRT-Studien angepasst werden. Des Weiteren wurden eigene Klassen für die zeitliche und funktionelle Steuerung von Experimenten, für die Ereignisprotokollierung von Stimuli und zur Echtzeit-Adaption von VR-Stimuli über Auswertinformationen zur direkten Kommunikation mit der Vision Engine Prozesskette entworfen und integriert.

Es sei darauf hingewiesen, dass die *Vision 3D Engine* nur die Einbindung des firmeneigenen *VSCENE* Dateiformats unterstützt. Dementsprechend können virtuelle Szenen oder Umgebungen nur in diesem Format in das *Vision Engine Framework* eingebunden werden. Aufgrund dieser Restriktion ist die unkomplizierte Einbindung von freien

Szenen nicht ohne weiteres möglich und kann nur durch vorherige Konvertierungsprozesse erreicht werden. Für die Verwendung der *Vision 3D Engine* wurde dem Institut für Biometrie und Medizinische Informatik, Medizinische Fakultät, Universität Magdeburg, eine Lizenz zur wissenschaftlichen Grundlagenforschung bereitgestellt.

6.7.2 Nvidia PhysX

Das Framework *PhysX* von der Firma *Nvidia Corp.* steht für einen sogenannten Physikbeschleuniger oder auch eine „Physics Processing Unit“. Dieses Framework wird hauptsächlich als sogenannte Physik-Engine in diversen Computerspielen eingesetzt. Die Aufgabe des Frameworks besteht vorrangig in der Berechnung physikalischer Effekte oder der Kollisionsdetektion in dreidimensionalen Umgebungen. Die *PhysX-Engine* ist weitestgehend in das *Nvidia CUDA-System* integriert. Dies erlaubt die vollständige Berechnung der Physik mit der „Graphical Processing Unit“ (GPU), was im Endeffekt zusätzliche Physikbeschleuniger optional macht. *Nvidia PhysX* ist weitestgehend über Plug-ins in die *Vision 3D Engine* eingebunden und erlaubt so die Simulation von physikalischen Effekten innerhalb einer 3D Vision Engine Szenenpräsentation (NVIDIA Corp. 2011).

Für den Einsatz in Echtzeit-fMRT-Experimenten wurden zudem mehrere Optimierungen innerhalb der Kollisionsprozesskette vorgenommen. Diese beziehen sich vordergründig auf Anpassungen und Abstimmungen innerhalb der Prozesskette zur Echtzeit-Adaption von VR-Paradigmen.

6.7.3 Nokia Qt - User Interface Framework

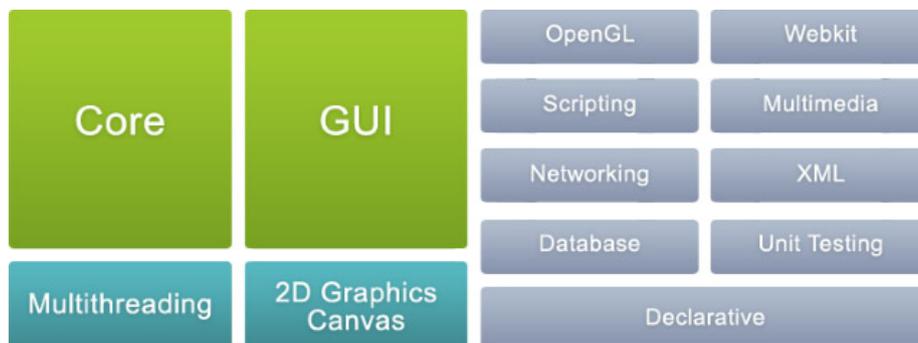


Abbildung 6.8: Die Abbildung zeigt die verschiedenen Module sowie Klassenbibliotheken des *Qt*-Frameworks (Nokia Corp. 2011).

Mit dem C++-Framework *Qt*, ausgesprochen engl. „cute“, von der Firma Nokia Corp. wurde ein Werkzeug zur Entwicklung plattformübergreifender Benutzeroberflächen verwendet (Nokia Corp. 2011). Das Framework besitzt im Wesentlichen zwei Lizenzsysteme. Zum einen eine Lizenz unter der *GNU General Public License* (GPL) und eine proprietäre Lizenz zur Entwicklung kommerzieller Software. *Qt* ist ein umfangreiches, modular aufgebautes Framework, welches eine Vielzahl unterschiedlicher Funk-

tionen bietet. Dies schließt sowohl die Modellierung von grafischen Benutzerschnittstellen als auch die Programmierung von Netzwerkkommunikationsroutinen ein (Blanchette und Summerfield 2009). Eine Übersicht zu allen Qt-Modulen kann in Abbildung 6.8 eingesehen werden. Für eine vollständige Auflistung aller im *Qt-Framework* bereitgestellten Funktionalitäten sei auf die Webseite des Herstellers verwiesen (Nokia Corp. 2011).

6.7.4 Autodesk 3ds Max

Die kommerzielle Anwendung *3ds Max* von der Firma *Autodesk GmbH* ist ein umfangreiches 3D-Computergrafik- und Animationsprogramm, welches vor allem zur Modellierung von dreidimensionalen Objekten und geometrischen Strukturen verwendet wird. Es findet seinen Einsatz in den Bereichen Design, Architektur und bei der Erstellung von Computerspielen. Überdies unterstützt *3ds Max* die Erstellung von Charakteranimationen, die Modellierung von Polygon- und mathematischen Objekten (NURBS) sowie das Rendering und die Darstellung diverser Partikelsysteme. Weitere Funktionalitäten betreffen das Montieren von mehreren Objekten oder Bildern zu einem Gesamtbild (Compositing) sowie diverse Lichtbrechungs- und Beugungseffekte an transparenten Objekten (Raytracing) (Autodesk GmbH 2011). Plug-ins werden ebenso unterstützt und können zur Erweiterung der Funktionalitäten eingebunden werden. Die *Vision 3D Engine* der Firma *Havok Inc.* liefert zudem ein Plug-in für *3ds Max*, welches den Export von 3D-Modellen in das *Vision*-Dateiformat (VSCENE) unterstützt.

Kapitel 7

Realisierung und technische Evaluierung der Applikation

In diesem Kapitel sollen die Realisierung, der Benutzerakzeptanztest sowie die zugehörigen Performanztests der vorliegenden Arbeit beschrieben werden. Zunächst wurden die Klassen der Datenzugriffsschicht implementiert. Im Anschluss folgte, darauf aufbauend, die Realisierung der Applikationsschicht und schließlich die Erstellung der Benutzeroberfläche. Die Integration der Stimulusanwendung in die globale MRT-Infrastruktur erfolgte im finalen Entwicklungsprozess. Aufgrund der plattformspezifischen Einschränkungen des *Vision Engine Frameworks* wurde *Microsoft Visual Studio 2008* (Microsoft Corp. 2011b) als Entwicklungsumgebung ausgewählt. Das Software Development Kit *Qt* und diverse Zusatzsoftware, wie z.B. *PhysX* und *DirectX*, wurden ebenfalls integriert und unterstützen die vorhandene Klassen in ihrer Funktionalität.

7.1 Realisierung der Datenzugriffsschicht

Wie bereits im Entwurfskapitel beschrieben, sollten in der Datenzugriffsschicht alle Funktionalitäten zur Kommunikation mit externen Softwareanwendungen oder Peripheriegeräten implementiert werden. Empfangene Informationen werden hier nicht verarbeitet, sondern ausschließlich zur Applikationsschicht weitergeleitet.

Durch die Aufteilung der Klassen nach den im Entwurf genannten Aspekten kam es zur Erstellung von zwei Hauptsteuerklassen, der *TcpController*-Klasse und der *CommController*-Klasse, sowie von insgesamt fünf Hauptfunktionsklassen (Abb. 7.1). Speziell bei Neurofeedback-Experimenten können simultane Aufrufe der TCP-Klassen und der Tastersteuerung Performanzprobleme innerhalb der Datenzugriffsschicht auslösen. Aus diesem Grund wurde zur Vereinfachung der TCP-Threadstruktur eine Datenzugriffsschicht mit zwei Steuerklassen eingeführt, welche möglichen *bottleneck*-Problemen entgegenwirken soll.

Die erste Klasse, welche vor allem die *DirectX*-basierte Tastersteuerung ermöglicht, ist die Klasse *DirectXInput*. Sie gestattet die Registrierung von Ereignissen, die bei der Betätigung von angeschlossenen Tastern entstehen und gewährleistet so die zeitlich korrekte Aufzeichnung der Ereignisse zur späteren Datenauswertung. Die Funktionalität

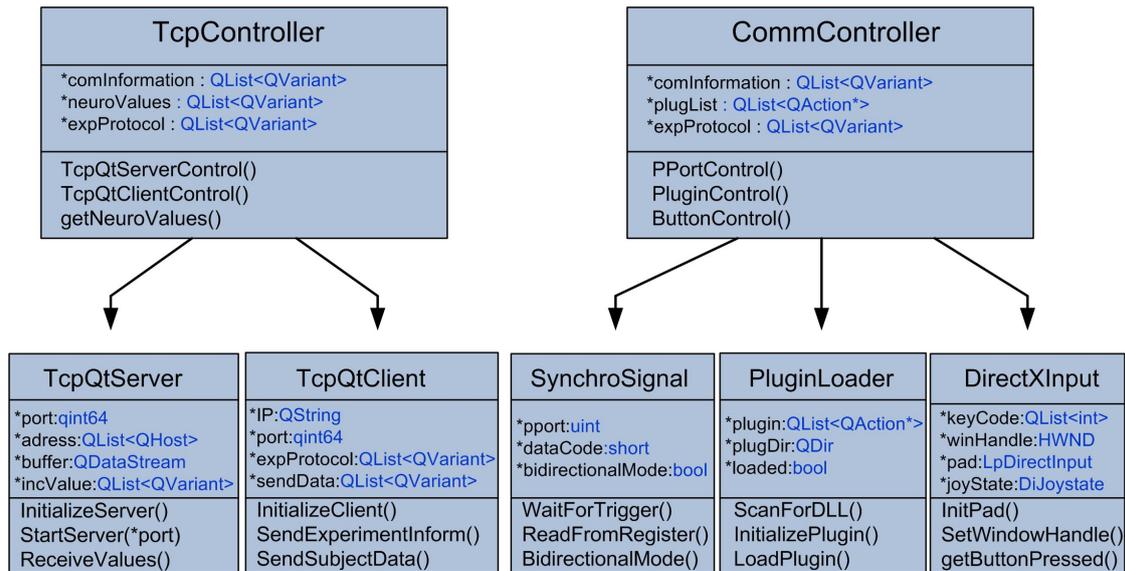


Abbildung 7.1: Die Struktur der Datenzugriffsschicht baut auf zwei Steuerklassen und fünf Hauptfunktionsklassen auf. Eine genauere Beschreibung der einzelnen Funktionalitäten findet sich im Text.

lität dieser Klasse baut zudem auf dem *DirectX Software Development Kit 9.29.1962* auf, welches in das Softwareprojekt eingebunden wurde (Microsoft Corp. 2011a). Die Eingabe mittels Tastern dient als Standardmethode zur Interaktion der Probanden mit der künstlichen Umgebung. Für die Verwendung der Taster-Funktionalität ist die Installation der *DirectX-Endbenutzer-Runtime* notwendig.

Die zweite Klasse in dieser Schicht ist die Klasse *SynchroSignal*. Sie dient der direkten Kommunikation zwischen VR-Stimulanwendung und dem Steuerrechner des Magnetresonanztomographen.

Zur Gewährleistung einer fehlerfreien Auswertung der fMRT-Bilddaten ist die Synchronisation zwischen Stimuluspräsentation und MRT-Messesequenz essentiell. Hier bietet das Steuer- oder Synchronisationssignal, welches beim Starten der Messesequenz vom Host-PC des MRT ausgesendet wird, eine Möglichkeit alle verbundenen Applikationen mit der Messesequenz zu synchronisieren. Speziell zur Synchronisation der Messesequenz mit der VR-Stimulanwendung orientierte man sich an den Gegebenheiten der lokalen MRT-Infrastruktur und realisierte ein Empfangsmodul zum Aufzeichnen dieses sogenannten Triggersignals über die parallele LPT-Schnittstelle des Rechners. Als Basisschnittstelle zum Ansprechen des LPT-Ports diente die freie Software *InpOut32*.

Die Inpout32-Bibliothek wurde dazu in das Projekt eingebunden und über die implementierte Steuerungsklasse des Synchronisationssignals *SynchroSignal* angesprochen. Die Funktionalität der *SynchroSignal*-Klasse wurde so implementiert, dass nach dem Starten einer VR-Präsentation die Stimulanwendung alle notwendigen Ladeprozesse durchführt und anschließend solange in einer Warteschleife verbleibt, bis das Steuerungssignal vom Host-PC empfangen wurde (Abb. 7.2).

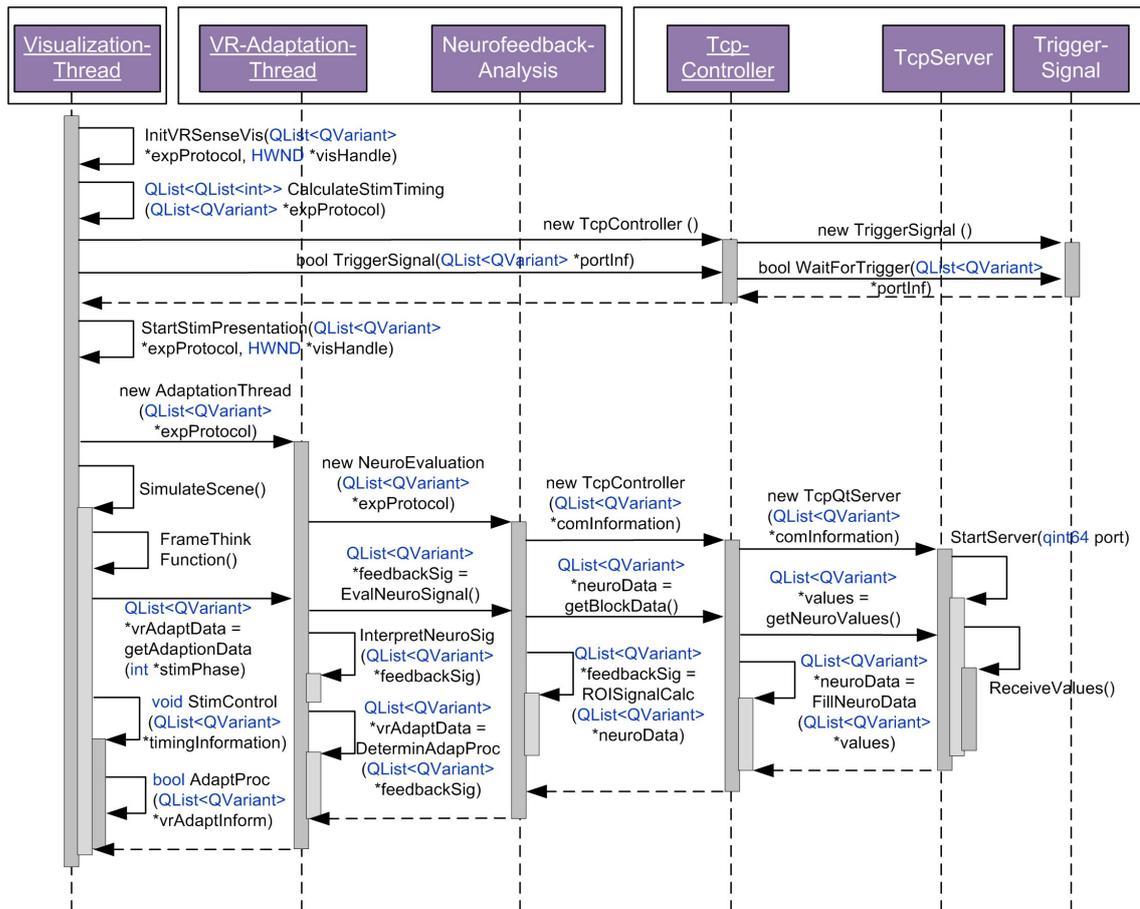


Abbildung 7.2: Das Sequenzdiagramm verdeutlicht grob die Interaktionsprozesse zwischen den Klassen der Applikationsschicht und den Klassen der Datenzugriffsschicht am Beispiel der Neurofeedback-Adaptionsfunktion mit Synchronitätssignal und TCP-Empfangsprozess.

Für das Setzen des Steuersignals wird auf das sogenannte *BUSY*-Bit der parallelen Schnittstelle zurückgegriffen, welches auf Pin 11 (25 Pin „female D“ PC-Verbinder) lokalisiert ist.

Eine weitere Klasse innerhalb der Datenzugriffsschicht ist die Klasse *PluginLoader*. Sie ist Teil des sogenannten Plug-in-Frameworks und fungiert als Schnittstelle für benutzerdefinierte Plug-ins, welche als DLL-Bibliothek eingebunden werden. Einen genauen Einblick zur Funktionsweise dieser Klasse liefert Kapitel 7.2.4, wo sämtliche Prozessabläufe des Plug-in-Frameworks gesamtlich dargestellt werden.

Die beiden letzten Klassen *TCPQtServer* und *TCPQtClient* sind Teil des im Institut für Biometrie und Medizinische Informatik, Universität Magdeburg, entwickelten Kommunikationsprotokolls. Sie werden im Folgenden gesondert vorgestellt.

7.1.1 Ein TCP-basiertes Kommunikationsprotokoll zur Informationsübermittlung

Grundlegende Anforderungen an die Entwicklung des auf TCP-basierenden Transportprotokolls waren ein hohes Maß an Erweiterbarkeit, eine breitgefächerte Kommunikationsfunktionalität sowie die Möglichkeit zum parallelen Senden und Empfangen von Informationen von der VR-Stimulusapplikation. Um die Variabilität dieser Funktionalität zu gewährleisten, wurde die Einbindung des Protokolls in zwei unterschiedliche Datenauswerteanwendungen realisiert.

Die erste Anwendung ist der im Institut für Biometrie und Medizinische Informatik, Medizinische Fakultät, Universität Magdeburg entwickelte und in *MATLAB* implementierte *rtExplorer*. Die zweite Anwendung ist der von *Brain Innovations* vertriebene und in C++ entwickelte *Turbo BrainVoyager*. Beide Anwendungen erlauben benutzerdefinierte Eingriffe und gewähren so die Möglichkeit eigene Kommunikationsroutinen zu integrieren. Nähere Informationen zu den TCP-Erweiterungen des *rtExplorers* finden sich in (Thoms 2011).

Für die Kommunikationserweiterungen des *Turbo BrainVoyager* musste ein gesonderter DLL-Plugin konzipiert und über die Plugin-Funktionalität des *Turbo BrainVoyager 3.0b* in die Anwendung integriert werden. *Brain Innovations* stellt dafür eine vordefinierte Plugin-Struktur sowie verschiedene interne Schnittstellen für den Zugriff auf laufende Echtzeit-fMRT-Prozesse zur Verfügung.

Wie bereits erwähnt, wurden in die Datenzugriffsschicht der VR-Anwendung zwei Hauptfunktionsklassen zum TCP-Kommunikationsprotokoll integriert.

Die Klassen *TCPQtServer* und *TCPQtClient* repräsentieren dabei die beiden Aspekte, die bei der Informationsübermittlung in Frage kommen.

Die Aufteilung in zwei separate Klassen diente vor allem zur besseren Strukturierung, aber auch zur erweiterten Differenzierung zwischen dem Empfang von Datenauswertinformationen während der Durchführung von Echtzeit-fMRT-Experimenten (Empfangsfunktionalität) (Abb. 7.2) und der Verbreitung von Experimentinformationen an externe Applikationen vor dem Experimentstart (Sendefunktionalität, Abb. 6.7).

Die *TCPQtServer*-Klasse wurde zudem an die Anforderungen der zu entwickelnden Echtzeit-Adaption von VR-Paradigmen sowie an die Weitergabe von Auswertinformationen zu den Klassen der Neurofeedback-Signalinterpretation angepasst (vgl. Abschnitt 7.2.2).

Des Weiteren wurde sie in die Thread-Struktur des *VRAdaptionThread* integriert und dementsprechend mit zusätzlichen Funktionen zur Parallelverarbeitung ergänzt.

Die grundlegende Implementierung des Kommunikationsprotokolls sowie die Send- und Empfangsmodalitäten wurden mit dem C++-Framework *Qt* realisiert und orientiert sich speziell an den benötigten Informationen zur Echtzeit-Stimulusadaption von VR-Paradigmen. Das *Networking*-Modul des *Qt*-Toolkits stellt bereits Routinen zur Informationsübertragung bereit und ermöglicht so die beschleunigte Realisierung dieser Teilfunktionalität. Während Send- und Empfangsfunktionalitäten damit sichergestellt waren, musste für die Verbreitung von Informationen ein geeigneter Informationsträger konzipiert werden. Durch die mehrfache Kombination der vom *Qt*-Framework bereitgestellten Konstrukte *QList* sowie *QVariant* konnte ein Datenkonstrukt erstellt werden,

welches eine wohlstrukturierte Informationsübertragung von Datenanalyseergebnissen möglich macht und die Datenhaltung der empfangenen Auswertergebnisse im Informationsprotokoll der VR-Stimulusapplikation ermöglicht. *QVariant* repräsentiert einen sogenannten abstrakten Datentyp, welcher aufgrund seiner Struktur unterschiedliche Variablentypen, auch *QList*-Konstrukte, in sich aufnehmen kann.

Der *QList*-Datentyp ist eine Container-Klasse und unterstützt das Abspeichern von unterschiedlichen Datentypen. Selbst ein *QVariant*-Datentyp, welcher bereits ein *QList*-Konstrukt mit *QVariants* beinhaltet, kann so erneut in ein *QList*-Datenkonstrukt abgespeichert werden. Dieses Konzept der Datenspeicherung verspricht durch diese multiple Verschachtelung von *QList*-Konstrukten eine hohe Effizienz und Skalierbarkeit bei der Realisierung von komplexen Datenstrukturen. Im Folgenden wurde dieses Konzept der Datenspeicherung mehrmals aufgegriffen und findet auch bei der Informationsübertragung zwischen den einzelnen Schichten des Stimulusframeworks seine Anwendung (vgl. Abschnitt 7.2.3). Detaillierte Beschreibungen zur Umsetzung des Kommunikationsprotokolls der VR-Anwendung und zum TCP-Plugin des *Turbo Brain Voyager* finden sich zudem in verschiedenen Veröffentlichungen (Luehrs et al. 2011, Mueller et al. 2011a; 2012b).

7.2 Realisierung der Applikationsschicht

Nach der Implementierung der Datenzugriffsschicht wurden die Klassen der Applikationsschicht realisiert und eingebunden. Diese Schicht ist hierarchisch über der Datenzugriffsschicht angesiedelt und enthält die eigentliche Applikationslogik der Anwendung. Hier wurden sämtliche Datenmanagement- und Datenhaltungsfunktionen, Funktionalitäten zur zeitlichen und funktionellen Experimentsteuerung, Zugriffsroutinen für benutzerdefinierte Funktionalitäten sowie unterstützende Klassen des *Vision Engine Frameworks* und des *PhysX-Frameworks* eingebunden. Ein weiterer wichtiger Bestandteil dieser Schicht ist die Experimentsteuerung, die eng mit den Klassen des *Vision Engine Frameworks* interagiert und im Folgenden näher erläutert wird.

7.2.1 Experimentsteuerung und Einbindung des Vision Engine Frameworks

Um die Einbindung des umfangreichen *Vision Engine Frameworks* möglichst effizient zu gestalten, sollten ausschließlich die benötigten Kernfunktionalitäten in das Projekt eingebunden werden. Hierfür stellt das *Vision Engine Software Development Kit* (SDK) bereits verschiedene Funktionen zur Verfügung, welche direkt in das zu entwickelnde Softwareprojekt eingebunden werden können. Dies erleichtert den Zugriff auf verschiedene Klassen der *PhysX*-Funktionalitäten und gestattet die Verwendung grundlegender Klassen für Partikeleffekte, Schattendarstellungen, Tiefenwahrnehmungssimulationen, Renderfunktionen, usw.

Für die komplette Verwendung der *Vision Engine*-Funktionalitäten mussten vorhandene Steuerklassen adaptiert und mehrere neue Steuerklassen implementiert werden. Zur Administration der allgemeinen, technischen Einstellungen der VR-Welt wurde

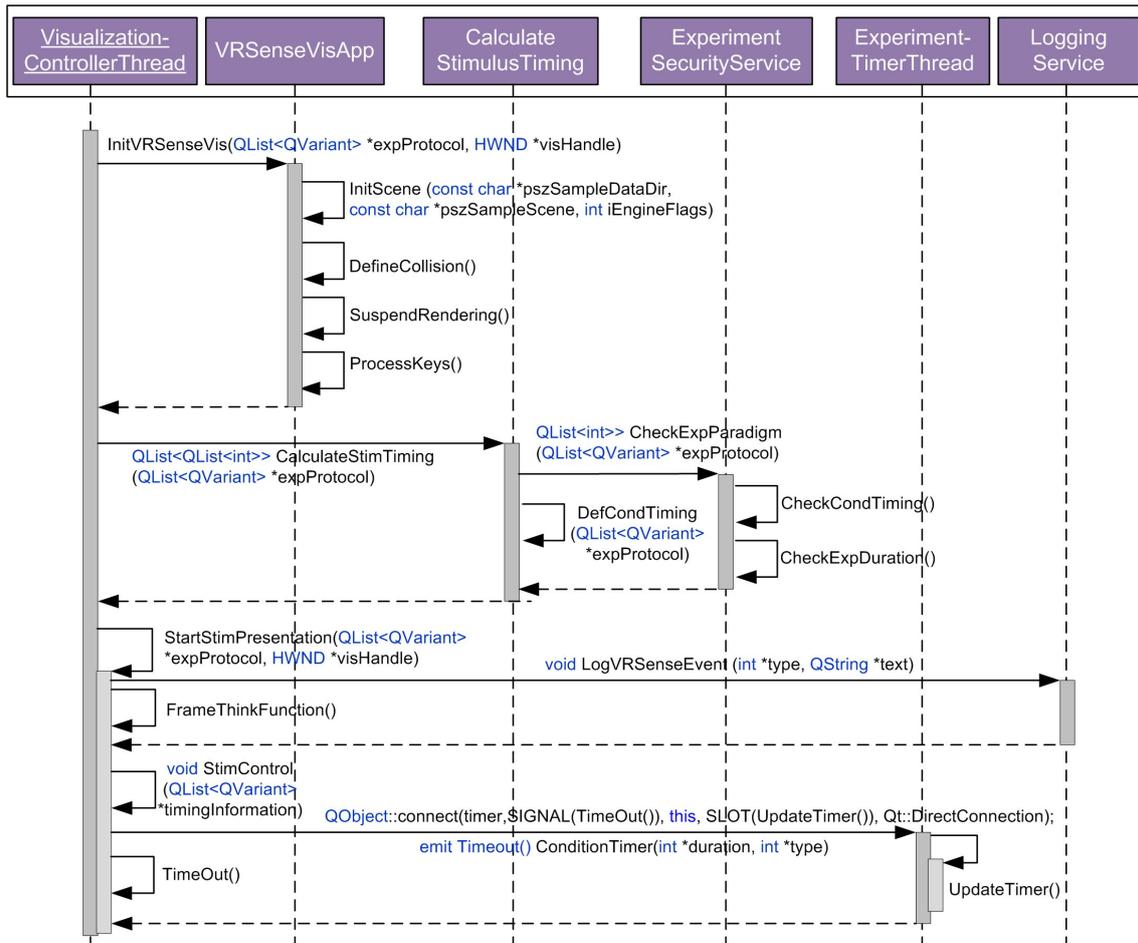


Abbildung 7.3: Das Sequenzdiagramm der Stimuluspräsentation verdeutlicht in groben Zügen die einzelnen Abläufe der VR-Konfiguration, die Berechnung des Experimentablaufs und der semantischen Fehlerüberprüfung sowie der eigentlichen Stimuluspräsentation mit der Ereignisprotokollierungsklasse *LoggingService* und der Zeitnehmer-Klasse *ExperimentTimerThread*.

die Struktur der vorhandenen *VisionApp*-Klasse an die Besonderheiten der neurowissenschaftlichen Stimuluspräsentation angepasst und mit zusätzlichen Funktionen zur Experimentkonfiguration versehen. Sämtliche Funktionserweiterungen innerhalb dieser Klassenstruktur dienten verschiedenen Verzeichnisstrukturadaptierungen, diversen Shader- und Texturerweiterungen, Präsentationsadaptierungen, der Konfiguration des implementierten Fenstermodus sowie den Einstellungen zur Steuerung des Avatars. Die resultierende *VisionApp*-Konfigurationsklasse wurde aufgrund der zahlreichen Änderungen in *VRSenseVisApp*-Klasse umbenannt (Abb. 7.3).

Eine weitere Klasse, die adaptiert und in die VR-Stimulusapplikation eingebunden wurde, ist die Klasse *VRSenseWeatherVisApp*. Sie ermöglicht die Simulation unterschiedlicher Wetterverhältnisse durch spezifische Partikeleffekte und wurde explizit für eine zeitnahe Adaption des fMRT-Paradigmas integriert. Durch diese Klasse können Sichtverhältnisse angepasst und die Aufmerksamkeit eines Probanden gezielt auf

verschiedene Punkte (saliente Reize) gelenkt werden. Essentiell für die Möglichkeit des Probanden, sich über einen Avatar in der Virtuellen Welt zu bewegen, ist die Klasse *FirstPersonEntity*. Mit ihr wurde eine Entität erstellt, welche sich anhand einer virtuellen Kamera in der Ich-Perspektive steuern lässt. Hier repräsentiert eine Entität ein eindeutig bestimmtes Objekt, über das Informationen gespeichert und verarbeitet werden. Jede Entität besitzt innerhalb des *Vision Engine Frameworks* einen sogenannten Entitätstyp sowie mehrere vordefinierte Zustände. Darauf aufbauend kann eine Entität beispielsweise einen Virtuellen Charakter, ein Virtuelles Objekt (Fahrzeug, Stuhl, Tür) oder auch einen definierten Zustand repräsentieren.

Die Klasse *FirstPersonEntity* greift zur Definition der Entitätszustände auf die Klasse *DirectXInput* der Datenzugriffsschicht zu und nutzt die über *DirectX* registrierten Tastereingaben als Kommandos für Bewegungszustandsänderungen des Avatars.

Viele Arbeitsabläufe innerhalb der Experimentsteuerung, aber auch innerhalb der gesamten Anwendung, müssen während der Laufzeit parallel ablaufen. Um hier keine Performanzverluste in Kauf nehmen zu müssen, wurden verschiedene Funktionalitäten innerhalb von parallel ablaufenden Prozessen (Threads) realisiert.

Das *Vision Engine Framework* besitzt bereits eine integrierte *ThreadManager*-Klasse, welche die Performanz der Szenenpräsentation optimiert. Um eine geregelte Interaktion zwischen *Vision Engine*-Klassen und Experimentsteuerungsklassen zu gewährleisten, wurden zur zeitlichen Steuerung und für den Empfang von Echtzeit-fMRT-Auswertinformationen verschiedene Thread-Klassen designet und eingebunden.

Berechnung des Experimentablaufs und zeitliche Steuerung

Die erste implementierte Verarbeitungsklasse zur Berechnung des VR-Experimentablaufs ist die *CalculateStimulusTiming*-Klasse. Sie erhält von der *VisualizationControllerThread*-Klasse sämtliche Stimulusinformationen und berechnet anschließend den Experimentablauf sowie die korrespondierende zeitliche Präsentation der Stimuluseignisse. Des Weiteren wird durch einen Aufruf der implementierten Klasse *ExperimentSecurityService* die Semantik der Experimentinformationen überprüft. Ergeben sich Diskrepanzen in den Experimentinformationen, wie z.B. undefinierte Stimulusintervalle, so tritt das entwickelte Sicherheitsprotokoll in Kraft, welches sofort die laufenden Berechnungen stoppt und über Benutzerdialoge Parameterkorrekturen vom Anwender einfordert (Abb 7.3).

Die Grundlage der Experimentsteuerung bildet die zentrale Steuerklasse *VisualizationControllerThread* (Abb. 7.3). Sie repräsentiert die Kontrollklasse, welche sämtliche Arbeitsschritte der Stimuluspräsentation steuert und die Szenenpräsentation startet oder beendet. Alle notwendigen Informationen zum Paradigma gelangen zuerst an sie und werden danach in geeigneter Form an die bearbeitenden Klassen weitergeleitet. Die Ergebnisinformationen dieser Datenverarbeitung werden anschließend an die *VisualizationControllerThread*-Klasse rückübermittelt und schließlich für die koordinierte VR-Szenendarstellung verwendet.

Für die zeitliche Steuerung der Stimuli wurde die *ExperimentTimerThread*-Klasse konzipiert. Diese erhält von der *VisualizationControllerThread*-Klasse die zeitlichen Informationen über die aktuelle Stimuluspräsentation und fungiert anschließend als Zeit-

nehmer. Nach Ablauf der jeweiligen Stimuluspräsentation wird vom *ExperimentTimerThread* ein *Stop-Signal* über die *QObject:connect*-Funktion emittiert und an die *VisualizationControllerThread*-Klasse rückübermittelt. Dies unterbricht die Präsentation des aktuellen Stimulus und zeigt den nachfolgenden Stimulus (Abb. 7.3).

7.2.2 Echtzeit-Stimulusadaption von Neurofeedback-Experimenten

Um die Varianz in der neurophysiologischen Reizantwort verschiedener Probanden zu berücksichtigen, wurde aufbauend auf dem Konzept aus Kapitel 6.5 ein zweistufiger Adaptionalgorithmus entworfen. Dabei wird zunächst eine Kalibrierungsmessung durchgeführt, bei der Referenzwerte für die neuronale Aktivierung des jeweiligen Probanden ermittelt werden. Im zweiten Schritt erfolgt dann die eigentliche Hauptmessung mit der Echtzeit-Auswertung der Hirnaktivierung und der Adaption der VR-Stimuli. Für die Referenzwertbestimmung und die spätere Adaption sollten das experimentelle Design der Kalibrierungsmessung und der Hauptmessung übereinstimmen.

Extraktion von Referenzwerten

In der Kalibrierungsmessung bzw. dem Vorexperiment werden zunächst die relevanten Hirnregionen für das spätere Neurofeedback-Training als region of interest (ROI) definiert. Nach dem Start der Messung erfolgen die Akquirierung und die kontinuierliche Übertragung der Datensätze zur Echtzeit-Datenauswertung. Für die Echtzeit-Datenauswertung von fMRT-Datensätzen wurde auf ein rekursives Allgemeines Lineares Modell zurückgegriffen, welches auch im *Turbo Brain-Voyager* (TBV) seine Anwendung findet, siehe Kapitel 2.3.2.

Neben der Anzeige der neuronalen Aktivität mittels des Parameterschätzers β unterstützt der TBV auch den Export der neuronalen Aktivitätsänderungen in Form des aus den MR-Bildern extrahierten Signals, d.h. der Grauwerte. Die Grauwertänderung eines MRT-Bildes über die Zeit kann dabei Aufschluss über die neuronale Aktivität eines Probanden liefern. Der TBV erlaubt auf Grundlage dieser Funktion die Mittelung von Grauwert-Clustern innerhalb einer festgelegten Hirnregion (ROI). In einem Vorexperiment werden daher zunächst entsprechende ROI als Referenzregionen definiert. Als wichtige Teilfunktionalität dient das entwickelte TCP-Kommunikationsprotokoll zur strukturierten Übertragung der Auswertinformationen der selektierten ROI an das interne Informationsprotokoll der VR-Stimulusapplikation. Eine gleichzeitige Auswahl von mehreren ROI wird vom TBV unterstützt. Im Rahmen des *VRAdaptionThread* greift der implementierte Echtzeit-Adaptionalgorithmus zunächst auf die Auswertinformationen mit den zugehörigen Grauwerten zu und speichert diese in einem Datenkonstrukt A ab:

$$A = \begin{pmatrix} a_{11} & \cdots & a_{m1} \\ \vdots & \ddots & \vdots \\ a_{1n} & \cdots & a_{mn} \end{pmatrix}. \quad (7.1)$$

Das Datenkonstrukt A wird dabei durch m der Anzahl der selektierten ROI und n der Anzahl der Messpunkte definiert. Informationen über die Experimentstruktur und

den Experimenttyp erhält der Algorithmus über die Informationsextraktion der Design-Matrix. Bei der Verarbeitung der Zeitreihen unterscheidet dieser zudem zwischen einer Signalauswertung für Block- und ereigniskorrelierte Experimentdesign.

Wurden alle Grauwerte für die Messpunkte n_S eines Stimulus S empfangen, so erfolgt die Berechnung des arithmetischen Mittels aller Grauwerte dieser Messpunkte. Dies wird für alle Stimuli S fortgesetzt, bis sämtliche Grauwerte der Vormessung empfangen wurden:

$$\bar{S} = \frac{1}{n_S} \sum_{i=1}^{n_S} \text{Grauwert}_i. \quad (7.2)$$

Da in Block-Design Experimenten ein einzelner Block mehrfach wiederholt wird, wurde für diesen Experimenttyp eine zusätzliche Optimierung eingeführt. So werden zur Verbesserung der Referenzwerte die arithmetischen Mittel aller korrespondierenden Stimuli zusammengefasst und zur Berechnung des globalen arithmetischen Mittels für jeden Stimulus verwendet. Für ereigniskorrelierte Experimentdesign ist per Definition eine solche Annahme nicht möglich.

Der letzte Schritt bei der Referenzwertextraktion ist die Berechnung und Abspeicherung des prozentualen Signalunterschieds p_n zwischen dem arithmetischen Mittel des Vorgängerstimulus \bar{S}_{n-1} und dem arithmetischen Mittel des Nachfolgerstimulus \bar{S}_n für ereigniskorrelierte Experimentdesign:

$$p_n(\bar{S}_n, \bar{S}_{n-1}) = 100 \left(\frac{\bar{S}_n}{\bar{S}_{n-1}} - 1 \right) \quad (7.3)$$

bzw. dem arithmetischen Mittel des aktuellen Stimulus \bar{S}_n und dem arithmetischen Mittel der Nullbedingungen \bar{S}_0 für Block-Design Experimente:

$$p_n(\bar{S}_n, \bar{S}_0) = 100 \left(\frac{\bar{S}_n}{\bar{S}_0} - 1 \right). \quad (7.4)$$

Echtzeit-Signalanalyse und -Stimulusadaption

Nach der Berechnung der Referenzwerte in der Kalibrierungsmessung durch Formel (7.2) bis Formel (7.4) findet in der Hauptmessung die eigentliche Echtzeit-Adaption eines VR-Stimulus statt. Dabei baut der Algorithmus auf den berechneten prozentualen Signalunterschieden p_n aus der Kalibrierungsmessung auf und vergleicht diese mit aktuellen prozentualen Signalunterschieden (Abb. 7.4). Sind die aktuellen Signalunterschiede größer oder kleiner als eine zuvor definierte Toleranzschwelle, erfolgt die Stimulusadaption zum nächsten definierten Zeitpunkt. Der wesentliche Unterschied zur Vormessung besteht darin, dass sämtliche Messwerte sofort nach dem Empfang über die TCP-Verbindung verarbeitet werden. Dies gestattet die Berechnung des Signalunterschieds zwischen zwei Stimuli in Echtzeit und ermöglicht die Anwendung des Entscheidungskriteriums zur Adaption eines Stimulus. Das Prinzip zur Berechnung der Signalunterschiede in der Hauptmessung arbeitet nach einem ähnlichen Prinzip wie in der Kalibrierungsmessung. Dabei werden zunächst alle Grauwerte aus den

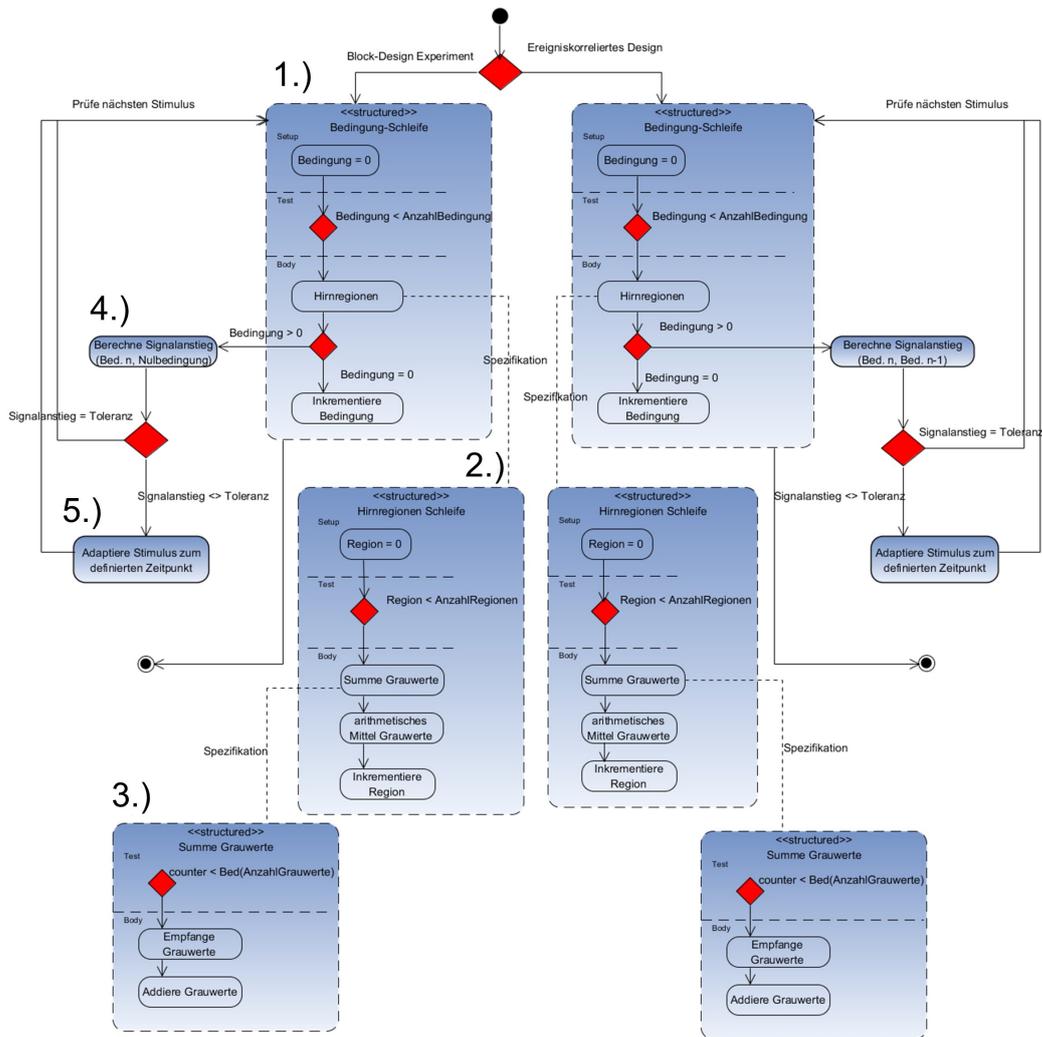


Abbildung 7.4: Das Aktivitätsdiagramm zum Adaptionalgorithmus. In der Hauptmessung analysiert der Algorithmus das Neurofeedback-Signal und vergleicht die Werte mit den Referenzwerten (Punkt 1 - 4). Bei Über- oder Unterschreitungen des Wertebereichs (Punkt 5) erfolgt eine Echtzeit-Stimulusadaption nach vordefinierten Kriterien.

angegebenen ROI (Punkt 2, Abb. 7.4) über TCP empfangen sowie deren Summe errechnet (Punkt 3). Anschließend erfolgt die Berechnung des arithmetischen Mittels für den jeweiligen Stimulus und für alle ROI (Punkt 2). Wurde bei der Abfrage (Punkt 1) für mindestens einen Vorgängerstimulus das arithmetische Mittel der Grauwerte bestimmt, so kann die Berechnung des prozentualen Signalunterschieds zwischen zwei Stimuli durch (Punkt 4) erfolgen. Aufgrund der Besonderheiten von Block-Design und ereigniskorrelierten Design unterscheiden sich hier die Berechnungsgrundlagen des Signalunterschieds.

Eine Abfrage durch den Entscheidungsknoten in Formel (7.5) entscheidet schließlich über die Anpassung des Stimulus (Punkt 5). Dabei wird der aktuelle Signalunterschied p_a mit dem vom Benutzer definierten Toleranzwert k und dem Signalunterschied p_n aus der Vormessung verglichen. Befindet sich der Wert des Signalunterschieds p_a nicht

in dem durch den Toleranzwert k eingegrenzten Wertebereich, so gilt $f(p_a) = 1$ und eine Stimulusadaption zum nächsten definierten Zeitpunkt wird eingeleitet. Durch die benutzerdefinierte Toleranzwerteingabe wurde zudem eine Möglichkeit zur optionalen Einflussnahme des Experimentleiters auf die Stimulusadaption geschaffen. Die Funktion $f(p_a)$ des Entscheidungskriteriums lässt sich hier wie folgt beschreiben:

$$f(p_a) = \begin{cases} 0, & \text{falls } (p_n - k) \leq p_a \leq (p_n + k) \\ 1, & \text{sonst.} \end{cases} \quad (7.5)$$

Neben dem eigentlichen Algorithmus stand vor allem die klassenspezifische Realisierung dieses Adaptionskonzeptes im Vordergrund. Ausgehend von der aspektorientierten Implementierung sowie der Strukturierung durch ein erweitertes Drei-Schichten-Architekturmodell wurden verschiedene Klassen innerhalb der Datenzugriffsschicht und der Applikationsschicht erstellt.

Hier sind insbesondere die *VRAdaptionThread*-Struktur, die *TcpController*-Klasse sowie die *TcpServer*-Klasse für den sicheren Empfang der Datenauswertinformationen verantwortlich. Wird der Adaptionsthread gestartet, so steuert er neben dem Empfang auch die Weitergabe der Auswertinformationen zur *NeurofeedbackAnalysis*-Klasse (Abb. 7.2).

Einzelne Werte des Neurofeedback-Signals werden hier zunächst analysiert und anschließend anhand des Entscheidungskriteriums aus Formel 7.5 interpretiert.

Die so gewonnenen Informationen können anschließend über die *EngineController*-Klasse zur *FirstPersonEntity*-Klasse oder zur *VRSenseWeatherVisApp*-Klasse weitergeleitet werden und erlauben damit die Adaption des laufenden Stimulus. Die Steuerung des Avatars durch neuronale Aktivierungen des Probanden oder die temporäre Präsentation von salienten Reizen zur Beeinflussung der Aufmerksamkeit sind über diese Prozessstruktur ebenfalls möglich.

7.2.3 Datenmanagementframework

Ein wesentlicher Aspekt bei der Realisierung dieser Anwendung war die Modellierung eines Datenmanagement- und Datenhaltungskonzeptes. Hierfür musste ein skalierbares Konstrukt geschaffen werden, welches folgende Funktionalitäten unterstützt: eine variierende Anzahl von Experimentinformationen strukturiert abspeichern (a), eine Menge von Experimentinformationen zuverlässig zu internen und externen Komponenten transportieren (b) und diese Experimentinformationen vor etwaigen Datenverlusten schützen (c).

Ein beständiger Schutz vor Datenverlusten durch Absturz oder Beenden eines Programms lässt sich in der Regel nur durch die Einbeziehung einer Datenbank- oder einer Dateisystemspeicherung realisieren. Zunächst wurde zur Datenspeicherung das Dateiformat *VREXP* konzipiert. Es stellt ein einfaches Konzept auf Basis einer regulären Textdatei dar und speichert sämtliche Experimentinformation unverschlüsselt und im Klartext. Zukünftig ist das Abspeichern von Experimentinformation in pseudonymisierter Form innerhalb einer Datenbank vorgesehen.

Ausgehend von den definierten Aufgaben des Drei-Schichten-Modells wurde für das

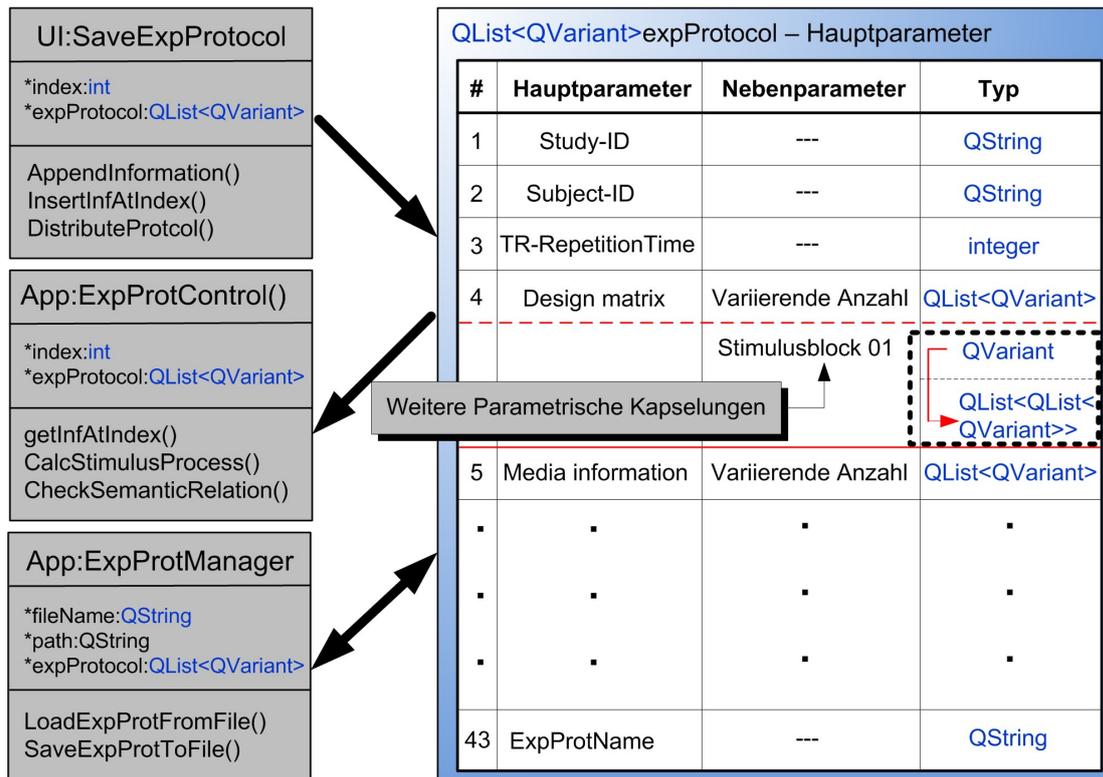


Abbildung 7.5: Das Datenhaltungskonzept *experimentProtocol* basiert auf einem `QList<QVariant>` Datenkonstrukt, welches Informationen durch parametrische Kapselungen strukturiert und geordnet abspeichert. Durch die Verwendung des `QVariant`-Datentyps werden zudem weitere Spezifizierungen durch `QList`-Datenkonstrukte ermöglicht (gestricheltes Kästchen).

Sammeln von Experimentinformationen die Präsentationsschicht ausgewählt. In der Klasse *SaveExpProtocol* werden hierfür die eingegebenen Experimentinformationen zusammengeführt und innerhalb eines Datenkonstrukts abgespeichert. Aufgrund der Vielzahl an Informationstypen innerhalb der Experimentinformationen und im Hinblick auf zukünftige Erweiterungen musste die Abspeicherung der Informationen innerhalb eines gesonderten Konstrukts erfolgen. Viele der Informationen liegen beispielsweise als numerischer Wert, als Zeichenkette, als Boolean-Wert oder als zusammenhängende Information innerhalb einer Matrix vor und können nicht einheitlich in einer Container-Klasse abgelegt werden.

Eine mögliche Lösung ist die Einbeziehung von sogenannten *Union-Datentypen* (discriminated - union wrapper), wie z.B. der *QVariant*-Datentyp (Blanchette und Summerfield 2009).

Ein *Union-Datentyp* ist eine Datenstruktur, welche eine Vielzahl unterschiedlicher Datentypen in sich aufnehmen kann und gleichzeitig alle Informationen über den ursprünglichen Datentyp archiviert. Dies erlaubt eine flexible Speicherung und eine zuverlässige Rekonstruktion des abgespeicherten Wertes. Für die strukturierte Speicherung und den Transport von Experimentinformationen wurde, ähnlich zum TCP-Kommunikationsprotokoll, das Datenkonstrukt `QList<QVariant> expProtocol` auf Basis

der *Qt*-Klassen *QList* und *QVariant* realisiert (Abb. 7.5). Die *QList*-Klasse dient dabei als übergeordnetes Konstrukt und erlaubt durch die indizierte Speicherung eine strukturierte Ablage von Informationen an festdefinierten Positionen. Komplexe Experimentinformationen werden zunächst innerhalb eines *QVariant*-Datentyps zwischengespeichert und anschließend in das *QList*-Konstrukt eingepflegt. Der Vorteil dieses Konzepts liegt in der flexiblen Struktur des *QVariant*-Konstrukts, welches wiederum ein verschachteltes *QList*-Konstrukt in sich aufnehmen kann und somit eine multiple Kapselung von Informationen ermöglicht. Dieses Konzept wurde auch für das Abspeichern der Matrix-basierten Paradigmeninformationen verwendet. Ausgehend von diesem Konzept wurden drei Hauptfunktionsklassen zur Interaktion mit dem Datenkonstrukt implementiert (Abb. 7.5).

Die Klasse *SaveExpProtocol* führt hier sämtliche eingegebenen Experimentinformationen zusammen und speichert diese dann innerhalb des Datenkonstrukts *expProtocol*. Die Klasse *ExpProtControl* innerhalb der Applikationsschicht greift mittels verschiedener Methoden auf diese Informationen zu und leitet sie gegebenenfalls zur semantischen Fehlerüberprüfung oder zur Berechnung des Experimentablaufs weiter. Ein weiterer Aspekt dieses Datenhaltungskonzeptes wird durch die Klasse *ExpProtManager* repräsentiert. Sie übernimmt sowohl das Einlesen von Experimentinformationen als auch die Datensicherung durch den Export der Experimentinformationen in das oben beschriebene *VREXP*-Dateiformat. Des Weiteren ermöglicht sie das strukturierte Auslesen sämtlicher Auswertinformationen aus dem TCP-Stream sowie das zielgerichtete Abspeichern innerhalb des internen Informationsprotokolls (vgl. Abschnitt 6.4).

7.2.4 Einbindung benutzerdefinierter Funktionalitäten

Eine weitere Komponente ist die sogenannte Plug-in-Schnittstelle, welche eine schnelle und unkomplizierte Einbindung von benutzerdefinierten Funktionalitäten über DLL-Dateien (direct link libraries) gestattet. Die Plug-in-Schnittstelle ist nicht ausschließlich auf die Applikationsschicht beschränkt, sondern besitzt ein globales Konzept, das sich durch alle Schichten der Anwendung zieht.

Das *Qt*-Framework stellt dahingehend bereits diverse Funktionalitäten zum Einbinden von C++ Bibliotheken zur Verfügung. Ausgehend von diesen Funktionalitäten wurde ein Plug-in-Framework modelliert, welches vollständig auf die implementierte Dreischichten-Architektur angepasst wurde.

Zur Erstellung und Einbindung benutzerdefinierter DLL-Bibliotheken muss zunächst eine vordefinierte Struktur eingehalten werden. Alle benutzerdefinierten DLL-Dateien die in die VR-Stimulusanwendung eingebunden werden, müssen spezifische Methoden in der Hauptklasse enthalten.

Dies ist zum einen die Funktion *Name()*, welche den Namen des Plug-in und den Konstruktor festlegt. Des Weiteren die Methode *Echo()*, die den Destruktor der Plug-in-Klasse beschreibt. Weitere grundlegende Methoden sind *PlugLog()*, für konkrete Einstellungen auf der Bedienoberfläche des Plug-in, und *getUiData()*, zum Abrufen der Konfigurationseinstellungen, welche auf der Bedienoberfläche des Plug-in vorgenommen wurden.

Die direkte Schnittstelle für benutzerdefinierte DLL-Bibliotheken befindet sich in der

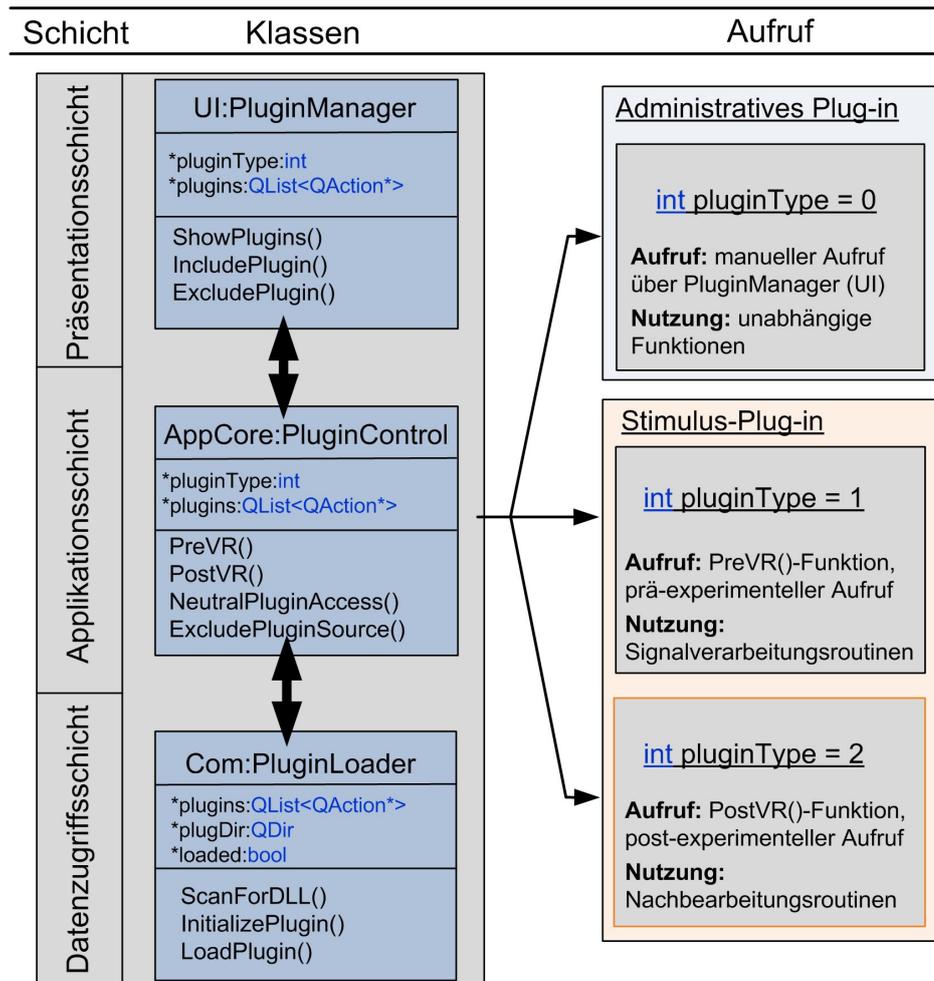


Abbildung 7.6: Benutzerdefinierte Erweiterungen (Plug-in) werden über die Datenzugriffsschicht eingebunden, über die Applikationsschicht aufgerufen und über die Präsentationsschicht konfiguriert. Des Weiteren sind im bisherigen Konzept drei verschiedene Plug-in-Typen vorgesehen, die sich vor allem anhand ihres Aufrufpunktes unterscheiden.

Klasse *PluginLoader* innerhalb der Datenzugriffsschicht. Die Anbindung der Plug-in-Funktionalitäten an die Funktionen der VR-Stimulusapplikation wurde basierend auf dem verwendeten Architekturmodell innerhalb der Applikationsschicht realisiert. Hierfür wurden zwei Schnittstellen innerhalb der *PluginControl*-Klasse definiert. Die Methoden *PreVR()* und *PostVR()* sind die entscheidenden Funktionen innerhalb dieser Plug-in-Struktur. Sie stehen für die eigentlichen Aufrufpunkte, in denen auf die jeweiligen Funktionen eines Plug-in zugegriffen werden. Zusammen mit dem integer-Wert *pluginType* wird zudem der Typ des jeweiligen Plug-in spezifiziert (Abb. 7.6). Plug-ins, welche durch den *pluginType = 1* definiert sind, werden vor dem Start der Szenenpräsentation in der Funktion *PreVR()* aufgerufen. Plug-ins, die dem *pluginType = 2* entsprechen, werden nach dem Ende der Szenenpräsentation in der Funktion *PostVR()* ausgeführt. Plug-ins mit dem *pluginType = 0* werden als administrative, eigenständige Erweiterungen in die Stimulusapplikation integriert und können

nur direkt über die Benutzerschnittstelle aufgerufen werden. Den Abschluss des Plug-in-Konzeptes bildet die Klasse *PluginManager*, welche innerhalb der Präsentationsschicht angesiedelt und für die Administration sowie die Verwaltung der verschiedenen Plug-ins zuständig ist.

Eine weitere Funktionalität die mit diesem Plug-in-Konzept eingeführt wurde, ist die Integration von spezifischen Oberflächenkomponenten. Die Plug-in-Struktur erlaubt das Erstellen und die Einbindung von Oberflächenstrukturen zur Administration benutzerdefinierter Funktionalitäten. Dazu wird der Quellcode dieser Oberflächenkomponenten über den *PluginManager* in die VR-Stimulusanwendung eingebunden und dem Benutzer anschließend präsentiert. Sämtliche Oberflächenkomponenten zur Administration von eingebundenen Plug-ins können an speziell dafür vorgesehenen Punkten der Benutzeroberfläche aufgerufen werden und visualisieren Informationen aus den in der DLL-Bibliothek definierten Methoden *PlugLog()* und *getUiData()*.

7.3 Realisierung der Präsentationsschicht

Die Präsentationsschicht oder Benutzeroberfläche repräsentiert in der Schichtenhierarchie die höchste Schicht. In ihr finden die Erhebung der Experimentinformationen, die syntaktische Analyse der Informationen, die Konfiguration sämtlicher Funktionalitäten sowie die eigentliche Stimuluspräsentation statt. Hier gestattet das *open source user interface framework Qt* zusammen mit dem *QT Designer*, eine Vielzahl von Möglichkeiten zur Modellierung einer Nutzeroberfläche. Des Weiteren wurde zum Konzept der Präsentationsschicht eine zusätzliche Fassadenschicht hinzugefügt, welche im Rahmen zukünftiger Erweiterungen eine schnellere und flexiblere Anpassung von GUI-Klassen ermöglicht.

7.3.1 Strukturierung der Oberflächenkomponenten

Im Mittelpunkt des Implementierungsprozesses stand hier zum einen die Skalierbarkeit der Präsentationsschicht und zum anderen die Modellierung der Nutzeroberfläche nach dem Arch-Modell. Zusätzlich dazu sollten sich die Struktur und die Prozessabläufe an der Modellierung neurowissenschaftlicher Experimente orientieren. Des Weiteren waren bereits einige zeitnahe Erweiterungen angedacht, die neben fMRT-Experimenten auch andere Modalitäten, wie z.B. EEG, unterstützen. Zur Unterstützung der post-experimentellen Auswertung von VR-Experimenten ist die Videoaufzeichnung des Probandenverhaltens innerhalb einer Virtuellen Umgebung eine sehr nützliche aber auch aufwendige Funktionalität. Um die Einbindung dieser und weiterer Eigenschaften in zukünftigen Inhaltserweiterungen zu vereinfachen, wurde sich für eine Hauptoberfläche mit vier Schaltflächen entschieden. Diese sollen die Einstiegspunkte für die aktuellen und die geplanten Kernfunktionalitäten der VR-Stimulusapplikation definieren.

Die Einstiegspunkte sind die Stimuluspräsentation (a), die Funktion zur zukünftigen videobasierten Auswertung (b), der Aufruf der Dokumentation (c) sowie die Funktionalität zur Administration der geplanten Probanden- und Studieninformationen (d).

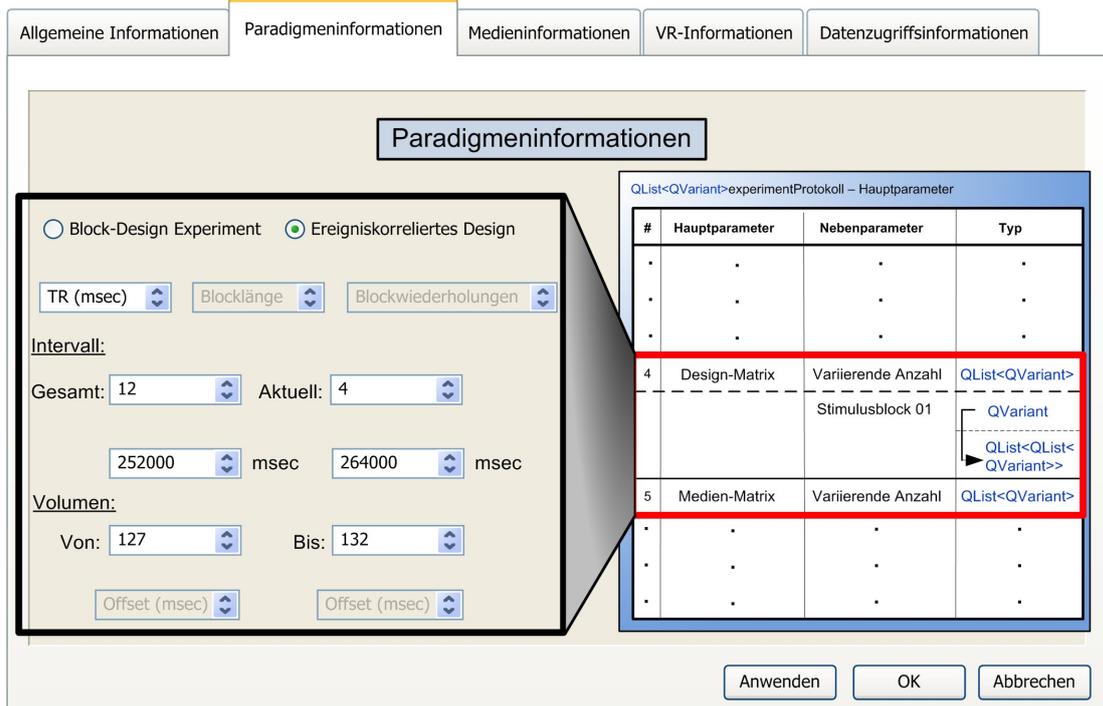


Abbildung 7.7: Das Konzept zur VR-Experimentmodellierung baut auf einem Registerkartenmodell auf. In Relation stehende Experimentparameter werden sowohl auf den Registerkarten als auch im Experimentprotokoll ortsnahe angeordnet und ermöglichen eine hohe Zugriffsgeschwindigkeit.

Die Schaltfläche zum Aufruf der Stimuluspräsentation ist der eigentliche Startpunkt zur Erstellung von VR-Experimentabläufen. Nach dem Aufruf der Startfunktion erscheint das Präsentationsfenster, welches sich in mehrere Bereiche gliedert. Im Zentrum befindet sich der eigentliche Präsentationbereich. Die Präsentation von VR-Stimuli kann dabei entweder innerhalb eines Fenstermodus oder im Vollbild durchgeführt werden. Aus Gründen einer potentiellen Aufmerksamkeitsverschiebung werden Stimuli regulär im Vollbild präsentiert.

Standardmäßig am linken Bildschirmrand befindet sich die Schnellstartleiste der Anwendung, welche die meist genutzten Funktionen beherbergt. Einen kompletten Überblick über alle Funktionalitäten bietet das Hauptmenü am oberen Rand des Präsentationsfensters. Die Menüstruktur orientiert sich an den Anordnungen konventioneller Stimulusanwendungen und ordnet die unterschiedlichen Kernfunktionalitäten hinsichtlich ihrer Applikationslogik. Die wichtigsten Menüpunkte sind *Communication*, *Logs* und *Plugins*. Der Menüpunkt *Communication* umfasst die Funktionen des TCP-Übertragungsprotokolls und Funktionen des Synchronisationssignals. Der Menüpunkt *Logs* beinhaltet Menüstrukturen zur Anzeige sämtlicher Experiment- und Ereignisprotokolle. Der letzte funktionspezifische Menüpunkt wird durch den Abschnitt *Plugins* repräsentiert. Hier sind die Funktionen des im vorherigen Unterkapitel beschriebenen Plugin-Managers angesiedelt. Die aktuelle Hauptfunktionalität des Präsentationsfensters ist die Erstellung von VR-fMRT-Experimenten, welche über den Menüpunkt *Crea-*

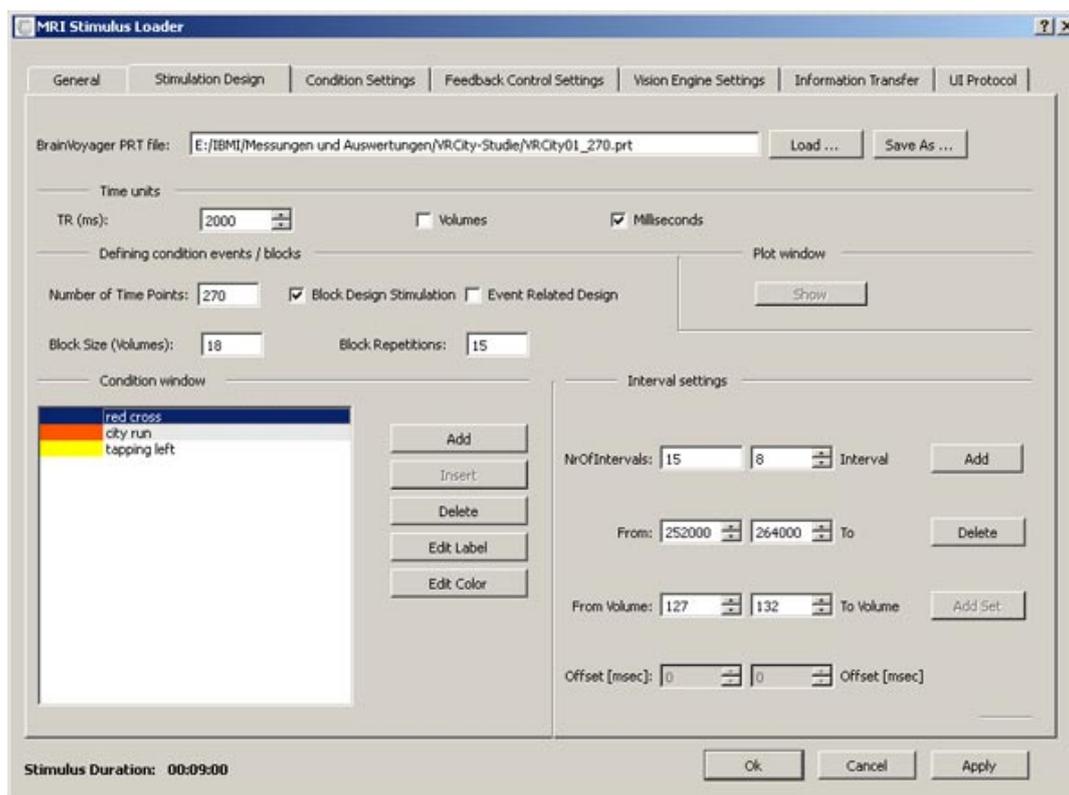


Abbildung 7.8: Die Stimulus-Registerkarte zur Modellierung von fMRT-Paradigmen. Im Zentrum steht das Festlegen des Experimentablaufs mit den entsprechenden Versuchsbedingungen.

te *fMRT-Stimulus* gestartet wird. In Folge dessen, gelangt man zum Dialog der Paradigmenerstellung. Ab hier wird die Eingabe der verschiedenen Experimenteinstellungen über ein Registerkartenmodell vollzogen (Abb. 7.7).

Auf sieben verschiedenen Registerkarten wird der Ablauf einer regulären Experimentmodellierung aufgegriffen und dem Nutzer die Möglichkeit gegeben, die einzelnen Aspekte der Modellierung separat durch Anklicken einer Registerkarte anzusteuern. Paradigmeninformationen, Stimulusinformationen, Probanden- und Studiedaten sowie die Experimenteinstellungen können hierüber gezielt eingetragen und gespeichert werden. Die Anordnung der Registerkarten bzw. die Eingabereihenfolge der jeweiligen Experimentinformationen richtet sich dabei nach dem Aufbau des im Kapitel 7.2.3 beschriebenen Experimentprotokolls (Abb. 7.7). In Relation stehende Informationen werden sowohl auf den Registerkarten als auch im Experimentprotokoll ortsnahe angeordnet und erlauben so, besonders bei komplexen Experimenten, eine hohe Zugriffsgeschwindigkeit auf vorhandene Experimentinformationen. Informationen zur Stimulusreihenfolge, zur Stimulusdauer und zum Stimulustyp werden auf der *Stimulus-Registerkarte* eingetragen und können von dort auch über PRT-Dateien importiert oder exportiert werden (Abb. 7.8).

Eine weitere wichtige Registerkarte ist die *Condition-Registerkarte*. In ihr werden sämtliche zuvor erstellten Versuchsbedingungen mit Mediendateien verknüpft. Die zu-

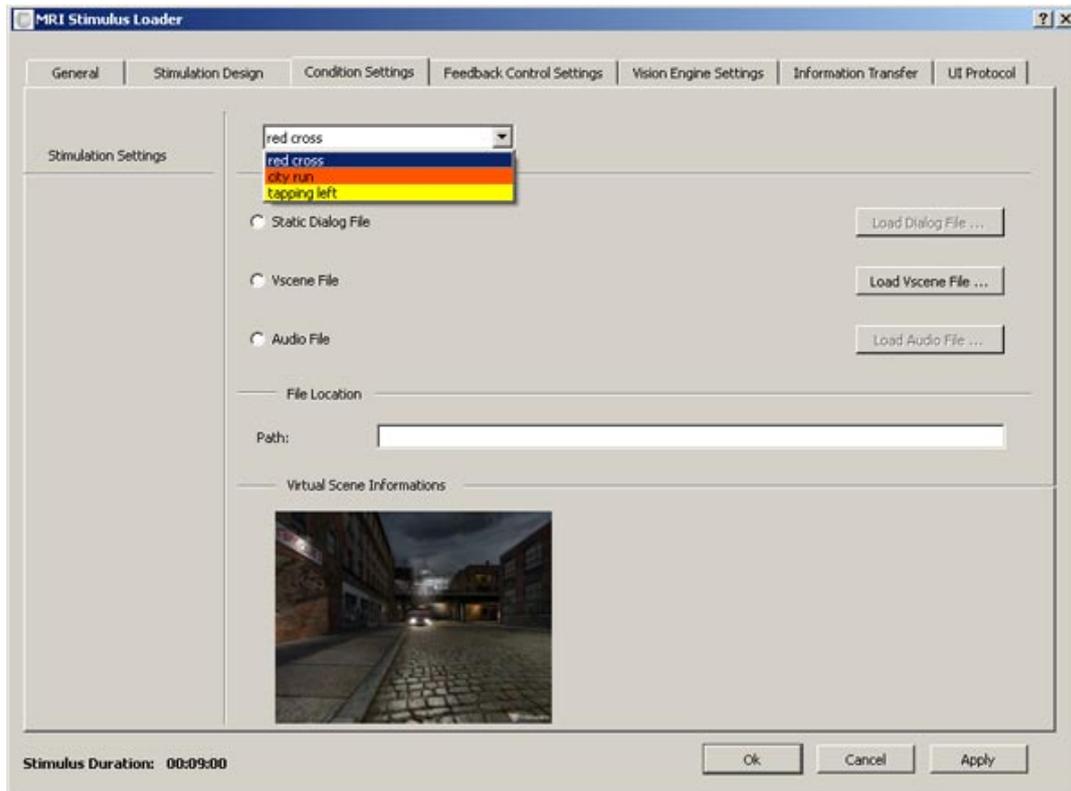


Abbildung 7.9: Die zuvor festgelegten Versuchsbedingungen auf der Stimulus-Registerkarte werden hier mit dem Stimulustyp und der entsprechenden Mediendatei verknüpft. Informationen auf dieser Registerkarte sind von den zuvor gemachten Angaben abhängig und werden automatisch aktualisiert.

vor erstellten Informationen zur Stimulusreihenfolge, Stimulusdauer, etc. werden dabei übernommen und farblich kodiert wiedergegeben (Abb. 7.9).

Zusätzlich dazu wurde mit der Registerkarte *Information Transfer* eine Möglichkeit zum Austausch von Experimentinformationen via TCP bereitgestellt. Durch ihre direkte Informationsübertragung verhindert diese Funktionalität Inkonsistenzen durch Schreib- und Lesezugriffe auf Textdateien. Des Weiteren kann über die Schnellstartoption der Registerkarte die bereits beschriebene TCP-Serverfunktion oder die Steuerung des Avatars mittels Tastern bequem zugeschaltet werden.

Die letzte Registerkarte *UI-Protocol* dient als Administrationswerkzeug der Anwendung. Sämtliche Experimentinformationen können so aus einer *VREXP*-Protokolldatei eingeladen werden und gestatten so den Import einer kompletten Konfiguration. Diese Funktionalität kann darüber hinaus als Ersatz für den netzwerkabhängigen Austausch von Informationen mittels TCP angesehen werden.

Neben der Benutzeroberfläche der Anwendung wurde eine weitere Menüstruktur innerhalb der Szenenpräsentation des *Vision Engine Frameworks* realisiert. Sie gewährleistet durch eine vereinfachte Menüstruktur eine unkomplizierte und schnelle Wiederholung definierter Stimuli und erlaubt durch eine vorgesetzte Instanzierung die augen-

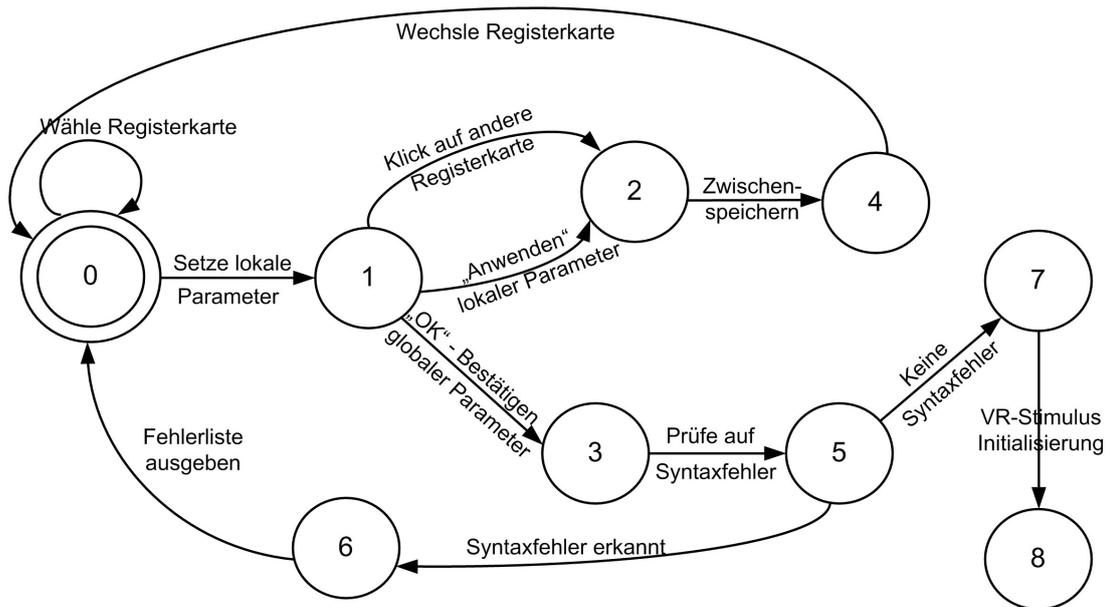


Abbildung 7.10: Die syntaktische Fehlerüberprüfung der Benutzerschnittstelle wurde mit einer kumulativen Zwischenspeicher-Funktion kombiniert. Experimentparameter werden innerhalb der syntaktischen Analyse zunächst hinsichtlich ihres Paramertyps und anschließend hinsichtlich eines vordefinierten Wertebereichs überprüft.

blickliche Unterbrechung laufender Stimuluspräsentationen. Die mehrfache Wiederholmöglichkeit eines Stimulus ohne erneutem Rückgang zu den Experimenteinstellungen wurde durch die implementierte Klasse *MonitorContext* ermöglicht. Sie gestattet per Knopfdruck, die Unterbrechung der Szenenpräsentation und die Einblendung von Einzelbildern während der Ruhebedingungen.

Syntaktische Fehlerprüfung

Ein wichtiger Aspekt der Benutzeroberfläche ist die durch das Arch-Modell spezifizierte Dialog-Komponente, welche die Interaktionen zwischen dem Anwender und dem UIMS steuert. Sie beinhaltet außerdem sicherheitstechnische Funktionen, die eine echtzeitnahe Überprüfung von eingegebenen Experimentinformationen durchführen. Diese syntaktische Analyse der Parameter wurde direkt in die Dialog-Komponente der Präsentationsschicht integriert und mit einer kumulativen Zwischenspeicher-Funktion zur Sicherung eingegebener Parameter versehen (Abb. 7.10). Nach jedem Klick zur Auswahl einer neuen Registerkarte werden zuerst die lokalen Werte der noch aktuellen Registerkarte in das Experimentprotokoll überführt. Nach der Bestätigung aller Parameterwerte auf der jeweiligen Registerkarte erfolgt die eigentliche syntaktische Prüfung der Werte. Dazu werden die Werte zunächst nach ihrem eingegebenen Variablentyp und anschließend hinsichtlich ihres vordefinierten Wertebereichs überprüft. Bei einer Verletzung der definierten Kriterien tritt ein Sicherheitsprotokoll in Kraft, welches die laufenden Prozesse stoppt und Sicherheitsmeldungen an den Anwender liefert.

Die syntaktische Fehlerprüfung dient als ergänzende Fehlersuche und soll bereits vor der semantischen Fehlerüberprüfung (vgl. Abschnitt 7.2.1) einfache Fehler bei den gemachten Experimentinformationen aufspüren und dem Anwender melden.

7.4 Evaluation der Softwareanwendung

7.4.1 Material und Methodik

Um einen repräsentativen Eindruck zur Zuverlässigkeit und zur Gesamtleistung dieser mehrschichtigen Applikation zu erhalten, wurde zunächst die Leistung der Einzelschichten analysiert und anschließend die Gesamtleistung in Relation zum kompletten Softwareframework betrachtet (Eeckhout 2010). Alle Leistungstests wurden auf zwei unterschiedlichen Computern durchgeführt. Die genauen technischen Spezifikationen dieser Computersysteme sind in Tabelle 7.1 dargestellt.

Die erste Hauptkomponente, die getestet wurde, war die Benutzerschnittstelle der Anwendung. Da hier die Erhebung, die Speicherung und die Weitergabe der Experimentinformationen realisiert wurde, prüfte man hier zunächst die korrekte Interaktion der involvierten Verarbeitungsprozesse. Dazu testete man die korrekte Erstellung und die Präsentation mehrerer unterschiedlicher Experimente. Insgesamt 40 Experimente im Block- und im ereigniskorrelierten Design mit einem Minimum von 30 und einem Maximum von 35 Versuchsbedingungen und einer festen Anzahl von je 1000 Messpunkten wurden hierfür erstellt. Während dieses Modellierungsprozesses wurden sämtliche Prozessaufrufe und die Speicherauslastung überwacht.

Aufgrund der Komplexität der Applikationsschicht wurden zwei verschiedene Leistungstests zur Evaluierung angesetzt. Hier sollte zum einen die korrekte Interaktion der selbstimplementierten Klassen mit den externen Klassen des *Vision Engine Frameworks* analysiert und zum anderen die Funktionen zur Berechnung des Stimulusverlaufs, zur semantischen Fehlerüberprüfung und zur Stimuluspräsentation getestet werden.

Zur Evaluierung der *Vision Engine*-Funktionen wurden drei unterschiedlich komplexe Szenen herangezogen. Zur Bestimmung ihrer Komplexität wurden folgende Indikatoren ausgewählt: die Anzahl der enthaltenen Texturen (a), die Anzahl der enthaltenen Modelle (b) sowie die Anzahl der von der *Vision Engine* verwendeten Index- und Vertex Buffers (c). Basierend auf diesen Indikatoren wurden die drei getesteten Szenen folgendermaßen klassifiziert: die *ViewerMap*-Szene als wenig detaillierte Szene, die *CrossingMap*-Szene als mittel detaillierte Szene und die *CityMap*-Szene als hoch detaillierte Szene.

Ausgehend von diesen Informationen wurden für den Test folgende Indikatoren zur Leistungsanalyse genutzt: Prozessorauslastung der *Central Processing Unit* (CPU) während der Eingabe der Experimentinformationen sowie während der Stimuluspräsentation (a), Prozessorauslastung der *Graphical Processing Unit* (GPU) während der Eingabe der Experimentinformationen sowie während der Stimuluspräsentation (b), Auslastung des Arbeitsspeichers vor und während der Stimuluspräsentation (c) sowie Auslastung der GPU vor und während der Stimuluspräsentation (d). Alle Tests bezogen

Komponente	Notebook-PC	Desktop-PC
Prozessor	Mobile Intel Core 2, P8400, 2.4 GHz	Intel Core 2, E8400, 3 GHz
Arbeitsspeicher	3048 MB RAM	3048 MB RAM
Netzwerkkarte	Broadcom NetLink Gigabit Ethernet	Realtek RTL 8168/8111 PCI-E Gigabit Ethernet NIC
Grafikkarte	Mobile Intel 4, 256 MB	Nvidia GeForce 9800GT, 512 MB
Betriebssystem	Microsoft Windows XP, Service Pack 3	Microsoft Windows XP, Service Pack 3
Parallelport- Device	Delock ExpressCard 66220 EPP	Standard EPP ATX 25-Pol DSUB

Tabelle 7.1: Die technischen Spezifikationen des zur Evaluierung verwendeten Notebook-Computer sowie des verwendeten Desktop-Computer. Zur Simulation einer Parallelport-Schnittstelle auf dem Notebook-Computer wurde zudem ein ExpressCard EPP-Gerät verwendet.

sich auf das Interaktionsverhalten mit dem VR-Stimulusframework, den Ladezeiten der 3D-Szenen, der Speicherauslastung des Arbeitsspeichers, der Speicherauslastung des GPU-Speichers sowie der Prozessorauslastung von CPU und GPU.

Zur Evaluierung der Funktionen für die Berechnung der Stimuluspräsentation sowie der semantischen Fehlerüberprüfung wurden insgesamt 24 Experimente, 12 ereigniskorrelierte und 12 Block-Design Experimente generiert und getestet. Hier sollten vor allem der Rechenaufwand in den involvierten Funktionen sowie die Speicherauslastung innerhalb des Experimentprotokolls evaluiert werden.

Der dritte Performanztest bezog sich auf TCP-Kommunikationsprotokoll aus der Datenzugriffsschicht. Da vor allem für Echtzeit-fMRT- und Neurofeedback-Experimente während des gesamten Experimentablaufs Daten empfangen und versendet werden, stand hier die Langzeitperformanz und die fehlerfreie Übertragung von Datenauswerteparametern im Vordergrund.

Aus sicherheitstechnischen Gründen sollte zur Evaluierung der Übertragungsfunktion zunächst jeder direkte Probandeneinfluss vermieden werden. Aus diesem Grund wurden insgesamt drei Echtzeit-fMRT-Experimente aufgezeichnet und für die Simulation im *Turbo BrainVoyager* vorbereitet. Jedes dieser Experimente beinhaltete 1000 fMRT-Volumen, wobei ein Einzelvolumen aus jeweils 32 Schichten bestand. Vor der Simulation des Experimentes wurden 10 zufällige *regions of interest* (ROI) selektiert. Aus diesen Regionen wurde im Folgenden jeweils ein Mittelwert aus den vorhandenen Grauwerten bestimmt und für das Senden über das TCP-Protokoll vorbereitet. Der T_R -Wert wurde gemäß der im IBMI und anderen Standort fMRT-Untersuchen mit 2s festgelegt. So konnten innerhalb eines Intervalls von 2s alle 10 Mittelwerte des aktuel-

len Volumens an die VR-Stimulusanwendung gesendet werden.

Um die Auslastung während eines Neurofeedback-Experimentes mit Echtzeit-Datenübertragung zu simulieren, wurde während dieses Sendeprozesses eine gleichzeitige Stimuluspräsentation mit der *CityMap*-Szene durchgeführt.

Um eventuelle zeitliche Verzögerungen aufzudecken, wurden die Eingangszeiten der empfangenen Werte durch die VR-Stimulusanwendung protokolliert und in einer Log-Datei abgespeichert. Zusätzlich dazu wurden die Funktionalitäten des Synchronitätssignals zwischen MRT und Stimuluspräsentation sowie die Funktionsweise der Tastersteuerung evaluiert.

7.4.2 Benutzerakzeptanztest

Um die Funktionalität und die Benutzerfreundlichkeit der VR-Stimulusanwendung zu evaluieren, wurde ein Benutzerakzeptanztest mit neun Wissenschaftlern durchgeführt. Alle Wissenschaftler waren mit der Durchführung von Neuroimaging-Experimenten vertraut und sollten neben Fragen zur beruflichen Qualifikation auch zwei Aufgaben zur Modellierung von Experimentabläufen mit der entwickelten VR-Stimulusapplikation bearbeiten.

Zu Beginn des Akzeptanztests erhielten alle neun Teilnehmer einen Fragebogen und einen Aufgabenzettel. Die Probanden mussten als erstes ihre neurowissenschaftlichen Grundkenntnisse einschätzen. Dazu wurden vier Fragen gestellt, welche die Probanden mittels einer Skala von eins (wenig Kenntnisse) bis 10 (exzellente Kenntnisse) beantworten sollten. Folgende Kriterien zur Bestimmung der Grundkenntnisse wurden herangezogen: Beschäftigungszeit im neurowissenschaftlichen Arbeitsbereich in Jahren (1), Kenntnisse bei der Erstellung von Experimentabläufen sowie deren ungefähre Anzahl (2), Erfahrung mit der Verwendung von neurowissenschaftlicher Stimulussoftware (3), Erfahrung mit der Verwendung von neurowissenschaftlicher Datenanalysesoftware (4).

Das Ziel dieser Eingangsbefragung war vor allem die Einschätzung der neurowissenschaftlichen Fähigkeiten aller Teilnehmer. Anhand dieser Einschätzung wurden die späteren Testresultate sowie die subjektive Bewertung der Software durch die Teilnehmer gewichtet und anschließend in Relation gesetzt. Die folgenden Wichtungen stehen in Beziehung zu den oben genannten Kriterien (1 - 4) und können anhand der Nummerierung miteinander verknüpft werden: (1) Beschäftigungszeit gewichtet mit Faktor 0.25, (2) Kenntnisse bei der Erstellung von Stimuli wurde abhängig von der Ja/Nein-Antwort (Faktor 0 oder 0.5) sowie mit der Anzahl erstellter Stimuli gewichtet (Faktor 2, 3 und 5), (3) Erfahrung mit Stimulussoftware wurde abhängig von der Antwort Nein/Ja/Ja mit Mehrfachnennung (Faktor 0, 0.25 oder 0.5) sowie mit der Anzahl der genannten Anwendungen gewichtet (Faktor 1 bis 10), (4) Erfahrung mit Datenauswertesoftware wurde abhängig von der Antwort Nein/Ja/Ja mit Mehrfachnennung (Faktor 0, 0.25 oder 0.5) sowie der Anzahl der genannten Anwendungen gewichtet (Faktor 1 bis 10).

Basierend auf den vier beschriebenen Eingangsfragen wurde zunächst für jeden Teilnehmer ein sogenannter *neurowissenschaftlicher Erfahrungswert* errechnet. Des Weiteren wurde zur Klassifizierung der Probanden ein Grenzwert festgelegt, welcher die

höchste, erreichbare Punktzahl eines fiktiven *unerfahrenen Anwenders* repräsentiert. Der Grenzwert erhielt für die Dauer des Nutzerakzeptanztests den Wert von 3.0 Punkten. Basierend auf diesem Grenzwert konnten die Nutzer anschließend in *unerfahrene Anwender* (Punktzahl < Grenzwert) und *erfahrene Anwender* (Punktzahl \geq Grenzwert) klassifiziert werden.

Alle Teilnehmer mussten innerhalb des Tests zwei Aufgaben lösen. Diese bezogen sich zum einen auf die Erstellung eines neurowissenschaftlichen Paradigmas und zum anderen auf die Adaption eines vorhandenen Experiments. Bei der Erstellung mussten die Teilnehmer ein VR-fMRT-Paradigma mit vordefinierten Werten in die VR-Stimulanwendung eingeben und anschließend dessen Funktionalität prüfen. Dabei waren zahlreiche Variablen bereits vorgegeben. Für eine genaue Übersicht zum Benutzerakzeptanztest sei hier auf Anhang B verwiesen.

Für die Adaption eines vorhandenen Paradigmas sollte zuerst eine vorhandene *Brain-Voyager Protokolldatei* (PRT-Datei) mit sämtlichen Experimenteinstellungen über die entwickelte Importfunktion eingeladen werden. Anschließend sollten komplexe Modifikationen am eingeladenen experimentellen Design vorgenommen und anschließend in angepasster Form, als neue PRT-Datei, abgespeichert werden.

Zum Abschluss des Benutzerakzeptanztests sollten alle Teilnehmer eine Bewertung zur Benutzerfreundlichkeit der Anwendung abgeben. Dazu wurden vier Indikatoren festgelegt: Anzahl der aufgetretenen Fehler während der Eingabe der Experimentinformationen (a), Anzahl der aufgetretenen Fehler während der Stimuluspräsentation (b), Bearbeitungszeit der beiden gestellten Aufgaben (c) sowie die subjektive Softwarebewertung des Probanden (d). Anhand einer Skala von eins (schlechte Benutzerfreundlichkeit) bis 10 (sehr gute Benutzerfreundlichkeit) konnten alle Probanden eine subjektive Bewertung zu den einzelnen Funktionalitäten vergeben. Kritische Applikationsfehler und einfache Funktionsfehler wurden ebenfalls im Rahmen dieses Tests registriert.

7.4.3 Ressourcennutzung und Performanztest

Stress- und Performanztest

Die Benutzerschnittstelle der Anwendung wurde in mehreren Performanz- und Stress-tests auf ihre Fehleranfälligkeit untersucht.

Sämtliche generierten Experimente im Block- oder ereigniskorrelierten Design konnten erfolgreich über die Komponenten der Benutzerschnittstelle eingegeben und syntaktisch überprüft werden. Die verschiedenen Registerkarten der Benutzerschnittstelle führten sicher durch den Erstellungsprozess des Paradigmas. Des Weiteren zeigte die Überwachung der Funktionsaufrufe und die Evaluierung der Speichernutzung keine Datenverluste oder Performanzeinbußen innerhalb der Präsentationsschicht weder für das Laptop- noch für das Desktop-Computersystem.

Während der Performanztests für die Applikationsschicht wurden die Funktionen zur Berechnung des Stimulusablaufs sowie die Interaktion zwischen externen Klassen der *Vision Engine* und Klassen zur Stimuluspräsentation evaluiert. Besonderes Augenmerk

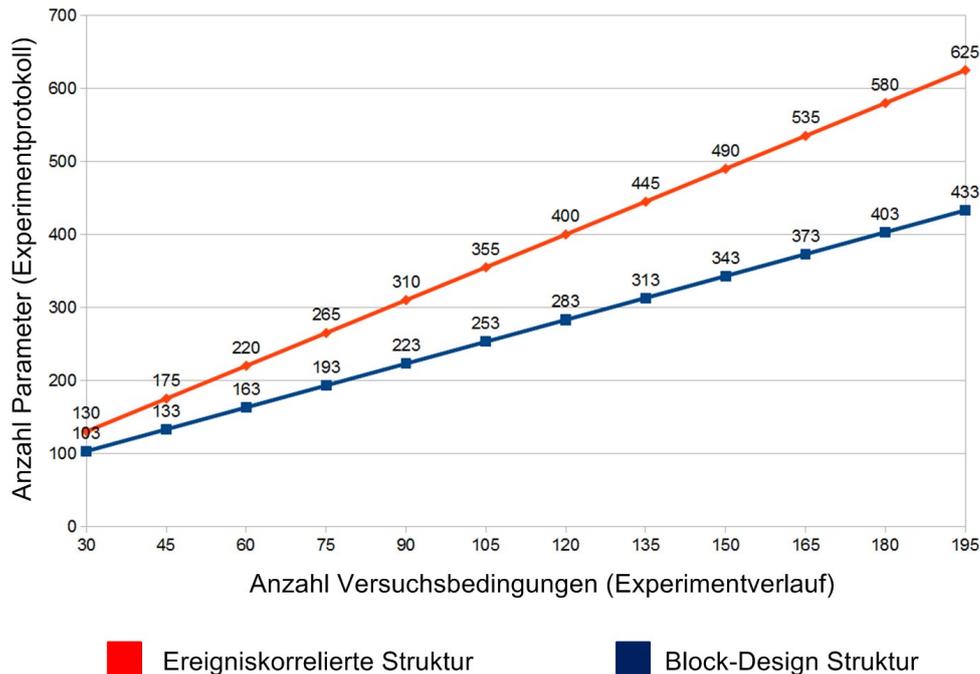


Abbildung 7.11: Die Anzahl der zu speichernden Experimentparameter variiert in Abhängigkeit der Experimentlänge (Anzahl der Versuchsbedingungen) und der verwendeten Experimentstruktur. Für ereigniskorrelierte Experimente steigt die Anzahl der Parameter über die Dauer der Messung schneller als bei Block-Design-Experimenten.

wurde auf die Ladezeiten der verschiedenen Szenen, auf die Prozessorauslastung und auf die Speicherauslastung gelegt. Hier zeigen die Ergebnisse eine Zeitersparnis beim zweiten Einladen einer Szene, welche aus den immer noch verfügbaren Daten des ersten Einladeprozesses resultiert. Alle Informationen der 3D-Szene stehen im Speicher der Grafikkarte sowie im Arbeitsspeicher des Rechners weiterhin zur Verfügung und beschleunigen so den Verarbeitungsprozess. Ein weiterer Unterschied zeigt sich bei der Performanz zwischen Laptop- und Desktop-Computersystem. Beide Computer benutzen im Test die gleiche Menge an Arbeitsspeicher. Allerdings zeigt sich wenig überraschend, dass der mit einer besseren Grafikkarte ausgerüstete Desktop-PC insgesamt eine weitaus bessere Performanz besitzt. Für die flüssige Präsentation von einfachen VR-Stimuli wird daher der Gebrauch einer leistungsstarken Grafikkarte mit mindestens 512 MB Speicherkapazität empfohlen.

Alle 24 generierten Experimentparadigmen konnten vom Experimentprotokoll sicher durch die Klasse *CalcStimulusTiming* eingelesen werden. Hier zeigte sich vor allem eine Diskrepanz in der Speicherbelegung zwischen ereigniskorrelierten und Block-Design Experimenten. Ist bei 30 Versuchsbedingungen die Speicherbelegung und die Anzahl der Experimentparameter zwischen ereigniskorrelierten und Block-Design Experimenten noch weitestgehend identisch, so driftet mit zunehmender Anzahl an Versuchsbedingungen die Anzahl der Parameter im Experimentprotokoll immer weiter auseinander (Abb. 7.11). Hier zeigt sich, dass ereigniskorrelierte Experimente weitaus komplexer sind und dass die Varianz in der Stimulusabfolge zu einem erheblichen

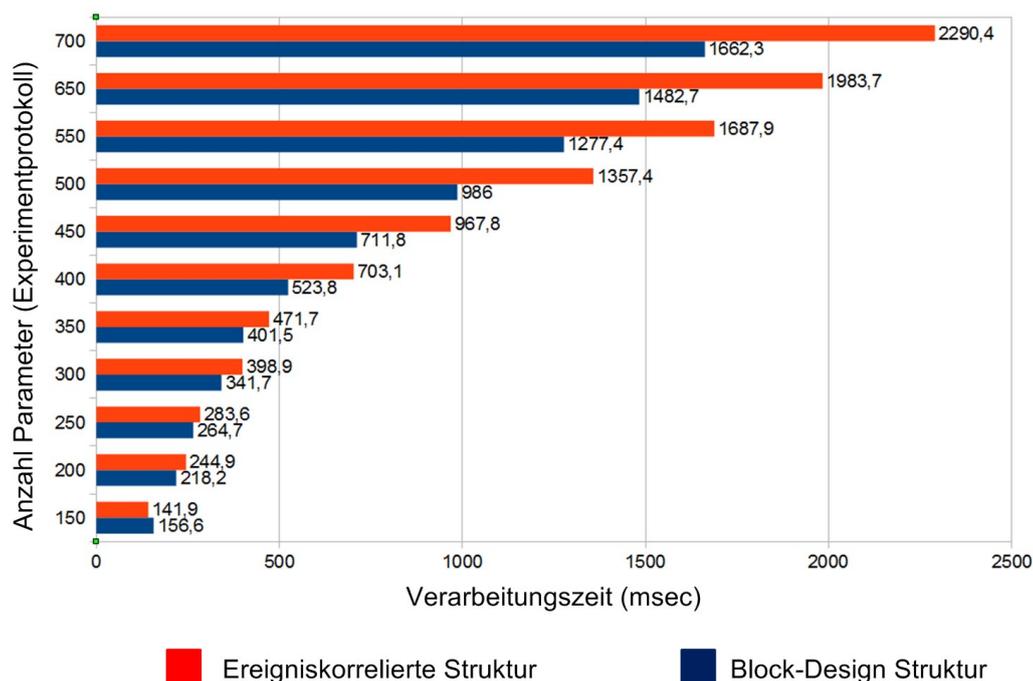


Abbildung 7.12: Die Länge bzw. Komplexität eines Experiments wirkt sich vor allem auf die Rechenzeit der semantischen Analyse sowie die Berechnung des Experimentablaufs aus. Die komplexere Experimentstruktur von ereigniskorrelierten Experimenten bedeutet zudem eine größere Anzahl von Experimentparametern, was sich mit zunehmender Experimentdauer in einer höheren Berechnungszeit widerspiegelt.

Mehraufwand in den Stimulusberechnungen und zu einem erhöhten Speicherbedarf führt (Abb. 7.12). Besonders auffällig ist hier der erhöhte Rechenaufwand für die syntaktische Fehlerüberprüfung bei ereigniskorrelierten Experimenten. Die Protokollierung der Funktionsaufrufe zeigt hier einen weitaus höheren Zugriff auf die einzelnen Stimulusinformationen des Experimentprotokolls. Ist beim Block-Design durch die konstante Wiederholung eines einzelnen Experimentblocks die Stimulusreihenfolge weitestgehend konstant, so zeigt sich bei ereigniskorrelierten Experimenten eine ständig variierende Stimulusreihenfolge, welche durch zusätzliche Informationen im Experimentprotokoll erfasst und für Stimulusberechnungen ständig abgerufen wird. Bei der Stimuluspräsentation sind diese Unterschiede allerdings nicht mehr präsent. Die Funktionen der Stimuluspräsentation konnten sowohl für ereigniskorrelierte als auch für Block-Design Experimente ohne Probleme auf die Stimulusberechnungen zugreifen und diese ohne zeitliche Verzögerungen exakt in der vorberechneten Zeit präsentieren. Des Weiteren zeigten die Funktionsaufrufe zwischen *Vision Engine*-Klassen und den Klassen der Stimuluspräsentation keine Interaktionsfehler. Alle protokollierten Zugriffe konnten ohne Probleme durchgeführt werden und lieferten sämtliche erforderlichen Übergabeparameter. Detaillierte Ergebnisse zum Performanztest der Applikationsschicht können in Tabelle 7.2 eingesehen werden.

Die TCP-Funktionalität der Datenzugriffsschicht wurde mit Hilfe von generierten Testdaten und mit echten Datenauswertinformationen aus Echtzeit-fMRT-Experimenten

Desktop-Computer						
Szene	Initialisierung (Sek.)		Speicherauslastung (MB)			CPU-Auslastung
	1.	2.	pVR	VR	VR-pVR	pVR → VR
„ViewerMap“	1.71	0.81	774	782	8	1 %
„CrossingMap“	2.82	0.54	772	869	97	14 %
„CityMap“	22.59	6.34	866	1221	355	32 %

Laptop-Computer						
Szene	Initialisierung (Sek.)		Speicherauslastung (MB)			CPU-Auslastung
	1.	2.	pVR	VR	VR-pVR	pVR → VR
„ViewerMap“	1.76	0.82	453	474	21	1 %
„CrossingMap“	2.62	0.94	458	664	206	17 %
„CityMap“	24.02	10.23	873	1286	413	37 %

Tabelle 7.2: Die Ergebnisse zeigen die Initialisierungszeiten der 3D-Szenen im ersten und zweiten Ladevorgang. Des Weiteren die Arbeitsspeicherauslastung vor der Szenenpräsentation (pVR), während der Szenenpräsentation (VR) und deren Differenz. Der Anstieg der Prozessorauslastung von pVR zu VR findet sich in der letzten Spalte.

auf eine fehlerfreie und zuverlässige Datenübertragung getestet. Die durch das Turbo BrainVoyager Plug-in definierten ROI ermöglichten zusammen mit der entwickelten TCP-Struktur die gleichzeitige Übertragung von mehreren gemittelten Grauwerten aus unterschiedlichen Signalverläufen innerhalb eines konstanten Intervalls von zwei Sekunden. So konnten zunächst in insgesamt drei simulierten Echtzeit-fMRT-Experimenten jeweils 1000 Grauwerte eines simulierten Voxel fehlerfrei versendet werden.

Des Weiteren wurden in allen drei Experimenten die gemittelte CPU-Auslastung während der Experimentmodellierung, während der alleinigen Stimuluspräsentation und während der Stimuluspräsentation mit gleichzeitigem Empfang der Auswertinformationen über TCP protokolliert (Abb. 7.13).

Hier zeigen die Ergebnisse einen Anstieg der CPU-Auslastung von der Modellierungsphase des Experiments zur Präsentationsphase mit der „Crossing-Map“ mit 18,8 Prozent beim Notebook-Computer und 11,1 Prozent beim Desktop-Computer. Des Weiteren ergab die Zuschaltung der TCP-Empfangsfunktion für Datenauswerteparameter nur einen im Vergleich geringen Anstieg der CPU-Auslastung. Für den Notebook-Computer wurde ein Anstieg der Prozessorauslastung um 3,5 Prozent und für den Desktop-Computer um 2,5 Prozent registriert. In allen Evaluationen zur Datenübertragung kam es zu keinen Verbindungsabbrüchen oder Datenverlusten.

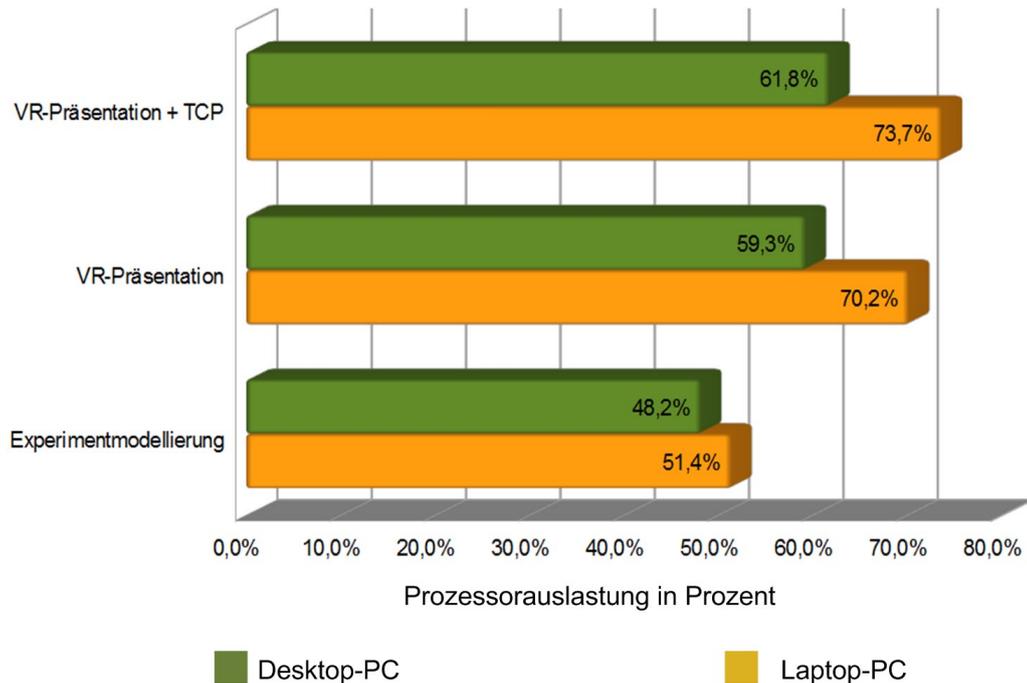


Abbildung 7.13: Beim Wechsel von Modellierungsphase zu Präsentationsphase ergeben sich für beide Computersysteme die höchsten Anstiege in der CPU-Auslastung. Dagegen zeigt die Zuschaltung der TCP-Übertragungsfunktion innerhalb der Präsentationsphase nur kleine Änderungen.

Ergebnisse - Benutzerakzeptanztest

Während des Benutzerakzeptanztests kam es zu keinen Stabilitätsverlusten oder Softwarefehlern. Alle experimentellen Einstellungen, entweder im ereigniskorrelierten oder Block-Design, konnten sicher erstellt und geladen werden. Die einzelnen Registerkarten führten die Anwender schrittweise durch den Modellierungsprozess und sorgten für eine wohlstrukturierte Experimentstruktur. Die Funktionen der syntaktischen Fehlerüberprüfung zeigten keine Implementierungsfehler und wiesen mittels verschiedener Warndialoge auf inkonsistente Experimenteinstellungen hin. Zudem vereinfachten implementierte Tooltips die Benutzerführung und wurden von vielen Testern positiv aufgenommen.

Alle Teilnehmer des Benutzerakzeptanztests konnten die Aufgaben erfolgreich bearbeiten. Gemessen an dem Ziel, die Benutzerfreundlichkeit sowohl für unerfahrene Anwender als auch für erfahrene Anwender zu bestimmen, wurden anhand des definierten Grenzwerts zunächst vier der neun Teilnehmer als unerfahrene Anwender und fünf als erfahrene Anwender klassifiziert (Tabelle 7.3).

Für die Durchführung des Benutzerakzeptanztests wurde keine feste Bearbeitungszeit definiert. Jedoch wurden zu Beginn des Tests alle Teilnehmer über die Protokollierung der Bearbeitungszeit informiert. Bei den gemessenen Bearbeitungszeiten zeigte sich zwischen beiden Gruppen kein signifikanter Unterschied. Ebenso zeigten die Ergebnisse zur Bewertung der Benutzerfreundlichkeit in Abhängigkeit des Erfahrungswertes

Anwender	Erfahrungsbewertung (Punkte)				Gesamt
	Beschäftigungszeit	Modellierung	Präsentation	Auswertung	
01	2.0	1.0	1.5	1.5	6.0
02	1.25	2.5	5.0	3.5	12.25
03	1.5	2.5	4.0	4.0	12
04	1.25	2.5	4.0	4.5	12.25
05	0.5	0.5	0.75	0	1.75
06	0.38	0.5	0	1.0	1.88
07	0.75	0.5	1.0	0	2.25
08	0.38	0.5	0	0.75	1.63
09	1.0	2.5	4.0	4.0	11.5

Tabelle 7.3: Das Gesamtergebnis der Eingangsbefragung zur Feststellung der Grundkenntnisse. Der Grenzwert zur Einteilung in unerfahrene und erfahrene Anwender wurde mit 3.0 Punkten festgelegt.

keine signifikanten Unterschiede zwischen erfahrenen und unerfahrenen Anwendern. Des Weiteren besaßen alle Teilnehmer die Option, am Optimierungsprozess der Anwendung teilzunehmen und über die schriftliche Befragung negative oder positive Auffälligkeiten anzumerken.

Mehrere Teilnehmer beschrieben hier die Benutzeroberfläche als gut strukturiert und die Tooltips zur Experimentmodellierung als sehr hilfreich. Zusätzlich werteten einige Teilnehmer den schnellen Import/Export von Experimentparadigmen über *Brain-Voyager* Protokoll-Dateien (PRT-Datei) und die automatische Berechnung von Block-Design Parametern als sehr positiv und hilfreich.

Kritikpunkte betrafen vor allem die optische Anzeige verschiedener Oberflächenkomponenten. So wurden aufgrund dieser Anmerkungen das fehlende, optische Hervorheben der derzeit aktiven Registerkarte sowie die fehlende Fortschrittsanzeige des Experimentmodellierungsprozesses nachträglich implementiert.

Anhand der Ergebnisse der Eingangsbefragung wird deutlich, dass Anwender 07 und Anwender 08 in zwei Bereichen keine Grundkenntnisse aufwiesen. Zudem zeigte sich durch verschiedene schriftliche Anmerkungen, dass der geführte Modellierungsprozess und die Tooltips die mangelnden Vorkenntnisse im Bereich der Experimentmodellierung bei diesen beiden Probanden nicht vollständig ersetzen konnte. Hier muss erneut geprüft werden, ob die Anwendung trotz Einstiegshilfen für Anwender ohne Grundkenntnisse geeignet ist. Alle Kritikpunkte und Verbesserungsvorschläge wurden vollständig geprüft und werden für eine kommende Aktualisierung der Stimulusapplikation berücksichtigt.

Kapitel 8

Evaluation des Stimuluskonzepts im Rahmen einer Pilotstudie

Das erfolgreiche Evaluierung des Zusammenwirkens aller entwickelten Teilaspekte im Rahmen einer Pilotstudie gilt als das Hauptziel dieser Arbeit. In diesem Kapitel sollen die zwei Anwendungsszenarien, welche im Rahmen der Pilotstudie zur Evaluation der VR-Stimulanwendung durchgeführt wurden, näher vorgestellt werden:

- die Planung, Durchführung und Auswertung eines VR-fMRT-Experiments mit der entwickelten VR-Stimulusapplikation, bei dem die neuronalen Mechanismen der räumlichen Navigation sowie des topographisch-episodischen Gedächtnisses (vgl. Kapitel 3) untersucht werden, und
- die sichere Validierung des entwickelten und bis dato neuen Echtzeit-Adaptionskonzeptes zur unmittelbaren Anpassung von VR-Stimuli mit MR-Daten aus dem ersten Anwendungsszenario sowie die Evaluierung möglicher unbekannter Einflussfaktoren auf Neurofeedback-Prozessabläufe.

Für die Durchführung der Pilotstudie wurde zunächst der komplette VR-fMRT-Experimentablauf innerhalb der entwickelten VR-Stimulusapplikation entworfen.

Hierfür wurde eine Virtuelle Umgebung eingeladen und mit dem modellierten experimentellen Setup ausführlich getestet. Nach dem erfolgreichen Testen der Experimentabläufe erfolgte die Durchführung der VR-fMRT-Experimente mit mehreren Probanden (Mueller et al. 2012b). Zudem wurden alle aufgezeichneten MR-Bilddaten direkt von der Echtzeit-Datenauswertesoftware *Turbo BrainVoyager* ausgewertet und gesichert.

Im Hinblick auf die Validierung des neuartigen Echtzeit-Adaptionskonzeptes mussten verschiedene sicherheitstechnische Aspekte und Richtlinien berücksichtigt werden. Zunächst durften aufgrund des Medizinproduktegesetzes nicht ohne weiteres Änderungen innerhalb der Siemens MR-Infrastruktur oder an deren Komponenten vorgenommen werden. Des Weiteren wollte man zur sicheren Evaluierung des Konzeptes zuerst das technische Setup optimieren und mögliche Belastungen für Probanden minimieren. Um dieses Konzept dennoch mit Probandendaten zu testen, entschied man sich für die

Simulation eines Echtzeit-fMRT-Experiments mit der *Turbo BrainVoyager*-Software und den zuvor aufgezeichneten Daten des VR-fMRT-Experiments. Für detaillierte Informationen zur Simulation des Echtzeit-fMRT-Experiments sei auf Abschnitt 8.2.3 und Abschnitt 8.2.4 verwiesen.

In den nachfolgenden Unterkapiteln sollen die technischen Spezifikationen, der Aufbau des Versuchsparadigmas und die spezifischen Parameter, die bei der Echtzeit- sowie bei der post-experimentellen Datenauswertung verwendet wurden, vorgestellt werden. Dieses Kapitel endet mit den Ergebnissen und der abschließenden Diskussion zu den Erkenntnissen beider Experimente. Die Ergebnisse dieser Studie mit beiden Anwendungsszenarien sowie die beinhaltenden technischen Konzepte wurden bereits an verschiedenen Stellen veröffentlicht (Mueller et al. 2010b; 2011a;b; 2012a;b).

8.1 Problembeschreibung

Das Ziel dieser Arbeit ist die Anwendung der entwickelten VR-Stimulusapplikation zur Identifizierung neuronaler Aktivitäten und kognitiver Mechanismen zur räumlichen Orientierung sowie zum episodischen und topographischen Gedächtnis. Durch die Erforschung dieser Netzwerke erhofft man sich Erkenntnisse auf dem Gebiet der altersbedingten Degeneration des räumlichen Gedächtnis oder bei der Bekämpfung neurodegenerativer Erkrankungen, wie z.B. Morbus Alzheimer (Pine et al. 2002).

Für die Erforschung topographischer Navigation und für das überwachte Training zur Bekämpfung altersbedingter, degenerativer Erkrankungen mit bildgebenden Verfahren sind VR-Stimuli gerade zu prädestiniert. So können trotz der Immobilität des Probanden im MRT Virtuelle Umgebungen zur Simulation von Bewegungs- oder Orientierungsabläufen und damit auch zur Aufzeichnung von relevanten Hirnaktivitäten verwendet werden.

Dieser vermeintliche Vorteil der VR-Präsentation kann allerdings auch zu nachteiligen bzw. komplexeren Experimentabläufen führen, bei denen sich neuronale Aktivierungen überlagern können. Neuronale Aktivierungen, die sich aus solchen Überlagerungen zusammensetzen, sind nur sehr schwer zu interpretieren (Maguire 1998, Burgess et al. 2002). Typische Überlagerungseffekte bei der visuellen Stimulation mit VR-Szenarien sind Tiefenwahrnehmungseffekte oder Effekte zur räumlichen Wahrnehmung (Wolbers et al. 2008, Baecke et al. 2009).

Um solche Störeffekte zu vermeiden, müssen im experimentellen Ablauf sämtliche Versuchsbedingungen und die zugehörigen Stimuli klar strukturiert sowie durch ausreichend lange Ruhebedingungen voneinander getrennt werden. Erst dadurch können spezifische neuronalen Aktivierungen den entsprechenden Hirnprozessen sicher zugeordnet werden.

Das zweite Anwendungsszenario dieser Studie, die interaktive Präsentation sowie die Adaption einer dynamischen VR-Präsentation mit Hilfe von Neurofeedback-Techniken, ist ein neuer Ansatz, welcher aufgrund der komplexen technischen Voraussetzungen nur wenig beachtet wurde (Mueller et al. 2012b). Die Grundlage dieses Anwendungs-

szenarios bildet die zeitkritische Datenanalyse durch Echtzeit-fMRT-Auswertetechniken. Die verschiedenen Einsatzgebiete der Online- bzw. Echtzeit-fMRT-Datenauswertung sowie deren Vor- und Nachteile wurden in Kapitel 3.8 bereits vorgestellt.

Für dieses Anwendungsszenario ist vor allem die Evaluierung des VR-Adaptionskonzepts zur Anwendung in zukünftigen Neurofeedback-Studien von entscheidender Bedeutung. Da es aufgrund dieses neuartigen Anwendungsszenarios nur wenig vergleichbare Studien gibt, muss auf die sichere Evaluierung des modellierten Experimentablaufs und die Validierung des entworfenen Adaptionsalgorithmus unter kontrollierten Bedingungen besonders großer Wert gelegt werden.

Zusätzlich dazu sollten die zeitkritischen Prozessabläufe der Echtzeit-Analyse, die Echtzeit-Übertragung des Neurofeedback-Signals und die Interpretation des Signals durch den Adaptionsalgorithmus evaluiert und auf mögliche Auswirkungen für klassische Neurofeedback-Strategien untersucht werden.

Dieses Anpassungskonzept unterstützt durch seine flexible Struktur die echtzeitnahe Adaption einer komplexen virtuellen Welt durch die Kombination von multiplen Neurofeedback-Signalen aus verschiedenen Hirnregionen. Zusätzlich dazu erlauben die implementierten Funktionen die Präsentation von effektiveren Stimuli sowie die bessere Anpassung des Paradigmas an die physiologischen Besonderheiten des Probanden. Die Evaluierung der ROI-Signalübertragungsfunktion und die Signalinterpretation innerhalb des VR-Adaptionsalgorithmus sind hier für die Durchführung zukünftiger Neurofeedback-Studien von entscheidender Bedeutung.

Detaillierte Informationen zum bereits vorgestellten Echtzeit-Adaptionskonzept finden sich in Kapitel 6.5 sowie in Kapitel 7.2.2. Zusätzlich dazu finden sich in Kapitel 3.5.3 weitere Ausführungen zu konventionellen Lösungen echtzeitnaher Experimentadaption.

8.2 Methodik und Materialien

In diesem Unterkapitel sollen die Methoden und Materialien der beiden durchgeführten Anwendungsszenarien vorgestellt werden. Im Zentrum stehen die technischen Spezifikationen der MRT-Infrastruktur, der experimentelle Ablauf des VR-fMRT-Experiments, die Methoden zur Evaluierung des VR-Adaptionskonzeptes und die Parameter der Echtzeit- sowie der post-experimentellen Datenanalyse.

8.2.1 Technischer Aufbau und Datenakquirierung

Alle Probandenexperimente wurden an einem Siemens „Trio“ Ganzkörper-MRT (Feldstärke: 3 Tesla, Numaris Syngo VA35, Siemens Medical Systems, Erlangen, Deutschland) an der Otto-von-Guericke-Universität Magdeburg, Klinik für Neurologie, durchgeführt. Zur Akquirierung der Bilddaten diente eine acht-Kanal Kopfspule sowie eine angepasste Messequenz zur Unterstützung des Echtzeit-Bilddatenexports.

Diese modifizierte EPI-BOLD Messequenz enthielt zusätzlichen Programmcode, welcher für den unmittelbaren Export des jeweiligen Datenvolumens sorgte (Weiskopf et al. 2007). Zunächst wurden alle Probanden mittels einer hochauflösenden T_1 -gewich-

Parameter	Wert
TR (<i>ms</i>)	2000
TE (<i>ms</i>)	30
Flipwinkel ($^{\circ}$)	77
Voxelgröße (mm) ³	3.0
Auflösung	64 x 64
Schichtenanzahl	34
Bandbreite (<i>Hz/px</i>)	1816

Tabelle 8.1: Die wichtigsten Parameter zur Durchführung des VR-fMRT-Experiments am 3T-MRT.

teten Messung untersucht. Die so gewonnenen Daten ermöglichten die präzise Zuordnung anatomischer Merkmale eines jeden Probanden bei der post-experimentellen statistischen Auswertung. Für alle funktionellen Messungen wurden Gradienten-Echo-EPI-Sequenzen eingesetzt. Innerhalb von 2s konnte damit der gesamte Kopfbereich mit einer Auflösung von $(3mm)^3$ aufgenommen werden (Tabelle 8.1). Der genaue Ablauf der Messung wird bei der Beschreibung des Experimentes dargestellt. Die erzeugten Bilddaten wurden sofort nach der Akquirierung durch die Siemens-eigenen Algorithmen bewegungskorrigiert und in Echtzeit auf die lokale Festplatte des Host-PC (MRT-Steuerrechner) exportiert. Alle fMRT-Bilddaten wurden vorher im geschützten Netzwerk des MRT-Systems freigegeben.

Der technische Aufbau des Echtzeit-fMRT-Systems sowie die Datenströme richten sich nach dem in Kapitel 6.1 beschriebenen Entwurf und sind in Abbildung 6.1 dargestellt. Die Anwendung *Turbo BrainVoyager* (Brain Innovation 2002-2006) wurde auf dem externen Rechner zur Datenauswertung (*Statistic-PC*) ausgeführt. Sie greift über das lokale Netzwerk direkt auf die freigegebenen Bilddaten zu und analysiert diese in Echtzeit. Die VR-Stimuluspräsentation erfolgt über einen separaten Rechner (*Stimulus-PC*), welcher visuelle Informationen via Beamer und über ein System von fest installierten Spiegeln dem Probanden anzeigt. Die Tastendruck-Ereignisse des Probanden werden über eine spezielle Glasfaserschnittstelle aus dem MRT-System an den *Stimulus-PC* übertragen und dort über eine USB-Schnittstelle empfangen und protokolliert. Detaillierte Informationen zur Stimuluspräsentation und zum VR-Paradigma finden sich im nachfolgenden Abschnitt (vgl. Abschnitt 8.2.2). Der *Stimulus-PC* und der *Statistic-PC* sind innerhalb dieser Versuchsstruktur über das implementierte TCP-Protokoll miteinander verbunden und können verschiedene Informationen darüber austauschen. Der *Turbo BrainVoyager* übernimmt in dieser globalen Struktur die Rolle der Echtzeit-Datenauswertung und sendet während der Experimente Auswertedaten zur VR-Stimulusanwendung, wo sie verarbeitet und interpretiert werden. Eine detaillierte Beschreibung der TBV-Einzelprozesse findet sich in Abschnitt 8.2.3.

8.2.2 Probanden und Paradigma

Im Rahmen dieser Studie wurden insgesamt 12 gesunde Probanden gemessen. 11 Probanden (23-31 Jahren, 6 weiblich, 5 männlich) wurden für die anschließende statistische Datenauswertung herangezogen und gaben schriftlich ihre Einwilligung zur Durchführung der Experimente, welche in Übereinstimmung mit den Bestimmungen der Ethikkommission der Medizinischen Fakultät der Universität Magdeburg durchgeführt wurden.

Sämtliche Einstellungen zum Experimentparadigma und die Präsentation des Versuchs wurden mit der implementierten VR-Stimulusanwendung durchgeführt.

Vor dem Beginn des Experiments erhielten alle Probanden eine grundlegende Einführung zu ihrer Versuchsaufgabe, jedoch nicht zum Forschungsziel des Experiments. Alle Probanden sollten innerhalb der Studie einen Avatar aus der Ich-Perspektive steuern und diesen durch ein künstliches, drei-dimensionales Stadtszenario navigieren. Hauptaufgabe innerhalb dieses Versuchs war die Erkundung und das Einprägen der baulichen Besonderheiten dieser virtuellen Welt. Während der Erkundung wurde der Proband durch keine Einschränkungen behindert. Der Proband konnte anhand seiner Entscheidungen die virtuelle Umgebung frei erkunden. Der Avatar wurde durch das Drücken von vier MRT-konformen Tastern gesteuert. Durch das Drücken der entsprechenden Taster konnte der Avatar folgende Bewegungen ausführen: vorwärts gehen, rückwärts gehen, nach links drehen, nach rechts drehen. Die verwendete, virtuelle Szene „City-Map“ wurde von Havok Inc. zur Verfügung gestellt und bestand aus drei miteinander verbundenen Straßenzügen, sowie mehreren Häusern und einigen Hinterhöfen (Abb. 8.1).

Das Experimentdesign des Versuchs wurde mit der implementierten VR-Stimulusapplikation prä-experimentell entworfen und für die Durchführung des Experiments sowie zur post-experimentellen Auswertung in eine *BrainVoyager*-konforme PRT-Datei abgespeichert.

Um die Daten dieses Experiments auch für die spätere Evaluierung des entwickelten VR-Adaptionskonzeptes zu nutzen, wurde das Experiment nach den Anforderungen des im Abschnitt 6.5 beschriebenen Experimentablaufs modelliert.

Hier musste zunächst die parametrische Varianz in der neurophysiologischen Reizantwort bei unterschiedlichen Probanden berücksichtigt werden. Aufgrund dessen wurden zwei aufeinanderfolgende Messungen entworfen. Zunächst fand eine Kalibrierungsmessung zur Lokalisierung der neuronalen Aktivität des jeweiligen Probanden und zur Berechnung von repräsentativen Signaländerungen für alle Ruhe- und Stimulusphasen statt.

Nach dieser Kalibrierungsmessung wurde die eigentliche Hauptmessung des Experiments durchgeführt. In ihr fand die topographische Navigation und die Aufzeichnung der Auswertedaten für die spätere Evaluierung des Adaptionsalgorithmus statt.

Das Experimentablauf für Kalibrierungs- und Hauptmessung bestand aus einem kategorialen Block-Design und beinhaltete zwei Versuchsbedingungen: Ruhebedingung (a) und Navigationsbedingung (b). Für die Ruhebedingung wurden 4 Messungen und

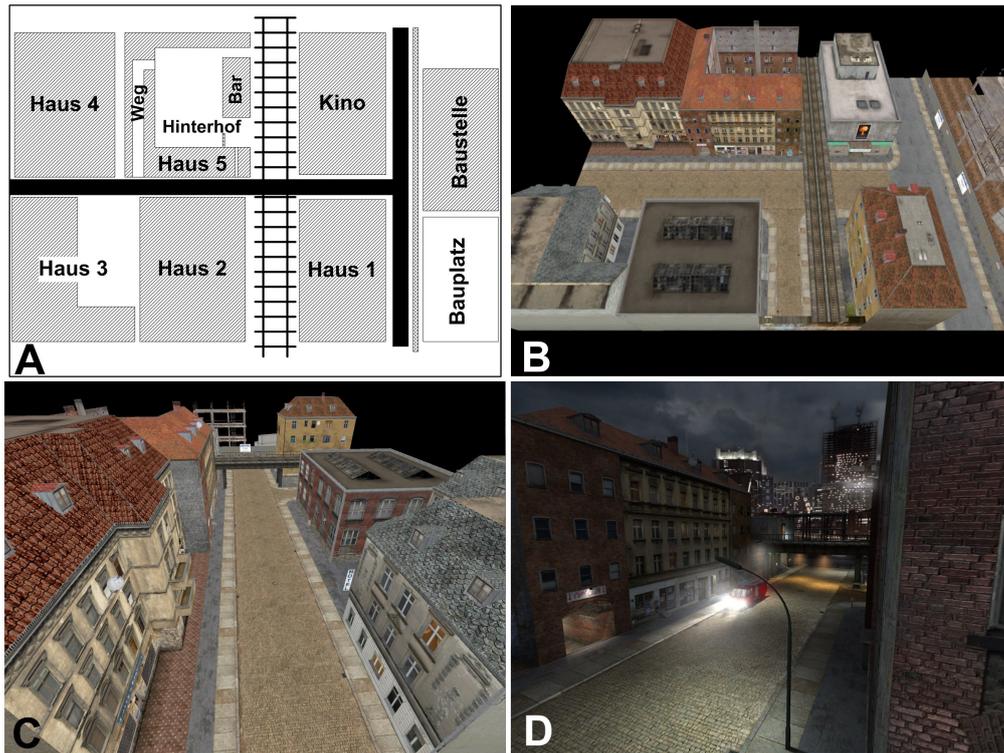


Abbildung 8.1: Die vereinfachte Grundstruktur des VR-Szenarios setzt sich aus verschiedenen Gebäuden und Straßenzügen zusammen (A). Die perspektivische Ansicht von oben (B) sowie die Straßenansicht (C) der modellierten Szene verdeutlichen die optischen Details und dienen als Basis für das Rendern der Szene im Experiment (D).

für die Aktivbedingung 7 Messungen eingeplant (Abb. 8.2). Aufgrund des Intervalls zwischen den Aufnahmen (T_R -Parameter) ergaben sich so die zeitliche Dauer der Ruhebedingung (8s) und die zeitliche Dauer der Aktivbedingung (14s) sowie die Gesamtdauer eines Stimulusblocks (22s).

Dieser modellierte Stimulusblock sollte in der Kalibrierungsmessung 6 mal und in der Hauptmessung 12 mal wiederholt werden (Abb. 8.2). Die Kalibrierungsmessung mit ihren 66 Aufnahmen im Abstand von 2s nutzt somit die dieselben Parameter wie die Hauptmessung (132 Messungen).

Während der Ruhephase wurde dem Probanden ein rotes Kreuz mit schwarzem Hintergrund angezeigt. Hier sollte der Proband nur das rote Kreuz fixieren und sämtliche motorischen Aktivitäten einstellen. Während der Einblendung dieses statischen Bildes waren alle Tastereingaben und Interaktionsmöglichkeiten des Avatars mit der virtuellen Umgebung gesperrt. Nach dem Ende der Ruhephase schloß sich das statische Bild und der Proband bekam die Möglichkeit innerhalb der virtuellen Welt zu navigieren (Abb. 8.2).

In der folgenden Navigations- bzw. Explorationsphase (Aktivbedingung) steuerte der Proband den Avatar für 14 Sekunden durch die künstliche Umgebung und prägte sich so die baulichen Besonderheiten der virtuellen Welt ein. Nach dem Ende der Aktivbedingung wurde die Navigation unterbrochen und es erfolgte die erneute Einblendung

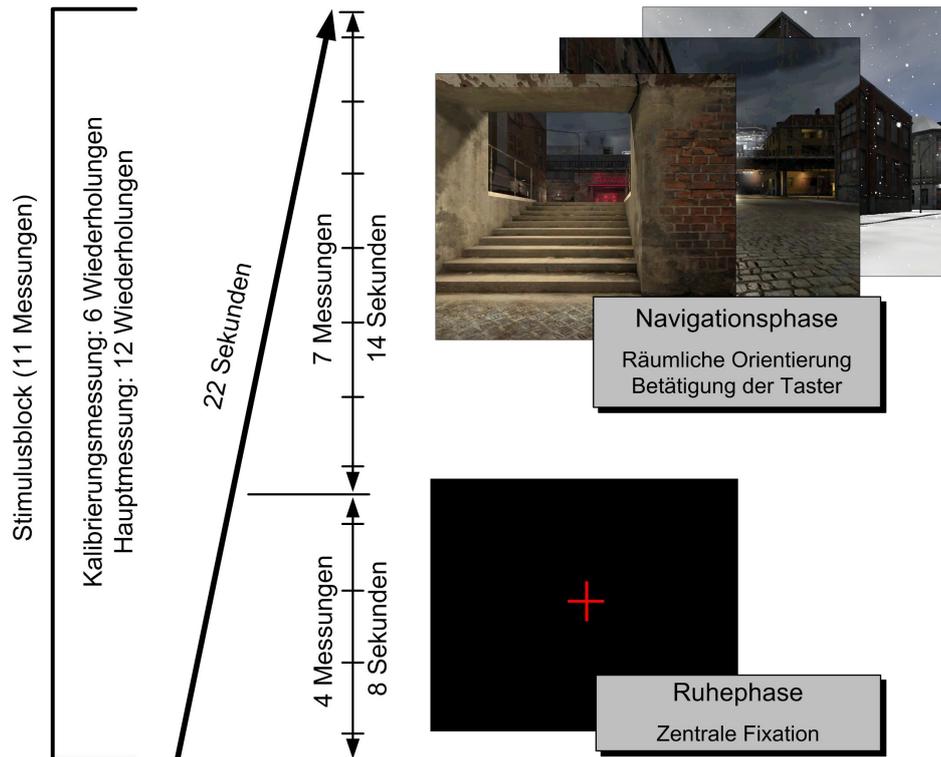


Abbildung 8.2: Der Stimulusblock des VR-fMRT-Experiments, welcher mehrmals wiederholt wird, startet mit der zentralen Fixation und Ruhephase (acht Sekunden). Die anschließende Navigationsphase (14 Sekunden) erlaubt das Navigieren innerhalb des VR-Szenarios durch Drücken der Taster.

des roten Kreuzes. Die Avatarposition blieb währenddessen unverändert, so dass der Proband zum neuen Stimulusblock die Erkundung von der vorherigen Position fortsetzen konnte.

Das Ziel der Experimente lag dabei auf der Kontrastuntersuchung zwischen Ruhephase und Aktivphase. Die Betrachtung der neuronalen Aktivität sollte in erster Linie die relevanten Hirnareale für das räumliche Gedächtnis identifizieren (Aguirre 1998, Burgess et al. 2002) sowie die Funktionalität der VR-Stimulusanwendung sicherstellen. Allen Probanden wurde nach der Durchführung des Experiments ein Fragebogen vorgelegt. Darin sollten etwaige Erfahrungen aus dem Bereich der Computerspiele und Erfahrungen über bisherige fMRT-Experimenten angegeben werden. Des Weiteren zielten die verschiedenen Fragen auf eine Bewertung und einen Vergleich des VR-Experiments mit herkömmlichen funktionellen fMRT-Experimenten ab. Weitere Fragen betrafen das Enkodieren der VR-Welt, das Entspannen zwischen den Navigationsphasen und die Bewertung der freien Exploration innerhalb der Virtuellen Welt (vgl. Anhang A).

Das Design dieses Paradigmas orientiert sich an vorhandenen VR-Studien zur räumlichen Navigation (Aguirre 1998, Burgess et al. 2002). Eigene Aspekte zur Steigerung des Interaktionsverhaltens wurden zusätzlich eingefügt.

8.2.3 Integration des Turbo BrainVoyager

Die kommerzielle Anwendung *Turbo BrainVoyager* (TBV) ist eine Echtzeit-Datenauswertesoftware, die im Rahmen der Pilotstudie zur Berechnung und Übermittlung von Datenauswerteeinformationen und zur Visualisierung funktioneller MRT-Datensätze genutzt wurde.

Des Weiteren besitzt die TBV-Software eine Simulations- und Datenaufzeichnungsfunktion, die es erlaubt, eine Echtzeit-fMRT-Studie auch post-experimentell erneut durchlaufen zu lassen. Auf diese Weise können die Ergebnisse der Echtzeit-Datenauswertung und deren Präsentation erneut geprüft und mit den Ergebnissen einer Offline-Auswertung verglichen werden.

Während der VR-fMRT-Studie wurden diese Funktionalitäten genutzt, um eine Echtzeit-Datenauswertung durchzuführen und um Daten für die spätere Evaluierung des Echtzeit-Adaptionskonzeptes aufzuzeichnen (vgl. Abschnitt 8.2.4).

Während der VR-fMRT-Studie lief der TBV auf dem separaten *Statistic-PC*, welcher im geschützten *Local Area Network* (LAN) der MRT-Infrastruktur eingebunden war. Die Funktionen der Echtzeit-Datenauswertung analysierten die bereitgestellten MR-Bilddaten und übermittelte die Auswerteparameter in Echtzeit an die VR-Stimulanwendung.

Sämtliche VR- und Kommunikationsereignisse wurden für die Offline-Analyse protokolliert und für die nachfolgende Evaluierung des Echtzeit-Adaptionskonzeptes im zweiten Anwendungsszenario verwendet.

Das zum Betrieb des TBV notwendige Experimentdesign wurde vor Beginn der Experimente erstmalig mit der VR-Stimulussoftware erstellt und anschließend in einer *BrainVoyager*-typischen Protokolldatei (PRT-Datei) abgespeichert. Die Protokolldatei diente im Folgenden sowohl für die VR-Stimulanwendung als auch für den *Turbo BrainVoyager* als Informationsträger.

Während der Experimente nahm der TBV via LAN ständigen Lesezugriff auf die neu akquirierten Volumendatensätze und analysierte diese in Echtzeit.

Die Analyse der Daten gliederte sich in zwei Phasen. Die erste Phase beinhaltete diverse Echtzeit-Vorverarbeitungsschritte, die bereits in Abschnitt 5.4.4 vorgestellt wurden. In der zweiten Phase wurde die eigentliche statistische Analyse der Daten durchgeführt. Hier verwendete der TBV das rekursive Allgemeine Lineare Modell (ALM).

Für die Visualisierung der aktivierten Hirnareale sowie die gleichzeitige Auswahl von mehreren Arealen zur Aktivierungswertanalyse stellt der TBV verschiedene Visualisierungsmöglichkeiten zur Verfügung und erlaubt so die Selektion und Betrachtung mehrerer „regions of interest“ (ROI) mit unterschiedlichen Voxelensembles. Durch das entwickelte und implementierte VR-TCP-Kommunikationsprotokoll wurde eine Möglichkeit geschaffen, die Werte dieser multiplen Signalverläufe aus unterschiedlichen ROI gleichzeitig und in Echtzeit zur VR-Stimulusapplikation zu übermitteln. Diese Funktionalität sollte im Rahmen der bevorstehenden Echtzeit-Datenauswertung ebenfalls evaluiert werden.

Zum Export der Auswerteeinformation pro ROI wurde auf die interne Mittlungsfunktion des TBV zurückgegriffen und ein gemittelter Grauwert pro selektierter ROI be-

stimmt. Hier wurde von allen in einer ROI enthaltenen Voxel das arithmetische Mittel der Grauwerte errechnet und im Folgenden an die implementierte Datenkommunikationsfunktion weitergegeben.

Durch die bereitgestellte Plugin-Schnittstelle des TBV konnte eine eigens dafür programmierte TCP-Kommunikationsfunktion in den TBV eingebunden und genutzt werden (Luehrs et al. 2011).

Dieses Plugin kam erstmalig in dieser Studie zum Einsatz und übernahm den Echtzeit-Transfer der gemittelten Grauwerte zur VR-Stimulusanwendung. Das entwickelte VR-TCP-Kommunikationsprotokoll zur Übertragung multipler Neurofeedback-Signalverläufe wurde sowohl im TBV-Plugin als auch im VR-Stimulusframework eingebunden (vgl. Abschnitt 7.1.1).

Das Ziel der Versuchsanordnung war vor allem das Sicherstellen und Testen der VR-Stimulusapplikation innerhalb einer Echtzeit-fMRT-Infrastruktur. Darin inbegriffen waren das Testen des Kommunikationskonzeptes zur sicheren Übertragung, die zuverlässige Verarbeitung der Auswertinformationen innerhalb der VR-Stimulusapplikation und die korrekte Protokollierung der eingegangenen Daten.

8.2.4 Echtzeit-Adaptionsframework

Für die Validierung des neuartigen Echtzeit-Adaptionsframeworks mussten verschiedene sicherheitstechnische Aspekte und Richtlinien berücksichtigt werden. Zunächst durften aufgrund des Medizinproduktegesetzes keine Änderungen innerhalb der Siemens MR-Infrastruktur oder an deren Komponenten vorgenommen werden. Des Weiteren wollte man zur sicheren Evaluierung des Konzeptes zuerst das technische Setup optimieren und mögliche Belastungen für Probanden und Patienten minimieren. Um dieses Konzept mit Probandendaten zu testen, nutzte man die Simulationsfunktion der *Turbo BrainVoyager*-Software. Auf deren Basis wurde die Echtzeit-fMRT-Studie erneut mit den zuvor aufgezeichneten Daten des VR-fMRT-Experiments simuliert (vgl. Abschnitt 8.2.3). Im Anschluss wurden die aufgezeichneten Probandendaten erneut ausgewertet und die resultierenden Auswertergebnisse über das VR-TCP-Protokoll zur VR-Stimulusapplikation übermittelt.

Die Funktionen des entwickelten Adaptionskonzeptes erhielten dadurch die notwendigen Auswertinformationen, welche im weiteren Verlauf zur Signalinterpretation verwendet wurden (vgl. Abschnitt 7.2.2).

Für die Berechnung der Referenzwerte wurden die fMRT-Daten der Kalibrierungsmessung mit 66 Messungen verwendet. Die eigentliche VR-Stimulusadaption auf Basis der Referenzwerte erfolgte anschließend in der definierten Hauptmessung mit 132 Messungen.

Im Hinblick auf zukünftige Neurofeedback-Studien, bei denen die Selbstregulierung neuronaler Aktivität durch die Kombination multipler Signalverläufe aus unterschiedlichen Hirnregionen angestrebt wird, sollten drei voneinander getrennte und funktionell unabhängige ROI definiert werden. Basierend auf der TBV-Anzeige der aktivierten Hirnregionen wurden drei gleichgroße ROI definiert und für die Signalextraktion ausgewählt. Die ROI wurden jeweils im rechten Motorischen Kortex (ROI 0), im linken Primären Auditorischen Kortex (ROI 1) und im medialen Präfrontalen Kortex (ROI 2)

definiert (Abb. 8.5 (C)). Der benutzerdefinierte Toleranzwert k , welcher für das Entscheidungskriterium des Adaptionsalgorithmus notwendig ist, wurde für alle Probanden mit 1,5 % Signaländerung festgelegt. Zur Initialisierung des Experimentablaufs wurde zudem die PRT-Protokolldatei des VR-fMRT-Experiments verwendet, welche durch die Importfunktion des TBV eingeladen wurde.

Die gesicherte Übertragung durch die TCP-Verbindung wurde vor dem Experiment sichergestellt.

8.3 Ergebnisse

Die VR-Stimulusapplikation konnte problemlos in die Siemens 3T „Trio“ MRT-Infrastruktur integriert werden. Keiner der Probanden beschrieb Probleme bei der Navigation oder beim Einprägen der topographischen Charakteristika innerhalb der VR-Umgebung. Sämtliche bereitgestellten Bedienelemente zeigten keinerlei Fehlfunktionen oder Beeinträchtigungen bei der Durchführung des Experiments.

8.3.1 VR-fMRT-Experiment - Echtzeit-Datenauswertung

Für die Evaluierung des VR-Adaptionsalgorithmus und die spätere Überprüfung der Offline-Datenauswertung wurden während der VR-fMRT-Experimente sämtliche Echtzeit-Datenauswerteergebnisse und Echtzeit-Protokolle des TBV aufgezeichnet.

Im Rahmen der Versuchsvorbereitung konnte zunächst der mit der VR-Stimulusapplikation prä-experimentell entwickelte Experimentablauf sicher in die TBV-Software eingeladen und die Kommunikationsfunktionen zur Übertragung von Datenauswerteeinformationen zur VR-Stimulusapplikation geprüft werden.

Alle aufgenommenen fMRT-Bilddaten wurden in Echtzeit vorverarbeitet und mittels des rekursiven Allgemeinen Linearen Modells (ALM) ausgewertet. Zeitgleich erfolgte die Präsentation der neuronalen Aktivität in Echtzeit (Abb. 8.3) sowie das Senden der gemittelten Grauwerte aus den drei selektierten ROI zur VR-Stimulusanwendung. Die genaue Lage der definierten ROI wurde bereits in Kapitel 8.2.4 beschrieben und ist in Abbildung 8.5 (C) dargestellt. Aufgrund der Protokollierungsdaten und der empfangenen Datenauswerteeinformationen konnte die Funktionsweise des implementierten VR-TCP-Kommunikationsprotokolls zur gleichzeitigen Übermittlung von multiplen Neurofeedback-Signalverläufen sichergestellt werden (vgl. Abschnitt 6.5 und Abschnitt 8.2.3). Während des Sendevorgangs traten keine Probleme bei der Datenübermittlung auf. Alle gesendeten Grauwerte konnten erfolgreich empfangen und zuverlässig im Netzwerkkommunikationsprotokoll der VR-Stimulusanwendung aufgezeichnet werden.

Die Ereignisprotokolle der VR-Stimulusapplikation protokollierten zudem sämtliche für die spätere Offline-Datenanalyse wichtigen VR-Ereignisse sowie die verschiedenen Berechnungsschritte des VR-Adaptionsalgorithmus mit den Grauwerten. Alle Probanden zeigten während der Experimente neuronale Aktivitäten im okzipitalen Bereich des Gehirns, im Cuneus, im Precuneus, im Mittleren Temporalen Gyrus und in

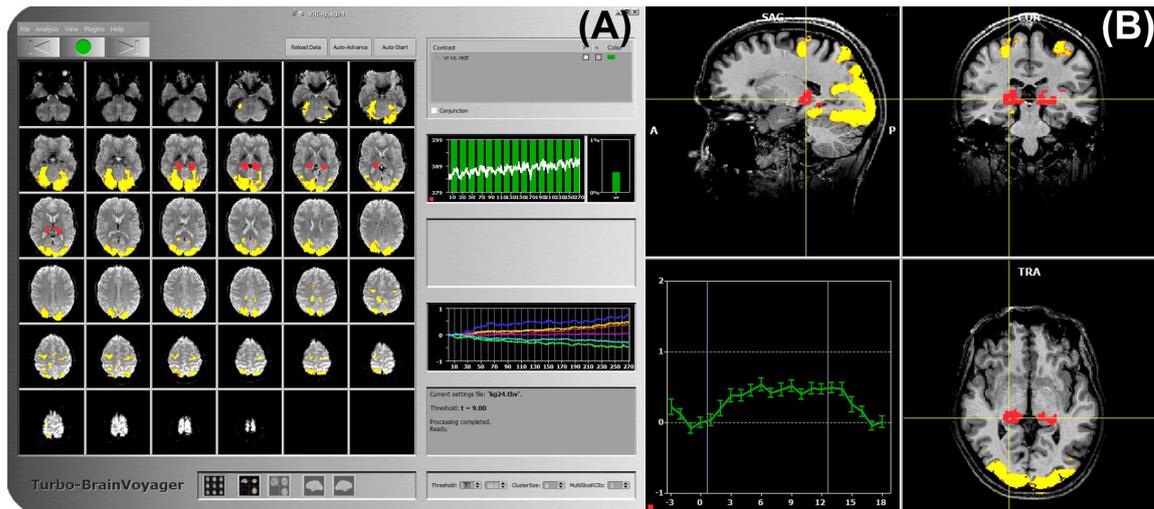


Abbildung 8.3: Die Echtzeit-Datenauswertung und die Präsentation der neuronalen Aktivität durch den TBV zeigt Aktivierungen im Cuneus, Precuneus und in den okzipitalen Arealen (A). Die detaillierte Ansicht zeigt zudem die neuronale Aktivität in den parahippokampalen Bereichen des Gehirns (B).

den parahippokampalen Bereichen. Die Ergebnisse dieser Auswertung wurden im Folgenden durch die post-experimentelle Datenauswertung sichergestellt.

8.3.2 VR-fMRT-Experiment - Post-experimentelle Datenanalyse

Zusätzlich zur Echtzeit-Datenauswertung mit dem *Turbo BrainVoyager* wurde zur eigentlichen Evaluierung der aufgenommenen Probandendaten eine post-experimentelle statistische Analyse mit der Datenauswerteanwendung *BrainVoyager QX 2.2* durchgeführt. Hier wurden alle mit der EPI-Sequenz akquirierten Bilddaten auf die T_1 -gewichtete anatomische Aufnahme koregistriert.

Die anatomische Aufnahme wurde zuvor normalisiert sowie in den von Talairach und Tournoux beschriebenen stereotaktischen Raum überführt.

Nach dieser Normalisierung wurden alle EPI-Bilder mit einem sogenannten FWHM-Gauss-Kernel (FWHM = full width at half maximum) von 4 mm räumlich geglättet.

Ein Allgemeines Lineares Modell wurde auf den zeitlichen Verlauf der Messung angewendet. Anhand des kategorialen Block-Design konnten so die Ruhe- und Navigationsphasen gegenübergestellt sowie die neuronale Aktivität des jeweiligen Probanden angezeigt werden. Die gemittelten Änderungen der Hirnaktivität aller Probanden wurden in einer Gruppenanalyse mit einem Fixed-Effect-Modell analysiert. Aufgrund der hohen Anzahl von Voxel innerhalb einer fMRT-Messung tritt das statistische Problem der multiplen Einzelvergleiche auf. Daher wurden sämtliche Aktivierungsbereiche mit dem FDR-Verfahren (*False Discovery Rate*) analysiert. Die signifikanten Bereiche ($p < 0.05$) sind in Abb. 8.4 dargestellt.

Signifikante Aktivierungen wurden über alle Probanden bilateral in frontalen, in parietalen und in medialen Bereichen des Gehirns festgestellt. Aufgrund der Gesamtzahl der signifikanten, aktivierten Voxel sowie des maximalen Clusterwertes zeigte sich ei-

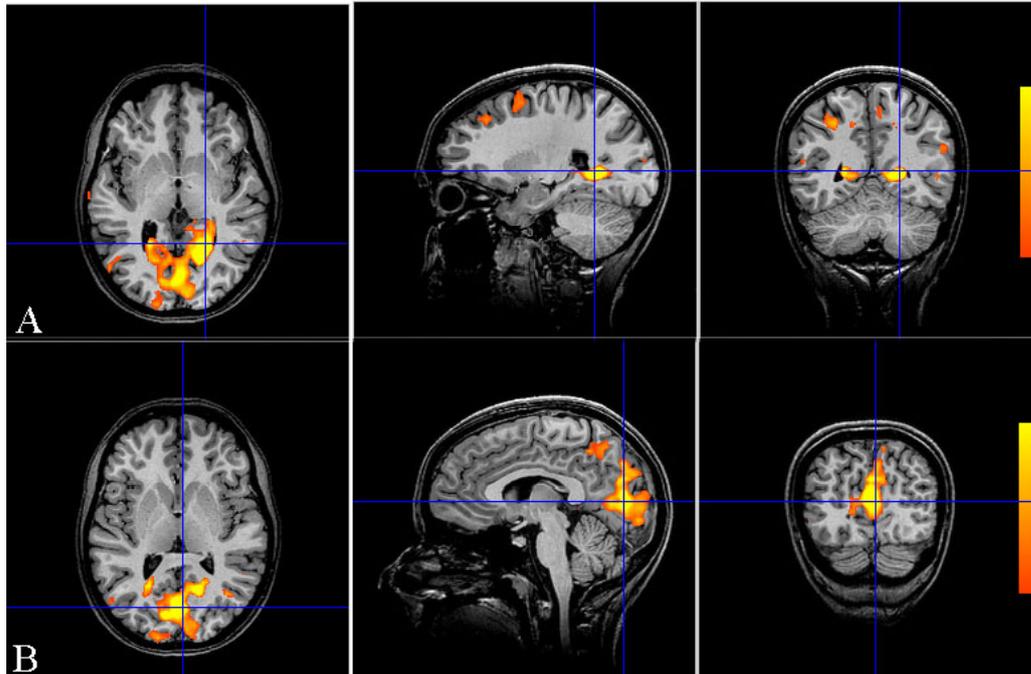


Abbildung 8.4: Die Resultate der Fixed-Effect Gruppenanalyse zeigen neuronale Aktivität im rechten und linken parahippokampalen Bereich (A) sowie im Lingualen Gyrus, im Cuneus, im Precuneus und dem Mittleren Temporalen Gyrus (B) (Mueller et al. 2012b).

ne leichte rechtsseitige Dominanz der Aktivierungen. Eine detaillierte Übersicht der aktivierten Hirnregionen bietet Tabelle 8.2.

In den okzipitalen Bereichen waren Aktivierungen beidseitig im Gyrus Lingualis, beidseitig im Cuneus sowie linksseitig im Gyrus Parahippocampalis sichtbar. Diese beschriebenen Aktivierungskuster weiteten sich in Richtung des Mittleren Okzipitalen Gyrus und des rechten Parahippokampalen Gyrus aus. Zusätzlich dazu wurden einige Aktivierungen im medialen Temporallappen registriert. Linksseitig zeigten sich Aktivierungskuster die vom Parahippokampalen Gyrus bis zur Intersektion mit dem Gyrus Fusiformis reichten. Signifikante geschlechtsspezifische Unterschiede konnten nicht festgestellt werden.

Die Ergebnisse des post-experimentellen Probandenfragebogens zeigten darüber hinaus eine hohe Akzeptanz des VR-Paradigmas bei den Probanden. Alle Versuchsteilnehmer hatten bereits früher an regulären fMRT-Experimenten teilgenommen und besaßen bereits diverse Erfahrungen im Umgang mit Computerspielen. Viele Versuchsaaspekte wurden von den Probanden als sehr positiv erachtet. Im Folgenden sind diese kurz zusammengefasst:

- hohe Interaktion mit dem Paradigma und der virtuellen Umgebung
- hohe Bewegungsfreiheit und die Möglichkeit selbstständig die Umgebung zu erforschen
- anfänglich hohe Aufmerksamkeit aufgrund vieler neuer Reize (permanente saliente Reize)

Region	Koord.			Voxel	Max-Wert	Brodmann Areal
	X	Y	Z			
RH Gyrus Lingualis	-21	-54	4	1125	10.144	18
LH Gyrus Lingualis	-19	-61	2	6	5.707	19
RH Cuneus	5	-90	10	16	5.262	18
LH Cuneus	-1	-75	12	491	8.768	23
LH Gyrus Parahippocampalis	-22	-52	4	77	6.160	30
RH Precuneus	6	-60	38	57	4.854	7
RH Mittlerer Gyrus Frontalis	48	8	41	114	5.316	8
RH Mittlerer Gyrus Occipitalis	36	-64	7	16	5.060	19
LH Mittlerer Gyrus Occipitalis	-51	-70	4	48	4.383	19
LH Gyrus Fusiform	-40	-61	-14	7	3.569	37

Tabelle 8.2: Die Tabelle zeigt die Ergebnisse der Fixed-Effect Gruppenanalyse des durchgeführten VR-fMRT-Experiments. Alle aufgelisteten Hirnregionen und Aktivierungscluster wurden FDR-korrigiert ($p < 0.05$) und zeigten während des räumlichen Einprägens verglichen zur Ruhe-Phase signifikante Aktivierungen (Mueller et al. 2012b).

- authentische und realitätsnahe „Welt“ (immersives VR-Szenario)

Des Weiteren wurden einige weniger gut empfundene Aspekte genannt:

- zu kleine Ausdehnung der virtuellen Welt
- Reduktion der Aufmerksamkeit nach der vollständigen Erkundung der VR-Umgebung

8.3.3 Echtzeit-VR-Stimulusadaption - Datenanalyse

Für die Initialisierung des Experimentablaufs wurden die bereits verwendeten Experiment- und Paradigmeninformationen des VR-fMRT-Experiments genutzt. Hierfür griff der TBV auf dieselbe PRT-Protokolldatei zu und lud die Experimentparameter in seine Prozessstruktur ein.

Zudem zeigten die Messungen der Übertragungszeiten keine Auffälligkeiten. Sämtliche Datenpakete konnten sicher übertragen werden.

Kalibrierungsmessung

Aufgrund der Evaluierung des VR-Adaptionkonzeptes durch eine simulierte Neurofeedback-Studie ohne direkten Probandeneinfluss musste für die Kalibrierungs- und Hauptmessung auf die Aspekte der Probandeninteraktion und der Selbstregulierung neuronaler Hirnaktivität durch operante Konditionierung verzichtet werden. Da aufgrund dieses Umstands eine Selbstregulierung neuronaler Aktivität mittels multipler Signalverläufe aus unterschiedlichen Hirnregionen nicht möglich war, entschied man sich zunächst für die Evaluierung des Konzeptes durch einen einzelnen Signalverlauf. Hierfür wurde aufgrund des beobachteten Signalverlaufs aus der post-experimentellen

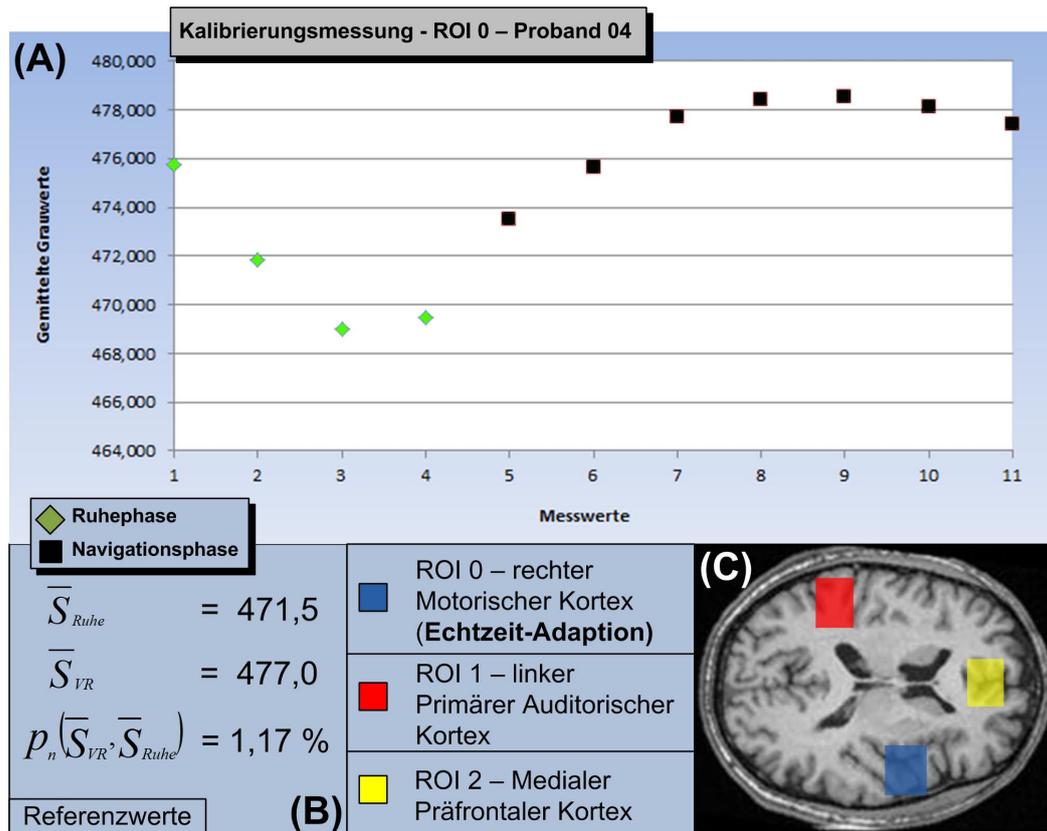


Abbildung 8.5: Exemplarisch für die Berechnung der Referenzwerte aller Probanden zeigt die Abbildung die zugrundeliegenden Informationen von Proband 04. Drei Hirnareale (ROI) wurden zur Neurofeedback-Signalanalyse für alle Probanden herangezogen (C). Als repräsentativ gilt der gemittelte Signalverlauf für ROI 0 (blaues Rechteck) (A), aus dem die wichtigsten Referenzwerte abgeleitet wurden (B) (Mueller et al. 2012a).

Datenauswertung die Hirnregion ROI 0 im rechten Motorischen Kortex ausgewählt sowie die Signalverläufe in ROI 1 und ROI 2 zur späteren Analyse aufgezeichnet (Abb. 8.5 (C)). Aufgrund dessen beziehen sich sämtliche nachfolgenden Berechnungen von Referenzwerten sowie die spätere Echtzeit-Analyse und der Vergleichsprozess auf die Hirnregion ROI 0. Detaillierte Informationen zur Auswahl der Hirnregionen finden sich in Abschnitt 8.2.4.

Nach dem erfolgreichen Synchronstart der Experimentsimulation durch den TBV und der Neurofeedback-Signalanalyse durch die Stimulusapplikation konnten beide Anwendungen die Neurofeedback-Parameter sicher senden und empfangen. Hier ergaben sich für die 66 Messwerte der Kalibrierungsmessung insgesamt 66 Verbindungsaufrufe, welche jeweils durch ein zwei Sekunden Intervall voneinander getrennt waren. Pro Verbindungsaufwurf wurden jeweils zeitgleich die drei gemittelten Grauwerte aus den drei definierten ROI für den aktuellen Volumendatensatz versendet. Alle 198 Parameter pro Probandenmessung konnten so sicher durch das implementierte VR-TCP-Übertragungsprotokoll übermittelt werden.

Nach Ende der Kalibrierungsmessung wurden die empfangenen Grauwerte des ersten

Stimulusblocks über alle ROI des Block-Design-Experiments automatisch zusammengefasst und gemittelt. Die entsprechenden Algorithmen wurden hier bereits in Kapitel 7.2.2 vorgestellt und sind in Abb. 7.4 abgebildet.

Aufgrund der spezifischen Struktur als Block-Design-Experiment mit einem repetitiven Ablauf, wiederholte der Algorithmus diese Prozedur für alle weiteren Stimulusblöcke und berechnete anschließend anhand der gemittelten Werte aller korrespondierenden Stimuli den gemittelten Signalverlauf für jeden einzelnen Probanden. Exemplarisch dafür ist in Abb. 8.5 (A) der gemittelte Signalverlauf des Probanden 04 (ROI 0) dargestellt.

Anschließend konnte der so gewonnene Signalverlauf zur Berechnung der repräsentativen Signaländerung p_n von Ruhephase zu Navigationsphase im ersten Stimulusblock verwendet werden (Abb. 8.5 (B)).

Ausgehend davon ergab sich zudem der zugrundeliegende Wertebereich für den eigentlichen VR-Adaptionsprozess in der Hauptmessung, welcher anhand des benutzerdefinierten Toleranzwertes k durch die untere Schwelle ($p_n - k$) und die obere Schwelle ($p_n + k$) eingegrenzt wird.

Diese Berechnungsprozesse konnten für alle 12 Probanden erfolgreich durchgeführt werden. Die ermittelten Referenzwerte wurden anschließend innerhalb des internen Experimentprotokolls und als weitere Replik innerhalb des Dateiformats *VREXP* abgespeichert, siehe Kapitel 7.2.3.

Hauptmessung

Analog zur Kalibrierungsmessung ergaben sich pro Hauptmessung 132 Verbindungsaufrufe in denen insgesamt 396 Parameter übermittelt wurden.

Im Unterschied zur Kalibrierungsmessung wurden hier die Parameter unmittelbar nach Erhalt des letzten zum Stimulusblock gehörenden Wertes in Echtzeit zusammengefasst und gemittelt. Alle beteiligten Verarbeitungsprozesse konnten hier erfolgreich und ohne Probleme auf die empfangenen Grauwerte zugreifen und diese verarbeiten.

Des Weiteren wurde nach der Mittlung der Werte im zweiten Stimulusblock auch die Echtzeit-Analyse zugeschaltet, welche die aktuellen Signaländerungen zwischen Ruhe- und Navigationsphase nun nach jedem weiteren Stimulusblock berechnete. Genauere Informationen zur mathematischen Verarbeitung der Daten finden sich in Kapitel 7.2.2.

Nach der Berechnung der aktuellen Signaländerungen wurde der Echtzeit-Vergleichsprozess gestartet. Hier konnte der Algorithmus erfolgreich auf die abgespeicherten Referenzwerte der Kalibrierungsmessung im Experimentprotokoll zugreifen und diese mit den aktuellen Signaländerungen zwischen Ruhe- und Navigationsphase vergleichen. Der Zugriff auf die Replik der Referenzdaten innerhalb der *VREXP*-Datei war hier nicht notwendig.

Zusätzlich dazu konnte im Rahmen des Vergleichsprozesses der vom Anwender definierte Toleranzwert k fehlerfrei in das zugrundeliegende Entscheidungskriterium, welches in Kapitel 7.2.2 Formel (8.5) vorgestellt wurde, eingebunden werden. Der Toleranzwert k blieb während dieser Experimente konstant und definierte zusammen mit dem Referenzwert p_n die untere sowie die obere Schwelle des zum Auslösen des Adapti-

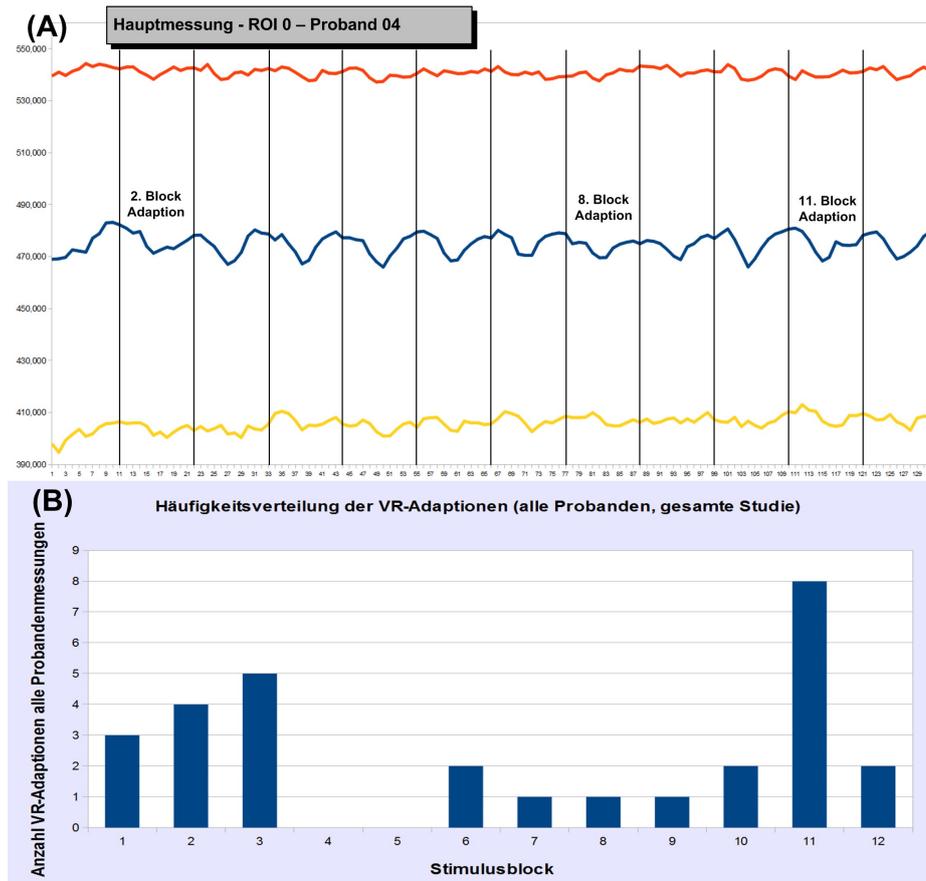


Abbildung 8.6: In der Hauptmessung analysiert der Algorithmus gleichzeitig die Signalverläufe aus mehreren ROI. Im vorliegenden Experiment wurde die VR-Stimulusadaption anhand der ausgewählten ROI 0 (blaue Kurve) durchgeführt (A). Die Signalverläufe in ROI 1 und ROI 2 (rote und gelbe Kurve) wurden nicht zur VR-Adaption herangezogen. Die Übersicht zur Häufigkeit von Stimulusadaptionen erlaubt das Aufdecken von Experimentstrukturschwächen oder möglichen Aufmerksamkeitsreduktionen, z.B. häufige Unterschreitung des Entscheidungskriteriums in Stimulusblock 11 in 9 von 12 Probandenexperimenten (B) (Mueller et al. 2012a).

onsprozesses notwendigen Wertebereichs.

Exemplarisch für den VR-Adaptionsprozess sind in Abbildung 8.6 (A) der Signalverlauf in ROI 0, ROI 1 und ROI 2 sowie die Zeitpunkte für detektierte Über- oder Unterschreitungen des Wertebereichs für Proband 04 dargestellt. Ausgehend vom Entscheidungskriterium des Echtzeit-Vergleichsprozesses wurde hier zu den drei markierten Zeitpunkten (ROI 0) eine nicht im Wertebereich liegende Signaländerung detektiert (Abb. 8.6 (A)). Als Resultat wurde eine parallel zur Stimulusadaption arbeitende Prozesskette aufgerufen, welche vor Beginn der nächsten Explorationsphase eine Anpassung des VR-Szenarios vornahm.

Die Gesamtergebnisse aller Probandendaten mit den wichtigsten Referenzwerten aus der Kalibrierungsmessung sowie den Signaländerungen und Adaptionszeitpunkten in der Hauptmessung sind in Tabelle 8.3 dargestellt. Hier wurden die verschiedenen Re-

Kalibrierungsmessung (ROI 0)				
Probanden	$\bar{S}_{Ruhe} \pm SD$	$\bar{S}_{VR} \pm SD$	p_n	$[p_n \pm k]$
01	485,1 ± 3,1	491,1 ± 2,2	1,24	1,09 – 1,39
02	505,6 ± 2,7	512,8 ± 2,4	1,42	1,27 – 1,57
03	492,7 ± 2,5	499,2 ± 2,2	1,32	1,17 – 1,47
04	471,5 ± 2,7	477 ± 1,7	1,17	1,02 – 1,32
05	492,4 ± 3,2	498,8 ± 2,2	1,3	1,15 – 1,45
06	491,5 ± 2,9	497,8 ± 2,0	1,28	1,13 – 1,43
07	502,8 ± 2,7	509,9 ± 2,2	1,41	1,26 – 1,56
08	471,9 ± 2,8	477,5 ± 2,2	1,19	1,04 – 1,34
09	484,1 ± 3,3	490 ± 2,4	1,22	1,07 – 1,37
10	501,2 ± 2,6	507,5 ± 2,1	1,25	1,1 – 1,4
11	510,3 ± 2,4	517 ± 1,9	1,31	1,16 – 1,46
12	487,4 ± 2,9	493,7 ± 2,2	1,29	1,14 – 1,44

Hauptmessung - VR-Adaption (ROI 0)				
Probanden	p_a (Stimulusblock - SB)			
01	1,04 (SB 03)	0,93 (SB 09)	1,03 (SB 12)	
02	1,25 (SB 11)			
03	1,11 (SB 01)	1,49 (SB 02)		
04	0,87 (SB 02)	0,69 (SB 08)	0,99 (SB 11)	
05	0,82 (SB 03)	0,95 (SB 11)		
06	1,09 (SB 02)	1,12 (SB 06)	1,06 (SB 10)	1,02 (SB 11)
07	1,21 (SB 01)	1,02 (SB 10)	1,11 (SB 11)	
08	1,01 (SB 03)	1,36 (SB 07)		
09	1,05 (SB 03)	1,04 (SB 06)	1,01 (SB 11)	1,05 (SB 12)
10	1,42 (SB 01)	1,08 (SB 11)		
11	1,02 (SB 02)			
12	0,98 (SB 03)	1,23 (SB 11)		

Tabelle 8.3: Die Gesamtergebnisse der wichtigsten Referenzwerte für ROI 0 aus der Kalibrierungsmessung (obere Tabelle) sowie die Zeitpunkte der VR-Adaption (Stimulusblock) mit der zugehörigen Signaländerung p_a in der Hauptmessung (untere Tabelle). Das Entscheidungskriterium greift auf die Referenzwerte aus der Kalibrierungsmessung zurück und analysiert die ermittelten Signaländerungen in der Hauptmessung. Über- oder Unterschreitungen des Wertebereichs $[p_n \pm k]$ führen hier zu Stimulusadaptionen vor Beginn der nächsten Navigationsphase (Mueller et al. 2012a).

ferenzwerte sowie die ermittelten Signaländerungen mit den Adaptionzeitpunkten in der Hauptmessung stichprobenartig überprüft. Die manuell errechneten Ergebnisse ergaben keinerlei Abweichungen zu den algorithmisch errechneten Referenzwerten und den vom Entscheidungskriterium detektierten Über- oder Unterschreitungen des Wertebereichs.

Die automatische Analysefunktion der Echtzeit-VR-Stimulusadaption verwendete die ermittelten Daten zur Generierung einer Häufigkeitsverteilung für alle durchgeführten VR-Stimulusadaptionen über alle Probanden. Die Analysefunktion markierte für alle Probanden sämtliche Adaptionzeitpunkte und deren spezifische Referenzwerte. Hier zeigte sich, dass vor allem im 3. Stimulusblock bei 5 und im 11. Stimulusblock bei 9 Probanden häufige Über- bzw. Unterschreitung des Wertebereichs $[p_n \pm k]$ auftraten (Abb. 8.6 (B)). So wurden verschiedene Aufmerksamkeitsreduktionen bei den Probanden sichtbar, die eine zu lange Gesamtmessdauer und eine zu kurze Dauer der Explorationsphase annehmen lassen. Spezifische Neurofeedback-Einflussfaktoren, die auf eine Untauglichkeit des Adaptionssystems in Neurofeedback-Experimenten hindeuten, wurden nicht gefunden.

8.4 Diskussion der Experimente

Das erste Ziel dieser Studie war die Anwendung der entwickelten VR-Stimulusapplikation zur Erforschung und Identifizierung kognitiver Mechanismen der topographischen Navigation sowie des räumlichen und semantischen Gedächtnisses. Das zweite Ziel dieser Studie bezog sich auf die Anwendung eines neuartigen Echtzeit-VR-Stimulusadaptionkonzepts in Neurofeedback-Experimenten bzw. in einer adaptiven Hirn-Computer-Schnittstelle. Hier sollte anhand eines Anwendungsszenarios die Frage geklärt werden, ob eine Nutzung von dynamischen VR-Stimulusreizen in Neurofeedback-Studien sowie deren Echtzeit-Anpassung durch Signalverläufe aus multiplen Hirnregionen durchführbar ist.

Zu diesem Zweck wurden während der Studie sämtliche Funktionen der VR-Stimulusapplikation und des integrierten VR-Adaptionframeworks erfolgreich evaluiert und getestet. Zuvor durchgeführte Performanztests garantierten die Funktionsfähigkeit der Stimulusapplikation (vgl. Abschnitt 7.4.3) und erlaubten damit die Evaluierung der Anwendung innerhalb eines Experiments mit mehreren Probanden.

Im ersten Anwendungsszenario wurden die neuronalen Aktivitäten der räumlichen Navigation und des topographisch-episodischen Gedächtnisses erfolgreich mit den entwickelten Funktionen der VR-Stimulusapplikation und deren Ereignisprotokollen untersucht.

Eine Vielzahl von VR-fMRT-Studien nutzten zur Stimuluspräsentation 3D-Modelle aus bestehenden Anwendungen und mussten in Folge dessen das experimentelle Setup an die Vorgaben des VR-Szenarios anpassen (Aguirre et al. 1996, Maguire 1998, Burgess et al. 2002, Pine et al. 2002, Andreano et al. 2009). Bei der Analyse dieser Studien wird deutlich, dass die verwendeten VR-Umgebungen viele Freiheiten durch die selbständige Exploration der topographischen Besonderheiten erlaubten, aber im Gegenzug auf die Protokollierung von auftretenden VR-Ereignissen während der Messung,

wie z.B. Bewegungen und Interaktionen des Probanden mit der Umgebung, verzichteten. Aufgrund dessen konnten spezifische Ereignisse, die innerhalb eines bestimmten Zeitraums stattfanden, nur schwer oder gar nicht mit den entsprechenden neuronalen Aktivierungen verknüpft werden.

Aktuelle Studien erkannten das Problem, aber veränderten stattdessen das experimentelle Setup, um die Bewegungsfreiheit und die eigenständige Exploration der Virtuellen Umgebung entweder durch eine schlauchartige Umgebungsstruktur, durch eine feste Aufgabenstellung oder durch vordefinierte Bewegungsrouten einzuschränken (Hassabis et al. 2009, Mellet et al. 2010). Mit der vorliegenden VR-Stimulusapplikation und der hier durchgeführten Studie wurden erstmals spezifische VR-Ereignisprotokolle und Kommunikationsfunktionen bereitgestellt, welche die statistische Auswertung von VR-fMRT-Experimenten nachhaltig unterstützen. Des Weiteren erlaubt die Einführung dieser Funktionen eine Kombination von komplexen VR-Stimuli mit Echtzeit-fMRT-Auswertetechniken, wie z.B. in Neurofeedback-Experimenten.

Die entwickelte VR-Stimulusapplikation ermöglicht die Aufzeichnung von VR-Ereignissen, Benutzerinteraktionen sowie Echtzeit-Auswerteeinformationen und gestattet dadurch die Online- sowie Offline-Datenanalyse mit VR-Ereignisprotokollen.

Während der Studie ermittelte die Echtzeit-Datenauswertung durch den *Turbo Brain Voyager* neuronale Aktivitäten in okzipitalen Bereichen, im Cuneus, im Precuneus, im Mittleren Temporalen Gyrus und in den parahippokampalen Bereichen des Gehirns. Diese Ergebnisse wurden durch eine erneute post-experimentelle Simulation der Studie mit dem TBV und der VR-Ereignisprotokolle bestätigt. Die Ergebnisse sowie die aufgezeichneten Echtzeit-Protokolle des TBV wurden im Folgenden zur Überprüfung der post-experimentellen Datenauswertung verwendet.

Die post-experimentelle Auswertung durch *BrainVoyager QX 2.2* bestätigte die Ergebnisse der Online-Auswertung und die Resultate vergleichbarer Studien (Aguirre 1998, Maguire 1998, Burgess et al. 2002, Andreano et al. 2009, Mellet et al. 2010). Die Ergebnisse zeigen neuronale Aktivierungen für topographische Navigation sowie Prozesse des räumlich-episodischen Gedächtnisses im Parahippokampalen Gyrus, im Posterioren Cingulus, im Medialen Parietalen Kortex (Precuneus) und in Okzipitotemporalen Strukturen.

Des Weiteren bestärken diese Ergebnisse die Resultate von Studien zum räumlich-episodischen Gedächtnis und zur Tiefenwahrnehmung, bei denen bereits gezielt VR-ähnliche Szenarios eingesetzt wurden (Aguirre et al. 1996, Maguire et al. 1997; 2006, Baecke et al. 2009).

Verschiedene Studien zum topographischen Gedächtnis, welche computergenerierte Umgebungen verwendeten, konnten in ihren Ergebnissen neuronale Aktivität im rechten Hippocampus nachweisen (Burgess et al. 2002, Hassabis et al. 2009, Marsh et al. 2010). Das Fehlen dieser Aktivierung in der hier durchgeführten Studie weist auf mögliche Aktivitätsunterschiede zwischen real empfundenen VR-Umgebungen und computersimulierten VR-Umgebungen hin. Diese wurden durch Maguire et al. (1998) erstmals theoretisiert und in weiteren Studien zur Beschreibung der neuronalen Aktivierungen verwendet (Aguirre 1998, Maguire et al. 2006, Wolbers et al. 2008, Adamovich et al. 2009, Marsh et al. 2010).

Konkrete Beweise für diese These wurden aufgrund fehlender Befragungen zum Rea-

litätsempfinden nicht gefunden. Im Rahmen der hier durchgeführten VR-fMRT-Studie zum topographischen Gedächtnis wurde post-experimentell erstmalig eine Probandenbefragung zum Realitätsempfinden und zu den immersiven Einflussfaktoren realisiert. Mit ihr sollten zunächst mögliche Immersionseffekte evaluiert und mögliche Aktivitätsunterschiede zwischen real empfundenen VR-Umgebungen und computersimulierten VR-Umgebungen charakterisiert werden.

Die Auswertung des post-experimentellen Probandenbefragung zeigte, dass alle Probanden bereits an fMRT-Experimenten teilgenommen sowie Erfahrungen mit diversen 3D-Computerspielen gesammelt hatten. Alle Probanden beschrieben eine höhere Aufmerksamkeit gegenüber herkömmlichen fMRT-Studien und beschrieben immersive Erfahrungen, die zu verschiedenen Zeitpunkten des Experiments auftraten.

Ein Vergleich der neuronalen Aktivierung zu den genannten immersiven Zeitpunkten lieferte weitere Anhaltspunkte, die auf eine mögliche Verbindung zwischen der fehlenden Aktivierung im rechten Hippocampus und der als real empfundenen VR-Umgebung hindeuten. Ein konkreter Beweis für diese These konnte aber aufgrund fehlender Vergleichsergebnisse noch nicht sicher dokumentiert werden. Hier müssen in zukünftigen Versuchen die expliziten Unterschiede zwischen immersiven und nicht-immersiven VR-Umgebungen stärker hervorgehoben sowie die resultierenden neuronalen Aktivierungen miteinander verglichen werden.

Ein mögliches Anwendungsszenario für eine solche Vergleichsstudie bieten sogenannte VR-Therapiestudien. Auch wenn die neuronalen Aktivierungen des sense-of-presence- oder Immersions-Effektes noch nicht näher untersucht wurden, so wird dieser Effekt bereits aktiv im Rahmen der Therapie genutzt (vgl. Abschnitt 3.5.3) (Rothbaum et al. 2000, Wiederhold et al. 2002, Andreano et al. 2009).

Ein Faktor, welcher von den Probanden positiv hervorgehoben wurde, war die hohe Bewegungsfreiheit innerhalb der Virtuellen Umgebung. Ausgehend vom Immersionsmodell nach Schubert (2003) ist die selbständige Exploration der Virtuellen Umgebung ein Hauptfaktor für die Involviertheit des Probanden (vgl. Abschnitt 3.5.1).

Des Weiteren berichteten mehrere Probanden von einem Aufmerksamkeitsverlust beim zweiten Experimentdurchlauf. Hier war oftmals die VR-Umgebung schon soweit erkundet, dass keine neuen salienten Reize mehr vom Probanden erfasst wurden (Mueller et al. 2012b). Anhand dessen lässt sich auch die Notwendigkeit zur Nutzung einer unbekannteren VR-Umgebung ableiten. Ähnlichen Schlussfolgerungen lieferten bisher nur die Studie von Mellet et al. (2010).

Mellet et al. (2010) nutzten die baulichen Strukturen eines bekannten, realen Gebäudes als Grundlage für die VR-Modellierung und verwendeten das Modell für eine Studie zur topographischen Navigation. Häufig durchgeführte VR-fMRT-Experimente mit der gleichen Umgebung oder die Nutzung einer bekannten Umgebung könnten demnach Auswirkungen auf die neuronalen Aktivierungswerte des Probanden haben und diese letztendlich negativ beeinflussen (Mueller et al. 2012b). Dies muss durch zukünftige Studien gesondert evaluiert werden.

Einen großen Vorteil dieser Anwendung stellt die neuartige integrierte VR-TCP-Kommunikationsfunktion dar, welche eine Vielzahl neuer Möglichkeiten eröffnet. Hier stehen vor allem die Echtzeit-Datenübertragung, die Anwendung auf Neurofeedback-Paradigmen oder der Informationsaustausch von Experimenteinstellungen zur effekti-

verer Gestaltung von neurowissenschaftlichen Studien im Vordergrund. Die Nutzung eines vergleichbaren Frameworks ist in keiner bisher veröffentlichten Arbeit dokumentiert und gestattete eine effektive und schnelle Durchführung der vorgestellten fMRT-VR-Studie.

Mit der Durchführung und der erfolgreichen Auswertung dieser VR-fMRT-Studie konnten neue Erkenntnisse gewonnen und bestehende wissenschaftliche Ergebnisse erfolgreich reproduziert werden. Die Ergebnisse erlauben die Identifikation verschiedener neuronaler Netzwerke, die in der Fachliteratur mit Funktionen zum räumlichen und semantischen Gedächtnis sowie mit topographischer Repräsentation in Verbindung gebracht werden (Schneider und Fink 2007).

Eine erweiterte Diskussion zur VR-Stimulusapplikation, welche bisher nicht behandelte Funktionalitäten einschließt, findet sich im nachfolgenden Kapitel.

Das zweite Hauptziel der Studie, die sichere Validierung unter kontrollierten Bedingungen eines bis dato nicht vorhandenen Echtzeit-Adaptionskonzeptes für dynamische VR-Stimuli in Neurofeedback-Experimenten sowie die Evaluierung von Neurofeedback-Einflussfaktoren auf die neu entwickelten, dynamischen VR-Präsentationsabläufe, wurde mit Hilfe der Probandendaten aus dem VR-fMRT-Experiment und der Simulationsfunktion des TBV ebenfalls erfolgreich umgesetzt (Mueller et al. 2012a).

Ein Vergleich dieses Anwendungsszenarios mit bereits veröffentlichten Studien ist nur bedingt möglich. Zum einen sind nur wenige Studien bekannt, bei denen VR-ähnliche Stimuli mit Echtzeit-Auswertetechniken kombiniert wurden. Zum anderen existiert nach dem aktuellen Kenntnisstand noch keine Neurofeedback-Studie, in der eine komplexe VR-Umgebung anhand der neuronalen Aktivität des Probanden in Echtzeit adaptiert wurde.

Ein Studie, die ausgehend von der Stimulusadaption ein ähnliches Konzept aufweist, wurde von deCharms et al. (2005) vorgestellt. Das Ziel dieser Neurofeedback-Studie war die schrittweise Reduktion von Schmerzempfindungen durch operante Konditionierung. In mehreren Echtzeit-fMRT-Experimenten bekamen Schmerzpatienten ihre eigenen neuronalen Aktivitäten aus dem rechten Anterioren Cingulum präsentiert und sollten durch Regulierungstechniken ihre neuronale Aktivität beeinflussen sowie ihr Schmerzempfinden schrittweise senken. Als visuelle Anzeige und Repräsentation der neuronalen Aktivität diente eine virtuelle Flamme, die sich abhängig von der Stärke der Aktivität entweder vergrößerte oder verkleinerte.

Auch wenn diese animierte Flamme keine komplexe virtuelle Umgebung darstellt, so ähnelt dieses stark vereinfachte Adaptionsprinzip in Grundzügen dem hier entwickelten Stimulusadaptionskonzept. deCharms et al. (2005) gehen leider nicht konkret auf die technischen Details, die verwendete Echtzeit-Analysesoftware und deren Auswertemethodik ein. Jedoch kann anhand der Beschreibungen zur Stimuluspräsentation davon ausgegangen werden, dass die Echtzeit-Datenauswertung und die Präsentation der animierten Flamme auf dem gleichen Computersystem liefen. Ein spezielles Kommunikationsprotokoll, welches Echtzeit-Datenauswerteeinformationen an die Stimuluspräsentation übermittelt und Auswerteergebnisse mit der animierten Flamme koordiniert, wurde nicht genutzt.

Hier unterscheidet sich dieser Ansatz vom hier entwickelten Konzept. Die Komplexität der hier verwendeten VR-Stimuli lässt aus Performanzgründen eine gemeinsame Nutzung von Rechenressourcen durch die VR-Stimulusapplikation und die Echtzeit-Datenauswertung auf einem Computersystem nicht zu. Des Weiteren finden in der Studie von deCharms et al. (2005) keine komplexen Benutzerinteraktionen mit der animierten Flamme statt. Die neuronale Aktivität aus einer definierten ROI wird durch die Echtzeit-Datenauswertung sofort analysiert und interpretiert. In diesem Zusammenhang wird lediglich die Nutzung eines Echtzeit-Datenauswertealgorithmus auf Basis einer optimierten Korrelationsanalyse erwähnt.

Nach der Signalinterpretation werden die gewonnenen Informationen sofort zur Stimulusadaption der Flamme weitergeleitet. Ausgehend vom beschriebenen Versuchsaufbau wird bei der Interpretation zwischen zwei Zuständen unterschieden. Bei einem Anstieg der neuronalen Aktivierung vom vorherigen zum aktuellen Messzeitpunkt wurde die virtuelle Flamme schrittweise vergrößert. Bei einem Signalabfall verkleinerte sich die Flamme auf eine vordefinierte Größe. Spezifische Benutzerinteraktionen des Probanden mit der animierten Flamme oder die Aufzeichnung von Ereignis- und Echtzeit-Auswerteprotokollen wurden nicht unterstützt.

Die hier implementierte VR-Stimulusapplikation verwendet eine komplexere Form der Stimuluspräsentation, welche vor allem durch dynamische Ereignisse und Benutzerinteraktionen in einer Virtuellen Umgebung geprägt ist. Des Weiteren ermöglichen die integrierten Ereignisprotokollen eine Protokollierung von Benutzerereignissen, Echtzeit-Auswerteeinformationen und Kommunikationsprozessen. Auf diese Weise kann eine konkrete Zuordnung von Benutzerinteraktionen oder VR-Ereignissen mit aufgezeichneten neuronalen Aktivierungen des Probanden durchgeführt werden.

Eine weitere Echtzeit-fMRT-Neurofeedback-Studie, welche Ähnlichkeiten zu dem hier entwickelten Adaptionalgorithmus aufweist, wurde von Sorger et al. (2012) vorgestellt. Das Hauptthema der Studie ist die Bereitstellung einer fMRT-basierten Buchstabieranwendung, die als Kommunikationsmittel für ALS- bzw. Locked-In-Syndrom-Patienten genutzt wird.

Dazu mussten zunächst die 26 Buchstaben des Alphabets und das Leerzeichen über 27 „dynamische Aktivierungsmuster“ kodiert werden. Zur Kodierung der 27 Zustände wurden $3 \times 3 \times 3$ verschiedene Informationseinheiten definiert, welche sich anhand der Kombination der folgenden Faktoren unterscheiden lassen. Der erste Faktor in diesem Kodierungsprozess ist die Versuchsaufgabe, welche entweder mentales Rechnen, mentale Motorbewegungen oder inneres Sprechen repräsentiert. Der zweite Faktor ist die zeitliche Verzögerung des Aufgabenstarts, welche entweder mit 0s, 10s oder 20s angegeben wird. Der dritte Faktor ist die Aufgabendauer, die sich entweder durch 10s, 20s oder 30s definiert.

Ein Patient der beispielsweise den Buchstaben „E“ schreiben möchte, müsste zunächst nach dem Start der Aktivbedingung 10s warten, anschließend eine mentale Motorbewegung durchführen und schließlich diese mentale Bewegung für 30s aufrechterhalten. Ein solches komplexes und zeitaufwendiges Kodierungsprinzip kann derzeit aufgrund der dynamischen und schnellen VR-Stimuluspräsentation mit ca. 30 fps (frames per second) nicht auf den Adaptionalgorithmus übertragen werden. Hier würden die um mehrere Sekunden verzögerte und die über mehrere Sekunden andauernde Ausführung

eines mentalen Prozesses die flüssige und authentische Präsentation von VR-Stimuli nachhaltig beeinflussen. In der Studie von Sorger et al. (2012) und in dem hier durchgeführten Anwendungsszenario findet eine gleichzeitige Verarbeitung von Signalverläufe aus multiplen Hirnregionen statt. Der große Unterschied beider Konzepte liegt in der unterschiedlichen Interpretation des Neurofeedback-Signals.

Der implementierte Adaptionalgorithmus für dynamische VR-Stimuli ist auf eine schnelle und realistische Anpassung von VR-Szenarien ausgerichtet und benötigt dahingehend einen schnelleren, weniger zeitaufwendigen Signalinterpretationsprozess. Die Signalinterpretation nach Sorger et al. (2012) muss in zeitaufwendigen Prozessen das Neurofeedback-Signal analysieren, die gefundenen Merkmale klassifizieren und dahingehend die statische Anzeige des jeweiligen Buchstabens durchführen.

Nach dem aktuellen Kenntnisstand wurden in sämtlichen Neurofeedback-Studien ausschließlich Anpassungen an 2D-Modellen oder einfachen Animationen vorgenommen (deCharms 2008, Coben und Evans 2010). Aufgrund dessen wurde während des Versuchs besonders auf unbekannte Störfaktoren bei der VR-Stimuluspräsentation geachtet. Im Mittelpunkt standen hier vor allem die zeitkritische Verarbeitung der Auswertinformationen zur Echtzeit-VR-Adaption und ein reale und kontinuierliche Anpassung des virtuellen Szenarios über einen angemessenen Zeitraum innerhalb der Explorationsphase. Spezifischen Störfaktoren, die auf eine Untauglichkeit des Adaptionalgorithmus in Neurofeedback-Experimenten hinweisen, wurden nicht gefunden. Allerdings konnten bisher aufgrund der Experimentsimulation durch den TBV nur die technischen Abläufe überprüft werden. Für zukünftige Neurofeedback-Experimente müssen vor allem dynamische Interaktionsprozesse mit Probanden und die Prozessabläufe der Selbstregulierung neuronaler Aktivitäten durch operante Konditionierung evaluiert werden.

Die implementierten Protokollfunktionen des VR-Adaptionalgorithmus zeichneten während der Kalibrierungs- und Hauptmessung automatisch alle empfangenen Auswertinformationen und sämtliche berechneten Werte des Algorithmus auf. (Sorger et al. 2012). Die Auswertung dieser Protokollierungen zeigte eine sehr häufige Unterschreitung des Entscheidungskriteriums in Stimulusblock 11 bei 9 von 12 Probanden (Abb. 8.6 (B)). Durch den individuellen Vergleich der Probandendaten wurden hier verschiedene Aufmerksamkeitsreduktionen sichtbar, welche auf eine zu lange Gesamtdauer und eine zu kurze Dauer der Explorationsphase hindeuten.

Eine ähnliche Protokollfunktion zur Analyse von Neurofeedback-Probandendaten wurde von Rief und Birbaumer (2006) für die EEG-Neurofeedback-Signalanalyse und zur Optimierung der operanten Konditionierung vorgestellt. Das hier implementierte Protokollierungskonzept baut im Unterschied dazu erstmalig auf den spezifischen Besonderheiten der Echtzeit-Datenauswertung von fMRT-Bilddaten auf und verwendet zudem die berechneten Werte des VR-Adaptionalgorithmus.

Nach der aktuellen Kenntnislage wurde das vorliegende Gesamtkonzept zur dynamischen Echtzeit-VR-Adaption erstmalig realisiert. Dieses Konzept stellt die Grundlage dar, um beispielsweise mittels virtuellen Charakteren neue Methoden zur Erforschung sozialer Interaktionen zu erschließen (Bailenson et al. 2008).

Auf Grundlage dieses Algorithmus konnte ein konkretes Anwendungsprinzip, die Adaptierung von Wetterverhältnissen innerhalb einer virtuellen Umgebung, erfolgreich

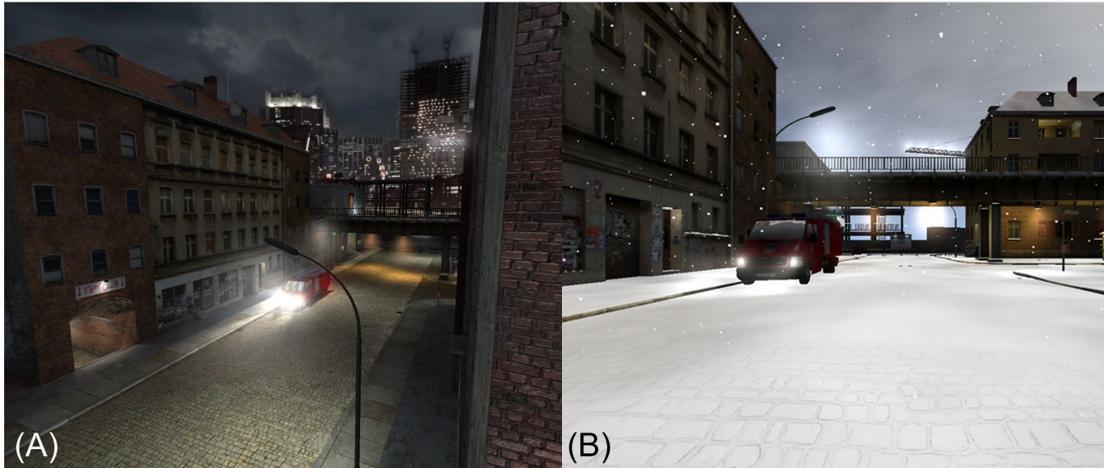


Abbildung 8.7: Das konkrete Anwendungsbeispiel des VR-Adaptionsframeworks. Der Algorithmus analysiert zunächst den Signalverlauf aus einer spezifischen Hirnregion und passt ein dynamisches VR-Szenario (A) durch die Einbettung von Schneepartikel-Effekten in Echtzeit an (B).

umgesetzt werden (Abb 8.7). In diesem Zusammenhang werden zunächst für 7s der Aktivphase Schneefalleffekte eingefügt. Nach den 7s stoppt der Schneefall und es setzt für die restlichen 7s der Aktivphase ein Tauteffekt ein, welcher den gefallenen Schnee kontinuierlich entfernt.

Zukünftige Verbesserungen des Adaptionsalgorithmus sollen vor allem die Effektivität und die Stabilität der statistischen Analyse steigern. Eine Möglichkeit wäre die Verwendung der β -Werte des rekursiven ALM zur VR-Stimulusadaption. Hier müssen für die Zukunft zunächst die spezifische Änderungen am VR-Adaptionsalgorithmus vorgenommen und die neuen Prozessabläufe hinsichtlich der zeitkritischen Abarbeitung überprüft werden.

Kapitel 9

Zusammenfassung und Ausblick

In diesem Kapitel sollen anhand der Zielstellung aus Kapitel 1.4 die Ergebnisse der Arbeit zusammengefasst und ein Ausblick auf zukünftige Erweiterungen eröffnet werden. Ausgehend von den Performanztests sowie den beiden durchgeführten Experimenten werden hier die implementierten Funktionen erörtert und mögliche Verbesserungsvorschläge in Aussicht gestellt.

9.1 Zusammenfassung und Diskussion

Mit dieser Arbeit konnte das Hauptziel dieser Arbeit, die Einbindung einer neuartigen VR-Stimulusapplikation in Echtzeit-fMRT-Prozessabläufe zur Beantwortung von konkreten neurowissenschaftlichen Fragestellungen erfolgreich umgesetzt werden. Zur Umsetzung dieses Ziels mussten zunächst die komplexen Funktionalitäten der Experimentsteuerung sowie die Prozesse zur Interaktion mit hochrealistischen, dreidimensionalen Umgebungen entworfen und implementiert werden.

Die Applikation wurde mit der Programmiersprache C++ realisiert und erlaubt das schnelle Einladen von vordefinierten 3D-Umgebungen anhand weniger Mausklicks. Des Weiteren wurden für die Anwendung Datenhaltungskonzepte sowie Steuerungsprozesse entwickelt, welche die zuverlässige VR-Stimuluspräsentation in neurokognitiven Studien ermöglichen. Im Zentrum stehen hier die komplexen Berechnungen des Experimentablaufs, die semantische Prüfung aller Experimentparameter sowie die zeitkritische Steuerung sämtlicher Stimulusprozesse.

Die Präsentation von authentischen VR-Stimuli spielt für ein immersives Probandenverhalten eine große Rolle. Daher wurde für die neurokognitive Stimuluspräsentation von detailreichen und hochrealistischen 3D-Szenarien die *Vision Engine*, der Firma *Havok Inc.*, erstmalig in ein erstelltes C++-Framework integriert.

Die Engine gewährleistet im Rahmen einer immersiven Stimuluspräsentation eine flexible Gestaltung von Paradigmen und garantiert damit die gezielte Ausrichtung der VR-Stimuli an das jeweilige Forschungsziel. Zusätzlich dazu wurde für den wissenschaftlichen Gebrauch der Applikation eine kostenlose Lizenz des Frameworks von der Firma *Havok* zur Verfügung gestellt. Im Rahmen dieser Lizenzvereinbarung können externe Forschergruppen auf die Stimulusapplikation zugreifen und diese in nicht-

kommerziellen, neurokognitiven Experimenten nutzen.

Ein kleiner Nachteil für die Stimuluspräsentation ist die Datenformat-Restriktion des *Vision Engine*-Frameworks. Hier erlaubt das Framework nur die Initialisierung des *Havok*-eigenen Datenformats *VSCENE*.

Weiterhin wurden zur Realisierung einer anwenderorientierten und strukturierten Benutzerschnittstelle die verschiedenen Aspekte der VR-Experimentmodellierung analysiert und schließlich im Rahmen der Oberflächenmodellierung zusammengeführt. Als Resultat entstand eine zentrale, auf einem Registerkarten-Konzept aufbauende, Oberflächenstruktur, welche sich an bereits bekannten Datenauswerteapplikationen orientiert und die Experimentmodellierung von Block- und ereigniskorrelierten Experimentstrukturen unterstützt.

Im Zentrum stand hier der Entwurf von benutzerakzeptierten Oberflächenstrukturen, die sich an Konzepten weitverbreiteter Stimulus- und Datenauswerteanwendungen orientieren und Anwendern aus dem medizinisch-technischen sowie neurowissenschaftlichen Forschungsbereich zu einem schnellen Einstieg verhelfen. Hier wurden im Rahmen der Problemanalyse verschiedene Stimulusanwendungen und Datenauswerteapplikationen aus dem neurowissenschaftlichen Kontext evaluiert sowie deren konzeptuelle Besonderheiten zusammengetragen. Ausgehend davon wurden eigene Konzepte zur Stimuluspräsentation erarbeitet und an die Besonderheiten einer VR-Stimuluspräsentation angepasst.

Im Entwurfsprozess der Anwendung wurde eine multimodale Stimulusapplikation modelliert, welche vor allem ein hohes Maß an Erweiterbarkeit zulässt und Schnittstellen für zukünftige Modalitätserweiterungen bereitstellt. Die Akzeptanz der entwickelten Oberflächenstrukturen wurde anschließend erfolgreich in einem Benutzerakzeptanztest mit neun Anwendern getestet. Die mit dieser Anwendung eingeführten Funktionalitäten vereinfachen die neurowissenschaftliche Verwendung dieser Modalität und stellen sie einer größeren, wissenschaftlichen Gemeinschaft zur Verfügung.

Zur unkomplizierten Modellierung von VR-Experimenten wurden zudem drei unterschiedlich komplexe 3D-Szenarien bereitgestellt und innerhalb von mehreren Perforanztests hinsichtlich ihrer Einbindung in VR-fMRT-Experimente evaluiert. Diese Tests ergaben keinerlei Auffälligkeiten bei der Nutzung der drei Szenen und gewährleisten damit die sichere Modellierung von VR-fMRT-Experimenten.

Die erfolgreiche Umsetzung der Experimente umfasste die Bearbeitung verschiedener Nebenziele. Hierfür wurden in verschiedenen Arbeitsschritten folgende Prozesse erfolgreich entworfen und realisiert:

- Basierend auf dem Softwareentwicklungsprozess durch das *V-Modell* wurde zur strukturierten Implementierung von Kernprozessen ein übergeordnetes Architekturmodell integriert. Das entworfene *erweiterte Drei-Schichten-Architekturmodell* gestattet hier eine strukturierte Aufteilung der Kernaspekte in Benutzeroberfläche, Applikationslogik und Datenzugriffsschicht. Die skalierbare Struktur dieser Schichten ermöglichte die erfolgreiche Evaluierung von spezifischen VR-fMRT-Stimuluspräsentationsfunktionen und gestattet zudem eine einfache

Erweiterung mit EEG-spezifischen VR-Präsentationsroutinen. Zusätzlich dazu erlaubt die integrierte Plug-in-Funktion die schnelle Erweiterung von Stimulus-funktionalitäten durch benutzerdefinierte Funktionen, welche mittels einer DLL-Bibliothek bereitgestellt werden.

- Aufgrund der neuartigen Paradigmenkonstellation mit dynamisch-interaktiven VR-Szenarien wurden verschiedene, experimentelle Versuchsabläufe analysiert und mit der Präsentation von VR-Stimuli in Relation gesetzt. Als Resultat wurde ein Oberflächenkonzept entwickelt und realisiert, welches Versuchsbedingungen automatisch registriert und diese jeweils mit statischen Bildern oder dynamisch-interaktiven VR-Stimuli verknüpft. Die syntaktische Fehlerprüfung innerhalb der Benutzeroberfläche signalisiert zudem mögliche Eingabefehler des Anwenders und verringert die Gefahr von experimentellen Inkonsistenzen, welche durch die semantische Fehlerprüfung innerhalb der Applikationslogik aufgedeckt werden.
- Im Zuge dieses Projektes wurde außerdem ein neuartiges Echtzeit-Adaptionskonzept zur Anpassung von dynamisch-interaktiven VR-Stimuli in Neurofeedback-Experimenten entworfen und realisiert. Das Konzept baut auf einem zweistufigen Anpassungsalgorithmus sowie auf dem entwickelten TCP-Kommunikationsframework auf und ermöglicht die Übertragung von Echtzeit-Datenauswerteparametern von der Datenauswerteapplikation *Turbo Brain Voyager* zur Stimulusapplikation. Dort werden diese Signalinformationen analysiert und interpretiert sowie zur Anpassung eines laufenden VR-Szenarios verwendet. Mit der konzeptuellen Planung, der Implementierung aller mathematischen Prozesse sowie der experimentellen Validierung dieses VR-Adaptionsframeworks wurde erstmalig eine Neurofeedback-Strategie zur Anpassung von dynamischen virtuellen Umgebungen in fMRT-Studien vorgestellt. Dieses Konzept erlaubt außerdem die Generierung von probandenspezifischen Konditionierungsprofilen im Neurofeedback-Training und unterstützt damit die Optimierung des Konditionierungsprozesses. Das Framework wurde unter sicheren Experimentbedingungen und ohne direkten Probandeneinfluss evaluiert (vgl. Abschnitt 8.4).
- Die Validierung und Evaluierung der entwickelnden VR-Stimulusapplikation unter Experimentbedingungen mit Hilfe eines neurowissenschaftlichen Pilotexperiments wurde ebenfalls erfolgreich durchgeführt. Vor der Durchführung des Experiments mit 12 Probanden wurden zunächst die Grundlagen zur räumlichen Orientierung sowie zum episodischen und topographischen Gedächtnis recherchiert und analysiert. Die Struktur und der experimentelle Ablauf des Experiments wurden dabei komplett innerhalb der VR-Stimulusapplikation modelliert und prä-experimentell getestet. Die Durchführung dieses Experiments verlangte außerdem die vorherige, erfolgreiche Durchführung von Performanz- und Anwendertests. Die Ergebnisse der Performanztests sowie die Resultate der post-experimentellen Auswertung werden im Folgenden näher vorgestellt.

Zur Evaluierung der VR-Stimulusapplikation wurden mehrere Performanz- und Nutzertests an dem zur Verfügung stehenden MRT-Akquirierungssystem durchgeführt und

ausgewertet. Hierzu wurden Funktionstests mit einem Phantom sowohl an einem Siemens 7T „Avanto-“ als auch an einem Siemens 3T „Trio“ System erhoben. Nach dieser erfolgreichen Evaluierung erfolgte die Durchführung des VR-fMRT-Experiments mit 12 Probanden am Siemens 3T „Trio“ MRT-System.

Alle fMRT-Bilddaten wurden unmittelbar nach der Akquirierung durch den Magnetresonanztomographen mit der Echtzeit-Datenauswertesoftware *Turbo BrainVoyager* durch das rekursive ALM erfolgreich analysiert und direkt präsentiert. Des Weiteren wurden pro Volumendatensatz die gemittelten Grauwerte aus drei vordefinierten Hirnregionen über die TCP-Kommunikationsfunktion in Echtzeit übertragen. Die so gewonnenen Echtzeit-Datenauswerteeinformationen wurden zur Evaluierung des Echtzeit-Adaptionskonzeptes in einem simulierten Neurofeedback-Experiment ohne direkten Probandeneinfluss verwendet.

Zusätzlich dazu wurden sämtliche Bilddaten aller Probanden post-experimentell mit der Datenauswertesoftware *BrainVoyager QX*, zunächst einzeln und anschließend durch eine Gruppenanalyse ausgewertet.

Die post-experimentelle Auswertung der fMRT-Bilddaten bestätigte die Resultate vergleichbarer Studien und zeigte neuronale Aktivität in Hirnarealen, die mit Prozessen des räumlichen Gedächtnisses, des episodischen Gedächtnisses und der topografischen Navigation in Verbindung gebracht werden (Maguire et al. 1997, Aguirre 1998, Burgess et al. 2002, Mellet et al. 2010). Hier wurden neuronale Aktivierungen im Parahippokampalen Gyrus, im Posterioren Cingulus, im Medialen Parietalen Kortex (Precuneus) und in Okzipito-temporalen Strukturen gefunden (Maguire 1998, Andreano et al. 2009).

Ein nach der Durchführung der MRT-Messungen vorgelegter Probandenfragebogen ermittelte die Motivation und die Aufmerksamkeit der Probanden sowie ihre jeweilige, persönliche Erfahrung mit neurokognitiven fMRT-Experimenten. Die Auswertung dieses post-experimentellen Probandenfragebogens zeigte, dass alle Probanden bereits an fMRT-Experimenten teilgenommen sowie Erfahrungen mit diversen 3D-Computerspielen gesammelt hatten. Alle Probanden beschrieben eine höhere Aufmerksamkeit gegenüber herkömmlichen fMRT-Studien und hoben besonders die hohe Bewegungsfreiheit positiv hervor. Dies ist ein Indiz dafür, dass mit VR-Stimuli eine effektivere Gestaltung von Experimenten möglich ist (Mueller et al. 2012b). Für detailliertere Informationen zu den Ergebnissen des VR-fMRT-Experiment sei hier auf Kapitel 8.4 verwiesen.

Des Weiteren konnte die Validierung unter kontrollierten Bedingungen eines bis dato nicht vorhandenen Echtzeit-Adaptionskonzeptes für dynamische VR-Stimuli in Neurofeedback-Experimenten erfolgreich durchgeführt werden (Abschnitt 8.4) (Mueller et al. 2012a). Die Evaluierung von Neurofeedback-Einflussfaktoren auf die neu entwickelten, dynamischen VR-Präsentationsabläufe zeigte zudem keine gravierenden Auswirkungen auf den Experimentablauf oder die Stimuluspräsentation. Auf Grundlage des entwickelten TCP-Kommunikationskonzeptes konnte so ein Echtzeit-Adaptionsalgorithmus zur automatischen Anpassung dynamisch-interaktiver VR-Stimuli geschaffen werden. Das zweistufige Algorithmuskonzept analysiert im ersten Schritt sämtliche empfangenen Datenauswerteparameter und verwendet diese zur Berechnung von probandenspezifischen Referenzwerten.

Im zweiten Schritt verarbeitet der Algorithmus, parallel zur laufenden Messung, die Werte der Echtzeit-Datenauswertung und vergleicht Signaländerungen zwischen vordefinierten Stimulusphasen auf Basis eines Entscheidungskriteriums. Das Entscheidungskriterium stellt hier die erste Implementierung eines für die Echtzeit-Vergleichsanalyse notwendigen Abarbeitungsprozesses dar. Nach dem aktuellen Kenntnisstand ist das vorliegende Konzept zur dynamischen Echtzeit-VR-Adaption erstmalig realisiert worden und bietet darüber hinaus die Möglichkeit, individuelle Neurofeedback-Konditionierungsprofile in die Signalanalysefunktionen zu integrieren. Für detailliertere Informationen zu den Ergebnissen des VR-Adaptionskonzeptes sei auf Kapitel 8.4 verwiesen.

Anhand der gewonnenen Ergebnisse konnte die Funktionalität der Anwendung sichergestellt werden. Die Anwendung erlaubt durch ihren Aufbau und ihre Funktionsweise eine zuverlässige Reproduzierbarkeit von wissenschaftlichen Ergebnissen aus VR-Studien und eine schnelle Erstellung von eigenen VR-fMRT- und Echtzeit-VR-fMRT-Paradigmen.

Ein notwendiger Bestandteil dieses Konzepts ist die schnelle Akquirierung von fMRT-Bilddaten zur Echtzeit-Datenauswertung. Die Grundlage des hier vorgestellten Konzepts ist die Aufzeichnung von fMRT-Bilddaten über eine modifizierte EPI-BOLD-Sequenz.

Für die zukünftige Anwendung der VR-Stimulusapplikation in Echtzeit-fMRT-Studien muss zunächst die Verwendung von noch schnelleren und verbesserten MR-Messsequenzen evaluiert werden.

Eine mögliche Alternative stellt das *Echo-Volumar-Imaging* (EVI) dar. Aufgrund der fortwährenden Entwicklung im Bereich der parallelen Bildgebung (parallel imaging) und der Bereitstellung immer besserer Gradientenspulen ermöglicht EVI die Reduzierung von Einfaltungsartefakten (aliasing), die Abschätzung von regionalen Unterschieden in der hämodynamischen Antwortfunktion und die Reduzierung von Intra-Scan Bewegungsartefakten durch eine hohe zeitliche Auflösung (Rabrait et al. 2008). Eine potentielle Akquirierungstechnik wurde durch Posse et al. (2012) vorgestellt und könnte zukünftig die Effizienz von fMRT-Studien mit dem TBV und der VR-Stimulusadaption verbessern. Hier muss eine detaillierte Evaluierung die notwendigen Änderungen am Echtzeit-Datenauswertesystem und mögliche Einschränkungen bei der dynamischen VR-Stimuluspräsentation ermitteln.

Neben der bereits entwickelten VR-Stimulusapplikation existieren bereits andere Applikationen, die eine VR-Stimuluspräsentation unterstützen. Jedoch bieten diese Applikationen nur einen Bruchteil der Funktionen, die mit dieser VR-Stimulusanwendung bereitgestellt wurden.

Viele neurowissenschaftliche Studien verwendeten bereits virtuelle Umgebungen zur Untersuchung ihrer Forschungsthemen (Aguirre et al. 1996, Maguire 1998, Hassabis et al. 2009, Marsh et al. 2010, Mellet et al. 2010, Mynatt et al. 2010, Doeller et al. 2010). Allerdings zeigte sich bei näherer Betrachtung, dass diese Studien meist unveränderbare Umgebungen aus kommerziellen Computerspielen verwendeten. Detaillierte Informationen zum technischen Setup sowie die speziellen Einschränkungen der

Umgebungen wurden nur teilweise angegeben und lieferte dadurch ein verzerrtes Bild von den technischen Anforderungen diverser VR-Stimuli.

Mit der realisierten VR-Stimulanwendung wurde ein Konzept etabliert, welches durch seine übersichtliche Struktur und die benutzerdefinierten Erweiterungsmöglichkeiten mehr Transparenz beim Einsatz neurowissenschaftlicher Stimuli verspricht. Zudem steht diese Anwendung, durch ihre für den wissenschaftlichen Gebrauch kostenlose Lizenz des *Vision Engine Frameworks*, für eine kostengünstige Alternative zum Einsatz diverser, kommerzieller Stimuluspräsentationen.

Eine kommerzielle VR-Stimulanwendung für neurologisch-verhaltenswissenschaftliche Forschungen ist *VR World 2* (Psychology Software Tools 2011). Die Modellierung von virtuellen Umgebungen basiert hier auf einem Drag & Drop Konzept und verspricht eine schnelle und einfache Erstellung von virtuellen Umgebungen. Dies stellt eine Vereinfachung des Modellierungsprozesses dar und ist ein Vorteil gegenüber der geplanten Modellierung von VR-Szenen mit *Autodesk 3ds Max*. Leider ist *VR World 2* nicht länger als eigenständige Software verfügbar. Virtuelle Umgebungen werden nur anhand konkreter Aufträge erstellt und können anschließend durch den Nutzer nicht weiter angepasst werden. Die Einbindung von Neurofeedback-Paradigmen ist ebenfalls nicht näher beschrieben. Es werden anpassbare Kommunikationsschnittstellen genannt, allerdings sind konkrete Informationen bzw. Spezifikationen nicht verfügbar. Hier sorgt mangelnde Transparenz für viele offene Fragen.

Dem gegenüber steht die freie neurowissenschaftliche VR-Stimulanwendung *ReactorMan* (RWTH 2011a, Beck et al. 2007). Sie verwendet für die Präsentation von VR-Stimuli das *ViSTA*-Framework, welches die Integration verschiedener VR-Technologien erlaubt.

Des Weiteren unterstützt *ViSTA* mehrere Geometrie-Datenformate und ist nicht auf ein einziges Datenformat eingeschränkt (RWTH 2011b). Diese Besonderheit unterscheidet *ReactorMan* von dem hier implementierten Konzept.

Laut der Softwarebeschreibung unterstützt *ReactorMan* weder die Kommunikation mit Echtzeit-Datenauswerteanwendungen noch die Durchführung von Neurofeedback-Studien.

Ein Nachteil den *ReactorMan* mit der Stimuluspräsentationssoftware *Presentation* verbindet, ist die rudimentäre Benutzeroberfläche (Neurobehavioral Systems 2011). Paradigmen und Stimuli werden anhand einer Skriptsprache manuell erstellt und in einer Textdatei abgespeichert. Anschließend werden die Textdateien in der Benutzeroberfläche eingeladen und zur Durchführung des Experiments verwendet. Eine Einarbeitung in die Semantik und Syntax dieser Skriptsprache ist absolut notwendig. Hier besitzt die implementierte VR-Stimulanwendung mit ihrem an bekannten Datenauswerteanwendungen angelehnten Oberflächendesign einen großen Vorteil.

Anwender, welche bereits neurowissenschaftliche Auswertungen durchgeführt haben, erkennen die Anordnung sowie die Funktionalitäten schnell wieder und können ihre vorhandenen Erfahrungen zur schnellen Erstellung von VR-Stimuli verwenden. Mehrfache Einarbeitungszeiten entfallen und sorgen für zeitlich effiziente Studienverläufe. Eine 3D-Präsentationsumgebung, welche explizit zur einfachen Modellierung und Anzeige von 3D-Geometrien konstruiert wurde, ist *Virtools* (3DVIA Corp. 2011).

Virtools liefert eine komplette Entwicklungs- und Präsentationsumgebung und kam be-

reits in diversen VR-fMRT-Studien zum Einsatz (Adamovich et al. 2009, Mellet et al. 2010). Da die Software nicht auf die Präsentation neurowissenschaftlicher Reize ausgelegt ist, fehlen im Konzept der Anwendung bereits die grundlegenden Funktionen zur Synchronisation, zur Ereignisprotokollierung und zur Experimentsteuerung. Virtuelle Umgebungen lassen sich mit der Anwendung zwar relativ schnell modellieren, sind aber zur Erforschung komplexerer neurowissenschaftlicher Fragestellungen ungeeignet. Die Durchführung von Neurofeedback-Studien wird von *Virtools* ebenfalls nicht unterstützt.

Alle genannten Stimulusanwendungen besitzen eine große Bandbreite der verschiedensten Funktionalitäten. Jedoch existiert keine Anwendung, die sowohl hochrealistische Umgebungen, Neurofeedback-Paradigmen, benutzerdefinierte Erweiterungsmöglichkeiten, eine einfache Einbindung virtueller Umgebungen und eine benutzerfreundliche Erstellung neurowissenschaftlicher Paradigmen ermöglicht. Des Weiteren weisen alle eingebundenen Grafik-Engines einige Defizite in der optischen Qualität auf. Aufgrund dessen wirken die virtuellen Umgebungen sehr künstlich und Probanden haben Probleme ein Immersionsgefühl aufzubauen (Hoffman et al. 2003, Mynatt et al. 2010). Zudem bestätigen einige Studien die Notwendigkeit des Immersionsgefühls, um während der Präsentation von VR-Paradigmen durchgängig eine hohe Motivation und Aufmerksamkeit beim Probanden zu erreichen (Slater et al. 1994, Schubert 2003, Sanchez-Vives und Slater 2005). Abhängig von diesen Anforderungen erfüllt das erarbeitete Konzept sowie die implementierte Stimulusanwendung alle Voraussetzungen, um immersive VR-Paradigmen professionell zu modellieren und zu präsentieren.

9.2 Ausblick

Mit dem vorgestellten Konzept wurde ein möglicher Ansatz zur Realisierung einer neuen, multimodalen VR-Stimulusanwendung gelegt, welcher weiterhin auf neue Erweiterungen angewiesen ist. Als zentrale Visualisierungskomponente wurde das *Vision Engine Framework* direkt in die Anwendung eingebettet. Durch seine Komplexität besitzt diese Grafik-Engine verschiedenste Visualisierungsmöglichkeiten, die für zukünftige Projekte interessante, neue Fragestellungen aufwerfen. Als Beispiel sei die neue Netzwerk-Infrastruktur des *Vision Engine Framework 8.1* genannt. Diese Funktionalität gestattet zusammen mit einer Netzwerk-Engine die Realisierung von sogenannten *Massively Multiplayer Online Games* (MMOG) und könnte im neurowissenschaftlichen Bereich zur Vernetzung mehrerer Modalitäten, wie z.B. EEG und MRT oder geografisch entfernter MRT-Systeme, verwendet werden.

Aufgrund der genannten Nachteile dieser Engine bleibt jedoch die Frage weiterhin offen, ob speziell die *Vision Grafik-Engine* die praktikabelste Lösung darstellt. Andere Frameworks, wie z.B. *ViSTA*, erlauben das Einladen von verschiedenen Geometrie-Dateien und können zudem auf Linux-Plattformen direkt eingesetzt werden. Zusätzlich dazu ist der Markt bei Grafik-Engines durch ständige Neu- und Weiterentwicklungen geprägt. Hier muss nach mehrjähriger Verwendung der *Vision Engine* eine mögliche Neuorientierung auf andere Engine-Frameworks in Betracht gezogen und neu bewertet werden.

EEG-VR-Stimuluspräsentation

Die konzeptuelle Entwicklung dieser VR-Stimulusanwendung war stets auf eine multimodale Stimuluspräsentation ausgelegt. Insbesondere die Einbindung der Elektroenzephalographie stellt aufgrund ihrer echtzeitnahen Aktivierungsverarbeitung und der hohen Bedeutung im Bereich der Neurofeedback-Studien eine lohnenswerte Ergänzung zum bisherigen Echtzeit-fMRT-Konzept dar. Für eine zukünftige Einbindung dieser Modalität wurden die Grundlagen bereits in der Entwurfsphase gelegt. So können die vorhandenen Klassen zur Experimentsteuerung, zur Verwendung der *Vision Engine* und zur TCP-Kommunikation auch für die Präsentation innerhalb einer EEG-Infrastruktur verwendet werden. Um die Durchführung von EEG-Experimenten vollends zu unterstützen, müssen jedoch weitere Oberflächenkomponenten zur Eingabe der EEG-Parameter sowie zusätzliche Algorithmen zur Verarbeitung EEG-relevanter Parameter implementiert werden. Des Weiteren muss für diese zusätzlichen Informationen das interne Datenhaltungs- und Datentransportkonzept erweitert werden.

3D-Szenenmodellierung durch Drag & Drop Konzept

Die Szenenmodellierung hat sich im Laufe dieser Arbeit als ein überaus komplexer und langwieriger Prozess dargestellt. Zusätzlich dazu verlangen die Erstellung und die Anordnung von dreidimensionalen Geometrie-Objekten einige Erfahrung im Bereich des Designs sowie der Verwendung von Designsoftware, wie z.B. *Autodesk 3ds Max*. Gerade für die Erstellung und Verwendung von virtuellen Umgebungen im Bereich der Neurowissenschaften kann dies mit sehr viel Aufwand verbunden sein. Oft fehlen die Zeit und die personellen Ressourcen, um eigene Szenen zu modellieren. Eine Möglichkeit ist der Kauf von Szenen direkt von der Computerspiel-Industrie oder von privaten Entwicklern. Eine weitere Lösung ist die Vereinfachung der hier angewendeten 3D-Szenenmodellierung. Hier ist eine zusätzlichen Anwendung angedacht, die eine schnelle Erstellung von 3D-Geometrien ermöglicht. Ähnlich des Drag & Drop-Prinzips von *VR Worlds 2* soll eine Anwendung in das vorliegende Konzept integriert werden, welche bereits vordefinierte Objekte bereitstellt und diese über ein einfaches Platzierungssystem in einer benutzerdefinierten Umgebung verteilen kann. Eine mögliche Integrierung in das vorliegende Konzept der Stimuluspräsentation ist zwar denkbar, aber aufgrund der unterschiedlichen Aspekte nicht empfehlenswert. Ein Kompromiss wäre die Einbettung des Applikationsaufrufs in einen sogenannten Intro-Dialog, bei dem der Anwender zu Beginn zwischen Funktionen der 3D-Szenenpräsentation (VR-Stimuluspräsentation) und der 3D-Szenenmodellierung entscheiden kann.

VR-Ereignisprotokollierung über Videoaufzeichnungen

Für die spätere statistische Auswertung von VR-fMRT-Experimenten stellt die Verknüpfung der Ereignisprotokollierung mit der visuellen Präsentation eine notwendige Funktionalität dar. Die Protokollierung von Bewegungs- und Interaktionsereignissen über Einträge in Textdateien galt in diesem Kontext als Minimalziel. Wirkliche Nachteile dieser Methodik stellten sich zwar nie ein, jedoch zeigte sich bei der Auswertung der VR-fMRT-Experimente die mangelnde Benutzerfreundlichkeit dieses Verfahrens.

Ereignisse die im Textlog anhand eines Zeitstempels vermerkt waren, konnten nur sehr umständlich mit den Auswertungen in der *BrainVoyager*-Anwendung verglichen werden. Um diesen Nachteil zu beheben, wurde eine zusätzliche Ereignisprotokollierung über Videoaufzeichnungen angedacht, allerdings aufgrund der festgelegten Dauer des Projektes, nicht rechtzeitig integriert. Das Framework *Qt* bietet mit dem *QtMultimediaKit* bereits eine mögliche Schnittstelle zur Implementierung dieser Funktionalität. Diese Schnittstelle gestattet sowohl das Aufnehmen als auch das Abspielen von Video-Dateien.

Erweiterung der echtzeitnahen Paradigmenadaption

Das Hauptziel bei der Realisierung eines Konzepts zur echtzeitnahen Paradigmenadaption war die Modellierung und die Implementierung der technischen Prozessabläufe. Echtzeit-fMRT-Auswertinformationen sollten via TCP direkt an die VR-Stimuluspräsentation übermittelt und dort von Signalinterpretationsfunktionen zur automatischen Adaption der Virtuellen Umgebung verwendet werden. Angedacht war hier die echtzeitnahe Einbindung von salienten Reizen, wie z.B. Wettererscheinungen, bauliche Strukturen oder nichtmenschliche Charaktere. Dieses Konzept wurde im bisherigen Verlauf des Projekts nur mit Wettererscheinungen umgesetzt und soll für weitergehende Forschungen ausgebaut werden.

Austausch von Experimentinformationen

Zusätzlich zu der realisierten Neurofeedback-Kommunikation über TCP wurde eine erweiterte Möglichkeit zum Austausch von Experimentinformationen etabliert. Diese Funktionalität erlaubt via TCP eine direkte Übermittlung von Probanden- und Studieninformationen zu einer entsprechenden Datenbank. Anonymisierte Probanden- und Experimentinformationen können dort bereitgestellt werden und erlauben dadurch eine effizientere Durchführung von Prozessabläufen. Auf die Vorteile einer direkten Kommunikation gegenüber einer dateibasierten Verbreitung wurde bereits in Kapitel 6.6 eingegangen. Sämtliche Client- und der Großteil der Serverfunktionalitäten wurden bereits implementiert, konnten aber aufgrund fehlender *Turbo BrainVoyager*-Schnittstellen noch nicht final getestet werden. Laut der Entwicklerfirma sind diese Schnittstellen für den Zugriff auf Experimentinformationen angedacht und sollen in zukünftigen Erweiterungen der Software bereitgestellt werden.

Literaturverzeichnis

Laboratory of Neurobiology, 2011. Cogent: Experimental stimuli design for brain imaging and psychophysical studies.

URL <http://www.vislab.ucl.ac.uk/cogent.php>

3DVIA Corp., 2011. Virtools.

URL <http://www.3dvia.com>

Adamovich, S. V., August, K., Merians, A., Tunik, E., 2009. A virtual reality-based system integrated with fMRI to study neural mechanisms of action observation-execution: A proof of concept study. *Restorative Neurology and Neuroscience* 27 (3), 209–223.

Aguirre, G. K., 1998. Neural components of topographical representation. *Proceedings of the National Academy of Sciences* 95 (3), 839–846.

Aguirre, G. K., Detre, J. A., Alsup, D. C., D'Esposito, M., 1996. The parahippocampus subserves topographical learning in man. *Cerebral Cortex* 6 (6), 823–829.

Andreano, J., Liang, K., Kong, L., Hubbard, D., Wiederhold, B. K., Wiederhold, M. D., 2009. Auditory cues increase the hippocampal response to unimodal virtual reality. *Cyberpsychology & Behavior* 12 (3), 309–313.

Ashdown, I., DiLaura, D., Mardaljevic, J., Rushmeier, H., Shakespeare, R., 1995. Near-field photometry: Measuring and modeling complex 3D light sources. *Sig-Graph 1995 Course Notes*, 1–15.

Autodesk GmbH, 2011. Autodesk 3ds Max.

URL <http://www.autodesk.de>

Baddeley, A. D., 2000. The episodic buffer: A new component of working memory? *Trends in Cognitive Sciences* 4 (11), 417–423.

Baecke, S., Luetzkendorf, R., Tempelmann, C., Mueller, C., Adolf, D., Scholz, M., Bernarding, J., 2009. Event-related functional magnetic resonance imaging (efMRI) of depth-by-disparity perception: Additional evidence for right-hemispheric lateralization. *Experimental Brain Research* 196, 453–458.

Bagarinao, E., Matsuo, T., Nakai, T., Sato, S., 2003. Estimation of general linear model coefficients for real-time application. *NeuroImage* 19, 422–429.

- Bailenson, J. N., Blascovich, J., Guadagno, R. E., 2008. Self-representations in immersive virtual environments. *Journal of Applied Social Psychology* 38 (11), 2673–2690.
- Bala, K., Dorsey, J., Teller, S., 1999. Radiance interpolants for accelerated bounded-error ray tracing. *ACM Transactions on Graphics* 18 (3), 213–256.
- Baumann, S. B., Neff, C., Fetzick, S., Stangl, G., Basler, L., Vereneck, R., Schneider, W., 2003. A virtual reality system for neurobehavioral and functional MRI studies. *Cyberpsychology & Behavior* 6 (3), 259–266.
- Baumgartner, C., Lindinger, G., Lurger, S., Aull, S., Bacher, J., Leutmezer, F., Olbrich, A., Pataria, E., Serles, W., Deecke, L., 1998. Das prolongierte Video-EEG-Monitoring in der Differentialdiagnose von Anfällen und in der prächirurgischen Epilepsiediagnostik. *Wiener medizinische Wochenschrift* 2, 2–8.
- BCI Competition, 2011. BCI Competition.
URL <http://www.bbci.de/competition/>
- Beck, L., Wolter, M., Mungard, N., Kuhlen, T., Sturm, W., 2007. Combining virtual reality and functional magnetic resonance imaging (fMRI): Problems and solutions. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (4799), 335–348.
- Beckermann, A., 2008. *Das Leib-Seele-Problem: Eine Einführung in die Philosophie des Geistes*, 1. Auflage. UTB GmbH, Stuttgart.
- Birbaumer, N., Ghanayim, N., Hinterberger, T., Iversen, I., Kotchoubey, B., Kübler, A., Perelmouter, J., Taub, E., Flor, H., 1999. A spelling device for the paralysed. *Nature* 398 (6725), 297–298.
- Birbaumer, N., Schmidt, R. F., 2006. *Biologische Psychologie*, 6. Auflage. Springer, Heidelberg.
- Birn, J., 2009. *Lighting & Rendering: 3D-Grafiken meisterhaft beleuchten - Realistische Texturen entwickeln*. München: Addison Wesley.
- Bischof, W. F., Boulanger, P., 2003. Spatial navigation in virtual reality environments: An EEG analysis. *Cyberpsychology & Behavior* 6 (5), 487–495.
- Bishop, C. M., 2006. *Pattern recognition and machine learning*. Springer, New York.
- Blanchard, G., Blankertz, B., 2004. BCI competition 2003 - data set IIa: Spatial patterns of self-controlled brain rhythm modulations. *IEEE Transactions on Biomedical Engineering* 51 (6), 1062–1066.
- Blanchette, J., Summerfield, M., 2009. *C++-GUI-Programmierung mit Qt 4: Die offizielle Einführung*, 2. Auflage. Addison-Wesley, München und Boston.

- Blankertz, B., Mueller, K., Curio, G., Vaughan, T., Schalk, G., Wolpaw, J., Schlaeggl, A., Neuper, C., Pfurtscheller, G., Hinterberger, T., Schroeder, M., Birbaumer, N., 2004. The BCI competition 2003: progress and perspectives in detection and discrimination of EEG single trials. *IEEE Transactions on Biomedical Engineering* 51, 1044–1051.
- Bloch, F., Hansen, W. W., Packard, M., 1946. Nuclear induction. *Physiological Review* (69), 127–128.
- Blumberg, B. M., Galyean, T. A., 1995. Multi-level direction of autonomous creatures for real-time virtual environments. *SigGraph Conference Proc.*, 47–54.
- Brain Innovation, 2002-2006. Turbo BrainVoyager.
URL <http://www.brainvoyager.com>
- Brill, M., 2009. Virtuelle Realität. Informatik im Fokus. Springer, Berlin.
URL <http://www.gbv.de/dms/bsz/toc/bsz287091465inh.pdf>
- Bröhl, A., 1995. Das V-Modell: Der Standard für die Softwareentwicklung mit Praxisleitfaden, 2. Auflage. Oldenbourg Verlag, München und Wien.
- Brown, M. A., Semelka, R. C., 2010. MRI: Basic principles and applications, 3. Auflage. Wiley-Blackwell, Hoboken (N.J.).
- Burdea, G., Coiffet, P., 2003. Virtual reality technology, 2. Auflage. J. Wiley-Interscience, Hoboken and N.J.
- Burgess, N., Maguire, E. A., O’Keefe, J., 2002. The human hippocampus and spatial and episodic memory. *Neuron* 35 (4), 625–641.
- Buxton, R., 2003. Introduction to functional magnetic resonance imaging: Principles and techniques. Cambridge Univ. Press, Cambridge.
- Carlson, C., 2003. A critical history of computer graphics and animation: Section 17: Virtual reality and artificial environments.
URL <https://design.osu.edu/carlson/history/lessons.html>
- Chapin, J., Moxon, K., Markowitz, R., Nicolelis, M., 1999. Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex. *Nature Neuroscience* 2, 664–670.
- Clausen, J., 2006. Ethische Aspekte von Gehirn-Computer-Schnittstellen in motorischen Neuroprothesen. *International Review of Information Ethics* 5, 25–32.
- Coben, R., Evans, J., 2010. Neurofeedback and neuromodulation techniques and applications, 1. Auflage. Academic Press, London.

- Codella, C., Jalili, R., Koved, L., Lewis, J., Ling, D., Lipscomb, J., Rabenhorst, D., Wang, C., Norton, A., Sweeney, P., Turk, G., 1992. Interactive simulation in a multi-person virtual world. Proceedings ACH CHI '92 Conference on Human Factors in Computing Systems, Tools & Architecture for Virtual Reality and Multi-User shared Data, 329–334.
- Coulouris, G., Dollimore, J., Kindberg, T., 2003. Verteilte Systeme: Konzepte und Design, 3. Auflage. Informatik. Pearson Studium, München.
URL <http://www.gbv.de/dms/hebis-darmstadt/toc/103536795.pdf>
- Cox, R. W., Jesmanowicz, A., Hyde, J. S., 1995. Real-time functional magnetic resonance imaging. *Magnetic Resonance in Medicine* 33 (2), 230–236.
- Craig, A. B., Sherman, W. R., Will, J. D., 2009. Developing virtual reality applications: Foundations of effective design. Elsevier/Morgan Kaufmann Publishers, Amsterdam and Boston.
- Debevec, P., Hawkins, T., Tchou, C., Duiker, H.-P., Sarokin, W., Sagar, M., 2000. Acquiring the reflectance field of a human face. Proc. of the 27th annual conference on Computer graphics and interactive techniques, 145–156.
- deCharms, R. C., 2008. Applications of real-time fMRI. *Nature Reviews Neuroscience* 9 (9), 720–729.
- deCharms, R. C., Maeda, F., Glover, G. H., Ludlow, D., Pauly, J. M., Soneji, D., Gabrieli, J. D., Mackey, S. C., 2005. Control over brain activation and pain learned by using real-time functional MRI. *Proceedings of the National Academy of Sciences* 102 (51), 18626–18631.
- Descartes, R., 1641. *Meditationes de prima philosophia: Meditationen über die erste Philosophie*, Neuauflage, 2008, 1. Auflage. Meiner, Hamburg.
- D'Esposito, M., Ballard, D., Aguirre, G., Zarahn, E., 1998. Human prefrontal cortex is not specific for working memory: A functional MRI study. *NeuroImage* 8 (3), 274–282.
- Doeller, C. F., Barry, C., Burgess, N., 2010. Evidence for grid cells in a human memory network. *Nature* 463 (7281), 657–661.
- Dössel, O., 2000. *Bildgebende Verfahren in der Medizin: Von der Technik zur medizinischen Anwendung*. Springer, Berlin.
URL <http://www.worldcat.org/oclc/612186161>
- Duda, R., Hart, P., Stork, D., 2004. *Pattern Classification*, 2. Auflage. Wiley-Blackwell, Hoboken.
- Eckstein, J., 1999. *Echtzeitfähige Kollisionserkennung für Virtual Reality Anwendungen*. Saarbrücken, Univ, München.

- Eeckhout, L., 2010. Computer architecture performance evaluation methods. Morgan & Claypool Publishers, San Rafael, California.
- Ehlers, W., Holder, A., 2007. Psychologische Grundlagen, Entwicklung und Neurobiologie. Klett-Cotta, Stuttgart.
- Ekstrom, A. D., Kahana, M. J., Caplan, J. B., Fields, T. A., Isham, E. A., Newman, E. L., Fried, I., 2003. Cellular networks underlying human spatial navigation. *Nature* 425 (6954), 184–188.
- Elbert, T., Rockstroh, B., 1987. Threshold regulation: A key to the understanding of the combined dynamics of EEG and event-related potentials. *Journal of Psychophysiology* 1 (4), 317–333.
- Ericson, C., 2005. Real-time collision detection. Elsevier; Morgan Kaufmann, Amsterdam.
- Esposito, F., Seifritz, E., Formisano, E., Morrone, R., Scarabino, T., Tedeschi, G., Cirillo, S., Goebel, R., Di Salle, F., 2003. Real-time independent component analysis of fMRI time-series. *NeuroImage* 20, 2209–2224.
- Fahrmeir, L., Künstler, R., Pigeot, I., Tutz, G., 2004. Statistik: Der Weg zur Datenanalyse, 5. Auflage. Springer, Berlin.
- Farin, G. E., Hansford, D., 2003. Lineare Algebra. Ein geometrischer Zugang. Springer, Berlin.
- Feinberg, A. W., Parker, K. K., 2010. Surface-initiated assembly of protein nanofabrics. *Nano Letters* 10, 2184–2191.
- Fetz, E. E., Finocchio, D., 1975. Correlations between activity of motor cortex cells and arm muscles during operantly conditioned response patterns. *Experimental Brain Research* 23, 217–240.
- Fisher, S., 1985. Virtual interface environment. Space Station Human Factors Research Review, Volume IV: Inhouse Advanced Development and Research, NASA Conference Publication 2426, Dec. 3-6, NASA Ames Research Center, Moffett Field, California, 94035.
- Fisher, S., 1986. Virtual interface environment. IEEE/AIAA 7th Digital Avionics Systems Conference, Fort Worth, Texas, October 13-16.
- Fowler, M., 1997. Analysis patterns: Reusable object models. Addison Wesley.
- Frackowiak, R. S. J., 2004. Human brain function, 2. Auflage. Elsevier Academic Press, Amsterdam und Boston.
- Frings, L., Wagner, K., Quiske, A., Schwarzwald, R., Spreer, J., Halsband, U., Schulze-Bonhage, A., 2006. Precuneus is involved in allocentric spatial location encoding and recognition. *Experimental Brain Research* 173, 661–672.

- Friston, K. J., 2007. Statistical parametric mapping: The analysis of functional brain images, 1. Auflage. Elsevier/Academic Press, Amsterdam und Boston.
- Friston, K. J., Holmes, A. P., Worsley, K. J., Poline, J. B., Frith, C. D., Frackowiak, R. S. J., 1995. Statistical parametric maps in functional imaging - a general linear approach. *Human Brain Mapping* 2, 189–210.
- Garcia-Palacios, A., Hoffman, H. G., Richards, T. R., Seibel, E. J., Sharar, S. R., 2007. Use of virtual reality distraction to reduce claustrophobia symptoms during a mock magnetic resonance imaging brain scan: A case report. *Cyberpsychology & Behavior* 10 (3), 485–488.
- Gedat, E., Braun, J., Sack, I., Bernarding, J., 2004. Prospective registration of human head magnetic resonance images for reproducible slice positioning using localizer images. *Journal of Magnetic Resonance Imaging* 20, 581–587.
- Gembris, D., Taylor, J., Schor, S., Frings, W., Suter, D., Posse, S., 2000. Functional magnetic resonance imaging in real time (FIRE): Sliding-window correlation analysis and reference-vector optimization. *Magnetic Resonance in Medicine* (43), 259–268.
- Goesele, M., 2004. New acquisition techniques for real objects and light sources in computer graphics, 1. Auflage. Books on Demand GmbH.
- Goodman, J. E., O'Rourke, J., 2004. Handbook of discrete and computational geometry. Chapman and Hall/CRC, Boca Raton.
- Goral, C., Torrance, K., Greenberg, D., Battaile, B., 1984. Modelling the interaction of light between diffuse surfaces. *Computer Graphics* 18 (3), 213–222.
- Green, M., 1984. Report on dialogue specification tools. *Computer Graphics Forum* 3 (4), 305–314.
- Grehn, J., 1998. Metzler-Physik. Gesamtband, 2. Auflage. Schroedel, Hannover.
- Grüningen, D. v., 2008. Digitale Signalverarbeitung: Bausteine, Systeme, Anwendungen. FO Print and Media, Zürich.
- Grzeschik, R., 2007. Entwicklung einer adaptiven VR-Umgebung für die Anwendung im realtime-fMRI. Diplomarbeit am Institut für Biometrie und Medizinische Informatik, Otto-von-Guericke Universität, Magdeburg.
- Guger, C., Ramoser, H., Pfurtscheller, G., 2000. Real-time EEG analysis with subject-specific spatial patterns for a brain-computer interface (BCI). *IEEE Transactions on Rehabilitation Engineering* 8 (4), 447–456.
- Haacke, E. M., 1999. Magnetic resonance imaging: Physical principles and sequence design. Wiley-Liss, New York (NY).
URL <http://www.gbv.de/dms/ohb-opac/266744052.pdf>

- Hassabis, D., Chu, C., Rees, G., Weiskopf, N., Molyneux, P. D., Maguire, E. A., 2009. Decoding neuronal ensembles in the human hippocampus. *Current Biology* 19 (7), 546–554.
- Havok Inc., 2012. Vision Engine Framework.
URL <http://www.havok.com/products/vision-engine>
- Heinrich, H., Gevensleben, H., Strehl, U., 2007. Annotation: Neurofeedback - train your brain to train behaviour. *Journal of Child Psychology and Psychiatry* 48 (1), 3–16.
- Hinterberger, T., Veit, R., Strehl, U., Trevorrow, T., Erb, M., Kotchoubey, B., Flor, H., Birbaumer, N., 2003. Brain areas activated in fMRI during self-regulation of slow cortical potentials (SCPs). *Experimental Brain Research* 152, 113–152.
- Hinterberger, T., Weiskopf, N., Veit, R., Wilhelm, B., Betta, E., Birbaumer, N., 2004. An EEG-driven brain-computer interface combined with functional magnetic resonance imaging (fMRI). *IEEE Transactions on Biomedical Engineering* 51 (6), 971–974.
- Hoffman, H., Patterson, D., Carrougher, G., 2000. Use of virtual reality for adjunctive treatment of adult burn pain during physical therapy: a controlled study. *The Clinical Journal of Pain* 16 (3), 244–250.
- Hoffman, H. G., Richards, T., Coda, B., Richards, A., Sharar, S. R., 2003. The illusion of presence in immersive virtual reality during an fMRI brain scan. *Cyberpsychology & Behavior* 6 (2), 127–131.
- Hollmann, M., Moench, T., Mueller, C., Bernarding, J., 2009. Predicting human decisions in socioeconomic interaction using real-time functional magnetic resonance imaging (rtfMRI). *Proceedings of the SPIE Medical Imaging, Lake Buena Vista, Florida, USA*.
- Hollmann, M., Moench, T., Mulla-Osman, S., Tempelmann, C., Stadler, J., Bernarding, J., 2008. A new concept of a unified parameter management, experiment control, and data analysis in fMRI: Application to real-time fMRI at 3T and 7T. *Journal of Neuroscience Methods* 175 (1), 154–162.
- Hopcroft, J. E., Motwani, R., Ullman, J. D., 2011. *Einführung in Automatentheorie, formale Sprachen und Berechenbarkeit*. Pearson Studium, München.
- Horowitz, A. L., 1995. *MRI physics for radiologists: A visual approach*, 3. Auflage. Springer-Verlag, New York.
- Huettel, S. A., Song, A. W., McCarthy, G., 2008. *Functional magnetic resonance imaging*, 2. Auflage. Sinauer Associates, Sunderland und Mass.
- Jäncke, L., 2005. *Methoden der Bildgebung in der Psychologie und den kognitiven Neurowissenschaften*, 1. Auflage. Kohlhammer, Stuttgart.

- Jeppard, P., Matthews, P. M., Smith, S. M., 2001. Functional MRI: An introduction to methods. Oxford University Press, Oxford and New York.
- Kajiya, J. T., 1986. The rendering equation. *Computer Graphics* 20 (4), 143–150.
- Kallay, M., 1984. The complexity of incremental convex hull algorithms in r^d . *Inform. Process. Lett.* 19:197.
- Kim, G. J., 2005. Designing virtual reality systems: The structured approach. Springer, London.
URL <http://www.gbv.de/dms/bsz/toc/bsz116108541inh.pdf>
- Kopetz, H., 2011. Real-time systems: Design principles for distributed embedded applications, 2. Auflage. Springer, New York.
- Krueger, M. W., 1983. Artificial reality. Addison-Wesley, Reading and Mass.
- Larman, C., 2006. Applying UML and patterns: An introduction to object-oriented analysis and design and iterative development, 3. Auflage. Prentice Hall [u.a.], Upper Saddle River and NJ.
URL <http://www.gbv.de/dms/bowker/toc/9780131489066.pdf>
- Lauterbur, P. C., 1973. Image formation by induced local interactions: Examples employing nuclear magnetic resonance. *Nature* 242 (5394), 190–191.
- Leins, U., Goth, G., Hinterberger, T., Klinger, C., Rumpf, N., Strehl, U., 2007. Neurofeedback for children with ADHD: A comparison of SCP and Theta/Beta protocols. *Biofeedback and Self Regulation* 32 (2), 73–88.
- Lem, S., 1981. *Summa technologiae*, 1. Auflage. Suhrkamp, Frankfurt am Main.
- Levoy, M., Hanrahan, P., 1996. Light field rendering. *Proc. of SigGraph 1996*, 31–42.
- Lindstrom, P., Turk, G., 1999. Evaluation of memoryless simplification. *IEEE Transactions on Visualization and Computer Graphics* 5 (2), 98–115.
- Luehrs, M., Mueller, C., Bernarding, J., 2011. A real-time fMRI communication interface for Turbo-BrainVoyager 3.0 using the TBV 3.0 plugin interface. *Proceedings of the Human Brain Mapping, Quebec-City, Canada, June 26-30, 2011*.
- Maguire, E. A., 1998. Knowing where and getting there: A human navigation network. *Science* 280 (5365), 921–924.
- Maguire, E. A., Frackowiak, R. S. J., Frith, C. D., 1997. Recalling routes around London: Activation of the right hippocampus in taxi drivers. *Journal of Neuroscience* 17 (18), 7103–7110.
- Maguire, E. A., Frith, C. D., Burgess, N., Donnett, J. G., OKeefe, J., 1998. Knowing where things are: Parahippocampal involvement in encoding object locations in virtual large-scale space. *Journal of Cognitive Neuroscience* 10 (1), 61–76.

- Maguire, E. A., Nannery, R., Spiers, H. J., 2006. Navigation around London by a taxi driver with bilateral hippocampal lesions. *Brain* 129 (11), 2894–2907.
- Mansfield, P., 1977. Multi-planar image formation using nmr spin echoes. *Journal of Physics C: Solid State Physics* 10 (3), 55–58.
- Marsh, R., Hao, X., Xu, D., Wang, Z., Duan, Y., Liu, J., Kangarlu, A., Martinez, D., Garcia, F., Tau, G. Z., Yu, S., Packard, M. G., Peterson, B. S., 2010. A virtual reality-based fMRI study of reward-based spatial learning. *Neuropsychologia* 48 (10), 2912–2921.
- Masic, N., Pfurtscheller, G., 1993. Neural network based classification of single-trial EEG data. *Artificial Intelligence in Medicine* 5 (6), 503–513.
- Mellet, E., Laou, L., Petit, L., Zago, L., Mazoyer, B., Tzourio-Mazoyer, N., 2010. Impact of the virtual reality on the neural representation of an environment. *Human Brain Mapping* 31 (7), 1065–1075.
- Mellinger, J., Schalk, G., Braun, C., Preissl, H., Rosenstiel, W., Birbaumer, N., Kübler, A., 2007. An MEG-based brain–computer interface (BCI). *NeuroImage* 36 (3), 581–593.
- Meseth, J., 2006. Towards predictive rendering in virtual reality. Dissertation, Universität Bonn, Bonn.
- Mezger, J., 2001. Effiziente Kollisionsdetektion in der Simulation von Textilien. Dissertation, Universität Tübingen, Tübingen.
- Microsoft Corp., 2011a. DirectX SDK.
URL <http://msdn.microsoft.com/de-de/directx/>
- Microsoft Corp., 2011b. Visual Studio 2008.
URL <http://www.microsoft.com>
- Mueller, C., Baecke, S., Luetzkendorf, R., Bernarding, J., 2010a. A human brain interface using a virtual environment system for real-time fMRI. German-Chinese Workshop on High Field MRI, April 26-27, Magdeburg, Germany.
- Mueller, C., Baecke, S., Luetzkendorf, R., Bernarding, J., 2012a. Ein Framework zur Echtzeit-Adaptierung von Virtual-Reality-Paradigmen in Neurofeedback-Experimenten. *Medizinische Bildverarbeitung für die computergestützte Diagnostik und Therapie, GI-Edition of Lecture Notes in Informatics (LNI) P-208*, 1727–1737.
- Mueller, C., Hollmann, M., Henniger, N., Bernarding, J., 2010b. A virtual environment system for real-time fMRI (3T, 7T) using direct communication via TCP/IP. *Proceedings of the Human Brain Mapping, Barcelona, Spain, June 6-11, 2010*.

- Mueller, C., Hollmann, M., Luetzkendorf, R., Baecke, S., Bernarding, J., 2009. A high-field human brain interface using a modular virtual environment system for real-time fMRI. Proceedings of the International Society for Magnetic Resonance in Medicine, Honolulu, Hawaii, USA, April 19-24, 2009.
- Mueller, C., Luehrs, M., Baecke, S., Adolf, D., Luetzkendorf, R., Luchtman, M., Bernarding, J., 2012b. Building virtual reality fMRI paradigms: A framework for presenting immersive virtual environments. *Journal of Neuroscience Methods* 209 (2), 290–298.
- Mueller, C., Luehrs, M., Baecke, S., Bernarding, J., 2011a. Stimulus application presenting virtual environments for real-time fMRI experiments. Proceedings of the European Society for Magnetic Resonance in Medicine and Biology, Leipzig, Germany, October 6-8, 2011.
- Mueller, C., Luehrs, M., Thoms, A., Baecke, S., Bernarding, J., 2011b. A Qt-based virtual environment stimulus application for rfMRI using an improved TCP/IP framework. Proceedings of the Human Brain Mapping, Quebec-City, Canada, June 26-30, 2011.
- Mynatt, E. D., Hudson, S. E., Fitzpatrick, G., 2010. Effects of interactivity and 3D-motion on mental rotation brain activity in an immersive virtual environment: Conference proceedings, Atlanta, Ga, USA, April 10-15, 2010. Association for Computing Machinery, New York (N.Y.).
- Neurobehavioral Systems, 2011. Presentation: Stimulus delivery and experimental control program for neuroscience.
URL <http://www.neurobs.com>
- Nicodemus, F. E., Richmond, J. C., Hsia, J. J., Ginsberg, I. W., Limperis, T., 1977. Geometrical Considerations and Nomenclature for Reflectance. Technical Report BMS Monograph 160, National Bureau of Standards.
- Nokia Corp., 2011. Qt: Cross platform application and UI framework.
URL <http://qt.nokia.com>
- NVIDIA Corp., 2011. PhysX.
URL <http://www.nvidia.com>
- Ogawa, S., 1990. Brain magnetic resonance imaging with contrast dependent on blood oxygenation. *Proceedings of the National Academy of Sciences* 87 (24), 9868–9872.
- Olsen, D., 1992. User interface management systems. Models and algorithms. M. Kaufmann Publishers, San Mateo.
- Othmer, S., Kaiser, D., 2000. Implementation of virtual reality in EEG biofeedback. *Cyberpsychology & Behavior* 3 (3), 415–420.

- Pfaff, G. E., 1985. User interface management systems. Springer, Berlin.
- Pfurtscheller, G., Leeb, R., Keinrath, C., Friedman, D., Neuper, C., Guger, C., Slater, M., 2006. Walking from thought. *Brain Research* 1071 (1), 145–152.
- Pfurtscheller, G., Neuper, C., 2001. Motor imagery and direct brain-computer communication. *Proceedings of the IEEE* 89 (7), 1123–1134.
- Pharr, M., Humphreys, G., 2004. Physically based rendering. Elsevier/Morgan Kaufmann.
- Pine, D., Grun, J., Maguire, E. A., Burgess, N., Zarahn, E., Koda, V., Fyer, A., Szeszko, P. R., Bilder, R. M., 2002. Neurodevelopmental aspects of spatial navigation: A virtual reality fMRI study. *NeuroImage* 15 (2), 396–406.
- Pollmann, S., 2008. Allgemeine Psychologie. Reinhardt, München [u.a.].
- Pollock, D. S. G., 1999. A handbook of time-series analysis, signal processing and dynamics. Academic, San Diego and London.
- Posse, S., Ackley, E., Mutihac, R., Rick, J., Shane, M., Murray-Krezan, C., Zaitsev, M., Speck, O., 2012. Enhancement of temporal resolution and BOLD sensitivity in real-time fMRI using multi-slab echo-volumar imaging. *NeuroImage* 61 (1), 115–130.
- Posse, S., Fitzgerald, D., Gao, K., Habel, U., Rosenberg, D., Moore, G. J., Schneider, F., 2003. Real-time fMRI of temporolimbic regions detects amygdala activation during single-trial self-induced sadness. *NeuroImage* 18 (3), 760–768.
- Preim, B., Bartz, D., 2007. Visualization in Medicine. Theory, Algorithms, and Applications, 1. Auflage. Morgan Kaufmann, Amsterdam.
- Preim, B., Dachselt, R., 2010. Interaktive Systeme: Band 1: Grundlagen, Graphical User Interfaces, Informationsvisualisierung, 2. Auflage. Springer, Berlin.
- Psychology Software Tools, I., 2011. VR Worlds 2.
URL <http://www.pstnet.com>
- Rabrait, C., Ciuciu, P., Ribes, A., Poupon, C., LeRoux, P., Dehaine-Lambertz, G., LeBihan, D., Lethimonnier, F., 2008. High temporal resolution functional MRI using parallel echo volumar imaging. *Magnetic Resonance Imaging* 27 (4), 744–753.
- Reddy, M., 1996. SCROOGE: Perceptually-driven polygon reduction. *Computer Graphics Forum* 15 (4), 191–203.
- Rief, W., Birbaumer, N. (Hrsg.), 2006. Biofeedback: Grundlagen, Indikationen, Kommunikation, praktisches Vorgehen in der Therapie, 2. Auflage. Schattauer, Stuttgart.
URL <http://catalogue.bnf.fr/ark:/12148/cb39971396j>

- Rothbaum, B. O., Hodges, L., Alarcon, R., Ready, D., Shahar, F., Graap, K., Pair, J., Hebert, P., Gotz, D., Wills, B., Baltzell, D., 1999. Virtual reality exposure therapy for PTSD vietnam veterans: A case study. *Journal of Traumatic Stress* 12 (2), 263–271.
- Rothbaum, B. O., Hodges, L., Smith, S., Lee, J. H., Price, L., 2000. A controlled study of virtual reality exposure therapy for the fear of flying. *Journal of Consulting and Clinical Psychology* 68 (6), 1020–1026.
- RWTH, 2011a. ReactorMan.
URL <http://www.rz.rwth-aachen.de>
- RWTH, 2011b. VR Software ViSTA.
URL <http://www.rz.rwth-aachen.de>
- Sanchez-Vives, M. V., Slater, M., 2005. Opinion: From presence to consciousness through virtual reality. *Nature Reviews Neuroscience* 6 (4), 332–339.
- Sanei, S., Chambers, J., 2007. EEG signal processing. John Wiley & Sons.
URL <http://books.google.de/books?id=vIuCV2IKwasC>
- Savoy, R. L., 1998. Encyclopedia of Neuroscience: Functional magnetic resonance imaging (fMRI). *Encyclopedia of Neuroscience*.
- Savoy, R. L., 2005. Experimental design in brain activation MRI: Cautionary tales. *Brain Research Bulletin* 67 (5), 361–367.
- Savoy, R. L., 2009. Experimental design: Good rules ... and when to break them. mbic-fMRI-Workshop, University of Maastricht, Netherlands, February 16-19, 2009.
- Schatten, A., 2010. Best Practice Software-Engineering: Eine praxiserprobte Zusammenstellung von komponentenorientierten Konzepten, Methoden und Werkzeugen. Spektrum Akademischer Verlag, Heidelberg.
- Schneider, F., Fink, G. R., 2007. Funktionelle MRT in Psychiatrie und Neurologie.
URL <http://dx.doi.org/10.1007/978-3-540-68558-6>
- Schubert, T. W., 2003. The sense of presence in virtual environments: A three-component scale measuring spatial presence, involvement, and realness. *Zeitschrift für Medienpsychologie* 15 (2), 69–71.
- Sherman, W. R., Craig, A. B., 2003. Understanding virtual reality. Morgan Kaufmann, San Francisco.
- Sitaram, R., Caria, A., Veit, R., Gaber, T., Rota, G., Kuebler, A., Birbaumer, N., 2007a. fMRI brain-computer interface: A tool for neuroscientific research and treatment. *Computational Intelligence and Neuroscience* 2007, 1–10.

- Sitaram, R., Zhang, H., Guan, C., Thulasidas, M., Hoshi, Y., Ishikawa, A., Shimizu, K., Birbaumer, N., 2007b. Temporal classification of multichannel near-infrared spectroscopy signals of motor imagery for developing a brain–computer interface. *NeuroImage* 34 (4), 1416–1427.
- Slater, M., Usoh, M., Steed, A., 1994. Depth of presence in virtual environments. *Design* 3 (2), 1–33.
- Smyser, C., Grabowski, T., Frank, R., Haller, J., Bollinger, L., 2001. Real-time multiple linear regression for fMRI supported by time-aware acquisition and processing. *Magnetic Resonance in Medicine* 45, 289–298.
- Sorger, B., Reithler, J., Dahmen, B., Goebel, R., 2012. A real-time fMRI-based spelling device immediately enabling robust motor-independent communication. *Current Biology* 22 (14), 1333–1338.
- SPM, 2008. SPM 2008: SPM - statistical parametric mapping.
URL <http://www.fil.ion.ucl.ac.uk/spm>
- Stanney, K. W., 2002. *Handbook of virtual environments: Design, implementation, and applications*. Lawrence Erlbaum Associates, Mahwah, NJ.
- Steuer, J., 1992. Defining virtual reality: Dimensions determining telepresence. *Journal of Communication* 42 (4), 73–93.
- Strobel, A., Debener, S., Sorger, B., Peters, J. C., Kranczioch, C., Hoehstetter, K., Engel, A. K., Brocke, B., Goebel, R., 2008. Novelty and target processing during an auditory novelty oddball: a simultaneous event-related potential and functional magnetic resonance imaging study. *NeuroImage* 40 (2), 869–883.
- Stroebe, W., Jonas, K., Hewstone, M. (Hrsg.), 2002. *Sozialpsychologie: Eine Einführung*, 3. Auflage. Springer, Berlin [etc.].
- Sturm, W., 2009. *Lehrbuch der klinischen Neuropsychologie: Grundlagen, Methoden, Diagnostik, Therapie*, 2. Auflage. Spektrum Akademischer Verlag, Heidelberg.
- Sutcliffe, A., 2003. *Multimedia and virtual reality. Designing multisensory user interfaces*. Lawrence Erlbaum, Mahwah, N.J.
- Tan, D. S., Nijholt, A., 2010. *Human-computer interaction series*, 1. Auflage. Springer, London.
- Tanenbaum, A. S., van Steen, M., Muhr, J., 2003. *Verteilte Systeme: Grundlagen und Paradigmen*. Pearson Studium Informatik. Pearson Studium, München.
- Teller, S., Bala, K., Dorsey, J., 1996. Conservative radiance interpolants for ray tracing. *Proc. of the Eurographics Workshop on Rendering Techniques*, 257–268.

- Thoms, A., 2011. Modellierung und Entwicklung einer Virtuellen Umgebung zum Einsatz einer MR-basierten Echtzeit Hirn-Computer-Schnittstelle. Diplomarbeit am Institut für Biometrie und Medizinische Informatik, Otto-von-Guericke Universität, Magdeburg.
- Thoms, A., Mueller, C., Bernarding, J., 2011. A brain-computer interface for navigation through a virtual environment in real-time fMRI studies. Proceedings of the Human Brain Mapping, Quebec-City, Canada, June 26-30, 2011.
- Tsichritzis, D., Klug, A., 1978. The ANSI/X3/SPARC DBMS framework: Report of the study group on database management systems. *Inform. Systems* (3), 173–191.
- Ulbricht, C., Wilkie, A., Purgathofer, W., 2005. Verification of physically based rendering algorithms. *Eurographics State of the Art Reports*, 95–112.
- Vidal, J., 1977. Real-time detection of brain events in EEG. *Proceedings of the IEEE* 65 (5), 633–641.
- Voyvodic, J. T., 1999. Real-time fMRI paradigm control, physiology, and behavior combined with near real-time statistical analysis. *NeuroImage* 10, 91–106.
- Walter, B. and Drettakis, G., Parker, S., 1999. Interactive rendering using the render cache. *Rendering Techniques* 10, 235–246.
- Wang, W., Degenhart, A. D., Collinger, J. L., Vinjamuri, R., Sudre, G. P., Adelson, P. D., Holder, D. L., Leuthardt, E. C., Moran, D. W., Boninger, M. L., Schwartz, A. D., Crammond, D. J., Tyler-Kabara, E. C., Weber, D. J., 2009. Human motor cortical activity recorded with micro-ECoG electrodes during individual finger movements. *Proceedings of IEEE Eng Med Biol Soc.*, 586–589.
- Weiskopf, N., Mathiak, K., Bock, S. W., Scharnowski, F., Veit, R., Grodd, W., Goebel, R., Birbaumer, N., 2004a. Principles of a brain-computer interface (BCI) based on real-time functional magnetic resonance imaging (fMRI). *IEEE Transactions on Biomedical Engineering* 51 (6), 966–970.
- Weiskopf, N., Scharnowski, F., Veit, R., Goebel, R., Birbaumer, N., Mathiak, K., 2004b. Self-regulation of local brain activity using real-time functional magnetic resonance imaging (fMRI). *Journal of Physiology* 98 (4-6), 357–373.
- Weiskopf, N., Sitaram, R., Josephs, O., Veit, R., Scharnowski, F., Goebel, R., Birbaumer, N., Deichmann, R., Mathiak, K., 2007. Real-time functional magnetic resonance imaging: methods and applications. *Magnetic Resonance Imaging* 25 (6), 989–1003.
- Whitted, T., 1980. An improved illumination model for shaded display. *Communications of the ACM* 23 (6), 343–349.

- Wiederhold, B. K., Jang, D. P., Gevirtz, R. G., Kim, S. I., Kim, I. Y., Wiederhold, M. D., 2002. The treatment of fear of flying: A controlled study of imaginal and virtual reality graded exposure therapy. *IEEE Transactions on Information Technology in Biomedicine* 6 (3), 218–223.
- Wiederhold, B. K., Wiederhold, M. D., 2005. *Virtual reality therapy for anxiety disorders: Advances in evaluation and treatment*, 1. Auflage. American Psychological Association, Washington (DC).
URL <http://www.worldcat.org/oclc/54816879>
- Wolbers, T., Hegarty, M., Buechel, C., Loomis, J., 2008. Spatial updating: how the brain keeps track of changing object locations during observer motion. *Nature Neuroscience* 11 (10), 1223–1230.
- Wolpaw, J. R., Birbaumer, N., McFarland, D. J., Pfurtscheller, G., Vaughan, T. M., 2002. Brain-computer interfaces for communication and control. *Clinical Neurophysiology* 113 (6), 767–791.
- Woods, W., 1970. Transition network grammars for natural language analysis. *Communications of the ACM* 13 (10), 591–606.
- Wuestenberg, T., Giesel, F. L., Strasburger, H., 2005. Methodische Grundlagen der Optimierung funktioneller MR-Experimente. *Der Radiologe* 45 (2), 99–112.
- Yoo, S., Fairney, T., Chen, N., Choo, S., Panych, L. P., Park, H., Lee, S., Jolesz, F. A., 2004. Brain-computer interface using fMRI: Spatial navigation by thoughts. *NeuroReport* 15 (10), 1591–1595.
- Zachmann, G., 2000. *Virtual reality in assembly simulation: Collision detection, simulation algorithms, and interaction techniques*. Dissertation, Technische Universität, Darmstadt.

Anhang A

Probandenfragebogen

Probandenfragebogen VRCity-Studie

Datum:

Probandenkürzel:

1. Spielen Sie Computerspiele? Ja Nein
Wenn ja, welche?

.....

2. Haben Sie bereits an anderen funktionellen MRT Studien teilgenommen? Ja Nein
Wenn ja, welcher Art waren diese? (visuell, auditiv, etc.)

.....

3. Konnten Sie sich die virtuelle Umgebung gut einprägen? Ja Nein
Wenn nein, warum?

.....

4. Konnten Sie während der Pausen gut entspannen und die Gedanken von der virtuellen Umgebung lösen? Ja Nein
Wenn nein, bitte kurz beschreiben warum?

.....

.....

5. Wie empfanden sie das „freie Bewegen“ und die selbstständige, aktive Kontrolle beim Navigieren durch die virtuelle Welt? Bitte kurz erläutern.

.....

.....

6. Gab es zeitliche Phasen während des Experimentes wo Ja Nein
ihre Aufmerksamkeit nachgelassen hat?
Beschreiben Sie kurz mit ihren Worten, warum ihre Aufmerksamkeit konstant blieb
oder warum sie nachgelassen hat.

.....
.....

7. Was ist Ihnen während der Messung negativ aufgefallen?

.....
.....

Anhang B

Benutzerakzeptanztest

Datum:

VR Sense Performance Test

1. Wie lange arbeiten Sie bereits im neurowissenschaftlichen Kontext?

.....

2. Besitzen Sie bereits Kenntnisse bei der Erstellung von neurowissenschaftlichen Stimuli? Ja Nein

- Wenn ja, schätzen Sie wie viele Stimuli Sie bereits erstellt haben?

1 bis 3 Stimuli

3 bis 10 Stimuli

Mehr als 10 Stimuli

3. Haben Sie bereits Erfahrung bei der Verwendung von neurowissenschaftlicher Stimulussoftware, wie z.B. Presentation oder rtPresent? Ja Nein

- Wenn ja, welche Software haben Sie verwendet und wie hoch stufen Sie jeweils Ihre Kenntnisse auf einer Skala von 0 bis 10 (0 = sehr wenig Kenntnisse bis 10 = perfekte Kenntnisse) ein?

.....

.....

4. Haben Sie bereits Vorkenntnisse bei der Verwendung von Datenauswertesoftware, wie z.B. BrainVoyager QX oder SPM? Ja Nein

- Wenn ja, welche Software haben Sie verwendet und wie hoch stufen Sie jeweils Ihre Kenntnisse auf einer Skala von 0 bis 10 (0 = sehr wenig Kenntnisse bis 10 = perfekte Kenntnisse) ein?

.....

.....

Der vorliegende Test gliedert sich in zwei Hauptaufgaben. In der ersten Aufgabe erstellen Sie einen VR-Stimulus, speichern die Einstellungen dazu in einer BrainVoyager Experimentdatei (PRT-Datei) ab und prüfen anschließend die Funktionalität des Stimulus.

In der zweiten Aufgabe, laden Sie einen bereits bestehenden Stimulus aus einer BrainVoyager PRT-Datei ein, passen diesen selbstständig an und prüfen ebenfalls die Funktionalität.

Achtung!!! Bitte vermerken sie zu Beginn die Startzeit und am Ende die Endzeit ihres Tests. Beachten Sie, viele Schaltflächen beinhalten Tooltips, die ihnen die Funktionalität näher bringen sollen.

Startzeit (Bitte eintragen):

- a. Öffnen Sie die Software VRSenses. Starten Sie das Designen eines neuen neurowissenschaftlichen Experimentes („Design an neuroscientific experiment“). Erstellen Sie anschließend ein neues fMRT Experiment mit dem Knopf oben links oder über das Tastenkürzel: **Ctrl+Alt+N**.
Gehen sie durch die einzelnen Tabs und richten Sie eine fMRT-VR Stimulation, mit folgenden Parametern, ein:

<i>General Tab</i>	
VR Sense Experiment File	
Study ID	VRSenseAccept
Subject ID	xx00
Experiment Name	UserAcceptTest
Virtual Scene File	Engine/Scenes/Citymap/Citymap.vscene
<i>Stimulation Design Tab</i>	
BrainVoyager PRT file	
TR	2000
Volumes	Aktiv
Event Related Design	Aktiv
Anzahl der Conditions	3
1. Condition Name	Baseline
1. Condition: Nr of Intervals	1
1. Condition: Start Volume	1
1. Condition EndValue	3
2. Condition Name	VR
2. Condition: Nr of Intervals	1
2. Condition: Start Volume	4
2. Condition EndValue	9
3. Condition Name	Baseline
3. Condition: Nr of Intervals	1
3. Condition: Start Volume	10
3. Condition EndValue	12
<i>Feedback Control Settings Tab</i>	
Avatar Control	Button Feedback – IBMI

- b. Nach dem Eingeben der Daten, sichern Sie diese mit dem „Apply“-Button. Anschließend speichern Sie die Paradigma Informationen im „Stimulation Design Tab“ als PRT-Datei. Hierzu wählen Sie den Ordner: **„PerformanceTest“** und speichern sie die Informationen als: **„UserAcceptTest.prt“** ab.
- c. Prüfen Sie alle gemachten Informationen durch Starten des Stimulus. Dazu bestätigen Sie alle gemachten Einstellungen mit „OK“ und starten den Stimulus mit dem „Start“-Button links oder mit dem Tastenkürzel: **Ctrl+Alt+P**. Prüfen sie die Funktionalität des Stimulus durch betätigen der Taster am Gamepad.
- 2.1 Rufen sie den „fMRI-Experiment Dialog“ („Zahnrad-Knopf“ am linken Bildschirmrand) auf und laden Sie im **„Stimulation Design Tab“** folgende PRT-Datei: **example.prt**
- 2.2 Ändern Sie die Paradigmeneinstellungen wie folgt und speichern Sie diese als PRT-Datei mit Namen **„AcceptTest“** im Ordner **„PerformanceTest“** ab. Beachten Sie, dass einige Bedingungen (conditions) gelöscht, umbenannt sowie die Anzahl der Intervalle angepasst werden muss.

General Tab	
Experiment Name	PrtTest
Stimulation Design Tab	
BrainVoyager PRT file	
Milliseconds	aktiv
BlockDesign	aktiv
1. Condition Name	Baseline
1. Condition: Nr of Intervals	1
1. Condition: Start Volume	1
1. Condition EndValue	5
2. Condition Name	VR
2. Condition: Nr of Intervals	1
2. Condition: Start Volume	6
2. Condition EndValue	20

- 2.3 Nutzen Sie nun die Funktionalität zur automatischen Block-Design Wertberechnung und fügen Sie jeweils zwei Intervalle zur **„Baseline“** und **„VR“** Bedingung hinzu. Selektieren Sie dazu nacheinander beide Bedingungen und betätigen Sie jeweils den **„Add“-Knopf**, rechts neben der **„Interval“** Schaltfläche, bis bei **„NrOfIntervals“** der Wert **„3“** erscheint. Bestätigen Sie anschließend die Einstellungen mit „OK“ und testen Sie den Stimulus mit dem Tastenkürzel: **Ctrl+Alt+P**.

Endzeit (Bitte eintragen):

Bitte bewerten Sie abschließend die Funktionalität und die Benutzerfreundlichkeit der Software.

Sind während ihres Tests schwerwiegende Funktionsfehler oder Softwareabstürze aufgetreten? Wenn ja, bitte näher beschreiben und Zeitpunkt angeben. Ja Nein

.....
.....
.....

Bewerten Sie die Benutzerfreundlichkeit der Software auf einer Skala von 1 bis 10 (1 = schlecht, 10 = sehr gut).

.....

Was ist Ihnen besonders gut oder besonders schlecht an der Software aufgefallen?

.....
.....
.....