

Softwareframework zur universellen
Methodenentwicklung für ein fMRT-BCI:
Adaptive Paradigmen und Echtzeitdatenanalyse

Von der Fakultät für Mathematik und Informatik der Universität Leipzig
angenommene

DISSERTATION

zur Erlangung des akademischen Grades

DOCTOR RERUM NATURALIUM

(Dr. rer. nat.)

im Fachgebiet

Informatik

vorgelegt von Diplom-Informatikerin Lydia Hellrung

geboren am 10.06.1981 in Leinefelde, jetzt Leinefelde-Worbis

Die Annahme der Dissertation wurde empfohlen von:

Prof. Dr. Martin Bogdan (Leipzig)

Prof. Dr. Arno Villringer (Leipzig)

Prof. Dr. Klaus-Robert Müller (Berlin)

Die Verleihung des akademischen Grades erfolgt mit Bestehen der
Verteidigung am **28.01.2015** mit dem Gesamtprädikat **magna cum
laude**.

Abstract

Softwareframework zur universellen Methodenentwicklung für ein fMRT-BCI: Adaptive Paradigmen und Echtzeitdatenanalyse

Die funktionelle Magnetresonanztomographie (fMRT) ist ein nicht-invasives Bildgebungsverfahren, mit dem Veränderungen der neuronalen Aktivität im Gehirn mit hoher räumlicher Auflösung erfasst werden können. Mit der fMRT-Bildgebung bei neurowissenschaftlichen Experimenten wurden in den letzten beiden Jahrzehnten bedeutende Erkenntnisse für die Hirnforschung und Medizin gewonnen. Mit Hilfe dieser Methode werden neuronale Aktivitätsunterschiede bei der Durchführung einer bestimmten Aufgabe, z. B. dem Betrachten von Bildern mit emotionalen Inhalten, erfasst und die Daten unabhängig von der Messung zu einem späteren Zeitpunkt statistisch ausgewertet. Mit Hilfe des technischen Fortschritts im letzten Jahrzehnt ist es darüber hinaus möglich geworden, fMRT-Daten direkt zur Aufnahmezeit zu verarbeiten und auszuwerten. Dies wird als Echtzeit-fMRT bezeichnet, wenn die Datenverarbeitung schneller erfolgt als die Aufnahme eines Hirnvolumens (aktuell ca. zwei Sekunden). Die Ergebnisse der Echtzeitdatenverarbeitung können dann wiederum als Steuerbefehle für verschiedene Anwendungen verwendet werden. Daher wird dies auch als Hirn-Computer-Schnittstelle (*Brain Computer Interface, BCI*) mittels fMRT bezeichnet. Die Echtzeitverarbeitung von fMRT-Daten ermöglicht mehrere neue Anwendungen. Dazu gehören unter anderem die Qualitätskontrolle zur Laufzeit von fMRT-Experimenten, die schnelle funktionelle Lokalisierung von Hirnarealen entweder für neurowissenschaftliche Experimente oder intra-operativ, die Kontrolle des Experimentes mittels des Probandenverhaltens und insbesondere die Möglichkeit, sogenannte fMRT-Neurofeedbackexperimente durchzuführen. Bei diesen Experimenten lernen Probanden, die Aktivität von definierten Hirnarealen willentlich zu beeinflussen. Das Ziel dabei ist, Veränderungen in ihrem Verhalten zu generieren. Die Umsetzung eines BCIs mittels Echtzeit-fMRT ist eine interdisziplinäre Aufgabenstellung von MR-Physik, Informatik und Neurowissenschaften um das Verständnis des menschlichen Gehirns zu verbessern und neue Therapieansätze zu gestalten. Für diese hard- und softwaretechnisch anspruchsvolle Aufgabe gibt es einen enormen Bedarf an neuen Algorithmen zur Datenverarbeitung und Ansätzen zur verbesserten Datenakquise.

In diesem Zusammenhang präsentiert die vorliegende Arbeit ein neues Softwareframework, das einerseits eine integrierte Gesamtlösung für die Echtzeit-fMRT darstellt und in seinen Teilmodulen eine abstrakte Basis für eine universelle Methodenentwicklung anbietet. Diese Arbeit

beschreibt die grundlegenden abstrakten Konzepte und die Implementierung in ein neues Softwarepaket namens '*Brain Analysis in Real-Time*' (*BART*). Der Fokus der Arbeit liegt auf zwei Kernmodulen, die für universelle Gestaltung von sogenannten adaptiven Paradigmen und die Einbindung von Echtzeit-fMRT-Datenverarbeitungsalgorithmen konzipiert sind. Bei adaptiven Paradigmen werden zur Laufzeit eines Experiments physiologische Parameter (z. B. Herzrate) oder Verhaltensdaten (z. B. Augenbewegungen) simultan zu den fMRT-Daten erfasst und analysiert, um die Stimulation eines Probanden entsprechend zu adaptieren. Damit kann die Zuverlässigkeit der Daten, die zur Auswertung zur Verfügung stehen, optimiert werden. Die vorliegende Arbeit präsentiert das dazu notwendige abstrakte Grundkonzept des neuen Softwareframeworks und die ersten konkreten Implementierungen für die Datenverarbeitung und adaptive Paradigmen. Das Framework kann zukünftig mit neuen methodischen Ideen erweitert werden. Dazu gehören die Einbindung neuer Datenverarbeitungsalgorithmen, wie z. B. Konnektivitätsanalysen und die Adaption von Paradigmen durch weitere physiologische Parameter. Dabei ist insbesondere die Kombination mit EEG-Signalen von großem Interesse. Außerdem bietet das System eine universelle Grundlage für die zukünftige Arbeit an Neurofeedbackexperimenten.

Das in dieser Arbeit entwickelte Framework bietet im Vergleich zu bisher vorgestellten Lösungsansätzen ein Ein-Computer-Setup mit einem erweiterbaren Methodenspektrum. Damit wird die Komplexität des notwendigen technischen Setups reduziert und ist nicht auf einzelne Anwendungsfälle beschränkt. Es können flexibel neue Datenverarbeitungsalgorithmen für ein fMRT-BCI eingebunden und vielgestaltige Anwendungsfälle von adaptiven Paradigmen konzipiert werden. Eine Abstraktion der Stimulation und die Kombination mit der Echtzeitauswertung ist bisher einzigartig für neurowissenschaftliche Experimente.

Zusätzlich zu den theoretischen und technischen Erläuterungen werden im empirischen Teil der vorliegenden Arbeit neurowissenschaftliche Experimente, die mit dem Softwarepaket *BART* durchgeführt wurden, vorgestellt und deren Ergebnisse erläutert. Dabei wird die Zuverlässigkeit und Funktionsweise der Implementierung in allen Teilschritten der Datenerfassung und -verarbeitung validiert. Die Ergebnisse verifizieren die Implementierung einer parallelisierten fMRT-Analyse. Weiterhin wird eine erste konkrete Umsetzung für ein adaptives Paradigma vorgestellt, bei dem zur Laufzeit die Blickrichtung der Probanden berücksichtigt wird. Die Ergebnisse zeigen die signifikanten Verbesserungen der Reliabilität der fMRT-Ergebnisse aufgrund der optimierten Datenqualität durch die Adaption des Paradigmas.

Zusammengefasst umfasst die vorliegende Arbeit eine interdisziplinäre Aufgabe, die sich aus der Verarbeitung von MR-Daten in Echtzeit, einem neuen abstraktes Softwarekonzept für Entwicklung neuer methodischer Ansätze und der Durchführung von neurowissenschaftlichen Experimenten zusammensetzt.

Danksagung

Diese Dissertation ist am Max-Planck-Institut für Kognitions- und Neurowissenschaften Leipzig entstanden. Beim Verfassen dieser Arbeit erfuhr ich von vielen Personen in meinem Umfeld auf unterschiedlichste Art und Weise Unterstützung. An dieser Stelle möchte ich mich bei allen dafür bedanken.

Besonderer Dank gilt Herrn Professor Villringer, der die Bearbeitung des Themas ermöglicht hat und die freie Ausgestaltung des Themas sehr unterstützt hat.

Ich bedanke mich bei Professor Martin Bogdan für die unkomplizierte und kompetente Zusammenarbeit, die die Erstellung dieser Arbeit sehr erleichtert hat.

Ein besonderer Dank gebührt Maurice Hollmann für zahllose konstruktive Diskussionen und das entgegengebrachte Vertrauen. Ich bedanke mich sehr bei Oliver Zscheyge für die großartige Zusammenarbeit. Ich danke auch allen weiteren Kollegen, die sich vom Projekt 'Echtzeit-fMRT' haben begeistern lassen, insbesondere Annette Horstmann, Christian Kalberlah, Hadas Okon-Singer und Elisabeth Roggenhofer.

Ein Dank an alle Kollegen am Max-Planck-Institut, die mir bei Fragen immer behilflich waren. Insbesondere danke ich den Kollegen der NMR-Gruppe, Torsten Schlumm, André Pampel, Jöran Lepsien, Toralf Mildner und Harald Möller für die viele Unterstützung bei der technisch aufwendigen Umsetzung der verrückten Idee 'Echtzeit-fMRT'. Außerdem danke ich Enrico Reimer für die sehr effiziente, lehrreiche und unterhaltsame Zusammenarbeit im '*friendly isis team*'.

Ein großer Dank gilt Silke Jahn und Gabriele Bixel, die mir persönlich mit Rat und Tat zur Seite standen und die Lesbarkeit der Arbeit deutlich verbessert haben.

Ein besonderer Dank gilt meinen Eltern, die mich jederzeit vollstens unterstützen und immer zur Seite stehen. Und ich danke ganz besonders Torsten für die viele Unterstützung.

Inhaltsverzeichnis

1	Einleitung	1
2	Grundlagen	5
2.1	Magnetresonanztomographie und funktionelle Magnetresonanztomographie . .	5
2.1.1	Das MR-Phänomen	6
2.1.2	MR-Bildgebung	7
2.1.3	Funktionelle MR-Bildgebung	10
2.2	Durchführung fMRT-Experimente	11
2.2.1	Stimulation, Paradigma und Design bei fMRT-Experimenten	11
2.2.2	Reaktionserfassung bei fMRT-Experimenten	15
2.2.3	Vorverarbeitung von fMRT-Daten	16
2.2.4	Statistische Auswertung von fMRT-Daten	17
2.3	Echtzeit-fMRT als Schnittstelle zwischen Gehirn und Computer	21
2.3.1	Schnittstellen zwischen Gehirn und Computer	21
2.3.2	Echtzeit-fMRT	23
2.3.3	Literaturübersicht zu fMRT-Neurofeedback als Applikation von Echtzeit-fMRT	28
2.3.4	Grenzen von Echtzeit-fMRT und fMRT-Neurofeedback	35
2.4	Adaptive Paradigmen	35
2.4.1	Literaturübersicht zu adaptiven Paradigmen	36
2.4.2	Adaptive Paradigmen für die vorliegende Arbeit	37
2.4.3	Grenzen der Anwendung adaptiver Paradigmen	38
3	Methoden und Ergebnisse: Softwarepaket ‘Brain Analysis in Realtime’ (BART)	40
3.1	Motivation	40
3.1.1	Vorhandene Softwarepakete für Echtzeit-fMRT	41
3.1.2	Grundanforderungen an das neue Softwaresystem BART	44
3.2	Methoden	45
3.2.1	Objektorientierte Programmierung	45
3.2.2	Entwurfsmuster in der Softwareentwicklung	46
3.2.3	Experiment Description Language (EDL)	50

3.2.4	Konkrete Werkzeuge für die Entwicklung	54
3.3	Ergebnisse	59
3.3.1	Softwaredesign BART	59
3.3.2	Technisches Setup	62
3.3.3	Hardware für das Softwarepaket BART	63
3.3.4	Echtzeit-System	63
3.3.5	Integriertes Stimulationssystem: konventionelle Paradigmen	68
3.3.6	Integriertes Stimulationssystem: adaptive Paradigmen	72
3.3.7	Gesamtaufbau: Adaptive Paradigmen und Echtzeit-Auswertung	77
3.4	Zusätzliche Entwicklungen im Rahmen dieser Arbeit	80
3.4.1	EDLEditor	80
3.4.2	Echtzeitexport	81
3.4.3	Bibliothek isis	85
3.5	Von der Softwareentwicklung zur Anwendung in fMRT-Experimenten	86
4	Methoden und Ergebnisse: fMRT-Experimente mit BART	87
4.1	Emotionale Netzwerke in fMRT-Experimenten	87
4.2	Experiment I: Adaptives Paradigma mit BART	90
4.2.1	Pilotexperiment	90
4.2.2	Hypothesen	91
4.2.3	Material und Methoden	91
4.2.4	Ergebnisse	95
4.3	Experiment II: Echtzeitauswertung mit BART	101
4.4	Experiment III: Adaptives Paradigma und Echtzeitauswertung mit BART	103
5	Diskussion	105
5.1	Diskussion Software	105
5.2	Diskussion fMRT-Experimente mit BART	109
5.2.1	Diskussion fMRT-Experiment I mit adaptivem Paradigma	109
5.2.2	Diskussion fMRT-Experimente II und III	111
5.3	Diskussion Adaptive Paradigmen für fMRT	112
6	Zusammenfassung und Ausblick	113
6.1	Zusammenfassung	113
6.2	Ausblick	114
A	Anhang	117
A.1	Anhänge zur Software BART	117
A.1.1	Klassendiagramm Datenmanagement	117
A.1.2	Klassendiagramm Konfiguration	118

Inhaltsverzeichnis	vi
A.2 Anhänge zu Spezifikationserweiterungen in EDL	119
A.2.1 XSD Spezifikation Bedingungen (Constraints)	119
A.2.2 XSD Spezifikation dynamische Regressoren	121
A.2.3 XSD Spezifikation Transferfunktion	122
A.3 Anhänge zum Experiment adaptives Paradigma	123
A.3.1 SAM Ratings	123
A.3.2 Stimulusbilder Übersicht	123
Literaturverzeichnis	133
Abbildungsverzeichnis	135
Tabellenverzeichnis	136
Auflistungsverzeichnis	137
Glossar	138
Abkürzungsverzeichnis	142
Bibliographische Angaben	144
Selbständigkeitserklärung	145

1 Einleitung

Die rasanten technischen Entwicklungen in den letzten Jahrzehnten haben schnell Einzug in die Medizin und die kognitiven Neurowissenschaften gehalten. Mit diesem Fortschritt wurden Möglichkeiten geschaffen, Computer direkt mit dem menschlichen Gehirn zu verbinden, was allgemein als Hirn-Computer-Schnittstelle, im Englischen als *Brain Computer Interface* (BCI), bezeichnet wird. Dabei werden im Gehirn gemessene Signale mit Hilfe eines Übersetzungsalgorithmus in einen Steuerbefehl für einen Computer gewandelt. Die Anfänge dieses Forschungsfeldes gehen auf die Entwicklung der Elektroenzephalographie (EEG) zurück, die zahlreiche vielversprechende Anwendungsmöglichkeiten, insbesondere in der medizinischen Therapie, anbietet. Dazu gehören unter anderem Möglichkeiten zur Kommunikation (Birbaumer et al. 1999) und Interaktion (Birbaumer et al. 2007) mit querschnittsgelähmten Patienten. Weiterhin wurden Therapiemöglichkeiten für Kinder entwickelt, die unter einem Aufmerksamkeitsdefizitsyndrom leiden (Monastra et al. 2005), oder Epilepsie-Patienten, die mittels EEG-basierten Therapiesystemen ihre Anfallshäufigkeit reduzieren konnten (Rockstroh et al. 1993). Diese Erfolge sind vielversprechend, jedoch sind viele Aspekte in der Erforschung weiterer Krankheiten, die Reproduzierbarkeit von Ergebnissen und der routinemäßige Einsatz noch ungelöst.

Eine der modernsten nicht-invasiven Techniken für Hirn-Computer-Schnittstellen ist die Funktionelle Magnetresonanztomographie (fMRT) in der speziellen Form der Echtzeit-fMRT¹. Mit der fMRT-Bildgebung können Signalveränderungen in Hirnarealen detektiert werden, die bei der Durchführung einer bestimmten Aufgabe (Paradigma) involviert sind. Die Methode bietet eine sehr hohe räumliche Auflösung, wodurch Hirnregionen, die an einem Prozess im Gehirn beteiligt sind, sehr gut lokalisiert werden können. In der Umkehrung zu diesem explorativen Ansatz können mit Echtzeit-fMRT ermittelte Gehirnaktivitäten als Neurofeedbacksignal für spezifische Hirnareale genutzt werden. FMRT-Neurofeedback besitzt ein großes Potential für die neurowissenschaftliche Forschung und neue therapeutische Ansätze. Allerdings gibt es bisher kaum standardisierte Methoden und Verfahren für die Anwendung und viele parallele Entwicklungen zahlreicher neuer, unabhängiger Methoden erschweren die Vergleichbarkeit und Bewertung von Ergebnissen (Weiskopf 2012). Die Forschung mit Echtzeit-fMRT-Experimenten steht noch am Anfang der Entwicklung. Für ein besseres Verständnis der Funktionsweise des Gehirns sind weitere Entwicklungen mit neuen algorithmischen und methodischen Ansätzen notwendig, um die

¹ Echtzeit bedeutet in diesem Zusammenhang, dass die Datenauswertung schneller erfolgt als die Messung des folgenden Hirnvolumens (aktuell ca. 2 s)

Experimente zu optimieren. In diesem Kontext ist die vorliegende Arbeit entstanden, in der ein flexibles und wiederverwendbares Framework für Echtzeit-fMRT konzipiert wird.

Ein weiterer Aspekt bei fMRT-Experimenten ist, dass die Antwortsignale im Gehirn nicht allein von der Aufgabe abhängen. Sie sind ebenfalls abhängig von der Kooperation des Probanden und allgemeinen physiologischen Fluktuationen. Für die Kooperation kann z. B. die Blickrichtung des Probanden bewertet werden, die einen erheblichen Einfluss auf die Ergebnisse bei visuellen Aufgaben hat (Pessoa et al. 2002a; Reynolds et al. 2004). Außerdem beeinflussen physiologischen Fluktuationen, wie z. B. der Puls oder die Atmung, die Antwortsignale im Gehirn so, dass eine applizierte Stimulation unterschiedlich wahrgenommen und verarbeitet wird (Fox et al. 2007; Krüger et al. 2001; Leopold et al. 2012; Linkenkaer-Hansen et al. 2004). Obwohl dies bekannt ist, werden fMRT-Paradigmen meist mit einem festen Ablaufschema zur Präsentation von Stimuli durchgeführt und die genannten Einflussfaktoren werden nicht berücksichtigt. Die Reproduzierbarkeit der Experimente könnte deutlich verbessert werden, wenn physiologische Parameter oder Verhaltensmuster berücksichtigt würden. Dazu könnten die Parameter im Anschluss an ein Experiment bei der Auswertung (post-hoc) berücksichtigt werden, um auf ihre Gültigkeit getestet und eventuell von der Analyse ausgeschlossen zu werden. Dies beinhaltet jedoch, dass die 'ungültigen' Daten zuerst erzeugt werden, was in diesem Kontext nicht nur Zeit und Geld verschwendet. Es ist vor allem problematisch wenn die Experimente mit Patienten durchgeführt werden oder die Auswertemethodik per Definition eine geringere statistische Teststärke hat, wie z. B. bei Echtzeit-fMRT. Außerdem können die vorhandenen algorithmischen Korrekturverfahren für physiologische Fluktuationen durch Interpolationen den Signalrauschabstand in den Daten verschlechtern.

Aus diesen Gründen wird in dieser Arbeit ein alternativer Ansatz verwendet, der die Parameter bereits zur Laufzeit mit einbezieht und das Paradigma bei Bedarf anpasst. Mit einem solchen a-priori Ansatz ist es möglich, dass Stimuli nur präsentiert werden, wenn objektive Kriterien erfüllt sind und damit die Daten immer gültig sind. Es wurden bisher nur wenige Ansätze zu dieser Thematik vorgestellt (Wilms et al. 2010; Yoo et al. 2012; Becker et al. 2011). Allerdings waren diese jeweils auf einzelne Anwendungsfälle, d. h. einen konkreten externen Parameter, ausgelegt und somit auf eine kleine Gruppe von Paradigmen eingeschränkt. Es gibt bisher kein universelles Softwareframework, das es ermöglicht, unterschiedliche Parameter zu integrieren und damit für eine Vielzahl an Paradigmen einheitlich und wiederverwendbar zu sein. Außerdem wurde bisher keine Kombination eines adaptiven Stimulationssystems mit einer Echtzeitauswertung von fMRT-Daten vorgestellt.

Zielsetzung der vorliegenden Arbeit: In der vorliegenden Arbeit werden Softwarekonzepte sowohl für die Echtzeit-fMRT-Datenverarbeitung als auch für die Integration externer Parameter zur Stimulation eines Paradigmas erarbeitet. Dazu wird ein abstraktes Konzept zur Beschreibung von möglichen Adaptionen und ein integriertes Framework zur universellen Gestaltung

von adaptiven Paradigmen konzipiert. Zudem wird ein Konzept für eine flexible Auswahl von Datenverarbeitungsalgorithmen vorgestellt. Die Zielstellung der vorliegenden Arbeit ist in Abbildung 1.1 illustriert. Es wird ein neues Softwareframework vorgestellt, welches beide Ansätze kombiniert, in einem BCI-Kontext arbeitet und flexible Schnittstellen für die Einbindung von Algorithmen zur Datenverarbeitung und externen Parametern zur Verfügung stellt. Die abstrakten Bausteine werden anschließend mit den ersten dafür entwickelten Algorithmen konkretisiert. Im empirischen Teil der Arbeit werden die Funktionalitäten des neu entwickelten Softwarepakets validiert. Die Ergebnisse belegen einerseits die Korrektheit der Algorithmen als auch die Vorteile, die sich durch die Adaption von fMRT-Paradigmen ergeben.

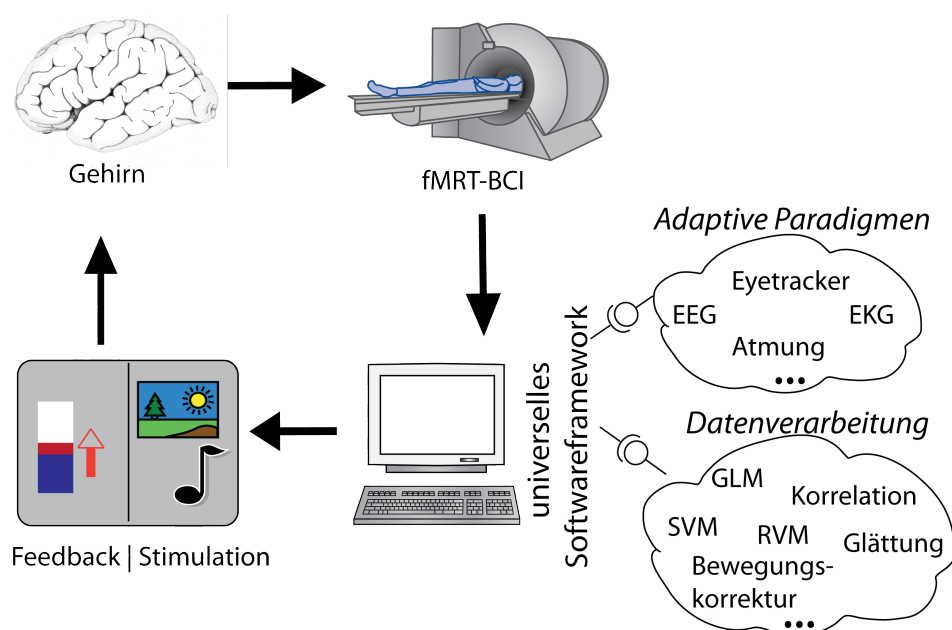


Abbildung 1.1: Die Zielsetzung der vorliegenden Arbeit ist die Entwicklung von neuen Konzepten für Schnittstellen zur flexiblen Datenverarbeitung und universellen Einbindung externer Parameter für adaptive Paradigmen in Echtzeit-fMRT-Experimenten. Die erstellten Konzepte werden in ein neues Softwareframework implementiert. Die Funktionalität dieser neu konzipierten Software wird im empirischen Teil der Arbeit mit fMRT-Experimenten validiert.

Aufbau der vorliegenden Arbeit: Diese Arbeit beschreibt eine interdisziplinäre Aufgabenstellung, die sich zusammensetzt aus Bereichen der MR-Physik, dem Softwaredesign, der Softwareentwicklung, der Durchführung neurowissenschaftlicher Experimente mit MR-Bildgebung und der Neuentwicklung experimenteller Methoden. Kapitel 2 und 3 führen jeweils in die Themengebiete ein, um die zum Verständnis der vorliegenden Arbeit notwendigen Aspekte zu erläutern. Kapitel 2 enthält eine Einführung in die Bildgebung mit der Magnetresonanztomographie und erläutert ausführlich die Durchführung konventioneller fMRT-Experimente. Danach werden Methodik und Besonderheiten für Echtzeit-fMRT erläutert und der Stand der Technik anhand der publizierten Literatur detailliert erklärt. Weiterhin wird in diesem Kapitel die Idee der adaptiven

Paradigmen für fMRT vorgestellt und anhand der aktuellen Literatur beschrieben. Im Kapitel 3 werden zuerst die notwendigen Grundprinzipien für das Softwaredesign vorgestellt und die Konzepte für die beiden Hauptkomponenten, das adaptive Stimulationsmodul und die Echtzeitdatenverarbeitung, erläutert. Die Konzepte wurden in ein neues Softwarepaket implementiert, das in diesem Kapitel detailliert erläutert wird. Die Ergebnisse präsentieren das technische Setup für ein BCI, das konkret realisiert wurde. Außerdem wird die Funktionsweise der Adaption erläutert und die Ergebnisse in Bezug auf Stabilität und Laufzeit präsentiert. Weiterhin werden in diesem Kapitel die Umsetzung und Ergebnisse der parallelisierten Algorithmen zur Datenverarbeitung vorgestellt. Abschließend dazu werden in diesem Kapitel zusätzliche Werkzeuge vorgestellt, die im Zusammenhang dieser Arbeit entstanden sind und notwendig sind für die praxistaugliche Funktionsweise des Gesamtsystems. Der empirische Teil der Arbeit ist in Kapitel 4 beschrieben. Dieses Kapitel beinhaltet die Durchführung von drei fMRT-Experimenten, die mit dem neu entwickelten Softwarepaket durchgeführt wurden. Die Experimente validieren die Funktionalität des neu entwickelten Systems. Im zentralen Experiment wurde ein Adaptives Paradigma angewendet und die Ergebnisse zeigen die Vorteile für die Zuverlässigkeit und Reproduzierbarkeit für die Ergebnisse von fMRT-Experimenten. Abschließend werden im Kapitel 5 alle Ergebnisse diskutiert und die zahlreichen Entwicklungs- und Anwendungsmöglichkeiten dieses Softwarepaketes aufgezeigt.

2 Grundlagen

Für das Verständnis dieser Arbeit sind Kenntnisse zur Durchführung von Experimenten mit funktioneller Magnetresonanztomographie (fMRT) notwendig. Daher wird in diesem Kapitel die Technik der Magnetresonanztomographie (MRT) und der fMRT vorgestellt. Die Technologie ist sehr umfangreich und wird aus diesem Grund hier nur zusammenfassend im Anwendungskontext erläutert. Weitergehende Literatur mit Informationen und Erläuterungen zu den physikalischen Details und Erweiterungen der Technik finden sich in Oppelt (2005) und Haacke et al. (1999). Nach einer Einführung der MR-Bildgebung in Abschnitt 2.1 wird in den Abschnitten 2.2 und 2.3 erläutert, wie fMRT-Experimente aufgebaut und durchgeführt werden. Eine weitergehende Einführung in die Durchführung von fMRT-Experimenten im Kontext von Psychiatrie und Neurologie findet sich in Schneider et al. (2007). Aufbauend auf den Erläuterungen zu konventionellen fMRT-Experimenten wird die Erweiterung zur Echtzeit-fMRT beschrieben. Anschließend wird ein ausführlicher Überblick über den aktuellen Stand der Technik und die bisherigen Anwendungen von Echtzeit-fMRT gegeben. Im letzten Abschnitt des Kapitels 2.4 wird die grundlegende Idee von adaptiven Paradigmen erläutert und die experimentelle Integration dieser Paradigmen zur Durchführung von Echtzeit-fMRT-Experimenten beschrieben.

2.1 Magnetresonanztomographie und funktionelle Magnetresonanztomographie

Magnetresonanztomographie (MRT) ist ein nicht-invasives bildgebendes Verfahren zur Darstellung von Strukturen im Inneren des Körpers. Das Verfahren wird seit ca. 40 Jahren fortlaufend weiterentwickelt. Mit Hilfe der fMRT lassen sich physiologische Funktionen im menschlichen Körper, insbesondere im Gehirn, erfassen. Sie beruht auf einer besonderen paramagnetischen Eigenschaft des Blutflusses, die erstmalig 1990 von Ogawa et al. (1990) am lebenden Gehirn erfasst werden konnte. Die MR-Technik basiert auf dem MR-Phänomen mit Hilfe dessen die MR-Bildgebung erfolgen kann.

2.1.1 Das MR-Phänomen

Magnetisierung: Alle Atomkerne mit ungerader Anzahl von Protonen (oder Neutronen) besitzen einen intrinsischen Eigendrehimpuls (Spin), der ein magnetisches Dipolmoment erzeugt. Da alle Protonen eine positive Ladung aufweisen, wird durch die Eigenrotation ein geringes Magnetfeld erzeugt. Im natürlichen Zustand sind die Drehachsen dieser Dipole zufällig ausgerichtet und gleichen so gegenseitig ihre jeweiligen magnetischen Momente aus. Werden diese allerdings in ein hohes statisches Magnetfeld (B_0) gebracht, so richten sich die meisten der Dipole, entsprechend der Boltzmann-Verteilung, parallel zu diesem äußeren Magnetfeld aus. Alle Kernspins drehen sich dann in einer Kreisbewegung mit der sogenannten Larmorfrequenz um die Achse von B_0 . Die Larmorfrequenz ergibt sich aus der kernspezifischen Konstante γ und der Feldstärke von B_0 . Durch die Kreisbewegung ergibt sich eine Längsmagnetisierung M_0 , welche jedoch nicht direkt für eine Messung zugänglich ist. Zudem ist die Bewegung unkoordiniert, das heißt, die Phasen der einzelnen Spins sind inkohärent (siehe Abbildung 2.1a,b).

Resonanz und HF-Impuls: Zur Erzeugung eines messbaren Signals wird der Gleichgewichtszustand dieser Kreisbewegung durch elektromagnetische Strahlung im Radiofrequenzbereich (*High-Frequency*, HF-Impuls) gestört. Entspricht dieser Impuls der Larmorfrequenz der Spins, dann nehmen die so angeregten Spins Energie auf und richten sich in einem antiparallelen Zustand zu B_0 aus (siehe Abbildung 2.1 c,d). Diese gerichtete Auslenkung führt zur Kohärenz der Phasen, was auch als Magnetresonanz bezeichnet wird. Nach dem Abschalten des Impulses geben die Spins die aufgenommenen Energie wieder ab und kehren in den Ausgangszustand zurück. Diese Energieabgabe kann in einer Empfängerspule detektiert werden.

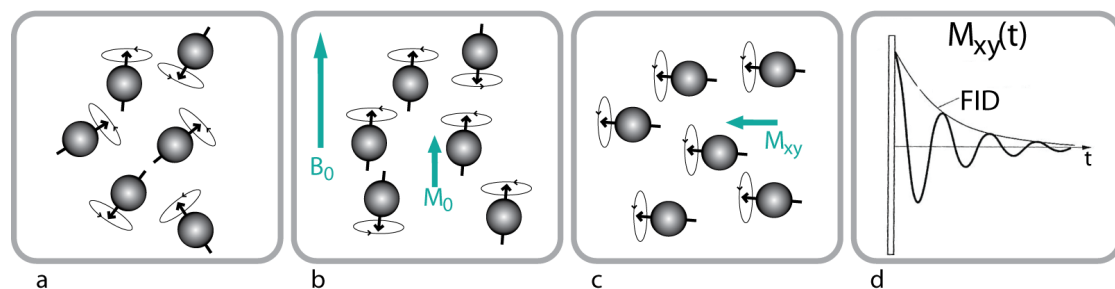


Abbildung 2.1: Das MR-Phänomen: Magnetresonanz basiert darauf, dass Protonen, die einen Eigendrehimpuls besitzen und im Normalzustand zufällig ausgerichtet sind (a), in ein statisches Magnetfeld B_0 gebracht werden (b). Dabei entsteht eine Längsmagnetisierung M_0 . Durch einen HF-Impuls werden alle Protonen gerichtet ausgelenkt wodurch die Magnetisierung M_{xy} entsteht (c). Nach der Anregung kehren die Protonen mit einem gewebespezifischen Verlauf in den Gleichgewichtszustand zurück. Das oszillierende Signal dieser Relaxation (Freier Induktionsverfall, FID) zerfällt im Idealfall mit der Zeitkonstanten T_2 , ist bei Inhomogenitäten allerdings von T_2^* überlagert (d²). Dieses Signal wird in einer Empfängerspule detektiert.

²Grafik d adaptiert aus Oppelt (2005, S. 154).

Relaxation (T_1 , T_2 und T_2^*): Die Rückkehr der Spins in die ursprüngliche parallele Ausrichtung wird als Relaxation bezeichnet und ist abhängig von der Protonenbeweglichkeit im Gewebe. Damit hat die Relaxation einen charakteristischen Verlauf je nach Gewebetyp und den detektierten Atomkernen³. Es ergeben sich verschiedene Signalintensitäten für unterschiedliche Gewebetypen im Körper, die von der Anzahl der Wasserstoffprotonen und ihrer molekularen Bindungen abhängen. Dies führt zu drei wesentlichen Parametern für die Beschreibung der Relaxation:

- T_1 definiert die Zeitkonstante mit der die longitudinale Relaxation charakterisiert ist. Diese beschreibt die Energieabgabe der Spins an das umgebende Molekülgitter und definiert die Zeit bis erneut angeregt werden kann;
- T_2 definiert die Zeitkonstante mit der die transversale Relaxation charakterisiert ist. Diese beschreibt die Energieabgabe der Spins untereinander durch Dephasierung der Spins;
- T_2^* : definiert die Zeitkonstante für den exponentiellen Signalabfall der durch die schnellere Dephasierung bei Änderungen der magnetischen Eigenschaften entsteht und ist immer kleiner als T_2 . Dieser Effekt entsteht aufgrund von Magnetfeldinhomogenitäten, die sowohl durch den technischen Aufbau großer Magnetfelder als auch durch lokale Magnetfeldänderungen der Probe (hier der Person im Scanner) entstehen. Dies überlagert T_2 . Der genaue Zusammenhang wird beschrieben durch $1/T_2^* = 1/T_2 + \gamma * \delta B_0$ (Möller 2006), wobei δB_0 die Feldinhomogenität bezeichnet.

MR-Bilder mit unterschiedlicher Gewichtung bezüglich dieser Parameter eröffnen die Möglichkeit, sehr unterschiedliche Gewebekontraste darzustellen. Wie in Abbildung 2.2 beispielhaft gezeigt, liefern T_1 -gewichtete Bilder ein starkes Signal für fetthaltige Gewebetypen, während T_2 -gewichtete Bilder Flüssigkeiten hervorheben. Zur Darstellung von Änderungen der magnetischen Eigenschaften im Gewebe (z. B. an Gewebegrenzen) eignen sich besonders T_2^* -gewichtete Bilder. Dies ist die Grundlage für die MR-Bildgebung der Hirnaktivität mit Hilfe des in Abschnitt 2.1.3 erläuterten BOLD-Effektes.

2.1.2 MR-Bildgebung

Mit dem beschriebenen Phänomen der Magnetresonanz wird ein messbares Resonanzsignal erzeugt. Damit daraus ein Bild berechnet werden kann, muss das gemessene Signal einem räumlichen Ursprung zugeordnet werden können. Als tomographisches Verfahren wird das zu messende Objekt hierbei in Schnittbildern erfasst. Dazu wird das Gewebe schichtweise angeregt und innerhalb dieser der Signalursprung bestimmt. Die Lokalisierung basiert auf dem Prinzip, dass die Larmorfrequenz proportional zur Magnetfeldstärke ist und eine Anregung von Protonen durch den HF-Impuls nur erfolgt, wenn die Frequenz der Larmorfrequenz entspricht. Wenn die Magnetfeldstärke ortsabhängig ist, kann das gemessene Signal, wie im Folgenden beschrieben, präzise einem Ursprung zugeordnet werden.

³ Bei MRT-Aufnahmen im medizinischen Kontext werden meistens Wasserstoffatome angeregt.



Abbildung 2.2: MR-Aufnahmen: Mit Hilfe der MR-Technik können Aufnahmen der Strukturen im menschlichen Gehirn gemacht werden. Mit unterschiedlichen Gewichtungen der MR-Aufnahmen können unterschiedliche Gewebetypen verschieden kontrastiert dargestellt werden (T_1 : starkes Signal für fetthaltiges Gewebe; T_2 : starkes Signal für Flüssigkeiten). Indirekt kann neuronale Aktivität mit dem T_2^* -Kontrast zur Erfassung magnetischer Änderungen im Gewebe erfasst werden.

Schichtanregung und Ortskodierung: Damit eine einzelne Schicht des Körpers angeregt werden kann, wird das homogene Magnetfeld B_0 durch eine zusätzliche Magnetspule mit einem sogenannten Gradienten überlagert. Das heißt, das Magnetfeld besitzt einen Anstieg in eine Richtung (z-Richtung), die eine kontinuierliche Änderung der Larmorfrequenz entlang der Richtung des Gradienten bewirkt. Damit kann nun mit einer bestimmten Anregungsfrequenz exakt eine Schicht innerhalb des Körpers angeregt werden. Dabei ist zu beachten, dass für die Verringerung der Schichtdicke die Gradientenstärke zunehmen muss. Zusätzlich zu dieser Schichtselektion wird das Magnetfeld mit zwei weiteren Gradienten in x- und y-Richtung überlagert. Damit kann das MR-Signals in die räumlichen Koordinaten aufgeschlüsselt werden. Diese werden mit Hilfe der Phasen- und Frequenzkodierung ermittelt, wobei die Details dazu hier nicht erläutert werden. Mit Hilfe der Frequenz und Phase kann jedes Volumenelement (Voxel) eindeutig charakterisiert und das gemessene MR-Signal räumlich zugeordnet werden.

Voxel: Eine dreidimensionale Volumeneinheit, die einem Pixel in einer zweidimensionalen Grafik entspricht. Die Voxel eines MR-Bildes enthalten den Wert des MR-Signals an der angeregten Position. Die räumliche Auflösung eines Bildes wird durch die Größe seiner Volumenelemente bestimmt (Weishaupt et al. 2006, S. 26).

Repetitionszeit (TR): Die Zeit zwischen den ersten HF-Impulsen aufeinanderfolgender Pulssequenzen zur Anregung der Protonen wird als Messwiederholzeit (*Repetition Time*, TR) bezeichnet.

MR-Pulssequenz: Die Abfolge und Ausrichtung von HF-Impulsen und Gradientenfeldern wird als Pulssequenz bezeichnet, d. h. eine Pulssequenz ist eine Kombination aus HF-Impulsen und Gradientenfeldern in verschiedenen Stärken, die mehrfach in jeder Sekunde in definierter Rei-

henfolge geschaltet werden. Durch verschiedene Kombinationen lassen sich nun die jeweils interessierenden Gewebekontraste abbilden und die Messzeit dafür minimieren. Für das Durchführen von MR-Experimenten, wie sie in dieser Arbeit beschrieben werden, sind zwei Sequenzen wichtig: T_1 -gewichtete Strukturmessungen vom menschlichen Gehirn und T_2^* -gewichtete funktionelle Messungen mit einer EPI-Sequenz:

Echo-Planar-Imaging (EPI): Bei der EPI-Sequenz werden alle Phasen- und Frequenzkodierschritte nach einer einzigen Pulsanregung ausgelesen, wie in Abbildung 2.3 dargestellt wird. Damit kann eine angeregte Schicht in weniger als 100 ms aufgenommen werden. Eine schnelle Akquisitionszeit ist notwendig für die funktionelle Bildgebung, um physiologische Fluktuationen im Körper erfassen zu können. Die Grenze des Verfahrens liegt in der begrenzten räumlichen Auflösung, die mit dieser Sequenz erreicht werden kann. In den meisten Messungen wird eine Matrix von 64 x 64 Voxeln aufgenommen. Zudem führt das schnelle Schalten der Gradienten zu einer deutlichen Lautstärkebelastung während der Messung. Mit einer EPI-Sequenz ist es möglich, das gesamte Volumens des menschlichen Gehirns - aufgeteilt in 30 Schichten mit jeweils einer Dicke von 3 mm und einer Matrix von 64 x 64 Voxeln - mit einer Repetitionszeit von 2 s zu messen.

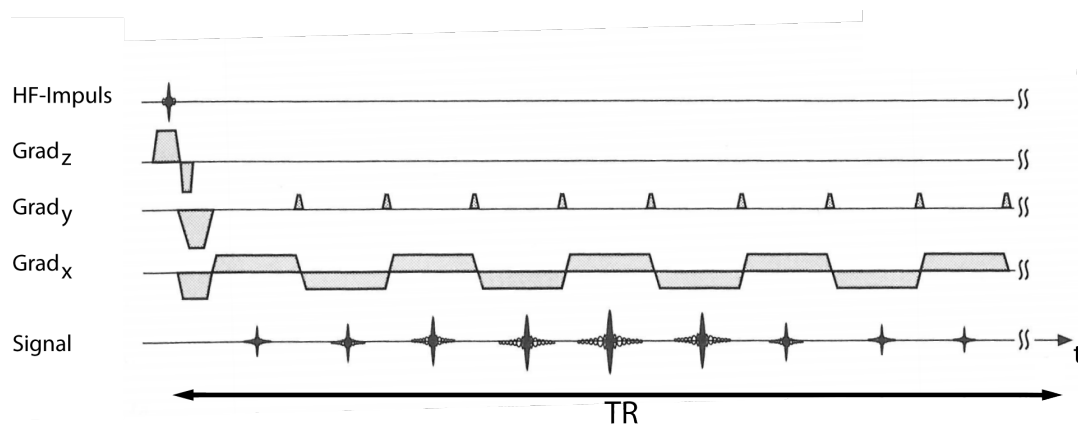


Abbildung 2.3: EPI-Sequenz: Für die Bildgebung mittels fMRT wird die schnelle Bildgebung mit dieser Sequenz verwendet. Hierbei wird die zu messende Schicht mit nur einem Puls angeregt und durch sehr schnell geschaltete Gradienten alle Phasen- und Frequenzkodierschritte ausgeführt. Mit dieser Sequenz können z. B. 30 Schichten im gesamten Hirnvolumens mit 3 mm Schichtdicke in 2 s angeregt werden.⁴

Strukturelle 3D-Bildgebung: Zusätzlich zu den relativ gering aufgelösten Daten, die mit der EPI-Sequenz aufgenommen werden, wird bei fMRT-Experimenten meist auch eine hoch aufgelöste, anatomische Strukturaufnahme gemessen. Damit können die späteren Ergebnisse der EPI-Daten gut lokalisiert und Artefakte der EPI-Sequenz teilweise korrigiert werden. Wichtig für strukturelle MR-Sequenzen sind einerseits eine hohe räumliche Auflösung bei akzeptabler Mess-

⁴Grafik adaptiert aus Oppelt (2005, S. 611)

zeit und ein guter Kontrast zwischen grauer und weißer Hirnsubstanz durch eine T_1 -Gewichtung (Schneider et al. 2007, S. 70). Dazu wird häufig eine sogenannte MP-RAGE-Sequenz (*Magnetization Prepared Rapid Acquisition Gradient Echo*) verwendet. Diese ermöglicht es, eine T_1 -gewichtete Aufnahmen des gesamten Gehirns mit einer Auflösung von ca. 1 mm^3 in weniger als 10 min aufzunehmen.

2.1.3 Funktionelle MR-Bildgebung

Als fMRT wird im allgemeinen die Bildgebung physiologischer Fluktuationen im Körper bezeichnet. Der Begriff beschreibt die indirekte Erfassung von neuronaler Aktivität, die mit Änderungen im Blutfluss in aktiven Hirnarealen einhergeht und durch den BOLD-Effekt beschrieben wird.

Blood Oxygenation Level Dependent (BOLD)-Effekt: Der BOLD-Effekt wurde erstmalig von Ogawa et al. (1990) beschrieben. Der Effekt basiert auf den unterschiedlichen magnetischen Eigenschaften von oxygeniertem (sauerstoffreichem) und desoxygeniertem (sauerstoffarmen) Blut. Eine lokale Aktivität eines Hirnareals bewirkt eine Überkompensation des Sauerstoffverbrauchs mit einem Nachschub an oxygeniertem Blut. Aufgrund der diamagnetischen Eigenschaft von desoxygeniertem Hämoglobin ist das MR-Signal dann zu dieser Zeit weniger gestört. Somit entsteht je nach Hirnregion ein Signalanstieg von bis zu 5 % bei einer Magnetfeldstärke von 3 T. Dieser Effekt ist eine indirekte Messung neuronaler Aktivität und wird von verschiedenen Mechanismen beeinflusst (Logothetis 2008). Die grundlegende Physiologie dieses Effektes ist noch nicht vollständig verstanden und ist immer noch Bestandteil der Forschung. Dennoch ist der zeitliche Verlauf sehr gut beschrieben. Das BOLD-Antwortsignal wird als hämodynamische Antwortfunktion (*Hemodynamic Response Function*, HRF) bezeichnet. Die Überkompensation mit oxygeniertem Blut erreicht erst ca. 4 s bis 6 s nach der eigentlichen neuronalen Aktivität ein Maximum. Bevor das Signal in den Gleichgewichtszustand zurückkehrt, kann ein Unterschreiten der Ausgangsaktivität (*Undershoot*) beobachtet werden, das bis zu 30 s nach der Aktivität andauern kann. Die Abbildung 2.4 veranschaulicht den Effekt und zeigt den charakteristischen Verlauf der HRF.

⁵Grafiken adaptiert aus Schneider et al. (2007, S. 77)

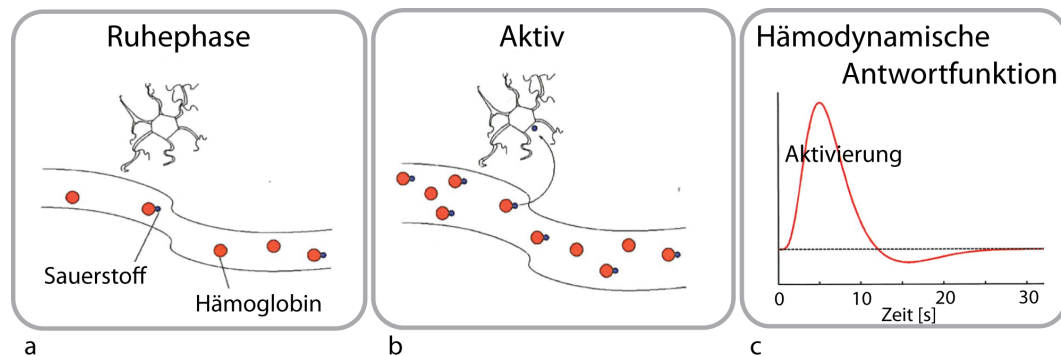


Abbildung 2.4: **BOLD-Effekt und HRF**: Wird das Gehirn durch einen äußeren Reiz angeregt, werden die bei der Verarbeitung des Reizes relevanten Hirnareale von der Ruhephase (a) in einen aktiven Zustand (b) versetzt. Die erhöhte neuronale Aktivität führt zu einem erhöhten Sauerstoffbedarf und zu einer höheren lokalen Durchblutung. Dadurch nimmt der Anteil an sauerstoffhaltigem (oxygeniertem) Hämoglobin in diesem Gehirnareal zu. Der verringerte Anteil an desoxygeniertem Hämoglobin führt zu einem Anstieg im messbaren MR-Signal. Der zeitliche Verlauf der Signalveränderungen wird als hämodynamische Antwortfunktion (HRF) bezeichnet (c).⁵

2.2 Durchführung von fMRT-Experimenten

2.2.1 Stimulation, Paradigma und Design bei fMRT-Experimenten

Mit fMRT-Experimenten können unter anderem Fragestellungen der neurologischen, psychiatrischen und kognitionswissenschaftlichen Forschung beantwortet werden. Dazu werden Hypothesen über die Beteiligung und Interaktion von Hirnarealen bei einem zu untersuchenden Prozess aufgestellt. Wird dieser mit einem Paradigma abgebildet, kann mit Hilfe der fMRT-Bildgebung die ausgelöste Hirnaktivität gemessen werden. Der folgende Abschnitt erläutert die dazu notwendigen Schritte und führt notwendige Begriffe und Definitionen ein. Eine Zusammenfassung über den allgemeinen Ablauf eines fMRT-Experiments ist in Abbildung 2.8 gezeigt.

Paradigma: Der Begriff bezeichnet hier die Abbildung des zu untersuchenden Prozesses in eine Aufgabe, die alle Probanden einer Studie während der MR-Messung durchführen müssen. An einem Beispiel erklärt heißt das: Zur Beantwortung der Frage, welche Hirnareale bei der emotionalen Verarbeitung von Gesichtsausdrücken aktiv sind, wird ein Paradigma mit Bildern von Menschen mit traurigen und neutralen Gesichtsausdrücken verwendet. Verschiedene Bilder, die jeweils diesen beiden Kategorien zugeordnet sind, werden den Probanden präsentiert. Dies wird auch als visuelle Stimulation bezeichnet. Die verschiedenen Bedingungen im Paradigma sollten sich dabei möglichst nur in dem zu untersuchenden Aspekt unterscheiden. Im folgenden Beispiel unterscheiden sich die Bilder nur im emotionalen Ausdruck. In Bezug auf Kontrast- und Helligkeitsmerkmalen oder dem Geschlecht der dargestellten Personen sind sie dagegen gleich. Das Paradigma beinhaltet die wiederholte Präsentation von Stimuli aus den unterschiedlichen

Kategorien (Bedingungen). Das fMRT-Signal, welches durch die sich wiederholende Präsentation der Stimuli ausgelöst wird, wird anschließend statistisch ausgewertet. Die Ergebnisse geben Aufschluss darüber, in welchen Hirnarealen ein verändertes Signal, d. h. neuronale Aktivität, in Abhängigkeit des präsentierten Stimulus zu beobachten ist.

Stimulation: Die Stimulation in einem Paradigma beschreibt die Applikation von Reizen (visuell, auditiv, taktil) bei den Probanden. Diese Reize sind in fMRT-Experimenten am häufigsten visuelle Stimuli (Bild, Video oder Text), da sich diese mit Hilfe eines Videoprojektors und einer Leinwand sehr einfach im Umfeld eines MR-Scanners aufbauen lassen. Probanden können die Stimuli dann über einen Spiegel, der an der Kopfspule angebracht ist, sehen. Eine akustische Stimulation (Musik oder Sprache) ist mit Hilfe von MR-geeigneten Kopfhörern ebenfalls möglich, allerdings wird der Ton durch die Geräusche des Scanners während der Messung überlagert. Weiterhin sind taktile Stimulationen durch Druck, Temperatur oder Elektrizität möglich. Mit speziell MR-geeigneten Aufbauten sind auch olfaktorische und gustatorische Reize applizierbar.

Design: Für die statistische Auswertung der fMRT-Zeitreihen werden die gemessenen Hirnsignale mit einem prädefinierten Signal verglichen, das sich aus den Zeitpunkten und Dauer der Stimulation ergibt. Diese Modellannahme für die erwarteten Signalantworten wird im Zusammenhang mit fMRT-Experimenten als Design bezeichnet. Ein Design ist somit die Abbildung eines Paradigmas in einem Modell für die Auswertung der gemessenen Zeitreihen. Im einfachsten Fall entsprechen die Zeitpunkte und die Dauer der Stimuli im Paradigma genau denen des Designs. Im Design werden zusätzlich die unterschiedlichen Kategorien (Bedingungen) abgebildet, denen die Stimuli zugeordnet werden.

Die Begriffe Design und Paradigma werden im Zusammenhang mit fMRT-Experimenten oft als Synonyme verwendet. Für die vorliegende Arbeit beschreibt der Begriff 'Paradigma' die Stimulation und 'Design' die Modellannahme der Stimulation für die Auswertung der Daten. In Anlehnung an die verwendete Literatur erläutert daher der folgende Abschnitt die Modellierung des erwarteten Signalverlaufs mit dem Begriff des 'Designs'. Je nach Dauer und Reihenfolge der Reizapplikationen werden zwei Arten von Designs unterschieden: Blockdesign und Event-related Design (nach Schneider et al. (2007, S. 123ff)):

Blockdesign: Werden die Stimuli in einer festen Reihenfolge von Ruhe- und Stimulationsperioden (Blöcken) mit mehreren Messwiederholungen präsentiert, wird dies in einem Blockdesign abgebildet. Die Blöcke sind deutlich länger als die Reaktionszeit der HRF (4 s bis 6 s). Ausgehend von der theoretischen Grundannahme, dass sich über den Zeitraum des Blocks die neuronale Aktivität linear aufsummiert, entsteht so ein stabiler kognitiver Zustand. Daraus ergeben sich auch die Vorteile des Blockdesigns: eine hohe statistische Teststärke (*Power*) und einfache Analysemöglichkeiten (z.B. Subtraktion oder Korrelation). Nachteilig wirkt sich vor allem die

Beeinflussung des Paradigmas auf die Erwartungshaltung der Teilnehmer und die Gewöhnung an den applizierten Reiz (Habituation) aus. Um diese Nachteile des Blockdesigns aufzulösen und um ein Paradigma flexibler gestalten zu können, werden ereignisbasierte (Event-related) Designs verwendet.

Event-related Design: Werden die Stimuli nur kurz präsentiert wird dies als Event-related Design bezeichnet. Damit ist die zeitliche Auflösung eines fMRT-Experiments deutlich gesteigert gegenüber einem Blockdesign, allerdings ist die erwartete Aktivierung im Gehirn deutlich geringer, d. h. die Sensitivität eines Event-related Design ist insgesamt geringer. Um diesen Effekt zu minimieren werden unterschiedliche Zeitabstände zwischen den Stimuli eingesetzt, die innerhalb eines *Inter-Trial-Intervalls (ITI)* liegen (z. B. im Intervall zwischen 4 und 10 Sekunden). Sind die gewählten Zeitabstände dabei keine Vielfachen der TR-Zeit, ergeben sich dadurch unterschiedliche Abstände zwischen der Stimuluspräsentation und dem Start der aktuellen Volumenmessung des Gehirns (Amaro et al. 2006). Diese Vorgehensweise wird als *jittern* bezeichnet und hat den Vorteil, das bei der mathematischen Auswertung verschiedene Zeitpunkte der HRF erfasst werden. Außerdem werden die unterschiedlichen Varianten der verwendeten Stimuli, z. B. negative und neutrale Bilder, bei diesen Designs meistens randomisiert präsentiert. Durch Randomisieren und *Jittern* sind die Art und der genaue Zeitpunkt des nächsten Stimulus für den Probanden nicht präzifizierbar. Damit können ungewollte Effekte, die durch eine Erwartungshaltung des Probanden entstehen, vermieden werden.

Um die jeweiligen Nachteile der beiden vorgestellten Designvarianten zu umgehen, sind auch Mischformen möglich, z. B. mehrere Events derselben Kategorie in einem Block.

Die Abbildung 2.5 zeigt schematisch ein Block- und ein Event-related Design, die als Modellannahme für das Signal dienen. Der Zeitpunkt zu der ein Stimuli auftritt, wird als Startpunkt (*Onset*) einer Hirnaktivität bezeichnet. Zur Verbesserung der Modellannahme mit Blöcken oder Events wird die Signalantwort im Gehirn in Form der HRF berücksichtigt. Daraus ergibt sich die Kombination dieser beiden Signale mit Hilfe der mathematische Operation einer Faltung der Signale. Sind alle Zeitpunkte für das Auftreten von Stimuli vordefiniert, ergibt sich über die gesamte Zeitreihe der Messung ein präzifizierter Signalverlauf in den bei der Verarbeitung des Reizes involvierten Hirnareale.

Faltung: Die Faltung ist ein mathematischer Operator für Funktionen, der den gleitenden Durchschnitt zweier Funktionen berechnet. In dem hier gegebenen Anwendungsfall wird die Faltung für diskrete Signale bestimmt und daher eine sogenannte diskrete Faltung durchgeführt. Diese kann mit Hilfe des Algorithmus der schnellen Faltung sehr zeiteffizient ermittelt werden, indem die zeitdiskreten, endlichen Funktionen mit einer diskreten Fourier-Transformation (DFT) auf ihre diskreten, periodischen Frequenzspektren abgebildet werden. Die Faltung der Funktionen entspricht einer einfachen Multiplikation ihrer Frequenzspektren. Mit der inversen diskreten

Fourier-Transformation wird das Ergebnis wiederum in den Zeitbereich transformiert (Gellert et al. 1972, S. 754).

Mehrfaktorielles Design und Designmatrix: Ein Paradigma setzt sich aus mehreren unterschiedlichen Bedingungen zusammen, die möglichst unabhängig voneinander sein sollten. Die Faltung von Stimulationsmodell und HRF wird dann für alle Bedingungen berechnet. Die einzelnen Bedingungen werden auch als Faktoren des Designs bezeichnet. Designs mit mehreren Bedingungen werden mehrfaktoriell genannt. Die Faktoren wiederum können entweder als eine Art von Stimuli definiert sein (kategorielles Design) oder als quantitatives Maß für das erwartete Signal (parametrisches Design). Je nach Art dieser Designklassifikation können unterschiedliche Vergleiche der Bedingungen in der Datenanalyse berechnet werden (Details finden sich in Schneider et al. (2007, S. 125ff)). Die prädierten Signale werden jeweils als Spalte einer Matrix dargestellt, die als Designmatrix bezeichnet wird. Als graphische Darstellung werden oft unterschiedliche Grauwerte verwendet. Ein Beispiel ist in Abbildung 2.7b gezeigt. Mit Hilfe dieser Matrix lässt sich vor Beginn eines Experiments die Orthogonalität eines Paradigmas berechnen, d. h. die Unabhängigkeit der verschiedenen Bedingungen in der Stimulation geprüft werden, die größtmöglich sein sollte. Unter der Annahme einer linearen Aufsummierung der Hirnaktivität bei der Durchführung einer Aufgabe, können die unterschiedlichen Bedingungen nach einer statistischen Auswertung der Daten zueinander in Bezug gesetzt werden, z. B. durch Subtraktion zweier Bedingungen. Diese Beziehung wird als Kontrast bezeichnet und zeigt die Aktivierung der relevanten Hirnareale.

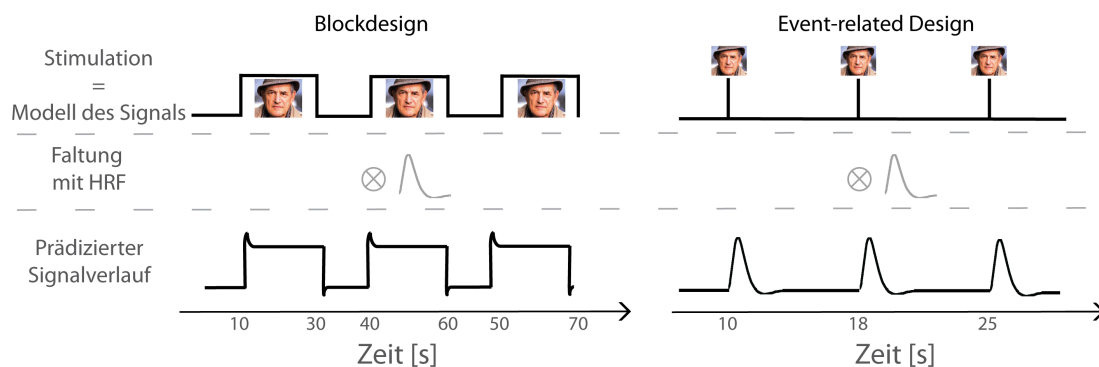


Abbildung 2.5: Stimulation und Design: Die Stimulation in einem fMRT-Experiment bezeichnet die Applikation eines Reizes an den Probanden, z.B. das Zeigen eines Bildes. Diese Stimulation kann für einen längeren Zeitraum (Block) oder nur für einen Zeitpunkt (Event) erfolgen. Die Stimulation wird in eine Modellannahme für die Auswertung, die Designmatrix übertragen. Je nach Art der Applikation somit wird in Block- oder Event-related Designs unterschieden. Aus der Modellannahme der Stimulation ergibt sich durch mathematische Faltung mit der zu erwartenden Signalantwort im Gehirn in Form der HRF ein prädiertes Signalverlauf für diese Bedingung des Paradigmas.

Ist die Stimulation eines fMRT-Paradigma vollständig definiert, kann dieses synchronisiert mit der Aufnahme der fMRT-Daten abgespielt und die Hirnaktivität der Probanden im Verlauf des Experiments erfasst werden.

2.2.2 Reaktionserfassung bei fMRT-Experimenten

Zusätzlich zu der Hirnaktivierung in den fMRT-Daten können auch weitere Reaktionen der Probanden, z. B. Antwortsignale des Probanden oder physiologische Verhaltensparameter, mit erfasst werden. Antwortsignale werden im einfachsten Fall mit Hilfe von Tastboxen erfasst, auf denen Probanden bestimmte Tasten drücken müssen. Damit können Antworten auf Fragen nach der Bewertung oder dem individuellen Empfinden von Stimuli während des Experiments erfasst werden. Ähnlich dazu können auch Joysticks zum Navigieren oder Gegenstände für Kraftmessungen verwendet werden. Somit lassen sich korrekte Antworten, Reaktionszeiten oder Bewertungen von Stimuli als Verhaltensparameter erfassen. Zudem können auch physiologische Reaktionen aufgezeichnet werden, wie der Herzschlag (mit Elektroden oder Pulsoxymeter), die Atmung (mit Atemgurt oder Nasenschlauch), der Hautleitwiderstand (mit Elektroden) und die Augenbewegungen.

2.2.2.1 Die Erfassung von Augenbewegungen bei fMRT-Experimenten

Augenbewegungen während eines fMRT-Experiments können entweder mit einem Brillensystem (Goggles) oder über Spiegelreflexionen erfasst werden. In beiden Fällen wird infrarotes Licht durch die Pupille in das Auge geleitet und die Reflexion des Lichtes an der Kornea gemessen. Das Maximum dieser Reflexion und die Größe der Pupille ergeben die aktuelle Blickposition des Auges. Damit die Blickrichtung korrekt bestimmt werden kann, muss das System vor jeder Messung kalibriert werden. Dazu werden mehrere Punkte auf eine Leinwand projiziert, die ein Proband zu bestimmten Zeitpunkten fixieren soll. Wird die Messung der Augenbewegungen mit dem Bildschirm synchronisiert, auf dem die Stimuli präsentiert werden, gibt dies Auskunft über die Fixationen und Augenbewegungen der Probanden. Abbildung 2.6 zeigt exemplarisch den experimentellen Ablauf für die Erfassung des Signals im MR-Scanner.

Insbesondere bei visuellen Paradigmen liefern die Augenbewegungen wichtige Information über die Bereitschaft des Probanden, die Stimulation wahrzunehmen, die Art der visuellen Erkundung der Stimuli und der Fixierung auf den präsentierten Stimulus. Weiterhin liefert die Anzahl und Dauer der fixierten Blickpunkte Informationen über die als relevant wahrgenommenen Bildbereiche. Zusätzlich können Pupillenreaktionen indizieren, wie ein Stimulus wahrgenommen wird, z. B. durch eine Pupillenvergrößerung bei emotionalen Bildern. Weitere Indikatoren ergeben sich aus dem Verlauf der Augenbewegungen (Sakkaden), die hier nicht im Detail ausgeführt werden sollen. Bei visuellen Paradigmen wird zwischen den Stimuli häufig ein Fixationskreuz

gezeigt, das als definierte Ruhebedingung angesehen wird. Mit Hilfe der gewonnenen Information aus den Augenbewegungen kann dies nach der Messung (post-hoc) überprüft werden. Damit können besonders starke Abweichungen als Fehlerquelle für die Analyse erkannt und verworfen werden. Dieser Ansatz zum Ausschluss von ungültigen Messungen ist die Grundlage für die erste Realisierung eines adaptiven Paradigmas, das in Abschnitt 2.4 vorgestellt wird. Die für diese Arbeit relevante Messmethodik und die nachfolgende Auswertung der Daten wird im empirischen Teil der Arbeit in Abschnitt 4.2.3 beschrieben.

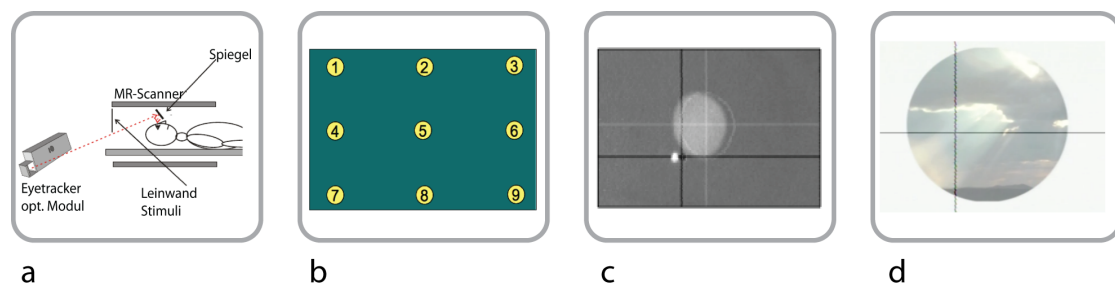


Abbildung 2.6: Eyetracker bei fMRT-Experimenten: Die Messung von Augenbewegungen mittels der Korneareflexion wird mit einem Eyetracker gemessen, der infrarotes Licht über einen Spiegel in das Auge leitet (a⁶). Für eine korrekte Blickrichtungsbestimmung muss das System vor der Messung kalibriert werden (b). Die Korneareflexion und Pupillendetektion geben die Koordinaten der aktuellen Blickrichtung an (c). In der Synchronisation mit dem Präsentationsbildschirm werden die Augenbewegungen während der Stimulation aufgezeichnet (d).

2.2.3 Vorverarbeitung von fMRT-Daten

Um den Abstand zwischen dem gewünschten Signal und dem Hintergrundrauschen zu optimieren, werden die gemessenen fMRT-Daten vor der statistischen Auswertung korrigiert. Dabei werden Korrekturverfahren angewendet, um Signalverzerrungen und Fehlerquellen zu minimieren. Zusätzlich können die verschiedenen Aufnahmemodalitäten (strukturelle und funktionelle Aufnahmen) aufeinander angepasst werden damit die Ergebnisse übertragbar sind. Die am häufigsten angewendeten Schritte für die Vorverarbeitung der fMRT-Daten sind die Folgenden:

Bewegungskorrektur: Da fMRT-Experimente in der Regel über einen Zeitraum von wenigen Minuten bis zu einer Stunde dauern, ist es unmöglich, geringe Kopfbewegungen der Probanden vollständig zu vermeiden. Dies kann mit einer Bewegungskorrektur ausgeglichen werden. Dabei wird ein Hirnvolumen der gemessenen Zeitserie als Referenz definiert, z. B. das erste Volumen der gesamten Zeitserie, und alle anderen Volumen der EPI-Zeitserie darauf registriert. Die Registrierung wird mit Hilfe von rigiden Transformationen (Translation und Rotation) berechnet und daher als Rigid-body-Transformation bezeichnet. Die Grenze dieses Korrekturverfahrens

⁶Abbildung entnommen und adaptiert aus Manual ASL Long Range Optics, www.asleyetracking.com

liegt in der festgelegten Voxelgröße, d. h. bei einer Auflösung von 3 mm^3 in der Echo Planar Imaging (EPI)-Sequenz kann eine Bewegung, die größer ist als 3 mm, nicht mehr dem korrekten Ursprungsvoxel zugeordnet werden.

Fieldmap-Korrektur: Das Magnetfeld B_0 wird durch das zu messende Objekt, den Kopf des Probanden, in seiner Homogenität gestört. Mit Hilfe einer zusätzlichen Sequenz können aktuelle Suszeptibilitätsartefakte gemessen werden und mit dieser sogenannten Fieldmap eine Korrektur dieser Signalverzerrungen berechnet werden.

Koregistrierung: Diese Vorverarbeitung dient nicht der Verbesserung des Signals, sondern der Kombination von funktionellen und strukturellen Daten. Die funktionellen Daten können durch nicht-rigide Transformationen auf die zugehörigen strukturellen Daten koregistriert werden. Die strukturellen Daten liefern eine wesentlich höhere räumliche Auflösung als die funktionellen Daten. Durch die Koregistrierung dieser Daten können die funktionellen Ergebnisse auf die strukturellen Daten übertragen werden. Damit ist eine exakte Lokalisierung der in der statistischen Auswertung ermittelten Hirnareale möglich.

Normalisierung: Ähnlich der Koregistrierung dient die Normalisierung dem Vergleich der funktionellen Aktivierungen zwischen verschiedenen Probanden. Dazu müssen alle funktionellen Daten auf einen einheitlichen strukturellen Raum registriert werden. Es können verschiedene Standardvorlagen verwendet werden. Eine der am häufigsten verwendeten Vorlagen wurde vom Montreal Neurological Institute (MNI) aus der Mittelung von 152 strukturellen Datensätzen erstellt. Durch die Normalisierung auf diese Standardvorlage sind die ermittelten Koordinaten (MNI-Koordinaten) von ermittelten Hirnregionen zwischen verschiedenen Probanden und Studien vergleichbar.

Hochpassfilterung: Eine Filterung bestimmter Frequenzen im Signal ist hilfreich zur Entfernung von zeitlichen Schwankungen im MR-Signal, z. B. einer Signaldrift durch Erwärmung des Magneten während der Messung.

Glättung: Eine Glättung der Daten minimiert den Einfluss von Signalfuktuationen durch Rauschen und wird daher für fMRT-Daten vor der statistischen Analyse zur Erhöhung des Signal-Rausch-Verhältnisses durchgeführt. Bei einer Voxelgröße von 3 mm^3 wird häufig eine Glättung mit einem Gaußfilter der Größe 4 mm bis 8 mm verwendet⁷.

2.2.4 Statistische Auswertung von fMRT-Daten

Zur Beantwortung einer Fragestellung, die mit einem fMRT-Experiment gestellt wird, werden die erfassten Daten statistisch ausgewertet, um Aussagen über die Aktivität von Hirnarealen nach einer spezifischer Stimulation treffen zu können. Grundsätzlich sind dazu zwei statistische Ansätze möglich:

⁷ Die Glättung von fMRT-Daten wird aktuell immer wieder sehr kritisch diskutiert. Zum aktuellen Zeitpunkt wird diese Vorverarbeitung in den meisten publizierten Studien durchgeführt.

Univariate Statistik: Dabei wird jeder Datenpunkt, hier also jedes Voxel, als eindimensional und unabhängig von allen anderen Voxeln betrachtet. Für jedes Voxel wird die gemessene Zeitreihe der fMRT-Daten mit dem zuvor erstellten Modell, der Designmatrix, abgeglichen. Zu den univariaten Verfahren gehören z. B. Korrelationsberechnungen, t-Tests und das Allgemeine Lineare Modell (siehe Abschnitt 2.2.4.1). Alle Verfahren sind mathematisch wenig komplex und können somit effizient in Algorithmen umgesetzt werden. Nachteilig bei diesen Verfahren ist, dass dabei räumliche Nachbarschaftsbeziehungen der Voxel nicht berücksichtigt werden und die Ergebnisse stark von dem zugrundeliegenden Modell abhängen. Dennoch ist die univariate Auswertung die momentan am häufigsten verwendete Analyseverfahren für fMRT-Experimente.

Multivariate Statistik: Neuere Ansätze für die Auswertung von fMRT-Daten betrachten die Voxel nicht unabhängig voneinander sondern fassen sie in einem multivariate Verfahren zusammen. Dabei werden Merkmale aus der Gesamtheit der Voxel extrahiert, mit Hilfe derer die Verfahren lernen, die Daten zu klassifizieren. Es können auch funktionelle Zusammenhänge von räumlich benachbarten Voxeln berücksichtigt werden. Zu den multivariaten Verfahren gehören z. B. Neuronale Netze, Support Vector Machine (SVM) und Relevance Vector Machine (RVM). Außerdem gibt es Verfahren, mit denen die Daten modellfrei exploriert werden können. Diese werden als strukturfindende Verfahren bezeichnet (z. B. Hauptkomponentenanalyse (PCA) und Unabhängige Komponentenanalyse (ICA)).

Multivariate Ansätze zur Auswertung von fMRT-Daten sind ein Forschungsfeld mit großem Potential aber derzeit noch vielen offenen Fragen, die mit dieser hochdimensionalen Art der Datenauswertung einhergehen. Die Details zu diesen Methoden sind nicht Bestandteil der vorliegenden Arbeit. Allerdings ist die Möglichkeit zur Integration neuer Analysealgorithmen wichtig für das Konzept der vorliegenden Arbeit. Das in dieser Arbeit entwickelte Softwarepaket, das im folgenden Kapitel 3 detailliert erläutert wird, beinhaltet aktuell eine Implementierung für die populärste Analyseverfahren für fMRT-Daten: das Allgemeine Lineare Modell.

2.2.4.1 Allgemeines Lineares Modell

Das Modell: Unter dem Begriff Allgemeines (Generalisiertes) Lineares Modell (GLM), im Englischen als *General Linear Modell* bezeichnet, werden ganz allgemein eine Vielzahl statistischer Testverfahren zusammengefasst, zu denen Mittelwertvergleiche, varianzanalytische Verfahren, Korrelations- und Regressionsrechnungen gehören. Die grundlegende Annahme für die Verwendung solcher Modelle ist, dass ein linearer Zusammenhang zwischen den gemessenen Daten und einer Variablen besteht. Abbildung 2.7a veranschaulicht das Prinzip: Aus dem Paradigma wird der Signalverlauf prädiziert (blau). Wenn das Modell und die tatsächlich gemessenen Daten einen ähnlichen Verlauf zeigen, kann geschlossen werden, dass die Hirnaktivierung in diesem Voxel durch die spezifische Stimulation hervorgerufen wurde. Im Zusammenhang mit der

Auswertung von fMRT-Daten wurde das GLM erstmalig von Friston et al. (1995) beschrieben. Mathematisch formuliert lautet das Modell für ein Voxel wie folgt:

$$\vec{y} = X * \vec{\beta} + \vec{\epsilon}$$

in ausführlicher Schreibweise:

$$\begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix} = \begin{bmatrix} x_{11} & \dots & x_{1k} \\ \vdots & \ddots & \vdots \\ x_{n1} & \dots & x_{nk} \end{bmatrix} * \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_n \end{pmatrix} + \begin{pmatrix} \epsilon_0 \\ \epsilon_1 \\ \vdots \\ \epsilon_n \end{pmatrix}$$

In dieser Gleichung beschreibt \vec{y} den Vektor der abhängigen Variable, in diesem Fall also die Werte der gemessenen Zeitreihe zu den Zeitpunkten 0 bis n in einem Voxel. Die Matrix X enthält die unabhängigen Variablen, d. h. die prädierten Signale für die Bedingungen des Paradigmas. Das Ziel der statistischen Auswertung besteht nun darin, die β -Parameter (auch β -Koeffizienten oder β -Gewichte) und den Fehlerterm ϵ so zu schätzen, dass der Fehlerterm minimal wird. Damit wird gleichzeitig die durch β -Koeffizienten erklärte Varianz maximal. Die Abbildung 2.7b veranschaulicht dieses mathematische Modell an einem Paradigma, das aus zwei aktiven Bedingungen und einer Ruhebedingung besteht. Diese sind als Spalten mit Grauwerten (X_1 und X_2) dargestellt. Die Ruhebedingung wird in diesem Fall implizit als Konstante c modelliert. Das gemessene Signal y ergibt sich als lineare Kombination dieser Prädiktoren, oder auch Regressoren, und dem Fehlerterm ϵ .

Regressoren: Als Regressor wird in diesem Zusammenhang eine Spalte innerhalb der Designmatrix bezeichnet, die für jeden Messpunkt in der fMRT-Zeitreihe einen Wert prädiert.

Statistische Interferenz: Mit den beim GLM berechneten β -Koeffizienten wird für jede Bedingung in jedem Voxel eine Effektgröße berechnet, die den Einfluss der Stimulation an der Position im Gehirn beschreibt. Diese Effektgrößen sind vergleichsweise gering, da der Signalanstieg des BOLD-Effektes bei 1 % bis 5 % liegt und damit im Rauschen des Signals verloren gehen kann. Für eine Aussage über die Hirnaktivität wird die Effektgröße mit dem Hintergrundrauschen in ϵ statistisch verglichen. Die statistische Interferenz prüft die Hypothese, ob der Koeffizient durch eine spezifische Stimulation bedingt wird oder ausschließlich Rauschen beinhaltet. Dazu werden die Regressoren positiv oder negativ gewichtet und aufsummiert. Das Ergebnis wird als Kontrast bezeichnet und kann alle enthaltenen Bedingungen, die in den Regressoren abgebildet sind, gegeneinander vergleichen.

⁸Abbildungen adaptiert aus Schneider et al. (2007, S. 124,141)

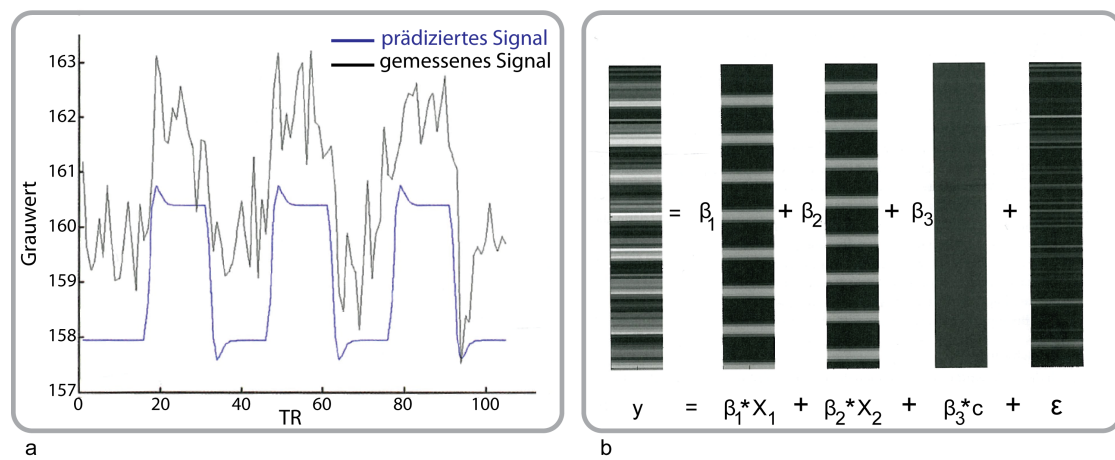


Abbildung 2.7: Erläuterung GLM: Mit Hilfe eines Modells kann für jede Zeitreihe eines Voxels die Genauigkeit des prädizierten Signals an das reell gemessene Signal (y) quantifiziert werden (a). Dazu werden β -Koeffizienten für alle Regressoren (X) des Designs und ein Fehlerterm ϵ ermittelt (b)⁸.

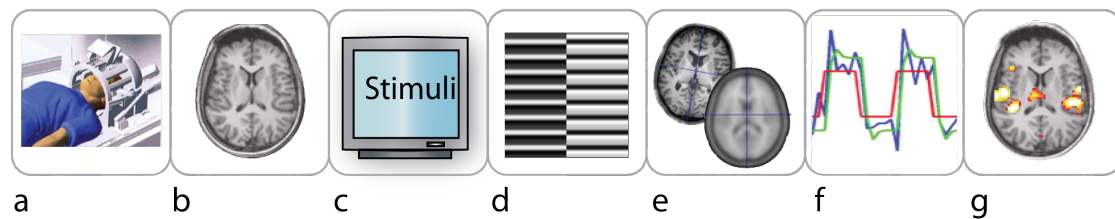


Abbildung 2.8: Der Ablauf eines konventionellen fMRT-Experiments: Ein Proband wird mit einer Kopfspule im MR-Scanner positioniert (a). Die Aufnahme einer fMRT-Zeitserie (b) erfolgt während Stimuli an den Probanden appliziert werden (c). Die Designmatrix beschreibt das Modell für die erwarteten Signaländerungen (d). Die fMRT-Daten werden auf einem externen Computer ausgewertet. Dazu gehört eine Vorverarbeitung, z. B. Bewegungskorrektur (e) und eine statistische Analyse, z. B. GLM-Analyse (f). Die Anzeige der als aktiv ermittelten Hirnregionen erfolgt farbig kodiert auf der Topographie des Gehirns (g).

2.3 Echtzeit-fMRT als Schnittstelle zwischen Gehirn und Computer

2.3.1 Schnittstellen zwischen Gehirn und Computer

Als eine Schnittstelle zwischen Gehirn und einem Computer (*Brain-Computer-Interface*, im Folgenden BCI genannt) werden alle Techniken bezeichnet, die eine Möglichkeit zur direkten Kommunikation zwischen Gehirn und Computer ermöglichen. BCIs verwenden neurophysiologische Signale, die im Gehirn detektiert wurden, um ein externes Gerät zu steuern. Werden diese Signale ohne direkten Eingriff in das Gehirn abgeleitet, wird dies als nicht-invasive Signalerfassung bezeichnet. Im Gegensatz dazu erfolgt die invasive Ableitung über implantierte Elektroden (Birbaumer et al. 2007) direkt im Gehirn.

Zur nicht-invasiven Erfassung der Hirnaktivität stehen die folgenden Modalitäten zur Verfügung (Haus et al. 2012):

- **Elektroenzephalographie (EEG):**
Ableitung elektrischer Potentiale an der Schädeloberfläche, die durch neuronale Aktivität erzeugt wurden,
- **Funktionelle Nah-Infrarot-Spektroskopie (fNIRS):**
Messung der charakteristischen Absorptionsspektren von kurzwellig infrarotem Licht zur Detektion von Änderungen der Hämoglobinkonzentration im Blut;
- **Magnetenzephalographie (MEG):**
Messung magnetischer Aktivität des Gehirns an der Schädeloberfläche, die durch neuronale Aktivität erzeugt wurde und
- **Funktionelle Magnetresonanztomographie (fMRT):**
siehe Abschnitt 2.1.3.

Mit Hilfe von BCIs können zwei interessante Ansätze realisiert werden:

Steuerung externer Systeme: Mit Hilfe einer solchen Schnittstelle können Muster von Hirnaktivität in Steuersignale für eine externe Rechneinheit übersetzt werden. Somit kann z. B. durch gezieltes, bewusste Veränderung der Hirnaktivität, wie Kopfrechnen oder mentales Lesen, ein Cursor auf einem Bildschirm navigiert oder eine computergesteuerte Prothese bewegt werden.

Operante Konditionierung und Neurofeedback: Aus der aufgezeichneten Hirnaktivität kann der Computer eine Antwort berechnen und diese als sogenanntes Feedback an Probanden zurück projiziert werden. Da das Signal eine Hirnaktivität repräsentiert, wird dies als Neurofeedback bezeichnet. Das Feedback wird mit den gleichen Modalitäten appliziert wie die Stimulation des Probanden, also z. B. über Bilder, Töne oder elektrische Reizung der Haut. Damit ergibt sich ein

interessanter Ansatz für neurowissenschaftlichen Experimente: In konventionellen Paradigmen wird die Hirnaktivität beim Ausführen einer Aufgabe beobachtet und anschließend analysiert. Beim Neurofeedback hingegen lernen Probanden ihre eigene Hirnaktivität willentlich zu beeinflussen und die daraus resultierenden Verhaltensänderungen können experimentell beobachtet werden. Diese Art der Experimente wird als operante Konditionierung bezeichnet.

Eine ausführliche und aktuelle Übersicht zu dem Thema Neurofeedback und Neuromodulation sowohl mit nicht-invasiven als auch mit invasiven Modalitäten ist in Coben et al. (2011) beschrieben. Bereits vor über 50 Jahren hat die Entwicklung von Neurofeedback mit EEG-basierten Systemen begonnen. Über die Zeit wurden verschiedene Möglichkeiten zur Feedbackgenerierung getestet. Dazu gehören u. a. das Trainieren der Probanden auf verschiedene Frequenzbänder des EEG-Signals oder auf die Kohärenz von Elektrodensignalen (Haus et al. 2012). Weiterhin gibt es Feedback mit den sogenannten Slow-Cortical-Potentials (SCPs), die maßgeblich von Niels Birbaumer etabliert und angewendet wurden (Birbaumer et al. 2007). Die Anwendung von Neurofeedback auf der Basis des EEG-Signals wurde u.a. bei den folgenden Erkrankungen etabliert:

- ADHS (Aufmerksamkeitsdefizit-/Hyperaktivitätsstörung):
Psychische Störung, die sich durch Probleme mit der Aufmerksamkeit, Impulsivität und Hyperaktivität auszeichnet;
- Autismus:
Wahrnehmungs- und Informationsverarbeitungsstörung des Gehirns, die sich durch Schwächen in sozialer Interaktion und Kommunikation sowie durch stereotype Verhaltensweisen und Stärken bei Wahrnehmung, Aufmerksamkeit, Gedächtnis und Intelligenz zeigt;
- Epilepsie:
Krankheitsbild bei dem spontan auftretende epileptische Anfälle auftreten, die durch synchrones Entladen von Neuronengruppen ausgelöst werden;
- Locked-in-Syndrom:
Zustand von Menschen, die z.B nach einem Schlaganfall zwar bei Bewusstsein sind, körperlich allerdings gelähmt sind und somit nicht mit der Umwelt kommunizieren können.

Trotz vielseitiger und erfolgreicher Anwendung von EEG-basierten Neurofeedbacksystemen können damit nicht alle Fragestellungen untersucht werden (Birbaumer et al. 2007). Das EEG-Signal wird an der Schädeloberfläche abgeleitet. Damit ist das Signal eine Summation der Aktivität aus verschiedenen Regionen, die nur indirekt ihrem genauen lokalen Ursprung im Gehirn zugeordnet werden kann. Außerdem ist die Signalerfassung auf kortikale Hirnareale beschränkt, d. h. subkortikale Hirnstrukturen können nicht präzise abgebildet werden. Um diese Einschränkungen zu umgehen wurden im Laufe der letzten Jahre auch weitere Modalitäten für BCIs verwendet. In der Literatur finden sich Ansätze zu BCI-Steuerungen und Neurofeedback mittels MEG (Birbaumer et al. 2007; Sudre et al. 2011), fNIRS (Abdelnour et al. 2009) und eine kombinierte Variante mit fNIRS und EEG (Fazli et al. 2012). Eine Vielzahl an realisierten BCI-

Systemen findet sich mit fMRT-Bildgebung. Wie im Abschnitt 2.1.2 vorgestellt, kann mit dieser Technik über alle Hirnstrukturen bereits eine räumliche Auflösung von ca. 3 mm^3 erreicht werden. Aufgrund dieser einzigartigen Ortsauflösung sind fMRT-BCI-Systeme, neben den populären EEG-Systemen mit hoher zeitlicher Auflösung, ein vielversprechender Ansatz, der sich in einer Vielzahl publizierter Arbeiten widerspiegelt. Eine Übersicht über die genannten Verfahren findet sich in Tabelle 2.1. Die folgenden Abschnitte stellen die Möglichkeiten von fMRT-BCI-Systemen mit Berücksichtigung der besonderen Anforderungen vor. Außerdem werden die in der Literatur beschriebenen methodischen Entwicklungen und Anwendungen vorgestellt.

Eigenschaft	EEG	MEG	BCI-Modalität	
			fNIRS	fMRT
Neuronale Aktivität	direkt	direkt	indirekt	indirekt
Zeitliche Auflösung	sehr hoch (ms)	hoch (ms)	niedrig (s)	niedrig (s)
Räumliche Auflösung	gering (cm)	mittel (mehrere mm)	mittel (mehrere mm)	hoch (wenige mm)
Erfassbare Hirnareale	kortikal	kortikal	kortikal	kortikal + subkortikal
Portabilität	ja	nein	ja	nein
Kosten	günstig	teuer	günstig	teuer

Tabelle 2.1: Die Tabelle vergleicht die wichtigsten Eigenschaften von nicht-invasiven Schnittstellen zwischen dem menschlichen Gehirn und Computern. Dabei sind zeitliche Auflösung im EEG und die räumliche Auflösung bei fMRT einzigartig und somit die am häufigsten verwendeten Modalitäten.

2.3.2 Echtzeit-fMRT

BCI-Systeme basierend auf fMRT-Messungen bieten eine einzigartige Ortsauflösung für die Erfassung neuronaler Aktivität. Wird die Schnittstelle zwischen Gehirn und einem externen Gerät (Computer) mit Hilfe von fMRT realisiert, muss die gesamte Datenverarbeitung und Auswertung zeitlich parallel zum Experiments stattfinden. Diese Art der Auswertung wird als Echtzeit-fMRT bezeichnet, wobei Echtzeit hier die Dauer der Aufnahme eines Hirnvolumens umfasst (TR-Zeit, wie in Abschnitt 2.1.2 erläutert meist um die 2 s). Vereinfacht gesagt ist Echtzeit-fMRT der geschlossene Kreislauf, der ein konventionelles fMRT-Experiment enthält und welches permanent wiederholt wird, wie in Abbildung 2.9 schematisch dargestellt. Genauer betrachtet impliziert diese Vorgehensweise, dass die Daten direkt vom MR-Scanner auf einen externen Rechner übertragen werden müssen. Weiterhin stehen für die Vorverarbeitung und Analyse zur Laufzeit des Experiments nur die zum aktuellen Zeitpunkt bereits aufgenommenen Daten zur Verfügung. Damit sind sowohl die Möglichkeiten der Korrektur, z. B. einer Drift im Gesamtsignal, eingeschränkt und auch die statistische Teststärke der Analyse geringer. Außerdem stellt diese Art der

Experimentdurchführung besonders hohe Anforderungen an die Zuverlässigkeit und Stabilität des verwendeten Setups. Die Datenverarbeitung ist mit einer Interaktion durch den Probanden verbunden, z. B. wenn der Proband aufgrund eines Feedbacksignals sein Verhalten und damit die messbare Hirnaktivität ändert. Die korrekte und stabile Datenverarbeitung zur Laufzeit des Experiments ist von großer Bedeutung. Im Gegensatz zu konventionellen Experimenten können unterschiedliche Parameter für die Vorverarbeitung nicht getestet werden oder die Analyse nach einem Softwareabsturz erneut berechnet werden. Eine Fehlfunktion oder der Abbruch des Experiments zur Laufzeit ist daher nicht nur zeit- und kostenintensiv, sondern führt auch dazu, dass die aktuell aufgenommenen Daten nicht weiter verwendet werden können. Zusammengefasst ergeben sich folgende besondere Herausforderungen an ein fMRT-BCI-System:

- Die fMRT-Daten müssen zuverlässig und schnell vom MR-Scanner zu externer Auswerteeinheit transportiert werden,
- die Vorverarbeitung und Auswertung der Daten muss innerhalb der verwendeten TR-Zeit abgeschlossen werden,
- das Systems muss zuverlässig und stabil arbeiten, da jede Unterbrechung zum Abbruch der aktuellen Messung führt und
- die methodische Anforderungen müssen berücksichtigt werden, z. B. die Beschränkung der Datenanalyse auf die zum aktuellen Zeitpunkt zur Verfügung stehenden Daten.

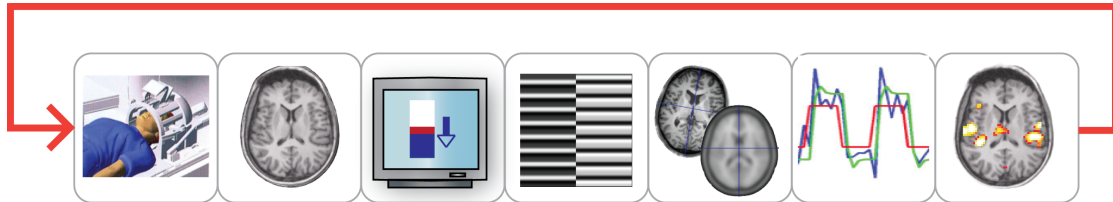


Abbildung 2.9: Echtzeit-fMRT: Als Echtzeit-fMRT wird die Verarbeitung aller Schritte eines konventionellen fMRT-Paradigmas innerhalb einer Repetitionszeit (TR, momentan meist um die 2 s) bezeichnet. Dieser geschlossene Kreislauf für die Verarbeitung stellt besondere Anforderungen an die Geschwindigkeit der Auswertung und die Zuverlässigkeit des Gesamtsystems.

Die Auswertung von fMRT-Daten in Echtzeit kann für folgende Anwendungen verwendet werden, die ausführlich und mit einer Übersicht über die bisher publizierten Arbeiten in den Abschnitten 2.3.2.2 und 2.3.3 erläutert werden:

- Qualitätssicherung während der Laufzeit von MR-Messungen,
- Funktionelle Lokalisierung,
- Steuerung externer Systeme und
- Neurofeedback.

2.3.2.1 Literaturübersicht zu technischen und methodischen Entwicklungen für Echtzeit-fMRT

Begonnen hat die Entwicklung von Echtzeit-fMRT bereits mit den Anfängen von neurowissenschaftlichen fMRT-Experimenten. Die erste Arbeit dazu wurde von Cox et al. (1995) vorgestellt. Diese Arbeit beschreibt eine rekursiv berechnete Korrelationsanalyse mit der die Daten sofort ausgewertet werden können. Allerdings ist diese Arbeit auf die Auswertung beschränkt und enthält kein vollständiges System zum Datentransport oder der Vorverarbeitung der Daten. Ein wesentlich umfangreicheres System wurde von Voyvodic (1999) entwickelt. Dieses beinhaltet bereits einen vereinfachten Algorithmus zur Bewegungskorrektur als Vorverarbeitung der Daten und analysiert die Daten mit einem t-Test. Allerdings wird die Rechenzeit hier als 'Nah-Echtzeit' bezeichnet, da diese mit 1 min bis 2 min noch deutlich über der TR-Zeit lag. Erwähnenswert an dieser Arbeit ist die integrierte Komplexität. Das System beinhaltet neben der genannten Datenverarbeitung auch eine mit dem MR-Scanner synchronisierte Stimulationskomponente und die simultane Erfassung von physiologischen Parametern wie Herzrate und Atmung. Ein ähnliches System, allerdings bereits mit einer rechentechnisch aufwendigeren GLM-Auswertung wurde wenig später von Smyser et al. (2001) vorgestellt. Diese Systeme werden noch einmal im Abschnitt 3.1.1 im Zusammenhang mit dem in dieser Arbeit entwickelten Softwarepaket diskutiert. Seitdem sind analog zum allgemein rasanten Fortschritt von Computertechnik auch die Techniken und Methoden für Echtzeit-fMRT weiterentwickelt worden. Einerseits wurden dazu die zugrunde liegenden EPI-Sequenzen verbessert, um die Qualität des fMRT-Signals zu verbessern und die Datenanalyse besser zu beherrschen. Dazu wurden Sequenzen vorgestellt, die das Signal-Rausch- bzw. Kontrast-Rausch-Verhältnis durch Multi-Echo-Sequenzen erhöhen (Posse et al. 2001; Posse et al. 2003; Posse et al. 2012) oder Sequenzen die eine Verzerrungskorrektur zur Verbesserung der Signalqualität beinhalten (Weiskopf et al. 2005).

Andererseits wurden Algorithmen für die Vorverarbeitung und die Auswertung der Daten entwickelt. Für die Echtzeitkorrektur von Kopfbewegungen wurden Algorithmen für Rigid-Body-Registrierungen von Cox et al. (1999) und Mathiak et al. (2001) vorgestellt. Weitere Echtzeitalgorithmen zur Korrektur von physiologischen Artefakten und räumlicher Glättung wurden von Posse et al. (2003) und für die Normalisierung der Daten auf einen stereotaktischen Raum von Lee et al. (2008) vorgestellt.

Für die statistische Analyse der fMRT-Daten in Echtzeit wurden mehrere der vorgestellten statistischen Verfahren neu implementiert, damit diese innerhalb der gegebenen Zeit berechnet werden können. Von den univariaten Verfahren wurden so Varianten von t-Tests (Voyvodic 1999) und Korrelationsanalyse (Cox et al. 1995; Posse et al. 2001) präsentiert. Für die Echtzeit-GLM-Analyse von fMRT-Experimenten findet sich eine Vielzahl an methodischen Varianten. Dazu gehören inkrementelle Berechnungen des Modells, d. h. die Berechnung der Ergebnisse des aktuellen Zeitschritts basierend auf den bereits errechneten Ergebnissen, wie von Bagarinao et al.

(2003) und Hinds et al. (2011) vorgestellt. Weiterhin gibt es Algorithmen, die rekursive Schätzungen verwenden, wie von Smyser et al. (2001) implementiert oder von Caria et al. (2007); Rota et al. (2009); Weiskopf et al. (2003) verwendet. Zusätzlich wird bei vielen Verfahren die Auswertung auf eine Teilmenge der Daten beschränkt, eine Vorgehensweise die in der Signalverarbeitung als ‘Fensterung der Daten’ bezeichnet wird (*sliding-window-Verfahren*) und erstmalig für die GLM-Berechnung von Gembris et al. (2000) vorgestellt wurde.

Darüber hinaus wurden bisher auch einige multivariate Ansätze für die Echtzeit-Auswertung angepasst und verwendet. In der Arbeit von Esposito et al. (2003) wurde ein Echtzeitalgorithmus zur Berechnung einer ICA für die fMRT-Daten vorgestellt. Für die Klassifikation von Daten in Echtzeit wurde von LaConte et al. (2007) ein Ansatz präsentiert, der vollständig in der Software des MR-Scanners integriert ist und mit Hilfe von SVMs die Daten klassifiziert. Für die Ermittlung einer Vorhersage von Entscheidungen beim Ultimatum-Game wurde von Hollmann et al. (2011) eine RVM zur Echtzeitklassifikation verwendet. Eine kürzlich veröffentlichte Arbeit von Koush et al. (2013) stellt erstmalig eine Echtzeitauswertung von Konnektivitäten zwischen Hirnarealen basierend auf dem Ansatz des *Dynamic Causal Modeling* (DCM) für die Verwendung in Neurofeedbackexperimenten vor.

Diese kurz vorgestellte Vielfalt an Methoden für Echtzeitverarbeitung der Daten verdeutlicht die permanente Entwicklung in diesem Forschungsgebiet. Mit der steigenden Performanz von Rechnerleistung ergeben sich weitere Möglichkeiten für die Adaption vorhandener Techniken und Methoden für die Echtzeit-Auswertung.

2.3.2.2 Literaturübersicht zu Anwendungsfeldern für Echtzeit-fMRT

Die Anwendungsfelder für die Methodik der Echtzeit-fMRT werden in diesem Abschnitt anhand veröffentlichter Studien erläutert.

Qualitätssicherung: Mit Hilfe von Echtzeit-fMRT kann das erfasste MR-Signal auf Artefakte geprüft werden. Artefakte in den Aufnahmen können durch Störungen im Magnetfeld entstehen, z. B. wenn elektrische Kabel im Scannerraum das Signal verzerren oder die Probanden während der Messung durch starke Kopfbewegungen die Datenauswertung unbrauchbar machen. Der Einsatz von Echtzeit-fMRT zur Qualitätssicherung bei der Durchführung von MR-Messungen ist ein pragmatischer Anwendungsfall durch den Kosten und Zeit eingespart werden können wenn technische Probleme für weitere Messungen direkt behoben oder artefaktbehaftete Messungen wiederholt werden können. Ein technischer Aufbau dazu wurde erstmalig von Weiskopf et al. (2007) beschrieben.

Funktionelle Lokalisierung: Der Begriff ‘funktionelle Lokalisierung’ bezeichnet die Definition eines Hirnareals anhand der Aktivierung bei Einsatz eines bestimmten Paradigmas, das heißt wenn ein Hirnareal zu einer bestimmten Funktion zugeordnet werden kann. Die Lokalisierung

von Hirnarealen in Echtzeit ist aus zwei Gründen notwendig. Erstens ist sie erforderlich für die Auswahl von Regionen für fMRT-Neurofeedbackexperimente, die im folgenden Abschnitt detailliert erläutert werden. Zweitens stellt sie einen hilfreichen Ansatz für die Neurochirurgie dar. Hier kann die funktionelle Zuordnung für die präoperative Planung und intraoperative Kontrolle verwendet werden, wie von Feigl et al. (2008) vorgestellt. In dieser Studie wurden die Operationen von zehn Hirntumorpatienten anhand der Detektion des funktionell lokalisierten primären motorischen Kortex durchgeführt, um Verletzungen dieser Areale während der Operation zu verhindern.

Steuerung externer Systeme: Für die Verwendung eines fMRT-BCI-Systems zur Steuerung eines externen Computers oder einer Hardwarekomponente müssen reproduzierbare Hirnaktivitäten erzeugt und in einen Steuerbefehl übersetzt werden. Erstmals wurde von Yoo und Kollegen eine Machbarkeitsstudie publiziert, bei der drei Probanden durch Ausführen verschiedener mentaler Strategien (Rechnen, Lesen, Vorstellung von Bewegung) einen Cursor durch ein Labyrinth navigierten (Yoo et al. 2004). In einer späteren Arbeit derselben Arbeitsgruppe wurde basierend auf diesem Prinzip ein Robotorarm gesteuert (Lee et al. 2009). Ein Einsatz ist in der Neurorobotik bzw. Neuroprothetik durchaus denkbar. Die korrekte Dekodierung und damit potentielle BCI-Basis einer visuell-räumlichen Aufmerksamkeitsverschiebung wurde von Andersson et al. (2012) publiziert. Die Studie unter Einsatz eines 7T-MR Scanners wurde mit neun Probanden durchgeführt, die ohne die Augen bewusst zu bewegen ihre Aufmerksamkeit in verschiedene Richtungen lenkten. Die Richtungsänderungen konnten mit 80 %-iger Wahrscheinlichkeit korrekt dekodiert werden.

Eine große Motivation zur Entwicklung von EEG-BCI-Systeme ist die Möglichkeit zur Kommunikation mit Patienten, welche am Locked-in-Syndrom leiden (Birbaumer et al. 1999). Zwischenzeitlich gibt es kommerziell erhältliche Systeme für Patienten, bei denen aus den EEG-Aktivitäten einzelne Buchstaben decodiert werden, die auf einem Bildschirm angezeigt werden. Diese Patienten können so wieder mit ihrer Umwelt zu kommunizieren. Allerdings kann dieser Ansatz nicht bei allen Patienten erfolgreich eingesetzt werden. Daher wurde die Idee auf ein fMRT-basiertes System ausgeweitet und von Sorger et al. (2012) umgesetzt. Die Studie zeigt die Funktionsweise eines fMRT-basierten Systems mit sechs gesunden Probanden. Alle Buchstaben des Alphabets sind dabei durch verschiedene mentale Strategien und eine unterschiedliche Länge der Ausführung dieser repräsentiert. Bei den Probanden hat die Zuordnung zum korrekten Buchstaben in 82 % der Fälle im ersten Versuch funktioniert. Diese experimentelle Vorgehensweise hat das Potential die Kommunikation mit diesen Patienten zu ermöglichen.

Für die Vorhersage von Entscheidungen in einem sozioökonomischen Spiel (Ultimatum Game) mit Hilfe von Echtzeit-fMRT wurde von Hollmann et al. (2011) eine Studie mit zehn Probanden vorgestellt. Dabei mussten die Probanden ein Geldangebot eines virtuellen Mitspielers annehmen oder ablehnen. Diese Entscheidung konnte nach dem Training einer RVM bereits vor der

physischen Antwort des Probanden mit 70 %-iger Wahrscheinlichkeit vorhergesagt werden. Dazu wurden Aktivierungen in Hirnregionen die für kognitive Kontrolle, Belohnung und Emotion verwendet. Mit dieser Art von Experimenten sind viele weitere Paradigmen mit sozialen Interaktionen denkbar, die mit solchen Informationen verbessert werden können.

Weiterhin wurde von Mueller et al. (2012) eine Softwareumgebung zur Erstellung von virtuellen Realitäten, die mit Hilfe von Echtzeit-fMRT gesteuert werden können, vorgestellt.

Ein ausführlicher Übersichtsartikel zu den Möglichkeiten und Herausforderungen bei der Dekodierung von Hirnzuständen mit Echtzeit-fMRT wurde von LaConte (2010) publiziert. Des Weiteren zeigte LaConte et al. (2007), dass es möglich ist, durch die Klassifikation von Hirnzuständen mit Hilfe einer SVM einen Cursor zu steuern. Die Studien zeigen ein breites Spektrum an Interaktionsmöglichkeiten zwischen dem menschlichen Gehirn und einer externen Recheneinheit, die mit Echtzeit-fMRT realisiert werden kann. Eine Zusammenfassung findet sich in Abbildung 2.10. Die Abbildung zeigt die Richtungskodierung durch visuelle Aufmerksamkeitsverschiebung (b1, Andersson et al. (2012)); die Richtungskodierung durch verschiedene mentale Strategien (b2, LaConte et al. (2007)); die Navigation durch eine virtuelle Realität (b3, Mueller et al. (2012)); die Steuerung eines Roboterarms (b4, Lee et al. (2009)); die Dekodierung von Buchstaben durch verschiedene mentale Strategien (b5, Sorger et al. (2012)) und die Navigation durch ein Labyrinth (b6, Yoo et al. (2004)).

Neurofeedback: In der Literatur werden bereits Neurofeedbackexperimenten im Kontext unterschiedlicher kognitiver Funktionen und auch Krankheiten beschrieben, die mit gesunden Probanden und Patienten durchgeführt wurden. Neurofeedback ist damit der häufigste Anwendungsfall von Echtzeit-fMRT und wurde daher von Weiskopf (2012) als die treibende Kraft zur Entwicklung von Methoden für Echtzeit-fMRT bezeichnet. Wie in Abschnitt 2.3.1 beschrieben werden dabei die Ergebnisse der Echtzeitauswertung als Feedback in einer gewählten Form an den Probanden zurück projiziert, wodurch eine operante Konditionierung stattfindet, die sich auf das zukünftige Verhalten auswirken kann. Dies eröffnet neue Möglichkeiten für neurowissenschaftliche Experimente sowie therapeutische Ansätze bei verschiedenen Erkrankungen. Eine Übersicht über die aktuelle Literatur wird in den folgenden Abschnitten vorgestellt.

2.3.3 Literaturübersicht zu fMRT-Neurofeedback als Applikation von Echtzeit-fMRT

Werden fMRT-Daten mit einer der in Abschnitt 2.3.2 vorgestellten Methoden während einer laufenden Messung ausgewertet, kann das Ergebnis als Feedback an den Probanden zurück projiziert werden. Für die Erzeugung und Präsentation des Feedbacks finden sich in der Literatur verschiedene Ansätze.

Generierung des Feedbacksignals: Im einfachsten Fall wird als Feedback das gemittelte Signal eines relevanten Hirnareals (*Region-of-Interest*, ROI) verwendet. Diese Vorgehensweise wird aktuell in den meisten Studien verwendet. Allerdings kann das Feedback auch aus einer Kombination von Hirnregionen ermittelt werden. Dazu müssen die jeweils ermittelte Aktivitätswerte mathematisch verknüpft (addiert oder subtrahiert) werden entsprechend einer Hypothese, die den Zusammenhang zwischen den Regionen beschreibt. Es wird angenommen, dass diese Vorgehensweise zu spezifischeren Ergebnissen führt, allerdings ist dazu noch weitere Forschungsarbeit notwendig. Alternativ zu dem Region of interest (ROI)-basierten Ansatz, kann das Feedback aus der funktionellen oder effektiven Konnektivität zwischen verschiedenen Hirnregionen ermittelt (Lee et al. 2012; Koush et al. 2013) oder aus klassifizierten Zuständen generiert werden.

Darstellung von Feedback: Ein ermitteltes Feedbacksignal kann mit Hilfe einer visuellen, auditiven oder taktilen Präsentation an den Probanden zurückgegeben werden. Dabei ist die visuelle Darstellung die am häufigsten verwendete Variante. Einzig bei Posse et al. (2003) findet sich ein auditives Feedback. Die Visualisierung des Feedbacks ist die am wenigsten störanfällige Methode in einem MR-Umfeld, da sie keine Magnetfeldstörungen durch zusätzliche Kabel mit sich bringt oder durch die vorhandenen Geräusche durch die MR-Gradienten beeinflusst wird. Nachteilig ist, dass die Probanden sich durch das Beobachten des Feedbacksignals abgelenkt fühlen, da das Feedback die Aufmerksamkeit der Probanden in Anspruch nimmt. Für die Präsentation eines visuellen Feedbacks sind zahlreiche Varianten möglich. Einige Beispiele aus bisher veröffentlichten Studien finden sich in Abbildung 2.10. Das Feedback wurde bisher bereits visualisiert als: ein Feuer das mit geringer werdender Aktivität in der Zielregion kleiner wurde (a1, DeCharms et al. (2005)), ein Signalverlauf in Kombination mit der farblich kodierten Anweisung (a2, Weiskopf et al. (2007)), Bargraph bzw. Thermometer (a3, Sitaram (2007)), Geldgewinn (a4, Bray et al. (2007)), Veränderung des Gesichtsausdrucks eines virtuellen Charakters (a5, Mathiak et al. (2010)), Schieberegler bei gleichzeitiger Präsentation von stimulierenden Bildern (a6, LaConte (2010)) und Balkendiagramm, wobei ein Balken sich bewegt und ein zweiter die Zielvorgabe definiert (a7, Zotev et al. (2011)).

Kontinuierliches oder periodisches Feedback: Ein Feedbacksignal kann kontinuierlich präsentiert werden, wobei es direkt nach der Berechnung, d. h. meist innerhalb der nächsten TR-Zeit direkt an den Probanden weiter gegeben wird. Alternativ dazu wurde vor kurzem eine periodische Präsentation beschrieben (Johnson et al. 2012). Dabei wurde am Ende des Zeitraumes, in dem die Aktivität der Zielregion erhöht werden sollte, ein Feedback über den gesamten Zeitraum gegeben. Die Autoren bezeichnen dies als *intermittent Feedback* und beschreiben den Vorteil dieser Variante, dass Probanden sich weniger abgelenkt fühlen als bei einer kontinuierlichen Feedbackpräsentation. Auch eigene Erfahrungen mit Neurofeedback bestätigen diese These, erfordern jedoch noch weitere systematische Analysen.

Strategie zur Aktivitätsregulierung: Für das Lernen der willentlichen Beeinflussung der Hirnaktivität muss eine Strategie gefunden werden, mit der sich die aktuelle ROI beeinflussen lässt. Ein einfaches Beispiel ist die Vorstellung von Bewegung zur aktiven Kontrolle von Hirnarealen im Motorkortex. Bei der Durchführung von Neurofeedbackexperimenten muss jeweils entschieden werden, ob die Probanden eine individuell funktionierende Strategie ohne Vorgabe finden müssen oder ob eine konkrete Strategie oder zumindest eine begrenzte Auswahl vorgegeben wird. Ein Beispiel ist die Vorgabe an die Probanden, dass diese sich an positive Ereignisse erinnern sollen. Wenn ein Proband die Strategie zur Regulation während des Neurofeedback frei wählen kann, entspricht dies der individuellsten Vorgehensweise für den Probanden. Damit werden die größtmöglichen Signaländerungen, das heißt hier Effekte für die Regulation der Hirnregion, erwartet. Allerdings müssen die Probanden dann ausreichend Zeit bekommen, um eine geeignete Strategie zu finden. Dies ist bei MR-Experimenten nicht immer möglich und kann außerdem zu einem heterogenen Verhalten bei den Probanden führen. Die Vergleichbarkeit der Ergebnisse wird dadurch erschwert. Mit einer fest vorgeschriebenen Strategie hingegen ist die Vergleichbarkeit gegeben, jedoch können eventuell nicht alle Probanden die gewünschten Regulationseffekte erzielen. Am obigen Beispiel heißt das, die Strategie zur Vorstellung einer Bewegung sollte z. B. nicht 'Tennis spielen' sondern 'Lieblingssport' sein, damit die zu erwartende Hirnaktivität maximiert wird. Bisher gibt es keine Standarddefinitionen für die Auswahl von Strategien.

Kontrollgruppen: Eine noch unzureichend beantwortete Frage ist die korrekte Auswahl einer Kontrollgruppe zur Quantifizierung des Effekts eines Neurofeedbacktrainings. In einem aktuellen Vortrag während der '1st Swiss rtfMRI Neurofeedback Conference'⁹ wurde von Christopher deCharms erläutert, dass eine Kontrolle erforderlich ist, welche die Lern- und Habituationseffekte quantifiziert. Er betont, dass die durch Neurofeedback erzielten Signaländerungen in den ROIs sind mit denen zu vergleichen sind, welche ausschließlich durch das Ausführen der gewählten Strategie (Aufgabe) erzielt werden. Ein Neurofeedbacktraining wird nur dann als sinnvoll erachtet, wenn diese sich signifikant voneinander unterscheiden (Sulzer et al. 2013).

Die übliche Vorgehensweise ist, dass eine zusätzliche Kontrollgruppe untersucht wird, die ebenfalls ein Feedbacksignal projiziert bekommt, welches jedoch nicht aus der definierten Hirnregion erzeugt wurde. Es können dazu entweder andere, unabhängige Hirnregionen oder Zufallswerte verwendet werden. Diese Vorgehensweise wird als 'Sham-Feedback' bezeichnet und die Erwartung ist, dass Probanden einer solchen Kontrollgruppe keine signifikante Beeinflussung der Hirnregion erlernen. Nachteilig ist jedoch, dass diese Art des Feedbacks auch andere Reaktionen, z. B. Frust durch den vermeintlichen Misserfolg, bei den Probanden auslösen kann, wodurch die Durchführung der eigentlichen Aufgabe beeinträchtigt wird.

Aktuelle Diskussion: Die aktuelle Diskussion in diesem Forschungsbereich ist in der Arbeit von Sulzer et al. (2013) zusammengefasst. Insbesondere wird der Generierung des Feedbacks

⁹ <http://www.relab.ethz.ch/rtfMRI2012>

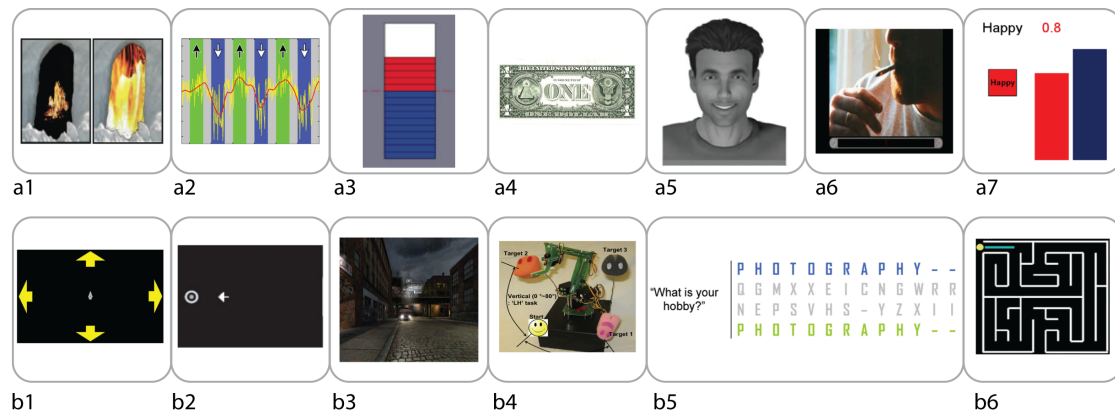


Abbildung 2.10: Für Neurofeedback wurden in verschiedenen Studien bereits unterschiedliche Visualisierungen verwendet (a): das Bild eines Feuers, das gelöscht werden musste (a1); der Signalverlauf (a2); als Bargraph bzw. Thermometer (a3); als Geldgewinn (a4); als Veränderung im Gesichtsausdruck eines virtuellen Charakters (a5); als Schieberegler zusammen mit Stimuli (a6); als Balkendiagramm (a7). Verschiedene Arten von BCI-Systemen, die mit Hilfe von Echtzeit-fMRT gesteuert werden (b): Richtungskodierung durch visuelle Aufmerksamkeitsverschiebung (b1); Richtungskodierung durch mentale Strategien (b2); Navigation durch eine virtuelle Realität (b3); Steuerung eines Roboterarms (b4); Dekodierung von Buchstaben (b5); Navigation durch ein Labyrinth (b6).

eine wesentliche Bedeutung für den Erfolg von Neurofeedback zugeordnet. Dabei liegt das Augenmerk auf der Berechnung des Feedback aus komplexeren Hirnzuständen, z. B. durch Konnektivitätsanalysen. In diesem Bereich ist weitere Forschungsarbeit notwendig, um ein besseres Verständnis des Einflusses von Neurofeedback und um Erkenntnisse zu relevanten Anwendungsmöglichkeiten zu erlangen. In der vorliegenden Arbeit wird dies berücksichtigt durch die Konzepte, die eine flexible Auswahl der Datenverarbeitung ermöglichen.

Die beiden folgenden Abschnitte fassen bisher publizierte Studien zum fMRT-Neurofeedback zusammen. Für eine bessere Übersicht werden diese kategorisch nach Funktionalität im Gehirn bzw. bei den Patientenstudien nach Erkrankung zusammengefasst.

2.3.3.1 Neurofeedbackstudien mit gesunden Probanden

Motorik: Erste Studien, die vor allem das Prinzip von fMRT-Neurofeedback zeigen sollten, wurden mit Arealen des Motor- und Somatosensorischen Kortex durchgeführt, da deren Funktionalität sehr gut untersucht sind. Aufgrund ihrer anatomischen Größe zeigen diese Areale sehr robuste Aktivierungen in fMRT-Studien.

Die erste Studie wurde von Yoo et al. (2002) vorgestellt, bei der fünf Probanden mit fMRT-Neurofeedback gelernt haben, ihre eigene Hirnaktivität willentlich zu beeinflussen. Auf diesem ersten Ansatz aufbauend stellte die Arbeitsgruppe eine Studie vor, in welcher die Probanden

gelernt haben, einen Roboterarm zu steuern. Das Anwendungsfeld ist hierbei die nicht-invasive Neuroprothetik wobei die Probanden durch Neurofeedback Befehle gelernt haben und durch Zustandsklassifikation die Steuerung des Roboterarms ausführen (Lee et al. 2009). Weitere Studien mit Arealen des Motorkortex, die als Machbarkeitsstudien zu werten sind, wurden von Weiskopf et al. (2003) und DeCharms et al. (2004) vorgestellt. Eine ausführlichere Studie wurde von Bray et al. (2007) publiziert. In dieser lernen Probanden mit Neurofeedback nach einem Signal entweder ein Finger- oder Zehenareal besser zu aktivieren als eine Kontrollgruppe. Das Besondere dieser Studie ist die Idee eines Belohnungsfeedbacks (Geldgewinn) bei Überschreiten eines Schwellwertes für die erzielte Signaländerung während des Trainings. Als Verhaltenseffekt wurde ermittelt, dass Probanden im Anschluss an das Neurofeedback eine längere Reaktionszeit benötigten, wenn diese auf die zuvor gelernten Signale invers reagieren sollten.

Sprachverarbeitung: Ein für die Sprachverarbeitung wichtiges Areal ist der rechte inferiore frontale Gyrus (IFG). In einer Studie aus dem Jahr 2009 zeigen Rota und Kollegen, dass Probanden nach einem Neurofeedbacktraining eine emotionale Betonung in Sätzen signifikant öfter richtig erkennen können. In einer Folgestudie wird weiterhin gezeigt, dass durch das Training auch die Konnektivität zum rechten IFG verändert wird. Die Konnektivität zum präzentralen Gyrus hat sich verstärkt, während die Konnektivität zu anderen, weiter verteilten Regionen, vermindert ist (Rota et al. 2009; Rota et al. 2011).

Hörvermögen: Für den primären auditorischen Kortex, Planum temporale, Putamen und superiores frontaler Gyrus wird von Yoo et al. (2006) beschrieben, dass Probanden gelernt haben, diese Areale mit Neurofeedback willentlich zu beeinflussen.

Emotionsverarbeitung und -kontrolle: An der Verarbeitung und Kontrolle von Emotionen sind eine Vielzahl von Regionen beteiligt, deren einzelne Aufgaben noch nicht endgültig erforscht sind. Dazu gehören u. a. die anteriore Insula, der IFG, der Precuneus, der orbitofrontale Kortex (OFC) und vor allem die Amygdala, eine subkortikale Struktur im limbischen System. Diese Hirnareale können mittels fMRT im Vergleich mit den genannten Modalitäten für BCIs mit der höchsten Ortsauflösung lokalisiert werden. Deshalb ist das fMRT-Neurofeedback im Bereich von Angst- und Emotionsstörungen ein erfolgversprechender Ansatz. Bereits in den ersten Arbeiten zu fMRT-Neurofeedback wurde die Möglichkeit untersucht, ob Probanden lernen können, die Aktivität der Amygdala willentlich zu beeinflussen. Von Phan et al. (2004) wurde die Machbarkeit der Echtzeit-Auswertung dieser subkortikalen Struktur bewiesen und in einer umfassenden Studie von Posse et al. (2003) beschrieben. Bei diesem Feedbackexperiment wurden Probanden instruiert, sich selbst in eine traurige Stimmung zu versetzen. Dabei bekamen diese per Kopfhörer nach jedem Durchlauf ein auditorisches Feedback vermittelt. Die Ergebnisse zeigen, dass Probanden prinzipiell lernen können diese Region zu beeinflussen. Eine weitere Region, die im Zusammenhang mit der Verarbeitung von Emotionen beschrieben wird, ist die anteriore Insula. Diese wurde als ROI von Caria et al. (2007) für ein Neurofeedbackexpe-

riment ausgewählt und reguliert, jedoch ohne Verhaltenseffekte des Neurofeedbacktrainings zu beschreiben. Eine sehr spezifische Region wurde des Weiteren von Hamilton et al. (2011) untersucht. Das Feedback wurde auf den subgenualen anterioren cingulären Kortex (sACC) beschränkt und die Aktivität dieses Hirnareal spezifisch mit positiven und negativen Gedanken hoch- bzw. herunterreguliert. Anschließend konnte für die Versuchsgruppe eine höhere Konnektivität der Region zum posterioren cingulären Kortex ermittelt werden als für die Kontrollgruppe. Die Regulation des sACC wird hier als Basis für Neurofeedback bei verschiedenen Arten von Depression vorgeschlagen. Da die Emotionsverarbeitung nicht auf eine einzelne Hirnregion beschränkt ist, wurde von Johnston et al. (2010) eine Studie mit verschiedenen Regionen des emotionalen Netzwerkes durchgeführt. Dazu gehörten u.a. der ventrolate präfrontale Kortex (VLPFC), die Insula und die Amygdala. Die Besonderheit der Studie ist, dass die Probanden keine vorgegebene Strategie zur Aktivitätskontrolle hatten. Diese Arbeit zeigt somit die Bedeutung einer individuellen Strategie, wobei dies im Kontext von Emotionen vor allem die Erinnerungen an individuelle, emotionale Ereignisse sind. Methodisch deutlich komplexer ist die von Sitaram et al. (2011) vorgestellte Studie zur Dekodierung von emotionalen Zuständen in 'glücklich', 'traurig' oder 'angeekelt' mit Hilfe einer SVM. Auch diese Studie basiert auf der Aussage, dass die Emotionsregulation ein Zusammenspiel aus mehreren Regionen ist und daher mit multivariaten Verfahren erfasst wurde. Den Probanden wurde visuell mitgeteilt in welchen emotionalen Zustand sie sich versetzen sollen. Mit einem erweiterten Ansatz für die SVM-basierte Klassifikation wurde aus der Genauigkeit des Klassifikators dann das Feedbacksignal bestimmt. Dieser Ansatz wird von Sitaram als Therapieansatz bei Störungen der Affektregulation, wie sie bei mehreren psychiatrischen Persönlichkeitsstörungen vorkommt, vorgeschlagen. Eine ausführliche Studie zum Neurofeedback basierend auf der Regulation der Amydalaaktivität wurde von Zotev et al. (2011) präsentiert. Dabei lernten die Probanden, die Aktivität mit Hilfe von positiven autobiographischen Erinnerungen zu erhöhen. Die Studie zeigt einen signifikanten Anstieg bereits nach drei Feedbackeinheiten im Vergleich zu einer Kontrollgruppe, die Feedback aus einer nicht emotionsbezogenen Hirnregion bekommen hatte.

2.3.3.2 Neurofeedbackstudien mit Patienten

Nachfolgend werden die Anwendungen von Neurofeedback im klinischen Einsatz mit Patienten vorgestellt.

Chronischer Schmerz: Die erste klinische Studie mit fMRT-Neurofeedback wurde von DeCharms et al. (2005) an Schmerzpatienten durchgeführt. Dazu wurde ein Feedbacksignal aus der Zielregion des rechten anterioren Cingulums (rACC) an die Patienten weiter gegeben. Die Aufgabe war, die neuronale Aktivität in dieser Region herunter zu regulieren. Das Feedback wurde mit einem Feuer visualisiert, das mit geringer werdender Aktivität im rACC 'gelöscht' wurde. Acht von zwölf Patienten berichteten im Anschluss an das Training, dass sie weniger

Schmerzen empfinden. Allerdings konnten diese Ergebnisse bis heute nicht repliziert werden. Die Versuche wurden auf mittlerweile 114 Patienten ausgeweitet wie in Hawkinson et al. (2011) berichtet wird. Allerdings kann chronischer Schmerz multiple Ursachen haben und mit vielen individuellen Symptomen einhergehen. Deshalb wurde von Chapin et al. (2012) vorgeschlagen, das Training auf Netzwerke von Hirnarealen und/oder Konnektivitäten zu erweitern und auch multivariate Ansätze für die Auswertung einzusetzen, um bessere Trainingseffekte zu erzielen.

Tinnitus: Eine klinische Studie mit sechs Tinnituspatienten wurde von Haller et al. (2010) vorgestellt, bei der Patienten gelernt haben, eine Region im primären auditorischen Kortex willentlich zu regulieren. Zwei von sechs Patienten zeigten eine signifikante Verbesserung der Symptome.

Parkinson: Subramanian et al. (2011) führten eine Neurofeedbackstudie mit zehn Parkinsonpatienten durch. Fünf dieser Patienten erhielten ein Feedback für die supplementär-motorische Rinde (SMA). Im Vergleich zur Kontrollgruppe konnten diese im Anschluss das Areal willentlich beeinflussen und zeigten auch in standardisierten Verhaltenstests für Parkinson eine bessere Leistung.

Schlaganfall: Eine aufwendig durchgeführte Studie mit zwei Schlaganfallpatienten (Läsionen in Capsula interna bzw. Corona radiata) und vier gesunden Probanden wurde von Sitaram et al. (2011) durchgeführt. Alle Teilnehmer bekamen ein dreitägiges Feedbacktraining, zusammen mit einem Videofeedback für den ventralen prämotorischen Kortex (PMv). Der Erfolg des Trainings wurde mit Hilfe der Erregbarkeit des primär motorischen Kortex (M1) durch transkraniale Magnetstimulation (TMS) getestet. Die Ergebnisse zeigen, dass die Schlaganfallpatienten nach Durchlaufen des Feedbacktrainings im Test deutlich besser abschnitten.

Depression: Zwei Patientenstudien mit fMRT-Neurofeedback wurden mit schwer depressiven Patienten durchgeführt. Beide Studien basieren auf dem Therapieansatz des Lernens von positiven Emotionen. Allerdings wurden unterschiedliche Hirnareale betrachtet. Bei Linden et al. (2012) wurde der ventrolaterale präfrontale Kortex ausgewählt, während bei Phillips et al. (2012) die Amygdala das Ziel des Therapieansatzes war. In beiden Studien haben die Patienten gelernt, das jeweilige Hirnareal zu regulieren. Bei Linden wurde der Effekt des Trainings mit Fragebögen erfasst, wodurch gezeigt werden konnte, dass sich die klinischen Symptome signifikant verbessert haben.

Diese Übersicht über eine Reihe von klinischen Studien zeigt, dass Neurofeedback mit nicht-invasiver Echtzeit-fMRT vielversprechende Therapieansätze eröffnet, methodisch jedoch weiter entwickelt werden muss, um die gewünschten Behandlungserfolge zu erzielen.

2.3.4 Grenzen von Echtzeit-fMRT und fMRT-Neurofeedback

Echtzeit-fMRT und Neurofeedback bieten neue Möglichkeiten in den Neurowissenschaften und neurologisch-psychiatrischer Therapie. Trotz rasant steigender Publikationszahlen in diesem Forschungsgebiet gibt es klare methodische Grenzen. Für fMRT-Neurofeedbackmessungen wird ein MR-Scanner benötigt. Diese Geräte sind in der Anschaffung und im Betrieb sehr kostenintensiv. Des Weiteren kann das Gerät nur stationär betrieben werden, d.h. Probanden und Patienten müssen zur Durchführung der Messungen eine Klinik oder Forschungsinstitut aufsuchen, welche über einen MR-Scanner verfügen. Die bisherigen Studien wurden bis auf wenige Ausnahmen mit sehr wenigen Probanden (< 10) und über einen kurzen Zeitraum (mehrere Durchläufe an einem Tag) durchgeführt. Aus diesen Ergebnissen lassen sich noch keine Schlussfolgerungen über Langzeiteffekte ableiten. Es ist bisher noch nicht erforscht, ob durch mehrtägiges oder sogar mehrwöchiges Training bessere Effekte möglich sind. Ein Schwachpunkt des fMRT-basierten Feedbacks ist die Verzögerung des Signalanstiegs um 4 s bis 6 s beim BOLD-Effekt. Eigene Versuche zeigen, dass Probanden sehr schnell lernen damit umzugehen. Für ein robustes Feedback ist es vorteilhaft, wenn Zustände des Gehirns abgebildet werden, die über mehrere Sekunden oder Minuten Bestand haben, d. h. die Probanden eine Strategie über diesen Zeitraum verfolgen um die Hirnaktivität zu beeinflussen. Eine zeitliche Auflösung wie beim EEG kann nicht realisiert werden. Weiterhin wird diese Methodik begrenzt durch den aktuellen Wissensstand über die Funktionsprinzipien im menschlichen Gehirn. An kognitiven Prozessen im Gehirn sind häufig eine Vielzahl an Hirnregionen mit unterschiedlichen Aktivitätsmustern beteiligt. Das Feedback aus einem Hirnareal kann daher nicht eine vollständige kognitive Funktion des Gehirns abbilden und beeinflussen. Welche Hirnregionen für ein individuelles Neurofeedbackexperiment geeignet sind, muss sorgfältig durch Studium publizierter Arbeiten sowie durch eigene Vorexperimente erarbeitet werden.

2.4 Adaptive Paradigmen

Die im Abschnitt 2.2.1 vorgestellten Paradigmen basieren alle darauf, dass die Zeitpunkte der Stimuluspräsentationen vor der Durchführung des Experiments festgelegt werden. Mit dieser Vorgehensweise soll sicher gestellt werden, dass allen Probanden eine identische Stimulation präsentiert wird. Allerdings ist damit nicht sicher gestellt, dass diese auch gleich wahrgenommen wird. Ein aufmerksames Verhalten der Probanden, bei einer visuellen Stimulation also die Hinwendung des Blicks auf die Stimuli, hat einen großen Einfluss auf die Ergebnisse (Pessoa et al. 2002b; Reynolds et al. 2004). Außerdem ist bekannt, dass im Gehirn erzeugte Signaländerungen nicht nur aufgabenspezifische Effekte sind. Sie sind zudem überlagert von physiologischen Fluktuationen, wie der Atmung oder dem Puls (Fox et al. 2007; Krüger et al. 2001; Leopold et al. 2012), die eine unterschiedliche Wahrnehmung und Verarbeitung der Stimulation

im Verlauf dieser Prozesse verursachen (Linkenkaer-Hansen et al. 2004). Obwohl dies bekannt ist, werden konventionelle fMRT-Experimente mit einem festen Stimulationsplan durchgeführt. Werden physiologische Parameter zur Laufzeit eines Experiments erfasst, können diese post-hoc bei der Analyse berücksichtigt werden, z. B. indem der Signalverlauf als Rauschen einbezogen wird oder Teile des Experiments aus der Analyse ausgeschlossen werden. Experimentelle Systeme, die die simultane Erfassung von physiologischen Parametern (Puls und Atmung) für fMRT-Experimente erlauben, wurden von Smyser et al. (2001) und Voyvodic (1999) vorgestellt. Dieser Ansatz hat jedoch den Nachteil, dass die mathematische Korrektur auch zu einem Signalverlust im Vergleich zur Originalmessung führt. Die Idee der adaptiven Paradigmen beschreibt dagegen einen a-priori Ansatz, der diese physiologischen Parameter bereits zur Laufzeit des Experiments berücksichtigt. Die Stimulation des Probanden wird zur Laufzeit des Experiments an einen Parameter, z. B. Herzrate, Atmung oder Augenbewegungen, gekoppelt und nach vorher definierten Kriterien adaptiert.

Ein weiterer Aspekt ist, dass konventionelle Experimente oft als künstliche Situation empfunden werden. Es wird angenommen, dass das Verhalten von Probanden nicht identisch zum Verhalten in einem realen Umfeld ist. Vor allem in interaktiven Situationen, in denen ein Verhalten von einem Gegenüber beeinflusst werden soll, ist dies eine Herausforderung für die Experimentgestaltung. Beispiele dafür finden sich in den sozialen Neurowissenschaften, die die Interaktionen zwischen Menschen untersuchen oder bei psychiatrischen Störungen wie Autismus, die durch ein gestörtes Sozialverhalten geprägt sind (Wilms et al. 2010).

2.4.1 Literaturübersicht zu adaptiven Paradigmen

Der Abschnitt erläutert die vorhandene Literatur zu diesem Thema und stellt abschließend die Möglichkeiten in der Kombination mit Echtzeit-fMRT vor.

Eine eingängige Idee für die Adaption einer Stimulation ist die Einbeziehung der Augenbewegungen des Probanden, besonders bei Experimenten mit sozialen Interaktionen. Dazu wurde von Wilms et al. (2010) ein experimenteller Aufbau für die Interaktion durch Blickkontakte mit einem virtuellen Experimentpartner entwickelt. Mit einem Eyetracker wurden die Augenbewegungen der Probanden erfasst, die Daten zur Laufzeit analysiert und mit den Ergebnissen die Augenbewegungen eines virtuellen Charakters (Avatars) adaptiert. Dies hat den Probanden eine reale Interaktion mit dem Charakter vermittelt, was sich in den Aktivierungsmustern der fMRT-Daten widerspiegelt.

Über diese externen Parameter hinaus ist es möglich, die Adaption direkt anhand der Hirnaktivität abzuleiten. In kürzlich publizierten Studien wurde dieses Prinzip realisiert. In der ersten Arbeit wurde von Becker et al. (2011) untersucht, ob spontane neuronale Oszillationen, die im EEG durch den α -Rhythmus erfasst wird, das fMRT-Antwortverhalten auf visuelle Reize verändert. Dazu wurden parallel EEG- und fMRT-Daten gemessen und gezeigt, dass die Aktivität in

visuellen Hirnarealen höher ist, wenn die Stimuli gleichzeitig mit einem hohen α -Zustand präsentiert werden. Dieser Zustand wird als erhöhte Aufmerksamkeit interpretiert. In der zweiten Studie von Yoo et al. (2012) wird diese Idee erweitert und das Paradigma anhand der fMRT-Aktivierung in einer Hirnregion angepasst. Es wurde ein Areal im parahippocampalen Kortex gewählt, das bei der Gedächtnisbildung und der Erinnerung an Objekte und Szenen eine Rolle spielt. In diesem Areal wurde die spontane Fluktuation gemessen und zeitgleich wurden Probanden Bilder von Objekten präsentiert, welche sie sich merken sollten. Dieselben Bilder wurden präsentiert, wenn das genannte Hirnareal in einem überdurchschnittlich aktivierten ('guten') oder deaktivierten ('schlechten') Zustand war. Die Studie ergab, dass die Probanden im Anschluss signifikant mehr Objekte wiedererkannten, die im 'guten' Zustand präsentiert worden waren. Soll ein Paradigma anhand der fMRT-Signale adaptiert werden, ist dazu die Datenauswertung in Echtzeit notwendig. Aus diesem Grund ist dieser Ansatz eng verbunden mit den bereits erläuterten Neurofeedbackexperimenten. Allerdings werden die Ergebnisse der Datenauswertung jeweils unterschiedlich eingesetzt. Beim Neurofeedback werden die Ergebnisse in ein Lernsignal übersetzt, wohingegen sie bei adaptiven Paradigmen einen Zustand beschreiben, aus dem sich die Präsentation der Stimuli ableitet.

2.4.2 Adaptive Paradigmen für die vorliegende Arbeit

In der vorliegenden Arbeit wird ein Softwarepaket vorgestellt, das ein Konzept für die flexible Umsetzung von adaptierten Paradigmen implementiert und eine abstrakte Beschreibung der Adaption und der externen Parameter einführt. Mit dem Softwaresystem wird es möglich sein, unterschiedliche adaptive Paradigmen zu realisieren. Für den experimentellen Teil der vorliegenden Arbeit wurde ein erstes adaptives Paradigma zur Validierung der Software realisiert. Des Weiteren werden die Vorteile adaptiver Paradigmen herausgearbeitet und experimentell belegt. Dazu wird ein Experiment mit emotionalen Bildern vorgestellt. Bei diesem experimentellen Ansatz ist es von besonderer Bedeutung, dass die Probanden den Inhalt der Bilder visuell korrekt wahrnehmen, um die erwarteten Aktivitätsänderungen in emotionsbezogenen Hirnarealen zu detektieren. Daher wurden die Augenbewegungen der Probanden während des Experiments gemessen und ausgewertet. Dadurch kann sichergestellt werden, dass die Probanden zum Zeitpunkt der gewünschten Stimuluspräsentation den Bildschirm fixierten. Nur wenn diese Bedingung erfüllt ist, wird ein Bild präsentiert. Andernfalls wird die Präsentation eines Bildes zeitlich verschoben, bis die Mitte des Bildschirms fixiert ist. Die Idee ist in Abbildung 2.11 veranschaulicht und wird im experimentellen Teil dieser Arbeit detailliert beschrieben.

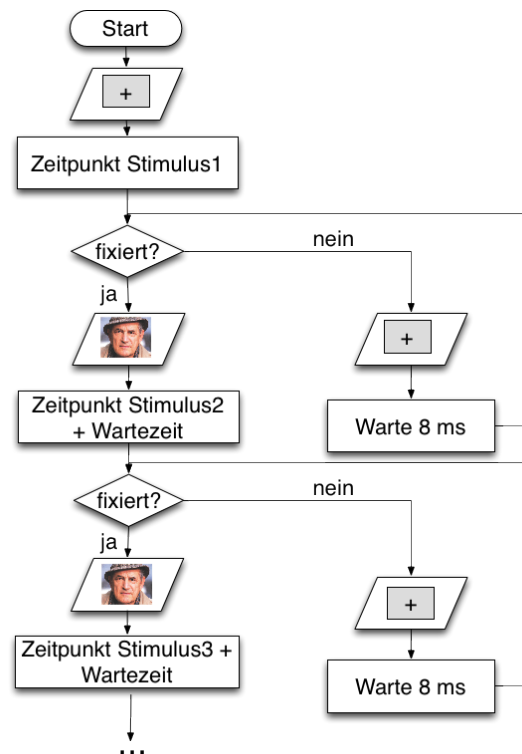


Abbildung 2.11: Zeitlicher Ablauf des in dieser Arbeit umgesetzten adaptiven Paradigmas: Im Gegensatz zu konventionellen Paradigmen wird der Zeitpunkt der Stimuluspräsentation an die Augenbewegung der Probanden gekoppelt: Jeweils zum Zeitpunkt der Stimuluspräsentation wird geprüft, ob die Probanden den Bildschirm fixieren, d. h. visuell aufmerksam sind. Nur dann wird das Bild präsentiert, andernfalls wird ein Fixationskreuz gezeigt und die Stimuluspräsentation solange verschoben bis die Probanden das Kreuz fixieren (hier 8 ms entsprechend der Frequenz des Eyetrackers).

2.4.3 Grenzen der Anwendung adaptiver Paradigmen

Die Möglichkeiten bei der Umsetzung adaptiver Paradigmen sind technisch begrenzt, da die Erfassung des externen Parameters in der MR-Umgebung möglich sein muss. Mittlerweile sind viele MR-taugliche Geräte für die Erfassung von u. a. EKG, Atmung und Augenbewegungen erhältlich. Der Aufbau eines gesamten Setup ist jedoch aufwendiger als in einem normalen Labor. Die Implementierung eines adaptiven Paradigmas ist auf die zuverlässige Arbeitsweise der einzelnen Komponenten angewiesen - sowohl der Auswertung der gewählten Variable als auch der Software zur Adaption der Stimulation. Die Interaktion muss zuverlässig und mit hoher Laufzeitstabilität erfolgen. Neben diesen technischen Aspekten ist es wichtig, eine starke a priori Hypothese zu haben über den Zusammenhang zwischen der neuronalen Aktivität und dem Parameter, der für die Adaption verwendet wird. Das ist von besonderer Bedeutung, wenn der Parameter für die Adaption direkt aus Hirnsignalen abgeleitet wird. Dabei muss streng darauf

geachtet werden, dass die Hirnsignale, die zur Adaption verwendet werden, unabhängig sind von denen, wo ein Effekt erwartet wird. Weiterhin sind für diese Technik auch neue Auswertemethoden notwendig, wie z. B. die Auswertung der BOLD-Effekt-Fluktuationen wie von Hinds et al. (2011) vorgestellt.

Mit den in diesem Kapitel vorgestellten Aspekten zur technischen und methodischen Entwicklung für Echtzeit-fMRT, Neurofeedback und der Idee der adaptiven Paradigmen wird im folgenden Kapitel die Entwicklung eines Softwarepakets für die Kombination der Ansätze und ihrer Weiterentwicklung vorgestellt.

3 Methoden und Ergebnisse: Softwarepaket 'Brain Analysis in Realtime' (BART)

Bisher wurde in den Abschnitten 2.3.2 und 2.4 die vorhandene Methodik und Techniken für Echtzeit-fMRT und adaptive Paradigmen vorgestellt. In diesem Kapitel werden die im Rahmen dieser Arbeit neu entwickelten Softwarekonzepte vorgestellt. Diese wurden im 'Brain Analysis in Real-Time' (BART) vollständig implementiert. Im Folgenden werden zuerst offene Fragestellungen des Forschungsfelds zusammengefasst und der Stand vorhandener Softwaresysteme für Echtzeit-fMRT beschrieben. Darauf basierend wird die Entwicklung des neuen Softwaresystems motiviert. Weiterhin wird die Anforderungsanalyse an die neue Software erstellt und die Entwicklerwerkzeuge vorgestellt. Darauf aufbauend werden das abstrakte Konzept zum flexiblen Laden der externen Komponenten, die Algorithmen zur Datenverarbeitung und deren Kombination detailliert erläutert.

3.1 Motivation

Die im vorangegangenen Kapitel zusammengefassten Arbeiten zu Methoden für Echtzeit-fMRT und die bisher publizierten Anwendungen zeigen die bereits vorhandene Vielfalt der Thematik von Echtzeit-fMRT. Allerdings hat dieser Überblick auch Fragen gezeigt, für deren Beantwortung noch weitere, umfangreichere Forschungsarbeit notwendig sein wird. Die offenen Fragen in Bezug auf die Methodik von Echtzeit-fMRT, die sich aus der aktuellen Literatur ergeben, können wie folgt zusammengefasst werden:

- Wie kann das technische Setup für potentielle Anwendungen im klinischen Alltag vereinfacht werden?
- Ist es vorteilhaft, Echtzeit-fMRT und Neurofeedback mit anderen Modalitäten, z. B. EEG oder physiologischen Parametern, zu kombinieren? Wie kann eine solche Kombination realisiert werden?
- Welche Algorithmen der univariaten und multivariaten Verfahren können für die Echtzeit-Datenauswertung verwendet werden?

- Wie können diese Algorithmen den Besonderheiten der Echtzeit-Auswertung besser gerecht werden?
- Mit welcher Art von Feedback (periodisch oder kontinuierlich) kann das Neurofeedback im konkreten Anwendungsfall optimal gestaltet werden?
- Kann die Analyse von Konnektivitäten (funktionelle und effektive Konnektivitäten) umgesetzt und für Feedback eingesetzt werden?
- Welche Darstellung des Neurofeedbacks ist für die individuelle Fragestellung am effizientesten?

Die Beantwortung dieser Fragestellungen ist nicht mit einem einzelnen Experiment möglich. Es ist eine Vielzahl an Fragen, die allerdings ausreichende Schnittmengen besitzen. Für eine systematische Erarbeitung von Lösungen ist es notwendig, ein flexibles Softwaresystem zu verwenden, mit dem sowohl bereits vorhandene und validierte Techniken als auch neu entwickelte Methoden für eine Fragestellung verwendet werden können. Zwei Beispiele zur Verdeutlichung: (1) Wenn der Einfluss von unterschiedlichen Arten des Feedbacks bei einem Experiment untersucht werden soll, dann muss diese Komponente flexibel austauschbar sein, ohne die Vorverarbeitung oder Analyse zu verändern. (2) Das gilt ebenfalls für die Entwicklung von neuen Methoden für die Auswertung der Daten in Echtzeit ohne die Beeinflussung anderer Komponenten.

Neben dieser Flexibilität ist das Ziel der vorliegenden Arbeit die Kombination der Echtzeitauswertung mit der Durchführung der im Abschnitt 2.4 vorgestellten adaptiven Paradigmen. Unter diesen Aspekten wird im folgenden Abschnitt der Stand von bereits vorhandenen Softwarepaketen diskutiert und abschließend die Anforderungen für das in dieser Arbeit neu entwickelte Softwarepaket vorgestellt.

3.1.1 Vorhandene Softwarepakete für Echtzeit-fMRT

Zum Zeitpunkt des Beginns der vorliegenden Arbeit existierten einige wenige Ansätze für Echtzeit-fMRT-Softwaresysteme, die in diesem Abschnitt kurz erläutert werden:

Turbo Brain Voyager¹⁰: Dies ist eine kommerziell erhältliche Software zur Durchführung von Neurofeedbackexperimenten, die in vielen der im Abschnitt 2.3.3 vorgestellten Studien verwendet wurde. Diese Software ist ein geschlossenes Gesamtpaket. Für die Anwendung in einem Neurofeedbackexperiment ist diese Software sehr gut geeignet, allerdings lässt sie keine eigene Methodenentwicklung zu. Das heißt, es können keine selbst implementierten Datenvorverarbeitungs- oder Analysealgorithmen in das Paket integriert werden. Zusätzlich beinhaltet dieses Paket keine Möglichkeit, adaptive Paradigmen mit Echtzeitauswertung umzu-

¹⁰ www.brainvoyager.com/TurboBrainVoyager.html

setzen. Dies widerspricht den Zielen der vorliegenden Arbeit, wodurch die Verwendung dieses Pakets nicht in Betracht gezogen wurde.

CIGAL und IOWA: In den Arbeiten von Voyvodic (1999) und Smyser et al. (2001) wurde jeweils ein umfangreiches Setup vorgestellt, das sowohl physiologische Parameter erfassen kann als auch jeweils einen Auswertalgorithmus für Echtzeit-fMRT umsetzt. Das IOWA-System von Smyser beschreibt eine Schnittstelle für die Erfassung der physiologischen Parameter und beinhaltet eine GLM-Echtzeit-Auswertung, verwendet allerdings sowohl ein selbst definiertes Datenformat als auch eine kommerzielle Entwicklungsumgebung und beschränkt das Setup auf MR-Scanner der Firma GE. Ähnliches gilt für das CIGAL-System von Voyvodic, das auf einer selbst entwickelten Skriptsprache basiert, die ausschließlich für die Arbeit verwendet wird. Der aktuelle Stand des Systems wurde in Voyvodic et al. (2011) vorgestellt. Diese Arbeit beschreibt die Verwendung des Systems ausschließlich zur Erfassung verschiedener physiologischer Parameter. Dies wurde allerdings auf Geräte des Herstellers *BIOPAC* beschränkt.

AFNI:¹¹ AFNI ist ein verbreitetes Programm für die Auswertung von konventionellen fMRT-Daten. Das System ist in der prozeduralen Programmiersprache C geschrieben, stellt eine Vielzahl an Methoden für fMRT-Datenverarbeitung zur Verfügung und der Quellcode ist frei verfügbar. Einer der Entwickler ist Robert Cox, der die Idee der Echtzeit-Auswertung von fMRT-Daten in der Arbeit Cox et al. (1995) vorgestellt hat. Dieses Paket enthält verschiedene rechenzeitperformante Algorithmen, z. B. für die Bewegungskorrektur. Außerdem bietet das Paket eine Routine für das Einlesen von Daten zur Laufzeit und eine Schnittstelle zur Skriptsprache Python, mit der sich Verarbeitungsabläufe für die fMRT-Daten einfach kombinieren lassen. Trotz dieser bereits vorhandenen Teile ist der Ansatz einer rein prozeduralen Vorgehensweise nachteilig im Vergleich zur objektorientierten Programmierung, die eine Wiederverwendung von Softwareteilen ermöglicht.

rtExplorer: Dieses Softwarepaket wurde von Maurice Hollmann entwickelt und in der Arbeit von Hollmann et al. (2011) verwendet. Der rtExplorer basiert auf der Entwicklungsumgebung MATLAB¹², die speziell für die Lösung mathematischer Probleme und zur vereinfachten graphischen Darstellung von Funktionen, Graphen und Daten entwickelt wurde. Besonders hervorzuheben ist die Unabhängigkeit vom Betriebssystem, da MATLAB auf allen gängigen Betriebssystemen lauffähig ist. Allerdings ist die Benutzung von rtExplorer an den Erwerb einer MATLAB-Lizenz gekoppelt. Das System bietet die Möglichkeit zur vollständigen Durchführung von Neurofeedbackexperimenten, einer Echtzeitklassifikation mit dem SVM-Ansatz und einen Plugin-Mechanismus¹³, der für die flexible Generierung und Gestaltung von Feedbacksignalen verwendet wird. Die Systemkonfiguration basiert auf der vorgestellten Beschreibungssprache EDL. Dieses Softwarepaket wird hier am Max-Planck-Institut für Kognitions- und Neu-

¹¹ <http://afni.nimh.nih.gov/afni>

¹² www.mathworks.de

¹³ Ein Plugin ist ein erst zur Laufzeit eingebundener Quellcode

rowissenschaften Leipzig für die ersten Neurofeedbackexperimente verwendet (zum jetzigen Zeitpunkt noch nicht publizierte Arbeiten). Dieses Paket existiert also parallel zu dem in dieser Arbeit entwickelten Softwarepaket BART. Damit war es neben der Softwareentwicklung bereits möglich, praktische Erfahrungen mit Neurofeedbackexperimenten zu sammeln. Die Entwicklung von BART wurde für die vorliegende Arbeit auf die Umsetzung von adaptiven Paradigmen konzentriert. Dazu wurde der Fokus auf eine integrierte Stimulationsumgebung gelegt, die beim rtExplorer nicht vorhanden ist. Das Paket rtExplorer wurde nicht für die Entwicklung von adaptiven Paradigmen weiterentwickelt. Dafür ist ein objektorientierter Ansatz vorteilhafter und die Programmierung mit MATLAB begrenzt das System in der Laufzeit- und Speicherperformanz für die Entwicklung von parallelisierten Algorithmen. Außerdem wurde für das Paket BART ein wesentlich flexiblerer Ansatz für das Einbinden von Plugins berücksichtigt und die Bedienung vereinfacht. Die Expertise, die mit der Anwendung des rtExplorer erworben wird, ist von großer Bedeutung für die Entwicklung des neuen Softwarepakets.

Eigenschaft	TBV	AFNI	IOWA und CIGAL	rtExplorer	BART
Echtzeitanalyse	✓	✓	(✓)	✓	✓
Neue Methoden integrierbar	-	✓	-	✓	✓
Ein-Computer-Setup	(✓)	-	-	(✓)	✓
Integrierte Stimulationsumgebung	-	-	-	-	✓
Physiologische Parameter erfassbar	-	-	✓	-	✓
Adaptive Paradigmen	-	-	-	-	✓
Open Source verfügbar	-	✓	-	-	✓

Tabelle 3.1: Eigenschaften vorhandener Softwaresysteme: - nicht vorhanden, ✓ vorhanden, (✓) je nach Anwendung bzw. Eigenschaft teilweise erfüllt.

Zusammenfassung: Die vorgestellten Systeme boten somit keine ausreichende Basis für eine Weiterentwicklung eines vorhandenen Systems. Insbesondere für die Implementierung einer flexiblen Lösung für adaptive Paradigmen ist die Entwicklung einer integrierten Stimulationsumgebung essentiell. Der dazu erforderliche Aufwand und bedingte Grad der Integration in das Gesamtsystem erfordert die Entwicklung eines eigenständigen Softwarepakets.

3.1.2 Grundanforderungen an das neue Softwaresystem BART

Allgemeine Anforderungen: Die Entwicklung des Pakets BART basiert auf der Intention, eine frei verfügbare, flexible Software zur Verfügung zu stellen, die zur vollständigen Durchführung von Echtzeit-fMRT-Experimenten genutzt werden kann. Dies soll mit der Möglichkeit für den Einsatz eines flexiblen Methodenspektrums gekoppelt werden. Daraus ergibt sich die allgemeine Anforderung, ein Setup anzubieten, das trotz der gegebenen Komplexität überschaubar bleibt. Damit wird die Bedienbarkeit verbessert und die Fehleranfälligkeit minimiert. Besonders hervorzuheben sind weiterhin die Anforderungen an die Stabilität, die Effizienz und die Flexibilität: Das System muss eine besondere Laufzeitstabilität aufweisen, da jeder Fehler zur Laufzeit des Systems zum Abbruch des gesamten Experiments führt und damit zeit- und kostenintensiv ist. Weiterhin muss das System in besonderem Maße rechenzeit- und speichereffizient arbeiten um die impliziten Zeitlimitierungen, sowohl für die (adaptive) Stimulation als auch die Datenverarbeitung, zu erfüllen. Neben diesen Aspekten ist die Möglichkeit für eine flexible Anwendung gefordert. Jede der zukünftigen Anwendungen der Software in Studien wird individuelle Anforderungen stellen, die Änderungen und/oder Erweiterungen der Software nötig machen. Daher muss die Software BART robust und vor allem abstrakt genug aufgebaut sein, um diesen Anforderungen gerecht werden zu können. Neben Abstraktionen ist die Modularisierung innerhalb des Gesamtpakets wichtig, um neue Methoden für einzelne Teilaufgaben schnell und einfach implementieren zu können.

Flexibles Setup: Für einen flexiblen Einsatz muss die Software abstrakt bezüglich der verwendeten MR-Scanner Hard- und Software sein. Die in dieser Arbeit vorgestellte Implementierung ist für MR-Geräte der Firma Siemens konzipiert. Zusätzlich sollen alle gängigen Datenformate für fMRT-Daten unterstützt werden.

Flexible Echtzeit-fMRT-Datenverarbeitung: Besonders für die Datenvorverarbeitung und -analyse der fMRT-Daten ist Flexibilität gefordert. Die erste konkrete Implementierung beinhaltet als Basisfunktionalität eine univariate Auswertung mit einer GLM-Analyse. Allerdings muss dies einfach erweiterbar sein um weitere Auswertelgorithmen zu integrieren. Analog dazu soll der Rahmen für die Einbindung von Vorverarbeitungsalgorithmen aufgebaut werden.

Flexible adaptive Paradigmen: Die Idee ein Framework für flexible adaptive Paradigmen zu erstellen, erfordert eine integrierte Stimulationsumgebung im Gesamtpaket. Sie ist einerseits mit besonderen Laufzeitanforderungen für die zeitlich korrekte Stimulation verbunden. Andererseits wird dieses hier erweitert mit der Anforderung, die Stimulation durch flexibel austauschbare, externe Parameter adaptieren zu können. Dazu ist es notwendig, sowohl die möglichen Adaptionen im Paradigma als auch die externen Parameter (Geräte) abstrakt zu beschreiben. In der ersten konkreten Implementierung wird dies mit einem Eyetracker zur Bestimmung der Blickrichtung des Probanden erreicht und das Paradigma entsprechend der Bildpräsentationszeit angepasst.

Kombination Echtzeit-fMRT und adaptive Paradigmen: Das Softwarepaket wird die beiden Konzepte, adaptives Paradigma und Echtzeit-Auswertung der fMRT-Daten, verknüpfen. Das beinhaltet die Übertragung der zur Laufzeit angepassten Stimulation in die für die Auswertung notwendigen Informationen.

Zur Erfüllung der genannten Anforderungen wurde ein Softwarepaket konzipiert und implementiert mit Hilfe der im folgenden Abschnitt erläuterten Werkzeuge und Konzepte. Im Anschluss wird die Funktionsweise der einzelnen Module des Softwarepaketes beschrieben.

3.2 Methoden zur Softwareentwicklung von BART

Dieser Abschnitt beschreibt die für die Entwicklung des Softwarepaketes BART verwendeten Technologien und konkreten Werkzeuge und begründet deren Auswahl.

3.2.1 Objektorientierte Programmierung

Objektorientierung ist ein Konzept der Softwaretechnik, das sich in den letzten zwei Jahrzehnten zur Beherrschung komplexer Softwaresysteme bewährt hat (Lahres et al. 2009, S. 13). Dieses Konzept enthält viele Prinzipien und Mechanismen, mit denen zahlreiche Aufgabenstellungen angegangen werden können. Die Varianten und Details sind in einer Vielzahl von Büchern ausgeführt. Die folgenden Ausführungen beruhen auf Lahres et al. (2009).

Das allgemeine Ziel des Einsatzes von Objektorientierung ist es, eine Software einfacher testbar, erweiterbar und wartbar zu machen. Das wird nicht allein durch die Anwendung der Technik erreicht, bietet aber hilfreiche Ansätze, die auf drei Grundkonzepten basieren:

- **Datenkapselung:** Die Kapselung von Daten in der Objektorientierung beruht darauf, dass es keine Trennung von Daten und deren Verarbeitung gibt, sondern beides in Objekten zusammengefasst wird. Diese Objekte definieren eine Schnittstelle von Operationen zum Zugriff auf die enthaltenen Daten und haben die Aufgabe, die Konsistenz der Daten sicher zu stellen. Ein alltägliches Beispiel zur Verständlichkeit ist ein Objekt, das ein 'Auto' beschreibt, welches immer vier Räder haben muss. Weiterhin kann das Objekt die Operation zum 'Tauschen der Räder' besitzen, die beim Ausführen der Methode¹⁴ immer alle vier Räder tauscht. Vorteil der Datenkapselung ist die damit garantierte Konsistenz des Objekts, die auch zur Korrektheit der Software notwendig ist. Ebenso wird damit der Aufwand bei Änderungen reduziert: Wenn sich interne Beschreibungen des Objektes ändern, z. B. die Anzahl der Räder, hat das keine Auswirkungen auf andere Objekte, die die Operation zum 'Tauschen aller Räder' ausführen wollen.

¹⁴ Eine Methode ist die konkrete Umsetzung einer definierten Operation, die nur für das aktuelle Objekt zuständig ist.

- **Polymorphie und Vererbung:** Polymorphie (wörtlich ‘Vielgestaltigkeit’) bezieht sich darauf, dass sich verschiedene Objekte beim Aufruf derselben Operation unterschiedlich verhalten können. Ein Beispiel zur Veranschaulichung ist eine Steckdose, an die verschiedene Elektrogeräte angeschlossen werden können. Die Geräte erfüllen alle eine andere Aufgabe, wie z. B. Wasser erhitzen oder ein Telefon laden, benutzen aber alle dieselbe Steckdose, um mit Strom versorgt zu werden. Hierbei sind Stecker und Steckdose die Schnittstelle, die für alle Geräte identisch ist. Dies wird in der Softwarearchitektur als Vererbung einer Spezifikation bezeichnet. Andererseits kann aber auch eine konkrete Umsetzung von Funktionalität vererbt werden. Mit diesen beiden Konzepten lassen sich komplexe Softwaresysteme besser beherrschen. Dazu werden Hierarchien von Objekten aufgebaut, die das gesamte Paket modularisieren und damit in übersichtlichere Teilaufgaben separieren (dem Prinzip ‘Teile und Herrsche’ folgend). Ebenso wird die Austauschbarkeit von Einzelteilen ermöglicht, die durch eine definierte Schnittstelle getrennt sind. Zusätzlich wird Redundanz vermieden, d. h., das Kopieren und Wiederholen von Objekten oder deren Funktionalität wird auf ein Minimum reduziert. Dadurch bleibt ein komplexes Gesamtsystem übersichtlicher und ist weniger fehleranfällig.

Die drei vorgestellten Grundkonzepte beschreiben die Vorteile für eine objektorientierten Vorgehensweise. Das Softwarepaket BART ist kein starres System, das exakt eine vollständig definierte Aufgabe lösen soll. Im Gegenteil, die Vielzahl der Möglichkeiten für die verschiedenen Aufgaben - Stimulation des Probanden, Einbindung externer Geräte, Vorverarbeitung, Analyse und Feedbackgenerierung - erfordert eine klare Modularisierung der Aufgaben. Diese Aufgaben sollen entsprechend der individuellen Anforderungen optimal gelöst und für neue Anwendungen erweiterbar sein.

Damit ein Softwaresystem objektorientiert umgesetzt werden kann, ist die Verwendung einer entsprechenden Programmiersprache erforderlich. Die hier hauptsächlich verwendete Sprache ist Objective-C, deren Auswahl in Abschnitt 3.2.4.1 begründet wird. Zusätzlich wird die Programmiersprache C++ in einer Teilbibliothek eingesetzt.

Außerdem werden weiterführende objektorientierte Konzepte eingesetzt, die als Entwurfsmuster bezeichnet werden. Diese bieten eine Art Schablone für die Lösung von immer wiederkehrenden Aufgaben in Softwarearchitekturen an. Der folgende Abschnitt erläutert die für diese Arbeit relevanten Entwurfsmuster.

3.2.2 Einführung zu Entwurfsmustern in der Softwareentwicklung

In der Softwarearchitektur treten einige Herausforderungen immer wieder auf, für deren Lösung mittlerweile eine Art Schablone existiert. Diese können immer wieder verwendet werden, um damit eine Softwareentwicklung kosten- und zeiteffizienter zu gestalten. Entwurfsmuster in der Softwarearchitektur (*Design Patterns*) beschreiben jeweils einen abstrakten Ansatz zur Lösung

von immer wiederkehrenden Anforderungen an Softwaresysteme beim Agieren mit Objekten. Diese Anforderungen lassen sich grob in drei Bereiche gliedern, für die Muster angeboten werden:

- Erzeugung von Objekten,
- Strukturierung von komplexen Softwaresystemen und
- Verhalten von Objekten in der Kommunikation untereinander.

Entwurfsmustern für die Softwarearchitektur wurden erstmalig in Gamma et al. (1994) vorgestellt. Diese Arbeit beschreibt eine Vielzahl von Entwurfsmustern auf dessen Grundlage die Entwurfsmuster permanent weiterentwickelt werden.

Die Entwurfsmustern vereinheitlichen die Beschreibung von Softwaresystemen. Das Benennen des verwendeten Entwurfsmusters reicht aus, um einzelne Teile unabhängig von der verwendeten Programmiersprache zu beschreiben. Die konkrete Umsetzung von Entwurfsmustern ist abhängig von der verwendeten Programmiersprache, die die Umsetzung der Entwurfsmuster jeweils unterschiedlich mit einfachen oder komplexen Konstrukten realisieren.

Die folgenden Erläuterungen beschränken sich auf die wichtigsten Muster mit denen das Softwarepaket BART umgesetzt wurde. Hier werden zuerst die abstrakten Muster eingeführt, deren Verwendung in Abschnitt 3.3.1 erläutert wird. Allerdings wird bereits hier der Bezug zur verwendeten Programmiersprache Objective-C und der Entwicklerbibliothek Cocoa berücksichtigt (siehe Abschnitt 3.2.4.1). Eine vollständige Erläuterung für Entwurfsmuster in Cocoa findet sich in Buck et al. (2009).

3.2.2.1 Model-View-Controller

Eines der ältesten und erfolgreichsten Entwurfsmuster für die Softwareentwicklung ist das zur Trennung von Modell, Präsentation und Steuerung (*Model-View-Controller*, MVC). Dieses Strukturmuster organisiert große Gruppen kooperierender Objekte in die folgenden drei Teilsysteme:

Model: Die zum Modell gehörigen Objekte sind zuständig für die Informationsverwaltung und Datenverarbeitung und bilden somit den Kern der Software.

View: Die zur Ansicht gehörenden Objekte präsentieren die Informationen des Modells an den Anwender, greifen allerdings nicht direkt auf die Informationen des zugrundeliegenden Modells zu.

Controller: Die beiden entkoppelten Komponenten Modell und Ansicht werden über die Controller-Objekte verknüpft. Das heißt, nur diese Objekte interagieren mit Modell und Ansicht(en). Damit können einerseits Änderungen bzw. Neuerungen an der Darstellung der Information gemacht werden ohne die internen Funktionalitäten zu beeinflussen. An-

dererseits ist dies auch für die interne Datenverarbeitung möglich, ohne die graphische Oberfläche neu gestalten zu müssen.

So plausibel die Idee auf den ersten Blick ist, so diszipliniert muss die Umsetzung erfolgen. Diese erfordert zusätzliche Klassen und Abstraktionslevel in der Hierarchie des Softwarepakets, die bereits im Design berücksichtigt werden müssen. Die Einhaltung dieses Strukturmusters ist vorteilhaft für die unabhängige Implementierung der Komponenten. Dies ist wichtig, da sich die Ansichten (Bedienoberflächen) einer Software öfter ändern als das zugrunde liegende Datenmodell oder das Datenmodell weiterentwickelt werden soll, ohne das Bedienkonzept zu verändern. Die Aufteilung ist für das in dieser Arbeit vorgestellte Softwarepaket sinnvoll und die hier eingesetzte und in Abschnitt 3.2.4.1 vorgestellte Bibliothek Cocoa unterstützt die Einteilung in diese drei Komponenten sehr effizient.

3.2.2.2 Singleton

Mit dem Begriff Singleton wird ein Muster zur Erzeugung von Objekten bezeichnet, die nur einmalig und damit global instanziiert werden können. Das heißt, dass nach der Erzeugung eines solchen Objektes immer wieder auf genau diese Instanz zugegriffen wird. Die Idee ist einerseits gegenläufig zur Idee der Objektorientierung, jedoch in wenigen Fällen erforderlich. Dazu gehört z. B. die Repräsentation von physischen Geräten, mit denen nur exklusiv kommuniziert werden kann, oder der Zugriff auf Konfigurationsinformationen, die in der gesamten Applikation relevant sind. Beim Einsatz dieses Musters ist Vorsicht geboten, da ein zu häufiger Einsatz die Prinzipien und damit die Vorteile des objektorientierten Ansatzes aufhebt.

3.2.2.3 Beobachter (Notifications)

Das Beobachtermuster beschreibt ein Verhalten für die Kommunikation zwischen Objekten, ohne dass diese direkt gekoppelt sein müssen. Dabei ist ein Objekt in der Lage, Informationen an beliebige Objekte zu verteilen, ohne weitere Kenntnisse über diese Empfängerobjekte zu besitzen. Die Empfänger registrieren sich selbständig für den Empfang, d. h. sie beobachten Änderungen in einem anderen Objekt, um ihrerseits auf diese reagieren zu können. Dieses Verhaltensmuster ist von großer Bedeutung bei der Umsetzung der losen Kopplung von Objekten. Dazu gehört z. B. die Umsetzung von MVC-Mustern, bei dem die Ansicht auf Änderungen der Daten im Modell reagiert, oder das Laden von Quellcode zur Laufzeit mit Hilfe einer abstrakten Fabrik.

3.2.2.4 Abstrakte Fabrik (Class Clusters) und Protocols

Die Abstrakte Fabrik ist ein Muster zur Erzeugung von Instanzen einer Familie von Objekten. Wie der Name besagt, beschreibt dies ein abstraktes Konstrukt zur Erzeugung von Produkten, in diesem Fall konkreten Instanzen von Objekten. Das erfordert eine zusätzliche Hierarchieebene von Klassen, die eine abstrakte Basis für die gewünschten Produkte enthält und eine Fabrik, die die Komplexität für das Erzeugen der Objektinstanzen vor der Hauptapplikation verbirgt. Zum besseren Verständnis wird dies hier exemplarisch für die Datenhaltung der fMRT-Daten erklärt: Es wird festgelegt, dass es ein Objekt *Datenelement* gibt, dessen Instanzen funktionelle oder strukturelle Daten halten können. Weiterhin gilt, dass die strukturellen Daten bereits als Datei auf dem lokalen Rechner vorhanden sind und diese in den Arbeitsspeicher geladen werden. Die funktionellen Daten dagegen müssen zur Laufzeit kontinuierlich über das Netzwerk empfangen werden. Die beiden Instanzen des abstrakten Objekts *Datenelement* müssen dafür unterschiedlich initialisiert werden. Beide Arten von *Datenelement* bieten aber den identischen Zugriff auf die enthaltenen Daten. Wird die Erzeugung mit einer abstrakten Fabrik realisiert, ist die konkrete Instanziierung dort enthalten. Damit wird die Komplexität von der Hauptapplikation entkoppelt, der Zugriff ist vereinheitlicht und wenn die konkreten Instanzen geändert oder gar vollständig ersetzt werden müssen, kann dies flexibel und unabhängig von der Hauptapplikation implementiert werden.

Dieses Muster ist zusammen mit dem MVC das wichtigste für das hier vorgestellte Softwarepaket. Alle Teile des Paketes, die flexibel austauschbar sein sollen, wurden mit diesem Muster realisiert. Die konkreten Umsetzungen werden in den Abschnitten 3.3.4.4, 3.3.5 und 3.3.6 erläutert. Ergänzend dazu bedarf es hier der Einführung des Begriffs *Plugin*. Ein Plugin ist ein extern zur eigentlichen Applikation erstelltes Softwaremodul, das erst zur Laufzeit der Applikation geladen wird. Dieser Teil kann unabhängig von der Gesamtapplikation entwickelt werden, z. B. ein neuer Auswertalgorithmus, und trotzdem von der Hauptapplikation genutzt werden. Dazu erforderlich ist einerseits, dass ein Plugin eine definierte Schnittstelle einhält, welche in dem verwendeten Entwicklerprogramm als Protocol bezeichnet wird. Andererseits muss es eine Möglichkeit geben, der Hauptapplikation mitzuteilen, ein spezifisches Plugin zu verwenden, was hier über die Konfiguration mit EDL gelöst wird.

3.2.2.5 Fassade

Mit dem Begriff Fassade wird ein Strukturmuster beschrieben, mit dem die Komplexität eines Subsystemes vor dem Gesamtsystem verborgen werden kann, wenn dieses nicht explizit angesprochen werden muss. Dieses Muster fördert die lose Kopplung, erfordert jedoch eine zusätzliche Hierarchiestufe. In Kombination mit der abstrakten Fabrik kann so eine flexible Schnittstelle zum Erzeugen von Systemobjekten unabhängig vom Subsystem angeboten werden. Die Subsysteme des Softwarepakets BART werden in Abschnitt 3.3.1 beschrieben.

3.2.3 Die *Experiment Description Language* zur Konfiguration des Softwaresystems

Wie im Abschnitt 2.2 erläutert, bestehen fMRT-Experimente aus mehreren Schritten von der Datenerzeugung bis zur vollständigen statistischen Auswertung. Für diese gibt es jeweils verschiedene Möglichkeiten für die Ausführung, die alle mit einer Vielzahl von Parametern konfiguriert werden. Das gilt sowohl für konventionelle als auch für Echtzeit-fMRT-Experimente. Eine Besonderheit für Echtzeit-fMRT ist, dass der gesamte Parameterraum für die einzelnen Schritte vor dem Beginn des Experiments definiert sein muss, um Stabilität und Korrektheit des Experiments zur Laufzeit zu garantieren. Die Parameter der einzelnen Schritten des Experiments hängen teilweise voneinander ab, z. B. das Stimulationsprotokoll und die Designmatrix für die Auswertung der Daten.

Zur besseren Strukturierung und Kontrolle dieses komplexen Parameterraum wurde von Hollmann et al. (2008) die *Experiment Description Language* (EDL) vorgestellt. Die Basis sowie EDL selbst werden im im Folgenden erläutert.

XML und XSD: EDL basiert auf der *Extensible Markup Language* (XML¹⁵). Dabei handelt es sich genau genommen um eine Metasprache, also eine Sprache zur Beschreibung von Sprachen (Erlenkötter 2003, S. 11). XML wird verwendet zur Beschreibung von hierarchisch strukturierten Daten. Ein sehr eingängiges Beispiel ist das Prinzip eines Telefonbuchs, das viele Adresseinträge enthält. Jeder Eintrag wiederum kann, wie beispielsweise in der Auflistung 3.1, aus einem Familiennamen, Vornamen und Telefonnummer bestehen:

```
<?xml version="1.0"?>
<Telefonbuch>
  <Eintrag>
    <Familiename>Muster</Familiename>
    <Vorname>Max</Vorname>
    <Telefon>0123</Telefon>
  </Eintrag>
  <Eintrag>
    <Familiename>Master</Familiename>
    <Vorname>Mux</Vorname>
    <Telefon>9876</Telefon>
  </Eintrag>
</Telefonbuch>
```

Auflistung 3.1: Ein Minimalbeispiel zur Veranschaulichung des Konzeptes von XML. Eine Sammlung von Einträgen, die wiederum aus vorgegebenen Elementen wie Namen und Telefonnummer besteht, wird hierarchisch strukturiert in einem Telefonbuch zusammengefasst. Eine solche Struktur kann von einer Software unabhängig von Plattform und Implementierung für verschiedene Zwecke (z. B. Anzeigen oder Drucken) benutzt werden.

In diesem einfachen Beispiel ist das oberste Strukturelement, welches als Wurzel bezeichnet wird, das Telefonbuch. Dieses hat beliebig viele Unterelemente, hier die Adresseinträge, die

¹⁵ <http://www.w3.org/standards/xml>

wiederum drei Unterelemente enthalten. Anfang und Ende der Elemente sind durch die spitzen Klammern und Schrägstrich eindeutig definiert. Damit sind die Daten vollständig beschrieben und jede beliebige Software, die dem XML-Standard folgt, kann die Einträge aus dem Telefonbuch extrahieren. Dieses einfache Prinzip kann beliebig komplex werden, je nachdem welche Datenstruktur (in dieser Arbeit: ein vollständiges Echtzeit-fMRT-Experiment) beschrieben wird.

Zusätzlich zur Strukturierung der Daten können Deklarationen für die einzelnen Elemente und Attribute festgelegt werden. Dazu wird das Prinzip der XML-Schema-Definition (XSD¹⁶) benötigt. Ein Schema hält XML-Beschreibungen fehlerfrei und logisch konform. Eine XSD beschreibt eine gültige Klasse von Dokumenten (in dieser Arbeit: EDL-Dokumente) mit Hilfe von:

- zulässigen Datentypen,
- der Reihenfolge von Elementen,
- den Häufigkeiten (Multiplizitäten) von Elementen,
- Attributen, die die Eigenschaften von Elementen beinhalten,
- Standardwerten oder Einschränkungen für solche Attribute und
- den Beziehungen von Elementen untereinander.

EDL: Mit diesen XML-Grundbausteinen wurde EDL als Beschreibungssprache für die strukturierte Zusammenfassung von Parametern für Echtzeit-fMRT-Experimente definiert. Zusätzlich dazu wurde nach dem gleichen Prinzip eine Wissensbasis deklariert, die mit Hilfe von Regeln Abhängigkeiten zwischen diesen Parametern repräsentiert. Die Struktur für EDL besteht aus dem Wurzelement *rtExperiment*, das in die folgenden Elemente unterteilt wird, die jeweils in ihrer weiteren Verschachtelung eine Kategorie von Parametern beschreiben:

- *environment*: Dateipfade, Logfile, Trigger u. ä.
- *experimentData*: Designmatrix für die Datenauswertung
- *mriParams*: MR-Parameter (TR, Anzahl Volumen u. ä.)
- *preprocessing*: Parameter für die Vorverarbeitungsalgorithmen
- *statistics*: Parameter für die Analyse
- *activationProcessing*: Parameter für die Feedbackgenerierung u.ä.
- *stimulusData*: beschreibt die Stimulation mit den verwendeten Medienobjekten und deren zeitlichem Ablauf
- *viewProperties*: Parameter für die Datenansichten

In der vorliegenden Arbeit sind vor allem die Beschreibung für die Stimulation und die Designmatrix wichtig, da diese durch neue abstrakte Konstrukte erweitert werden (siehe Abschnitt 3.3.6.1). Daher wird hier am Beispiel von Medienobjekten ein Ausschnitt aus dem EDL-Schema gezeigt. Die Auflistung 3.2¹⁷ zeigt die Definition von Medienobjekten und deren Referenzierung in der Struktur für die Designmatrix. Medienobjekte sind hier die Definition für die Art der

¹⁶ <http://www.w3.org/standards/xml/schema>

¹⁷ Zur Vereinfachung wurde die Definition hier allerdings gekürzt.

Stimulation und die dazugehörige konkrete Verknüpfung zu einer Datei. Das Schema definiert den komplexen Typ Medienobjekt, der eine Auswahl an referenzierten Unterelementen (*contentText*, *contentSound*, *contentImage*, *contentVideo*, *contentComplex*) jeweils einmal oder gar nicht enthalten kann. Weiterhin werden als Attribute eine eindeutige Identifikation (ID) und ein Datentyp festgelegt. Die Medienobjekte werden mit Hilfe ihrer ID in einem anderen EDL-Element, dem Stimulusevent (*stimEvent*), verwendet. Der Event definiert den Zeitpunkt und die Dauer der Präsentation eines Medienobjektes und kann beliebig oft in einer Sequenz von Stimulusevents benutzt werden (*blockStimulusDesign*). Mit diesem Schema lässt sich ein individuelles EDL-Dokument erzeugen, in dem z. B. der Inhalt wie in Auflistung 3.3¹⁸ enthalten sein kann. Dieses Beispiel zeigt die Definition der Medienobjekte *INTRO* und *FIXCROSS* strukturiert in einer Liste (*mediaObjectList*). Diese Objekte werden durch Referenzierung in den Stimulusevents, die in dem übergeordneten Element zur Beschreibung der Stimulation (*blockStimulusDesign*) zusammengefasst sind, verwendet.

```
<xs:element name="mediaObject">
  <xs:complexType>
    <xs:sequence>
      <xs:choice>
        <xs:element maxOccurs="1" minOccurs="0" ref="contentText">
        </xs:element>
        <xs:element maxOccurs="1" minOccurs="0" ref="contentSound">
        </xs:element>
        <xs:element maxOccurs="1" minOccurs="0" ref="contentImage">
        </xs:element>
        <xs:element maxOccurs="1" minOccurs="0" ref="contentVideo">
        </xs:element>
        <xs:element maxOccurs="1" minOccurs="0" ref="contentComplex">
        </xs:element>
      </xs:choice>
    </xs:sequence>
    <xs:attribute name="moID" type="xs:ID" use="required"></xs:attribute>
    <xs:attribute name="type" use="required">
      <xs:simpleType>
        <xs:restriction base="xs:token">
          <xs:enumeration value="TEXT"></xs:enumeration>
          <xs:enumeration value="SOUND"></xs:enumeration>
          <xs:enumeration value="IMAGE"></xs:enumeration>
          <xs:enumeration value="VIDEO"></xs:enumeration>
          <xs:enumeration value="COMPLEX"></xs:enumeration>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:complexType>
</xs:element>

<xs:element name="blockStimulusDesign">
  <xs:complexType>
    <xs:sequence>
      <xs:element maxOccurs="unbounded" minOccurs="1" ref="stimEvent">
```

¹⁸ ebenfalls verkürzt

```

    </xs:element>
  </xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="stimEvent">
  <xs:complexType>
    <xs:sequence>
      <xs:element maxOccurs="1" minOccurs="1" name="mObjectID"
        type="xs:IDREF"></xs:element>
    </xs:sequence>
    <xs:attribute name="time" type="xs:unsignedLong" use="required">
    </xs:attribute>
    <xs:attribute name="duration" type="xs:unsignedInt" use="required">
    </xs:attribute>
  </xs:complexType>
</xs:element>

```

Auflistung 3.2: Dieser Ausschnitt aus dem EDL-Schema definiert Medienobjekte und deren Verwendung in einem Blockdesign, das wiederum aus Stimulusevents besteht. Das Schema definiert den komplexen Typ Medienobjekt, der eine Auswahl an referenzierten Unterelementen 0 oder 1 mal (minOccurs, maxOccurs) enthalten kann. Attribute sind hier die ID und ein Typ aus einer Aufzählung möglicher Typen. Die Medienobjekte werden mit Hilfe ihrer ID in einem anderen EDL-Element verwendet, dem stimEvent. Dieser beschreibt den Zeitpunkt und die Dauer der Präsentation eines Medienobjektes. Aus einer beliebigen Anzahl von stimEvents setzt sich ein blockStimulusDesign zusammen.

```

<mediaObjectList>
  <mediaObject moID="INTRO" name="Intro" type="TEXT">
    <contentText>
      <text>Gleich geht's los!</text>
      <tColor>
        <tcRed>0.0</tcRed>
        <tcGreen>0.0</tcGreen>
        <tcBlue>0.0</tcBlue>
      </tColor>
      <tSize>30</tSize>
      <posX>250</posX>
      <posY>300</posY>
    </contentText>
  </mediaObject>
  <mediaObject moID="FIXCROSS" name="Fixcross" type="IMAGE">
    <contentImage>
      <imageFile>FixationCross.jpg</imageFile>
      <posX>0</posX>
      <posY>0</posY>
    </contentImage>
  </mediaObject>
</mediaObjectList>

<blockStimulusDesign>
  <stimEvent duration="5000" time="0">
    <mObjectID>INTRO</mObjectID>
  </stimEvent>
  <stimEvent duration="4000000" time="5000">

```



```
<mObjectID>FIXCROSS</mObjectID>  
</stimEvent>  
</blockStimulusDesign>
```

Auflistung 3.3: Dieser Ausschnitt aus einem EDL-Dokument zeigt die Deklaration von zwei Medienobjekten (INTRO und FIXCROSS), die in konkreten Stimulusevents referenziert werden und in der Auflistung aller stimEvents, dem konkreten blockStimulusDesign genutzt werden.

Mit dieser Art von Regeln und Abhängigkeiten wird geprüft, ob alle Medienobjekte, die für die Stimulation benutzt werden, vollständig definiert sind. Es wird garantiert, dass zur Laufzeit des Experiments keine unbekanntes Medienobjekte benutzt werden. Weitere Abhängigkeiten sind zusätzlich in den verwendeten MR-Parametern für die Anzahl der zu messenden Aufnahmen und der TR-Zeit enthalten. Zum Beispiel gilt hier die Regel, dass die Gesamtzeit aller Messungen mindestens der Zeit entsprechen muss, die für die Stimulation benötigt wird. Diese und viele weitere Elemente, Attribute und Regeln umfasst die Schema- und Regeldefinition für EDL.

Die strukturierte Abbildung des Parameterraums und die Wissensrepräsentation zu den Abhängigkeiten dieser Parameter untereinander sind große Vorteile für die Korrektheit der Konfiguration einer Echtzeit-fMRT-Software. Aus diesen Gründen wird EDL als Konfigurationssprache in BART verwendet. Die Einbindung von EDL und die im Rahmen dieser Arbeit neu entwickelten abstrakten EDL-Konstrukte, mit denen adaptive Paradigmen abgebildet werden, sind in Abschnitt 3.3 erläutert.

3.2.4 Konkrete Werkzeuge für die Entwicklung

3.2.4.1 Programmiersprache Objective-C und Entwicklerframework Cocoa

Objective-C ist eine objektorientierte Programmiersprache, die die prozedurale Programmiersprache C um Sprachmittel zur Objektorientierung erweitert. Die Sprache wurde in der aktuellen Version 2.0 im Jahr 2007 mit dem Betriebssystem Mac OS X 10.5 vorgestellt. Viele Konstrukte sind nicht direkt in der Sprache enthalten, sondern durch das Entwicklerframework¹⁹ Cocoa²⁰ ermöglicht, dessen primäre Sprache Objective-C ist. Neben den bereits vorgestellten Grundkonzepten der Objektorientierung (Datenkapselung, Polymorphie und Vererbung) setzen Objective-C und Cocoa zwei weitere Grundkonzepte um, die den Einsatz dieser Sprache deutlich von anderen objektorientierten Sprachen (wie z. B. C++) unterscheiden und sie sehr leistungsfähig machen:

¹⁹ Als Framework wird hier eine Grundstruktur bezeichnet, die grundlegende Objekte für die Programmierung zur Verfügung stellt.

²⁰ <https://developer.apple.com/technologies/mac/cocoa.html>

- **Dynamisches Typsystem:** Beim dynamischen Typsystem erfolgt die Prüfung von konkreten Datentypen zur Laufzeit des Programms im Gegensatz zum statischen Typsystem, bei dem dies zur Übersetzungszeit (Compilezeit) des Programms durchgeführt wird.
- **Dynamisches Binden:** Dynamisches Binden ermöglicht es, die Aufrufe von konkreten Methoden erst zur Laufzeit zu generieren und gestattet damit eine flexible Kommunikation von Objekten, d. h. Polymorphie über Klassenhierarchien hinweg.

Ein ausführlicher Vergleich findet sich in Lahres et al. (2009, S. 93ff). Diese beiden dynamischen Grundkonzepte unterstützen die lose Kopplung von Objekten innerhalb eines Softwarepakets, bieten eine hohe Flexibilität und vermeiden Redundanz von Quellcode. Im Gegenzug für diese Flexibilität wird allerdings die Notwendigkeit für Laufzeittests verstärkt, da Fehler erst zur Laufzeit erkennbar werden. Für die komplexen Anforderungen eines großen Softwarepakets wie BART, vor allem in der Flexibilität der Weiterentwicklung z. B. das Einbinden neuer Auswertalgorithmen, ist die Programmiersprache Objective-C mit dem Entwicklerframework Cocoa sehr gut geeignet. Durch die Verwendung wird Redundanz in der Programmierung und die damit verbundene Zeit für die Entwicklung verringert. Außerdem unterstützt das Framework Cocoa die Entwurfsmuster MVC, abstrakte Fabrik und Beobachter sehr gut, welche vielfach in der erstellten Software BART eingesetzt werden. Zusätzlich kann Objective-C externe Bibliotheken einbinden, die in den Sprachen C oder C++²¹ geschrieben sind. Des Weiteren enthält Cocoa eine Vielzahl an Klassen, die hilfreiche Entwurfsmuster bereits implementieren und somit direkt verwendet werden können. Dazu gehören u. a. Muster für die Verwaltung von Objekten, das dynamische Erzeugen von Objekten und Vereinfachungen für den Zugriff auf Container von Objekten.

3.2.4.2 Entwicklungsumgebung Xcode

Für die Softwareentwicklung unter dem Betriebssystem Mac OS X steht die umfangreiche und kostenfreie Entwicklungsumgebung Xcode zur Verfügung. Die Entscheidung zur Verwendung von Xcode beruht vor allem auf der umfangreichen Unterstützung der Softwareentwicklung mit dem Cocoa-Framework und den zahlreichen Werkzeugen zur Verhaltens- und Performanzanalyse von Software. Dazu gehören u. a. Werkzeuge zur Kontrolle der Arbeitsspeichernutzung, der Interface Builder zur Erstellung von graphischen Oberflächen und die integrierte Testbibliothek OUnit (siehe Abschnitt 3.2.4.5). Für die Entwicklung des Softwarepaketes BART wird aktuell die Version Xcode 4.4 verwendet. Momentan wird der Compiler *clang* verwendet.

²¹ mit Hilfe von Objective-C++, für Details siehe http://developer.apple.com/library/mac/#documentation/DeveloperTools/gcc-4.2.1/gcc/index.html#//apple_ref/doc/uid/TP40007091

3.2.4.3 Parallelisierungsbibliothek Grand Central Dispatch (GCD)

Die Bibliothek *libdispatch*²² liefert die Schnittstelle zur Verwendung der *Grand Central Dispatch* (GCD)-Technologie, die für die vereinfachte Parallelisierung von Software von der Firma Apple entwickelt und 2009 erstmals vorgestellt wurde. Die Bibliothek ist unter der freien Softwarelizenz Apache 2.0 freigegeben und kann unter Mac OS X und UNIX Betriebssystemen benutzt werden. Das Grundprinzip dieser Technologie beruht darauf, einzelne Aufgaben zu definieren, die unabhängig (d. h. parallel) ausgeführt werden können. Dazu werden diese in Abarbeitungskonstrukte übergeben, die von der GCD verwaltet werden. Die Verteilung auf die vorhandenen Prozessoren wird dabei von der GCD übernommen und somit auf die aktuell verfügbare Hardware angepasst. Diese Abstraktion erleichtert die Entwicklung paralleler Software durch den gezielten Einsatz von wohldefinierten Aufgabenblöcken im Vergleich zur manuellen Threadprogrammierung²³. Ein Thread bezeichnet in diesem Zusammenhang einen parallelen und unabhängigen Strang in der Ausführung der Software, der manuell verwaltet wird und in allen seinen Komponenten unabhängig ist von anderen Threads. Dabei ist einerseits der Programmieraufwand höher und andererseits benötigt das Anlegen, Starten, Beenden und Verwalten paralleler Threads zusätzliche Rechenzeit. Diese Vorgehensweise ist fehleranfälliger und kann zu Seiteneffekten führen, die die Stabilität der Software beeinträchtigen. Außerdem ist die Threadprogrammierung weniger flexibel, da diese explizit programmiert werden und nicht zur Laufzeit an die vorhandene Hardware angepasst werden kann. Trotz dieser Nachteile sind auch manuell angelegte und verwaltete Threads in der Gesamtapplikation notwendig, da diese aus verschiedenen, unabhängigen Teilaufgaben besteht. Bei der Parallelisierung von Algorithmen dagegen ist die Programmierung mit den Konstrukten der GCD wesentlich effizienter und weniger fehleranfällig im Bezug auf Seiteneffekte und Ergebnisverwaltung. Der in Abschnitt 3.3.4.4 beschriebene Algorithmus zur Datenauswertung wurde mit Hilfe der GCD parallelisiert.

3.2.4.4 Schnittstellentechnologie OpenCL

Eine ebenfalls neue Technologie in der Programmierung ist die Schnittstelle *Open Computing Language* (OpenCL)²⁴, die auf eine einheitliche Grundstruktur für die Parallelisierung von Algorithmen auf Haupt- und Grafikprozessoren abzielt. Eines ihrer Ziele ist es, proprietäre Schnittstellen (wie z. B. CUDA von Nvidia²⁵) durch eine einheitliche Schnittstelle zu ersetzen, um Software auf andere Hardware übertragen zu können. Ein zweites Ziel ist es, die Parallelisierung auf Haupt- und Grafikprozessoren zu vereinheitlichen, um die Parallelisierung auf die vorhandene Hardware optimieren zu können. Die erste Implementierung der OpenCL-Schnittstelle

²² <https://libdispatch.macosforge.org/>

²³ <https://developer.apple.com/library/mac/#documentation/General/Conceptual/ConcurrencyProgrammingGuide>

²⁴ <http://www.khronos.org/opencl/>

²⁵ http://www.nvidia.de/object/cuda_opencl_new_de.html

wurde von der Firma Apple in der Version 1.0 im Jahr 2009 mit dem Betriebssystem Mac OS X 10.6 vorgestellt. An dieser Stelle sei ein Ergebnis vorweg genommen, das in der restlichen Arbeit nicht weiter betrachtet werden soll. Die Möglichkeit zur Nutzung dieser Technologie war ein wichtiges Kriterium für die Entwicklung des Softwarepaketes BART mit dem Betriebssystem Mac OS X und den hier vorgestellten Werkzeugen. In der aktuellen Version kommt diese Technologie jedoch aus zwei Gründen nicht zum Einsatz: (1) Die Implementierung einer GLM-Analyse, die mit OpenCL für die Grafikkartenprozessoren parallelisiert wurde, brachte keine Laufzeitvorteile gegenüber der in Abschnitt 3.3.4.4 beschriebenen Realisierung. Durch den Mehraufwand für die Speicherorganisation und die Threadverwaltung hat sich der Geschwindigkeitsgewinn bei der Berechnung aufgehoben. (2) Die Implementierung zum Zeitpunkt des Tests entsprach nicht dem vollständig beschriebenen Funktionsumfang der Schnittstelle. Die Ergebnisse wurden noch nicht mit der aktuellsten Version von OpenCL repliziert. Nichtsdestotrotz ist die vorhandene Anbindung für zukünftige, wesentlich rechenaufwendigere Algorithmen zur Datenauswertung in der Echtzeit-fMRT von Vorteil.

3.2.4.5 Softwaretestframework *SenTestKit*

Software- bzw. Unittests sind ein Instrument um die Funktionalität von Software auf Korrektheit zu verifizieren, indem die selbst erzeugten Objekte in Testobjekten benutzt werden. Die Tests umfassen jede angebotene Operation des Objektes und kennen die korrekten Ergebnisse für die Verifikation. Mit dem in Xcode integrierten Framework *SenTestKit* lassen sich Softwaretests für Objective-C sehr einfach realisieren. Das Framework vereinfacht das Schreiben von Tests für selbst entwickelte Frameworks, Applikationen und Bundles²⁶. Mit Unittests wird nicht nur die Korrektheit geprüft, sondern auch die Fehlerbehandlung und damit die Laufzeitstabilität optimiert. Diese Vorgehensweise kann sehr komplex sein, wenn Objekte auch Daten verarbeiten, z. B. eine GLM-Analyse gerechnet wird. In diesem Fall kann die mathematische Korrektheit der Implementierung anhand einer Referenzimplementierung des Algorithmus oder durch einen Binärdatenabgleich überprüft werden.

Dieser nicht unerhebliche Mehraufwand bei der Implementierung ist notwendig, um die Korrektheit der Software automatisch und jederzeit wiederholt prüfen zu können. Dies kann bei komplexen System und Algorithmen nicht mehr manuell erfolgen. Die Qualität der Software verbessert sich, da das Laufzeitverhalten und die Behandlung von Fehleingaben geprüft wird. Vor allen Dingen ist es gewünscht, dass das Softwarepaket BART sich ständig weiterentwickeln soll. Dabei können unerwünschte Seiteneffekte auftreten, die durch solche automatisierten Tests schnell erkannt und behoben werden können.

Außerdem ist die Anwendung der Software BART immer mit einem Probandenexperiment verbunden. Diese sind mit Kosten für Scanner- und Probandenzeit verbunden und können bei be-

²⁶ Von Apple definierter Begriff für extern eingebundene Softwarepakete.

grenzter Anzahl an Probanden (oder sogar Patienten) sehr aufwendig und kostenintensiv sein. Die Tests für BART wurden mit dem beschriebenen *SenTestKit* implementiert, da dieses eine einfache Integration in Cocoa-Projekte bietet. Die Tests existieren als sogenannte Testbundles neben der eigentlichen Software und werden parallel zur Erstellung der Software ausgeführt.

3.2.4.6 Numerikbibliothek GSL, Schnittstelle BLAS und Accelerate Framework

Die *GNU Scientific Library* (GSL²⁷) ist eine Open-Source-Software für numerische Berechnungen in der Programmierung mit C und C++. Das Paket bietet mit mehr als 1000 Funktionen ein großes Spektrum an mathematischen Routinen von Zufallszahlengeneratoren, über lineare Algebra bis hin zu Differentialgleichungen. Der für diese Arbeit relevante Bestandteil sind Vektor- und Matrixoperationen. In diesem Zusammenhang ist eine weitere Schnittstelle zu nennen, die als *Basic Linear Algebra Subprograms* (BLAS) bezeichnet wird. Diese bietet alle elementaren Operationen für Vektor- und Matrixmultiplikationen für die es eine Vielzahl an Implementierungen in unterschiedlichen Programmiersprachen gibt. Der Vorteil der Schnittstelle BLAS ist einerseits die Wiederverwendbarkeit von Programmen und andererseits die Möglichkeit die Routinen für eine bestimmte Hardware zu optimieren und damit rechenzeiteffizienter zu gestalten. Die Bibliothek GSL wiederum setzt auf den Grundrechenarten dieser BLAS-Schnittstelle auf und implementiert Funktionen für komplexere Berechnungen. In der vorliegenden Arbeit wird die BLAS-Implementierung des *Accelerate*-Frameworks des MAC OS X Betriebssystems zusammen mit der GSL verwendet. Die genannten Bibliotheken werden in der vorliegenden Arbeit für die Implementierung des GLM-Algorithmus verwendet, dessen Ergebnisse im Abschnitt 3.3.4.4 vorgestellt werden.

3.2.4.7 Bildverarbeitungsbibliothek ITK

Das Softwarepaket *Insight Segmentation and Registration Toolkit* (ITK²⁸) ist eine Open-Source-Software, welche eine Vielzahl an Algorithmen für die medizinische Bild- und Signalverarbeitung zur Einbindung in eigene Softwarepakete anbietet. Insbesondere für die Segmentierung medizinischer Bilddaten und der Registrierung von Daten gleicher oder unterschiedlicher Aufnahmemodalitäten bietet dieses Paket Implementierungen der neuesten Algorithmenentwicklungen an. In der vorliegenden Arbeit wird diese Bibliothek für die Registrierungen bei der Bewegungskorrektur und Koregistrierung struktureller und funktioneller Daten verwendet. Zusätzlich bietet die bereits vorhandene Einbindung dieser Bibliothek noch zahlreiche weitere Möglichkeiten für die weitere Entwicklung vor allem von Algorithmen zur Vorverarbeitung von fMRT-Daten.

²⁷ <http://www.gnu.org/software/gsl>

²⁸ <http://www.itk.org>

Mit den in diesem Abschnitt vorgestellten Grundkonzepten der Softwareentwicklung, Entwurfsmustern und Werkzeugen wird im folgenden Abschnitt das im Rahmen dieser Arbeit entstandene Softwarepaket BART beschrieben.

3.3 Ergebnisse der Softwareentwicklung

Das Softwarepaket BART kombiniert die Lösung verschiedener Teilaufgaben, aus denen Echtzeit-fMRT-Experimente zusammengesetzt sind. Zuerst werden in diesem Abschnitt Übersichten zum Softwaredesign, dem technischen Setup des Gesamtsystems und der Hardware vorgestellt. Anschließend ist die Präsentation der Ergebnisse in die folgenden Teilaufgaben unterteilt:

- Integration einer Stimulationsumgebung inklusive der Einbindung externer Geräte (Abschnitt 3.3.6),
- Echtzeit-Verarbeitung der Daten (Abschnitt 3.3.4) und
- Kombination von adaptiven Paradigmen und Echtzeitauswertung (Abschnitt 3.3.7).

Die Entwicklung einer Software im wissenschaftlichen Kontext bringt einige Besonderheiten mit sich. Die Entwicklung ist keine abgeschlossene Produktentwicklung für externe Anwender, sondern dient der eigenen Nutzung bei wissenschaftlichen Experimenten ohne andere Anwender und Anwendungsmöglichkeiten auszuschließen. Das bedeutet, die Entwicklung hat das Ziel einer momentan definierten Anwendung, muss jedoch abstrakt genug sein um zukünftige Anwendungsmöglichkeiten zu unterstützen. Die folgenden Ergebnisse beschreiben insgesamt die erarbeiteten Konzepte, beziehen sich allerdings für das bessere Verständnis in einigen Abbildungen und Beschreibungen bereits auf das im empirischen Teil der Arbeit durchgeführte Experiment (siehe Abschnitt 2.4.2). BART ist eine Open-Source-Lösung wobei der Quellcode auf den Servern des Hosting-Dienstes *github* zu finden ist: <https://github.com/bart-group/BART>. Eine Beschreibung des hier vorgestellten Softwarepakets ist zusammen mit den empirischen Daten, die in Kapitel 4 vorgestellt sind ebenfalls in englischer Sprache in einer Fachzeitschrift publiziert (Hellrung et al. 2015).

3.3.1 Softwaredesign BART

Das Gesamtsoftwarepaket BART ist entsprechend der unterschiedlichen Teilaufgaben in einzelne, voneinander unabhängige Teilpakete untergliedert. Jedes dieser Teilpakete definiert eine Schnittstelle, die mit dem Entwurfsmuster der Fassade entkoppelt voneinander sind. Damit wird die Komplexität der Teilaufgaben vor der Gesamtapplikation verborgen. Die Paketstruktur ist als UML-Diagramm²⁹ in Abbildung 3.1 dargestellt mit den Bezeichnungen der einzelnen

²⁹ UML ist eine universelle graphische Beschreibungssprache für Software.

Aufgaben, die von den Paketen übernommen wird und der Angabe der Schnittstellenklasse(n). Das gesamte Softwarepaket BART besteht aktuell aus der Hauptapplikation, sechs Bibliotheken (Konfiguration, Datenmanagement, Stimulation und Design, Analyse, Vorverarbeitung, Registrierung) und einem extra Framework (Serielles I/O Framework), die in den folgenden Abschnitten ausführlich erläutert werden. Die Teilpakete Hauptapplikation und das Stimulationsmodul sind in sich wiederum als Modell-View-Controller organisiert.

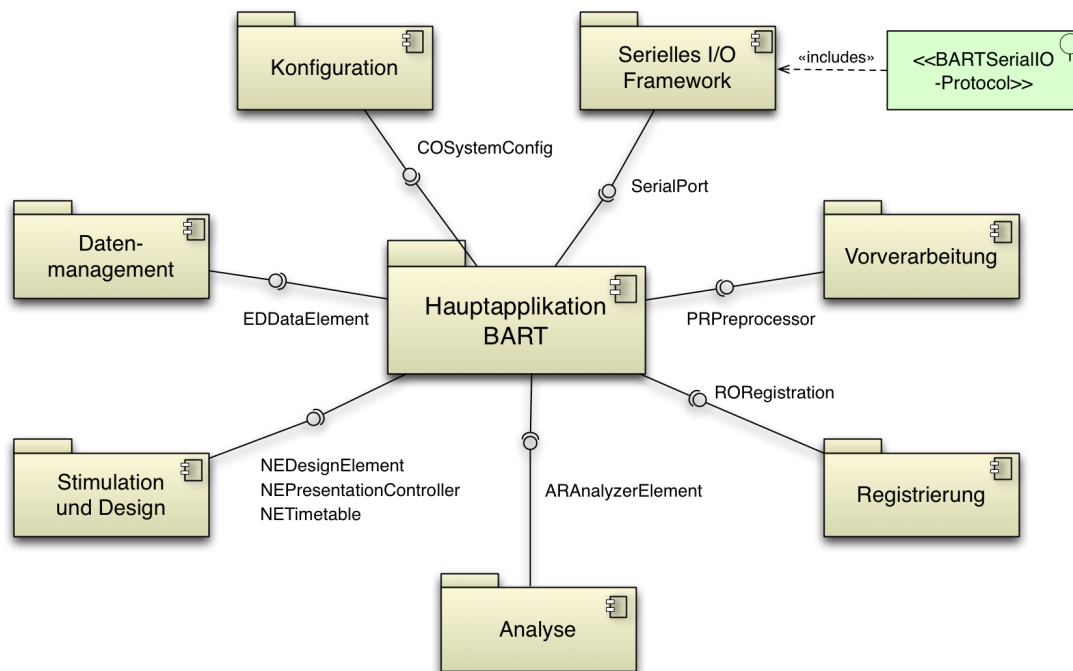


Abbildung 3.1: Übersicht über die einzelnen Teilpakete, in die das Softwarepaket BART zur besseren Aufgabentrennung und Beherrschung der Komplexität aufgeteilt wurde.

Hauptapplikation BART: Die Hauptapplikation BART führt alle Teilaufgaben zusammen und organisiert den Gesamtprozess. In einem Objekt zur Beschreibung einer Session (*BA*Session) können ein oder mehrere Experimente enthalten sein. Jedes Experiment (*BA*Experiment) entspricht einem konkreten Typ von Experiment (z. B. einem Neurofeedbackexperiment). Jedes Experiment wird durch ein EDL-Dokument beschrieben, das dieses individuell konfiguriert, und besteht aus Teilschritten (*BA*ProcedureSteps), die den einzelnen Teilaufgaben aus den Paketen entsprechen. Dazu gehören das Laden und Verwalten der Daten, Präsentation der Stimulation, Vorverarbeitung (inklusive Registrierung) und Analyse der Daten. Die Hauptapplikation übernimmt die Aufgaben für die Steuerung und Anzeige (View und Controller). Das Modell der eigentlichen Datenverarbeitung ist dagegen in den Teilschritten verborgen. Jeder Teilschritt ist damit eine Fassade, hinter der sich eine komplexe Aufgabe verbirgt. Die interne Klassenstruktur ist in Abbildung 3.2 graphisch dargestellt.

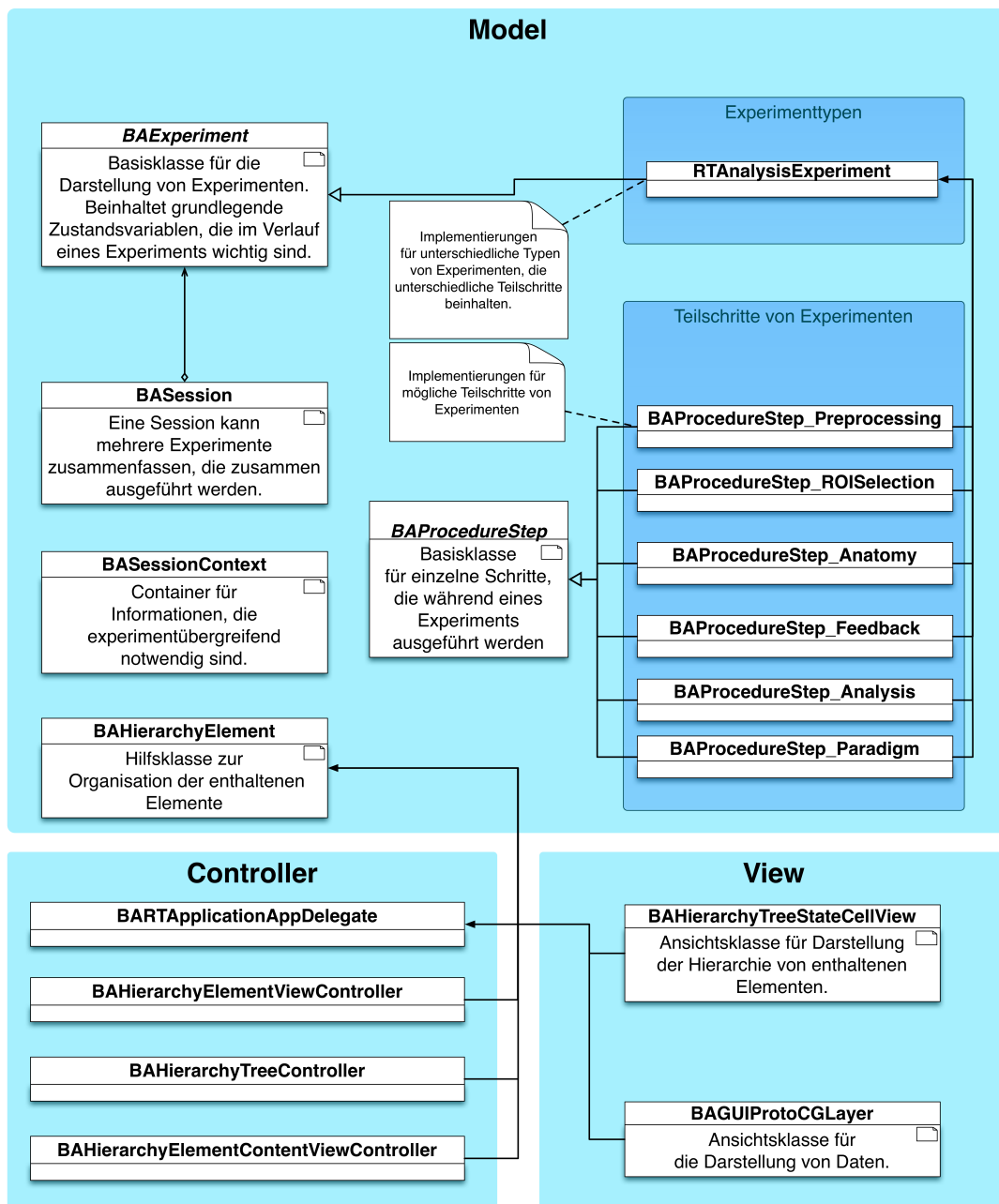


Abbildung 3.2: Das Klassendiagramm veranschaulicht den Aufbau der Hauptapplikation des Softwarepakets BART, die alle anderen Teilpakete (Bibliotheken) zusammenführt und zu Verarbeitungsschritten in Experimenten zusammenfügt. Mehrere Experimente können dann wiederum in einer Session nacheinander abgearbeitet werden.

3.3.2 Technisches Setup

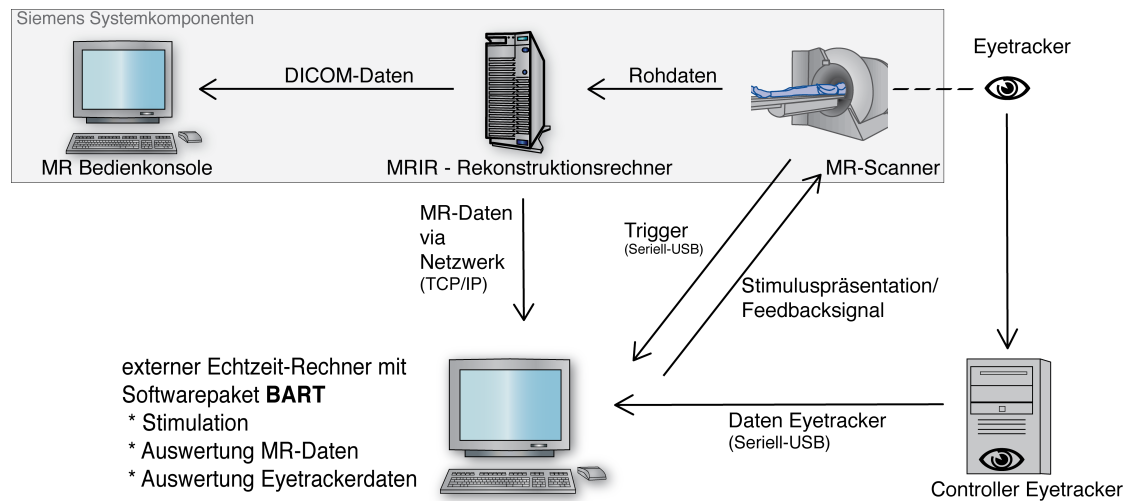


Abbildung 3.3: Das Schema zeigt das technische Setup mit dem neu entwickelten Softwaresystem BART, das für den empirischen Teil dieser Arbeit verwendet wurde. Das Softwarepaket BART läuft auf einem externen Rechner und übernimmt parallel die Aufgaben zur Stimulation des Probanden, die vollständige Datenauswertung und die Auswertung der Eyetrackerdaten zur Adaption des Paradigmas.

Die Abbildung 3.3 zeigt das gesamte technische Setup an einem Scanner der Firma Siemens, wie es für das Hauptexperiment (Abschnitt 4.2) realisiert wurde. Die MR-Daten werden hier direkt vom MR-Rekonstruktionsrechner, der innerhalb der Siemens-Systemkomponenten die Rohdaten empfängt und zu Bilddaten rekonstruiert, an den externen Rechner, auf dem das Softwarepaket BART installiert ist, gesendet. Dieser Datenexport wurde extra für diese Anwendung entwickelt und wird in Abschnitt 3.4.2 beschrieben. Der externe Rechner, der mit BART ausgestattet ist, übernimmt parallel die Präsentation der Stimulation an die Probanden, die gesamte Vorverarbeitung und Auswertung der MR-Daten und die Feedbackgenerierung. Der Rechner empfängt vom Scanner Triggersignale³⁰ und Signale von Tastboxen, die vom Probanden bedient werden. Beide werden über optische Signale übertragen, die außerhalb des MR-Raums auf ein serielles Signal umgelenkt werden und über ein *Universal Serial Bus (USB)*³¹-Kabel am externen Rechner empfangen werden. Zusätzlich wird ein Eyetracker zur Erfassung der Augenbewegungen der Probanden verwendet. Eine Komponente des Eyetrackers befindet sich im Scannerraum und kommuniziert mit einem externen Controller, der die Messdaten in kalibrierte, auswertbare Daten umrechnet. Diese Eyetrackerdaten werden vom Controller an BART weitergeleitet, wo die Daten ausgewertet und die Steuerung des aktuellen Paradigmas abgeleitet werden.

³⁰ Trigger bedeutet hier, dass jeweils zum Beginn jeder TR ein Signal zur Synchronisation gesendet wird

³¹ ein universelles Steckersystem für Computer

3.3.3 Hardware für das Softwarepaket BART

Die vorgestellten Ergebnisse der Software BART basieren auf dem Betrieb unter den Betriebssystemen Mac OS X 10.6 / 10.7 und 10.8 auf der Hardware eines MAC Pro mit 8 GB RAM und zwei Quadcore-Prozessoren und ist momentan auf die Verwendung von MAC Pro Hardware begrenzt. Allerdings existieren für alle gängigen Betriebssysteme Compiler für die Sprache Objective-C und, zumindest für Unix-Betriebssysteme, ist ein kongruentes Entwicklerframework für Cocoa verfügbar. Der Quellcode der GCD ist freigegeben und kann auf andere Systeme übertragen werden. Die Schnittstellentechnologie OpenCL ist eine Definition von Funktionalität, die für alle Betriebssysteme implementiert werden kann. Es besteht theoretisch die Möglichkeit, das gesamte Softwarepaket für alle Betriebssysteme umzusetzen. Das wäre jedoch aufgrund der vielen verschiedenen Bibliotheken, Compiler und Details der einzelnen Betriebssysteme mit einem erheblichen Aufwand verbunden. Daher war ein plattformunabhängiges System im Rahmen dieser Arbeit nicht realisierbar.

3.3.4 Echtzeit-System

Die erste Teilaufgabe, die im Softwarepaket BART implementiert wurde, ist die Echtzeit-Auswertung der fMRT-Daten. Wie im Schema 3.4 dargestellt, umfasst dieses Teilmodul den kontinuierlichen Empfang der fMRT-Daten vom MR-Scanner über das Netzwerk, die Vorverarbeitung, die Analyse und die Anzeige der Ergebnisse.

Für alle Teile des Paketes sind zwei Aufgaben zentral: das Management der Daten und die EDL-Konfiguration des aktuellen Experiments. Daher werden diese beiden Teile zuerst kurz erläutert.

3.3.4.1 Konfiguration mit EDL-Dokument

Die Konfiguration der Applikation ist in einem EDL-Dokument (siehe Abschnitt 3.2.3) gespeichert. Dieses Dokument wird von dem zuständigen Teilpaket beim Start der Applikation geladen und in einem Singleton-Objekt für alle anderen Module zur Verfügung gestellt. Damit ist der Zugriff auf eine eindeutige Konfiguration gesichert und alle Änderungen, die zur Laufzeit gemacht werden, werden zentral gespeichert. Die Hauptklasse *COSystemConfig* des Teilpakets löst die XML-Pfade der EDL-Datei auf und stellt alle Elemente zur Verfügung. Die Kapselung in ein eigenes Teilpaket entkoppelt die Gesamapplikation von den Details der in EDL-definierten Strukturen. Zur Vollständigkeit ist das UML-Klassendiagramm im Anhang in Abbildung A.2 abgebildet, da dies für das Verständnis dieser Arbeit nicht benötigt wird.

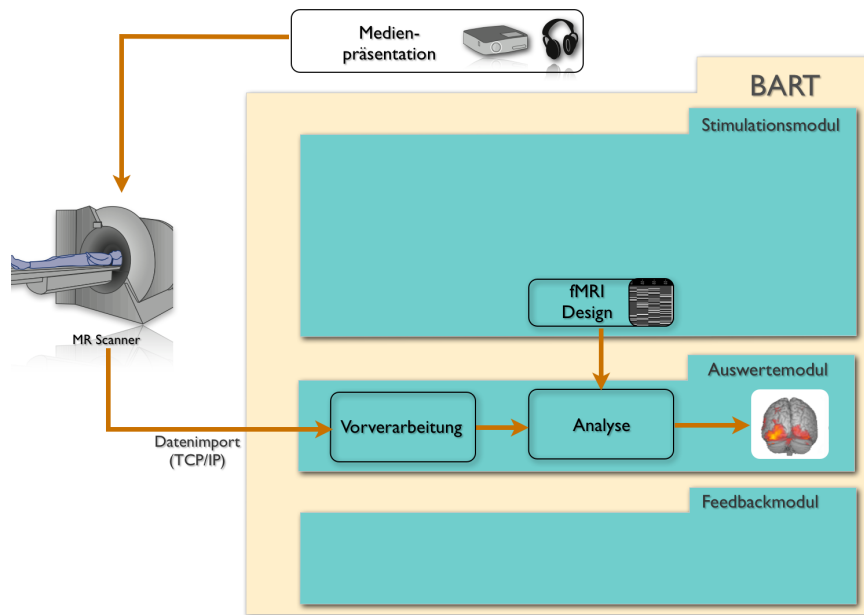


Abbildung 3.4: Die schematische Darstellung zeigt das Modul aus dem Softwarepaket BART, das die Echtzeitauswertung beinhaltet. Dazu gehört das kontinuierliche Einlesen der Daten vom Scanner, die Vorverarbeitung, statistische Analyse und die Anzeige der Ergebnisse. Die Designmatrix für die Analyse ist in diesem Anwendungsfall bereits vordefiniert und wird aus dem Stimulationsmodul geladen.

3.3.4.2 Datenmanagement

Für den Umgang mit den Daten im Kontext der Echtzeitauswertung ist eine effiziente Arbeitsweise im Bezug auf Zugriffszeiten und Speicherverwaltung essentiell. Im Softwarepaket BART wird dazu die Bibliothek *isis* verwendet. Die Bibliothek wurde parallel zum Softwarepaket BART entwickelt, beinhaltet eine typunabhängige Datenhaltung und wird in Abschnitt 3.4.3 erläutert. *Isis* bietet die Möglichkeit zum Lesen und Schreiben der gängigen Datenformate (DICOM, nifti) mit einer vereinheitlichten Schnittstelle. Dadurch kann diese Funktionalität durch weitere Plugins erweitert werden. Für BART ist damit das Einlesen der exportierten Daten vom MR-Scanner mit einem individuellen Plugin für das proprietäre Siemens-Datenformat gelöst. Für die Verwendung mit anderen MR-Scannern kann dies einfach erweitert werden. Die Funktionalität der Bibliothek *isis* ist durch die abstrakte Basisklasse (Wrapperklasse) *EDDataElement* unabhängig von der internen Arbeitsweise der Software BART. Für die Verwaltung der fMRT-Daten ist die konkrete Spezialisierung *EDDataElementIsis* zuständig. Für den Empfang der Daten über das Netzwerk ist zusätzlich die Klasse *EDDataElementRealtimeLoader* implementiert. Diese liest die Daten über das Netzwerk kontinuierlich in einem parallelen Thread ein und stellt die Gesamtdaten für die weitere Verarbeitung zur Verfügung. Das Klassendiagramm dieser Teilbibliothek ist im Anhang der Arbeit in der Abbildung A.1 zu finden.

3.3.4.3 Generierung Designmatrix

Vorbemerkung Softwarepaket Lipsia: Am Max-Planck-Institut für Kognitions- und Neurowissenschaften Leipzig wurde bereits vor einigen Jahren das Softwarepaket Lipsia von Lohmann et al. (2001) entwickelt, das für eine vollständige fMRT-Experimentdurchführung verwendet werden kann. Das Paket ist eine Open-Source-Software, die in der Programmiersprache C geschrieben ist und prozedural realisierte Einzelapplikationen für die Verarbeitung der fMRT-Daten anbietet. Für die vorliegende Arbeit wurden zwei dieser Einzelprogramme als Grundlage verwendet, einerseits für die Berechnung der Designmatrix inklusive der Faltung mit der HRF und die Implementierung des GLM. Die Algorithmen aus dem Lipsia-Paket wurden vollständig restrukturiert und unter dem Aspekt der Laufzeitminimierung neu implementiert. Dazu wurden die Algorithmen für alle geeigneten Berechnungen parallelisiert und auf effiziente Speichernutzung optimiert. Die Verifikation dieser Neuimplementierung erfolgt anhand der Ergebnisse der ursprünglichen Algorithmen.

Parallelisierte Berechnung der Designmatrix: Die Designmatrix ergibt sich wie im Abschnitt 2.2.1 beschrieben aus der Faltung des Stimulationsmodells mit der Hemodynamic Response Function (HRF). Dabei kann die HRF mathematisch unterschiedlich als Kernelfunktion beschrieben werden. In der vorliegenden Arbeit wird analog zum Softwarepaket Lipsia die mathematische Beschreibung nach Glover (1999) verwendet. Allerdings ist dies mit einer abstrakten Fabrik als Entwurfsmuster für die Kernelfunktionen realisiert. Damit kann diese Definition bei Bedarf auf andere Kernelfunktionen angepasst werden. Der Ablaufplan für die Stimulation, d. h. die Onsetzeiten und Dauer aller Stimulusevents, sind für die konventionellen Paradigmen in der EDL-Konfiguration abgelegt und werden aus dieser eingelesen. Daraus werden die Regressoren erzeugt, die als Listen der zeitlich sortierten Trials gehalten werden. Jeder Regressor kann eine individuelle Kernelfunktion verwenden. Für die Faltung der beiden Funktionen werden diese, wie in Abschnitt 2.2.1 beschrieben, im Frequenzspektrum multipliziert und das Ergebnis in den Zeitbereich zurück transformiert. Die Transformationen sind hier mit der Bibliothek `fftw3`³² implementiert. Die Objektstruktur der Implementierung ist im Klassendiagramm 3.8 enthalten.

Für konventionelle Paradigmen ist die Laufzeit dieser Operation selbst im Kontext von Echtzeit-fMRT kein kritischer Parameter, da die Designmatrix vor dem Beginn des eigentlichen Experimentes berechnet werden kann. Dies gilt allerdings nicht für adaptive Paradigmen mit Echtzeitauswertung, bei denen der Ablaufplan der Stimulation zur Laufzeit angepasst wird. Dafür ist es notwendig, dass die Faltungsoperation zur Laufzeit des Experiments wiederholt ausgeführt werden kann. Außerdem ergibt sich die Anforderung, dass in den Regressoren einzelne Events hinzugefügt und entfernt werden können. Dies ist mit sortierten Listen als Datenstruktur effizient möglich. Die Performanz zur Minimierung der Rechenzeit wird durch parallele Berechnung der Faltung für die Regressoren optimiert, da diese unabhängig voneinander sind. Außerdem ist die

³² <http://www.fftw.org/>

Speichernutzung so optimiert, dass zur Laufzeit keine aufwendigen Operationen, wie Speicher reservieren oder freigeben, notwendig sind. Die Laufzeit des Algorithmus zur Neuberechnung der Designmatrix beträgt im Durchschnitt 15,19 ms (Mittelwert aus 140 Berechnungen, Standardabweichung 3,5 ms).

3.3.4.4 Analyse allgemein und parallelisiertes GLM

Für die Analyse der fMRT-Daten gibt es, wie in Abschnitt 2.3.2 erörtert, eine Vielzahl an Algorithmen. Zentral für die vorliegende Arbeit ist die Flexibilität bei der Auswahl der Methodik. Daher wird für das hier entwickelte Softwarepakets BART ein Pluginsystem mit einer abstrakten Fabrik für die Analyseimplementierung verwendet. Dies ermöglicht weitere Methoden entwickeln zu können beziehungsweise bereits entwickelte andere Algorithmen zu integrieren. Damit können die zu verwendende Analysemethode flexibel ausgewählt werden und unabhängig vom Gesamtsystem neue Algorithmen entwickelt werden.

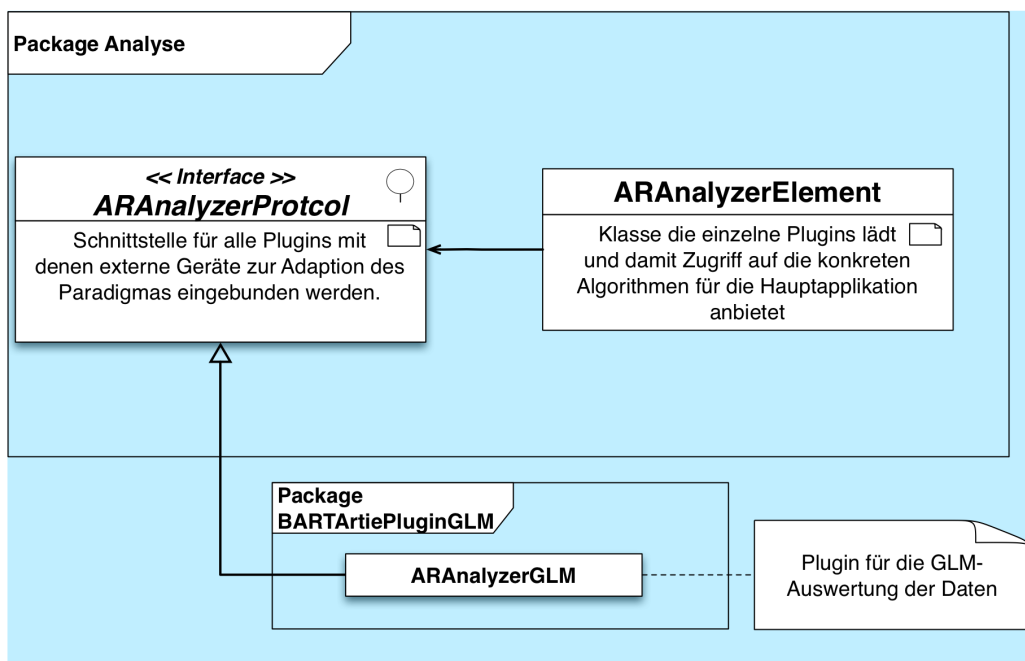


Abbildung 3.5

Die vorliegende Arbeit beinhaltet einen Algorithmus zur Berechnung eines Echtzeit-GLM, der mit paralleler Datenverarbeitung implementiert ist. Die Vorlage der Implementierung ist aus dem Lipsia-Paket. Die parallele Berechnung einer GLM-Analyse ist möglich, da die Zeitserien der einzelnen Voxel als voneinander unabhängig betrachtet werden und dadurch mehrere unabhängige Matrix- und Vektormultiplikationen parallel berechnet werden. Für die einzelnen Operationen werden Funktionen der Bibliothek GSL verwendet, die durch die Verknüpfung zum Accelerate-

Frameworks die Matrixmultiplikationen prozessoroptimiert ausführen und so die Rechenzeit minimieren. Für die Parallelisierung der einzelnen Voxelberechnungen wurde die Implementierung durch Elemente der GCD-Bibliothek strukturiert, die jede Berechnung in einen Block einteilen und auf allen verfügbaren Prozessoreinheiten verteilen.

Datensätze: Für die Evaluierung des GLM-Algorithmus wurden vier unterschiedliche Datensätze verwendet. Der erste Datensatz wurde bereits während der Implementierung als Referenz verwendet. Die funktionellen Daten basieren auf einem visuellen, event-related Design mit 720 Zeitschritten mit je 10 Schichten pro Hirnvolumen. Die Datensätze 2 und 4 beinhalten die Daten, die auch im empirischen Teil der Arbeit in den Abschnitten 4.3 und 4.4 verwendet und dort inhaltlich erläutert werden. Datensatz 2 basiert auf einem Blockdesign und enthält 130 Zeitschritte mit jeweils 30 Schichten pro Hirnvolumen. Datensatz 4 enthält die größte Datenmenge mit 34 Schichten und 1076 Zeitschritten. Datensatz 3 enthält einen Probandendatensatz aus dem in 4.2 vorgestellten Experiment mit 683 Zeitschritten und hat die analogen Aufnahmeparameter wie Datensatz 4.

Die Berechnung des GLM wurde als *sliding-window* und *growing-window* Variante getestet. Die Ergebnisse zeigen daher auch Laufzeiten für eine Teilmenge der Zeitschritte. Die Übersicht zu den erzielten Laufzeiten des Algorithmus ist in der Übersicht 3.2 wiedergegeben. Der Vergleich zur Laufzeit mit der Software Lipsia zeigt, dass die Laufzeiten auf 8 % bis 14 % der Vergleichslaufzeit reduziert sind.

	Zeitschritte					Vergleichs- laufzeit
	50	100	400	650	1000	
Datensatz 1 (10 Schichten, 720 TRs)	50 ms	100 ms	150 ms	250 ms	-	4 s
Datensatz 2 (30 Schichten, 130 TRs)	120 ms	150 ms	-	-	-	1 s
Datensatz 3 (34 Schichten, 683 TRs)	150 ms	200 ms	450 ms	750 ms	-	8,5 s
Datensatz 4 (34 Schichten, 1076 TRs)	150 ms	200 ms	500 ms	850 ms	1,3 s	15 s

Tabelle 3.2: Rechenzeiten des neu implementierten, parallelisierten GLM-Algorithmus im Vergleich mit der Laufzeit in Lipsia, die alle Zeitschritte beinhaltet. Durch Parallelisierung ist die Analyse bis zu 15 mal schneller. Außerdem kann die neue Implementierung eine sliding-window-Analyse berechnen, die eine definierte Anzahl an Zeitschritten einbezieht und daher eine konstante Rechenzeit hat.

3.3.4.5 Vorverarbeitung

Für die Implementierung der Vorverarbeitung gelten die gleichen Anforderungen in Bezug auf die Flexibilität der Methodenimplementierung. Daher ist auch dieses Teilpaket als Pluginsystem mit einer abstrakten Fabrik realisiert. Bisher implementiert ist ein Algorithmus zur Berechnung einer Bewegungskorrektur mit Funktionen aus Bibliothek ITK, der eine Rechenzeit von 200 ms benötigt und damit für die Echtzeitauswertung benutzt werden kann. Allerdings wurde diese Bewegungskorrektur noch nicht für die in dieser Arbeit durchgeführten Experimente verwendet und wird daher hier nicht im Detail vorgestellt.

3.3.5 Integriertes Stimulationssystem: konventionelle Paradigmen

Dieser Absatz beschreibt das Stimationsmodul, das einerseits die konventionelle Präsentation von Stimuli ermöglicht und zusätzlich auch das neue Konzept der adaptiven Paradigmen realisiert. Das Schema in Abbildung 3.6 veranschaulicht das Modul im Gesamtkontext.

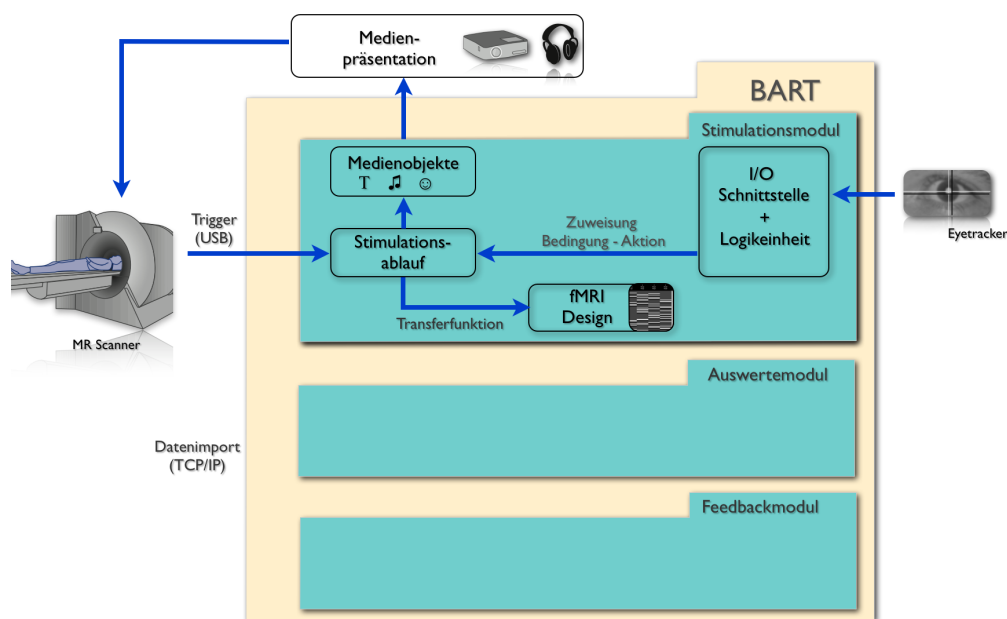


Abbildung 3.6: Die schematische Darstellung zeigt das Modul aus dem Softwarepaket BART, indem die integrierte Stimulationsumgebung beinhaltet ist. Die Integration verringert die Komplexität des Systemaufbaus und ermöglicht die Adaption durch externe Parameter. Die Vorteile von adaptiven Paradigmen werden im empirischen Teil der Arbeit gezeigt. Die I/O-Schnittstelle ist universell gestaltet, so dass die Einbindung externer Parameter (z. B. Eye-tracker, EEG, EKG) beliebig erweitert werden.

3.3.5.1 Allgemeiner Ablauf der Stimulation

Das Stimulationsmodul hat allgemein die Aufgabe, dem Probanden den definierten Stimulationsablauf synchronisiert mit den MR-Messungen zu präsentieren. In dem UML-Klassendiagramm 3.8 ist das Design für dieses Modul abgebildet. Zentral sind die zwei Klassen für die Verwaltung des Stimulationsablaufs (*NETimetable*) und die Steuerung der Anzeige (*NEPresentationController*). Die gesamte Stimulation läuft in einem Thread, der parallel zur Hauptapplikation gestartet wird, um sowohl die Stimulation als auch alle anderen Prozesse auf dem System ausführen zu können. Die Klasse *NETimetable* hält das Modell der Stimulation in einer Datenstruktur, die wie ein Wörterbuch funktioniert. Für jedes Medienobjekt werden alle dazugehörigen Events mit ihrem Start- und Endzeitpunkt zeitlich sortiert gespeichert. Die Steuerung der Stimulation überprüft im Takt von einer Millisekunde diesen Zeitplan auf neu zu startende oder zu beendende Events und sendet die erforderlichen Steuerbefehle über Nachrichten an diese Objekte. Die zu startenden Medienobjekte werden dann wiedergegeben, wobei alle visuellen Objekte (Text, Bild, Video) auf dem Bildschirm dargestellt werden müssen. Dafür ist die Anzeigeklasse *NEPresentationView* zuständig, die alle notwendigen Informationen über die Medienobjekte für die Anzeige auswerten kann. Zusätzlich zu der Anzeigefläche für die Stimuli ist es für die Benutzung der Software notwendig, den Zeitplan für die Stimuli anzuzeigen. Diese Anzeige übernimmt die Klasse *NETimelineView*. Außerdem gibt es ein Steuerungsfenster zur Interaktion für den Nutzer mit der Stimulation (*NEControlWindowController*). Über diese kann ein Nutzer per Maus oder Tastatureingabe die Stimulation ändern.

Zusätzlich zur Präsentation der Stimuli ist es essentiell für die spätere Auswertung der Daten, alle generierten Ereignisse in einer sogenannten Log-Datei zu protokollieren. Das ist Aufgabe der Klasse *NEPresentationLogger*, die als Singleton erzeugt wird. Damit ist das Protokollieren aller Einträge zentral und übergreifend über alle Laufzeitkomponenten sicher gestellt. Das heißt, dass alle Einträge, egal von welchem Objekt sie erzeugt werden, garantiert in der selben Struktur gespeichert werden. Die Einträge werden zur Laufzeit im Arbeitsspeicher gesammelt, mit einem Zeitstempel markiert und durch expliziten Aufruf der Schreibmethode am gewünschten Pfad abgelegt. Zur Vereinfachung wurde eine Hilfsklasse für die Formatierung von verschiedenen Arten von Einträgen (*NELogFormatter*) implementiert, damit diese einheitlich formatiert ausgegeben werden.

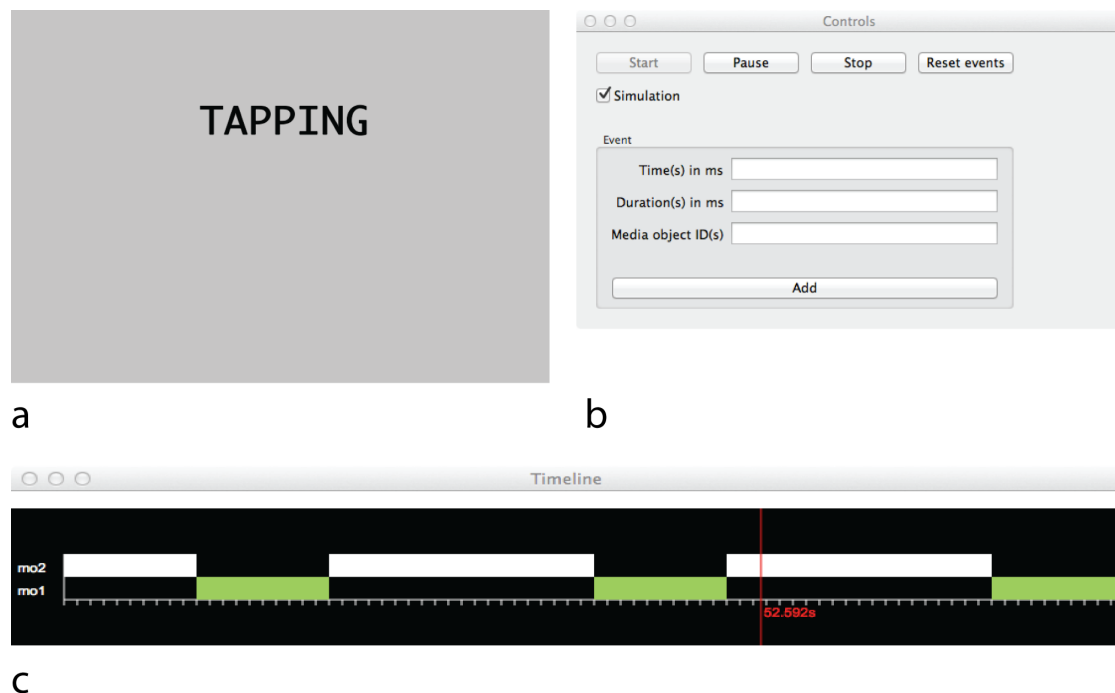


Abbildung 3.7: Die Ansichten (Views) des Stimulationsmoduls BART: Präsentationsfläche der Medienobjekte, die hier einen Anweisungstext abbildet (a). Kontrollfenster für den Bediener zur externen Steuerung der Präsentation (Start, Stopp, Eingabe weiterer Events) (b). Ansicht des Ablaufs der Präsentation aller Medienobjekte mit aktueller Zeitangabe (rote Linie) (c).

3.3.5.2 Medienobjekte

Medienobjekt ist der verallgemeinerte Begriff für die internen Repräsentationen von Stimuli. Momentan sind drei verschiedene Arten von Medienobjekten (Bilder, Text und Sound) implementiert. Die Realisierung der Objekte im Softwarepaket BART ist im UML-Klassendiagramm 3.8 abgebildet. Die abstrakte Basisklasse *NEMediaObject* ist als abstrakte Fabrik organisiert. Von dieser sind verschiedene Unterklassen abgeleitet, um konkrete Instanzen für die unterschiedlichen Stimuli zu erzeugen. Nur die abgeleiteten Klassen haben die Information über die notwendigen Eigenschaften, z. B. die Position des Textes oder die Schriftfarbe, und wissen, wie ein Medienobjekt zu präsentieren ist. Die vorhandenen Medienobjekte können auch sehr einfach erweitert werden. Damit z. B. Videos angezeigt werden können, muss ausschließlich die konkrete Klasse zum Anzeigen von Videos implementiert werden. An dem gesamten Ablauf der Stimulation oder dem Aufbau von Events sind dagegen keine Änderungen notwendig.

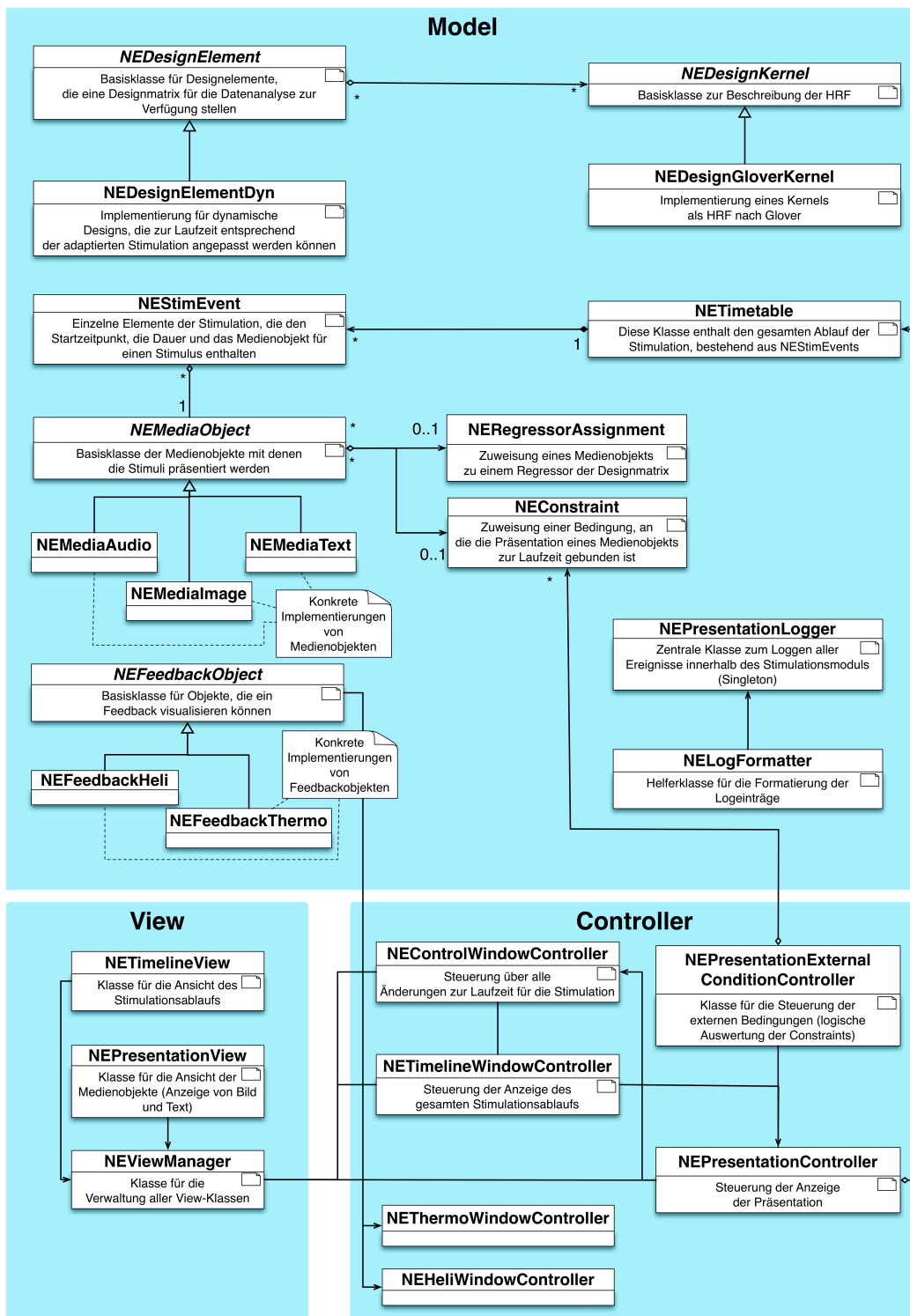


Abbildung 3.8: Das Klassendiagramm zeigt den Aufbau des Softwaremoduls, mit dem die adaptive Probandenstimulation implementiert ist.

3.3.6 Integriertes Stimulationssystem: adaptive Paradigmen

Für die Umsetzung von adaptiven Paradigmen wurde das Stimulationssystem zusätzlich zu dem vorgestellten Ablauf so erweitert, dass zur Laufzeit externe Parameter geprüft werden können und deren aktueller Zustand den weiteren Ablauf beeinflussen kann. Für das bessere Verständnis werden die Beispiele in diesem Abschnitt auf die in 2.4.2 beschriebene Anwendung bezogen. Insgesamt betrachtet sind drei zusätzliche Komponenten notwendig:

- Es müssen externe Parameter eingebunden werden können, deren aktueller Zustand das Paradigma beeinflussen kann. Das erfordert eine einheitliche I/O-Schnittstelle und entsprechende Plugins.
- Die Adaption der Stimulation muss beschrieben werden. Dazu müssen die Bedingungen (*Constraints*) für ein Stimulusevent definiert sein, die eine Stimulation zulassen bzw. verhindern. Zur Laufzeit müssen diese überprüft werden. Außerdem müssen die daraus entstehenden Änderungen für das Paradigma (*StimulusActions*) definiert und zur Laufzeit adäquat ausgeführt werden.
- Diese Konstrukte müssen in EDL definiert werden.

3.3.6.1 Erweiterungen in EDL zur Realisierung adaptiver Paradigmen

In der vorgestellten Beschreibungssprache EDL wurden die neuen Elemente *Constraint*, *SystemVariable* und *StimulusActions* eingeführt, die hier exemplarisch in den Auflistungen 3.4 und 3.5 mit einem konkreten Beispiel erläutert werden. Die vollständigen Schemadeklarationen für diese Elemente finden sich im Anhang in den Auflistungen A.1 und A.3. Die Deklaration ergibt sich aus den folgenden Lemmas:

- Die Ausführung der Präsentation von Medienobjekten wird durch *Constraints* eingeschränkt.
- Ein *Constraint* definiert genau eine Bedingung, die sich aus mehreren Variablen durch Konjunktion zu einem Wahrheitswert (*true* oder *false*) ergibt.
- Ist der Wahrheitswert *true* werden die Aktionen ausgeführt, die optional durch *stimulusActions_then* definiert sind, ansonsten werden die Aktionen in *stimulusActions_else* ausgeführt.
- Ein Medienobjekt kann nur mit maximal einem *Constraint* verknüpft werden.
- Eine *SystemVariable* ist durch Zuordnung zu einer Quelle und einem Variablennamen eindeutig definiert.

Das Beispiel in Auflistung 3.4 zeigt die Definition des *Constraints col*, der auf den Eyetracker als Quelle referenziert. Dabei definiert die Systemvariable *sVI* die interne Variable *eyePosIsFixated* als Bedingung. Das Beispiel beschreibt damit die Einschränkung der Stimulation durch die Bedingung der Fixation. Nur wenn die Variable *eyePosIsFixated* den Wahrheitswert *true*

liefert, wird der Stimuli präsentiert. Andernfalls wird die Alternativaktion *stimulusAction_else* ausgeführt. Das bedeutet, dass ein neuer Event mit dem Medienobjekt *FIXCROSS* für 8 ms in den aktuellen Ablauf eingefügt wird und alle noch ausstehenden Stimuli zeitlich verschoben werden. Dieser *Constraint* kann dann von allen Medienobjekten (hier Bildern) verwendet werden, die damit nur gezeigt werden sollen, wenn der Proband in der Bildschirmmitte fixiert. Die Verknüpfung wird, wie in Auflistung 3.5 am Beispiel gezeigt, als Attribut im Medienobjekt gekennzeichnet,.

```
<constraints>
  <systemVariables>
    <systemVariable source="de.mpg.de.ASLEyetracker" systemVariableID="sV1"
      systemVariableName="eyePosIsFixated"/>
    <systemVariable source="de.mpg.de.ASLEyetracker" systemVariableID="sV2"
      systemVariableName="eyePosX"/>
    <systemVariable source="de.mpg.de.ASLEyetracker" systemVariableID="sV3"
      systemVariableName="eyePosY"/>
  </systemVariables>
  <constraint constraintID="co1">
    <conditions>
      <condition systemVariableRef="sV1"/>
    </conditions>
    <stimulusActions_else>
      <stimulusAction>
        <insertNewStimulusEvent duration="8" mediaObjectRef="FIXCROSS"
          shiftFollowingStimulusEvents="true"/>
      </stimulusAction>
    </stimulusActions_else>
  </constraint>
</constraints>
```

Auflistung 3.4: Das Beispiel zeigt die Definition eines *Constraint* *co1*, das die Ausführung einer *Stimulation* von dem Wert der Systemvariablen *sV1* abhängig macht, die aus den *Eyetracker*daten ermittelt wird. In diesem Beispiel wird wenn die *Bedingung* (*condition*) nicht erfüllt ist, eine *Aktion* ausgeführt, die einen neuen *Stimulus* zum aktuellen Zeitpunkt einführt. Dabei wird das *Medienobjekt* *FIXCROSS* für 8 ms präsentiert und alle folgenden *Bilder* anpasst (*stimulusAction_else*).

```
<mediaObject moID="BildID1" name="1019" type="IMAGE" useConstraint="co1" >
  <contentImage>
    <imageFile>pics/1019.jpg</imageFile>
    <posX>0</posX>
    <posY>0</posY>
  </contentImage>
</mediaObject>
```

Auflistung 3.5: Das Beispiel aus einem *EDL-Dokument* zeigt die Verwendung des *Constraints* *co1* als Attribut in einem *Medienobjekt*. Die *Präsentation* dieses *Medienobjekts* in einem *Event* ist damit abhängig davon, ob der *Proband* zu dem gewünschten Zeitpunkt die *Mitte* des *Bildschirms* fixiert.

Dieser umfangreiche, abstrakte Ansatz ist notwendig, damit die *Präsentation* der *Stimuli* von den externen Einflussparametern entkoppelt ist. Das heißt, das *Stimulationsmodul* der *Software*

BART ist unabhängig von einem Eyetracker und kann genauso gut durch ein Blutdruckmessgerät oder durch ein EEG-Gerät adaptiert werden (siehe folgenden Abschnitt). Weiterhin ist es denkbar, andere Parameter, die erst extern berechnet werden, für die Adaption der Medienobjekte zu verwenden. Zum Beispiel ist es möglich, dass nicht die Bildpräsentation zeitlich verschoben wird, sondern das Bild an den Koordinaten der aktuellen Blickrichtung des Probanden anzuzeigen. Dazu können die vom Eyetracker ermittelten Systemvariablen *sv2* und *sv3* benutzt werden, um in der *StimulusAction* die Positionseigenschaften des entsprechenden Bildes anzupassen.

3.3.6.2 Die I/O-Schnittstelle zum Einbinden externer Variablen

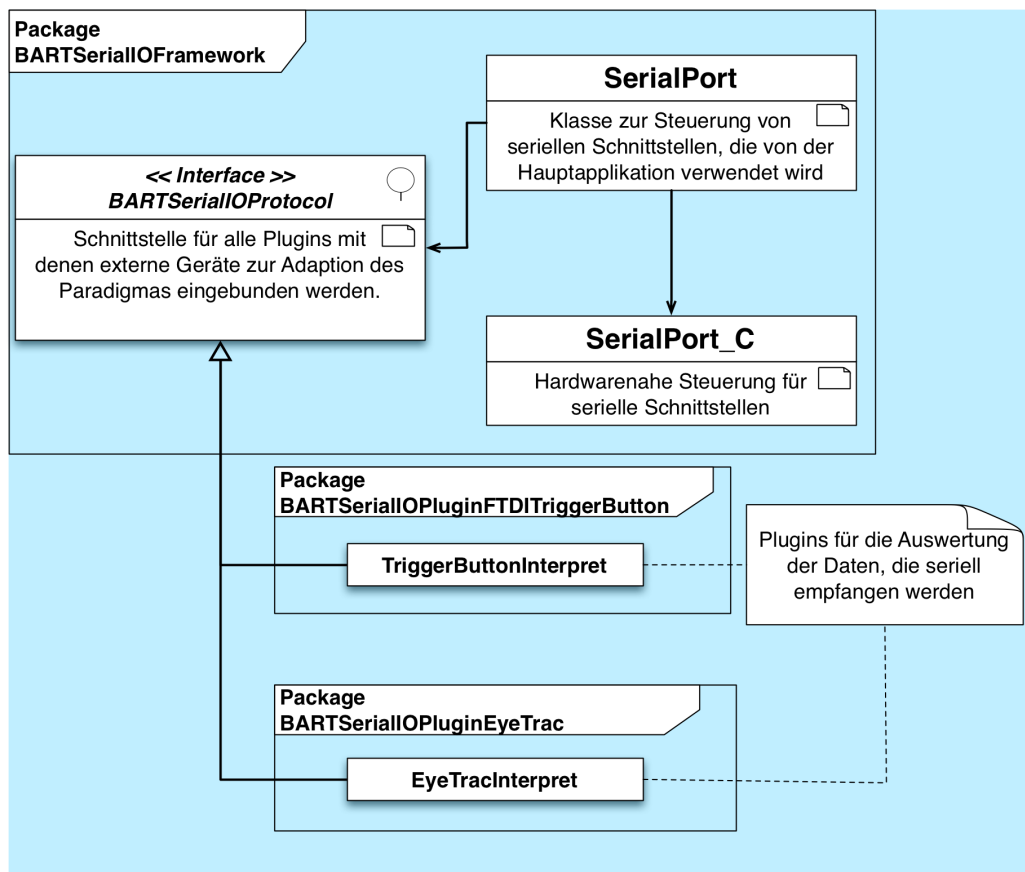


Abbildung 3.9: Aufbau des Frameworks *BARTSerialIOFramework*: Das Klassendiagramm zeigt den Aufbau des Frameworks für die Einbindung serieller Schnittstellen und die abgeleiteten Plugins, die jeweils für die Auswertung von Daten zuständig sind. Dargestellt sind hier der Empfang von Trigger- und Tastboxsignalen (*TriggerButtonInterpret*) und die Auswertung der Eyetrackerdaten (*EyeTracInterpret*).

Damit die externen Geräte, die zur Adaption des Paradigmas verwendet werden, unabhängig von ihrer konkreten Art eingebunden werden können, muss die Kommunikation mit diesen Geräten

abstrahiert werden. Dazu wurde die Schnittstelle zum Lesen und Schreiben (*I/O-Schnittstelle*) der Daten von den externen Geräten durch die Definition eines *Protocols* abstrahiert. Das *Protocol* definiert die Methode *evaluateConstraintForParams*, die von der Klasse implementiert werden muss, um ein externes Gerät abzubilden und die empfangenen Signale zu interpretieren. Das bedeutet, dass momentan ein Framework implementiert ist, das alle Arten von seriellen Ports³³ abbildet. Diese Art von Port ist bei vielen Geräten zum Empfang und Senden von Daten vorhanden und kann trotz der seriellen Eigenschaften mit einem USB-Stecker an einen Rechner angeschlossen werden. Das Framework *BARTSerialIOFramework* ist als Klassendiagramm in Abbildung 3.9 dargestellt und besteht neben der Definition des *Protocols* (*BARTSerialIOProtocol*) aus zwei Klassen für die Kommunikation mit den seriellen Ports. Die Klasse *Serial_Port* ist dabei die Schnittstelle zum einfachen Initialisieren und Öffnen bzw. Schließen eines solchen Ports. Die darunter liegende Klasse *Serial_Port_C* übernimmt die hardwarenahe Übersetzung dieser Kommandos in C-Funktionen. Ein solcher serieller Port empfängt die Daten von den externen Geräten und hat die Aufgabe, diese an Objekte weiterzuleiten, die diese Daten verarbeiten können. Das geschieht hier durch Plugins, deren Hauptklassen sich jeweils als Observer an einem seriellen Port angemeldet haben. In dieser Implementierung existieren Plugins für den Empfang der Triggersignale vom Scanner (Plugin *TriggerButtonInterpret*) und der Eyetrackerdaten (Plugin *EyeTracInterpret*). Die jeweiligen Plugins verarbeiten die Daten, die am seriellen Port ankommen und berechnen so die Systemvariablen, die von den *Constraints* definiert werden. Das Plugin für den Eyetracker empfängt also kontinuierlich die Daten des Eyetrackers, kann diese interpretieren und somit die Information ermitteln, ob der Proband zum aktuellen Zeitpunkt das Bild fixiert. Die Hauptapplikation und die Auswertung der Eyetrackerdaten sind damit vollständig voneinander entkoppelt und können unabhängig entwickelt werden.

3.3.6.3 Überprüfen der externen Bedingung und Anpassen des Paradigmas

Mit den vorgestellten Konstrukten der *Constraints* und *stimulusActions* und der abstrakten Schnittstelle kann eine Stimulation nun zur Laufzeit adaptiert werden. Wie im Klassendiagramm 3.8 dargestellt, sind die *Constraints* als Instanzen der Klasse *NEConstraint* im Stimulationsmodul abgebildet, die insgesamt von der Klasse *NEPresentationExternalConditionController* verwaltet werden. Diese Verwaltung ordnet den *Constraints* ihre zugehörigen seriellen Ports zu und generiert die Abfrage des externen Plugins mit allen notwendigen Parametern. Außerdem wird die Antwort empfangen und eventuelle Ergebniswerte in Datenstrukturen hinterlegt, die wiederum vom *PresentationController* verwendet werden. Durch das hohe Maß an Abstraktion der einzelnen Konstrukte ändert sich der grundlegende Ablauf im *NEPresentationController* durch das Hinzufügen einer Abfrage mit einer 'Wenn-Dann-Sonst'-Funktion gering. Das heißt, zur

³³ Serielle Ports ermöglichen den Datenaustausch zwischen Computern und Peripheriegeräten durch die serielle Übertragung von Bits über eine Leitung. Der Begriff Port wird hier anstelle von Schnittstelle verwendet, um Verwechslungen mit der in diesem Abschnitt beschriebenen Softwareschnittstelle zu vermeiden.

Laufzeit der Präsentation wird für jedes Event überprüft, ob das dazugehörige Medienobjekt an einen *Constraint* gebunden ist. In diesem Fall wird die Abfrage der Bedingung an das externe Plugin weitergeleitet und durch das Ergebnis eine der *stimulusActions* ausgelöst. Momentan sind die folgenden Aktionen möglich:

- **Parameter eines Medienobjekts ändern:** Die Position des Bildes kann entsprechend der Blickrichtung des Probanden verändert.
- **Medienobjekt ersetzen:** Das aktuelle Event kann ausgetauscht werden, z. B. der Austausch des anzuzeigenden Bildes.
- **Stimulusevent neu erzeugen:** Damit kann ein neues Event erzeugt werden, das zum aktuellen Zeitpunkt präsentiert wird. Zum Beispiel kann damit ein Fixationskreuz angezeigt und die eigentliche Bildpräsentation zeitlich verschoben werden.

3.3.6.4 Die Laufzeitgenauigkeit der Präsentation

Damit die Auswertung eines Experimentes valide ist, muss die Präsentation von Stimuli zeitlich exakt erfolgen. Die Zeit vom Auslösen eines Stimulusevents bis zur wirklichen Präsentation an den Probanden muss minimiert werden. Gründe für Verzögerungen können z. B. sein: (1) Ein zu präsentierendes Bild wird erst zum Zeitpunkt der Anzeige in den Arbeitsspeicher geladen. (2) Die Prozessoren des Rechners sind ausgelastet und die Anzeige kann nicht ausgeführt werden. Oder (3) das Datenmanagement der Software ist ineffizient.

Präsentationssysteme, die auch für EEG-Experimente verwendet werden, müssen eine Präsentationsgenauigkeit im Sub-Millisekundenbereich einhalten. Meist erfordern diese Softwarepakete für ein stabiles und zuverlässiges Verhalten die exklusive Ausführung auf einem Rechner, das heißt es dürfen keine weiteren Anwendungen laufen und auch keine Netzwerkverbindungen vorhanden sein. Für fMRT-Experimente ist die Laufzeitgenauigkeit der Präsentation wichtig aber mit einer Toleranz ausführbar. Dies ergibt sich aus dem BOLD-Effekt, der sich über mehrere Sekunden aufbaut und dessen Daten bei der Auswertung mit einer Abtastrate von 20 ms diskretisiert werden. Daher ergibt sich dieser Wert auch als maximaler Toleranzbereich für die Präsentation der Stimuli.

Für das hier vorgestellte Softwaresystem BART wurde das Stimulationssystem in das Gesamtsystem integriert. Dazu werden mehrere Threads, der Datenempfang über das Netzwerk und die gesamte Verarbeitung der Daten, gleichzeitig auf den Prozessoren des Rechners ausgeführt. Das Stimationsmodul ermöglicht es, dass bei den adaptiven Paradigmen neue Stimulusevents zur Laufzeit eingefügt werden können. Die Stabilität und Laufzeitgenauigkeit des integrierten Stimulationssystems ist trotzdem garantiert. Zur Quantifizierung der Laufzeitgenauigkeit wurden verschiedene Tests durchgeführt, bei denen das Stimationsmodul ausgeführt wurde, während alle Prozessoren des Rechners mit anderen Threads belastet wurden. Dann wurden die Zeiten

vom Auslösen des Events bis zur wirklichen Präsentation erfasst. Die Tests wurden auf einem Mac Pro (Dual Quadcore, 8 GB RAM) wie folgt durchgeführt:

- Mit der Software Cinebench³⁴ wurden 16 Threads gestartet, die die Prozessoren ausgelastet haben. Gleichzeitig wurde eine Stimulation mit Bildern, Text und Musik abgespielt.
- Weiterhin wurden in mehreren Versuchen zu einem beliebigen Zeitpunkt 200 neue Events eingefügt.
- Mit der Software SystemLoad³⁵ wurden alle 16 Prozessoren unter Last gestellt und eine Simulation abgespielt.
- Ebenfalls wurden hier zu einem beliebigen Zeitpunkt 200 neue Events eingefügt.

Die beschriebenen Versuche wurden mehrfach wiederholt und zeigen eine sehr hohe Laufzeitgenauigkeit (siehe Tabelle 3.3).

	BART + Cinebench (16 Render threads)	BART + SystemLoad (16 Prozesse)	BART (alle Prozesse)
konventionelle Ausführung	< 2 ms	< 2 ms	< 2 ms
adaptive Ausführung	< 7 ms	< 7 ms	< 2 ms

Tabelle 3.3: Zur Verifikation des Stimulationsmoduls wurde die Ausführung der Präsentation durch Auslastung der Prozessoren mit Benchmarksoftware getestet. Die Ergebnisse zeigen eine sehr hohe Genauigkeit auch bei maximaler Auslastung aller vorhandenen Prozessorkerne.

3.3.7 Gesamtaufbau: Adaptive Paradigmen und Echtzeit-Auswertung

Die beiden vorherigen Abschnitte haben die beiden Module zur Echtzeit-Auswertung und der Stimulation unabhängig voneinander beschrieben. Ein weiteres Ziel dieser Arbeit ist es, diese beiden Module zu kombinieren. Dazu sind die folgenden Komponenten notwendig, die in diesem Abschnitt beschrieben werden:

- die Definition dieser Konstrukte in der Experimentbeschreibungssprache EDL,
- die Übertragung der adaptierten Stimulation in entsprechende dynamische Regressoren mit Hilfe von Transferfunktionen und
- die Aktualisierung der Designmatrix in Echtzeit.

³⁴ <http://www.maxon.net/products/cinebench/overview.html>, Version 11.5

³⁵ <http://www.bresink.de/osx/SystemLoad-de.html>, Version 1.3

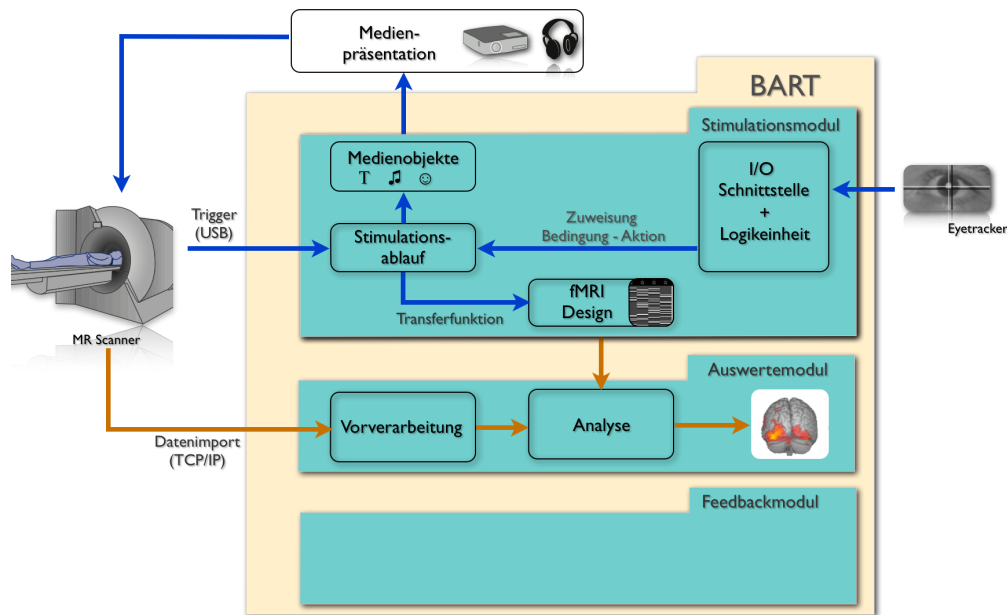


Abbildung 3.10: Die schematische Darstellung zeigt die Module des Softwarepakets BART, die das integrierte Stimulationsmodul simultan mit der Echtzeitauswertung ausführen. Die Verbindung der beiden Module entsteht durch die Übertragung der (adaptierten) Stimulationsereignisse zum Zeitpunkt ihrer Präsentation in die Designmatrix, die für die Auswertung verwendet wird.

3.3.7.1 Erweiterungen in EDL zur Realisierung einer dynamischen Designmatrix

Für die Übertragung neu erzeugter Events in die Regressoren der Designmatrix wurden in der Beschreibungssprache EDL weitere Elemente deklariert. Zum besseren Verständnis werden auch diese Elemente anhand von Beispielen erläutert. Die vollständigen Spezifikationen finden sich im Anhang in den Auflistungen A.2 und A.3.

Das Element *dynamicDesignStruct* ist eine Struktur, die ein vollständiges dynamisches Design beschreibt. Dieses definiert die maximale Länge des Designs und besteht aus einem oder mehreren Regressoren. Diese sind als *dynamicTimeBasedRegressor* bezeichnet und definieren nur beschreibende Attribute des Regressors (z. B. den Faltungskern, vergleiche Abschnitt 2.2.1). Sie enthalten keine Events, da diese erst zur Laufzeit generiert werden. Das Beispiel in Auflistung 3.6 zeigt die Definition einer solchen dynamischen Struktur mit vier dynamischen Regressoren (*REG-NEGDYN*, *REG-NEGSTAT*, *REG-NEUTDYN* und *REG-NEUTSTAT*).

```
<paradigm ignoreScansAtStart="20">
  <dynamicDesignStruct maxLength="4000000">
    <dynamicTimeBasedRegressor length="4000000" name="neg_dynamic"
      regressorID="REG_NEGDYN" scaleHeightToZeroMean="false" useRefFct="
      gloverKernel" useRefFctFirstDerivative="false"
      useRefFctSecondDerivative="false"/>
```

```

<dynamicTimeBasedRegressor length="4000000" name="neg_static"
  regressorID="REG_NEGSTAT" scaleHeightToZeroMean="false" useRefFct="
  gloverKernel" useRefFctFirstDerivative="false"
  useRefFctSecondDerivative="false"/>
<dynamicTimeBasedRegressor length="4000000" name="neut_dynamic"
  regressorID="REG_NEUTDYN" scaleHeightToZeroMean="false" useRefFct="
  gloverKernel" useRefFctFirstDerivative="false"
  useRefFctSecondDerivative="false"/>
<dynamicTimeBasedRegressor length="4000000" name="neut_static"
  regressorID="REG_NEUTSTAT" scaleHeightToZeroMean="false" useRefFct=
  "gloverKernel" useRefFctFirstDerivative="false"
  useRefFctSecondDerivative="false"/>
</dynamicDesignStruct>
</paradigm>

```

Auflistung 3.6: Das Beispiel aus einem EDL-Dokument zeigt die Definition einer Struktur für ein dynamisches Design mit vier dynamischen Regressoren (*REG-NEGDYN*, *REG-NEGSTAT*, *REG-NEUTDYN* und *REG-NEUTSTAT*). Im Gegensatz zu bisherigen Designs ist hier nur die maximale Länge der Regressoren definiert, das Einfügen der eigentlichen Events erfolgt zur Laufzeit über die Zuordnung von Transferfunktionen zu Medienobjekten.

Die Übertragung der Stimulationsevents in die Designmatrix erfolgt über die sogenannten Transferfunktionen. Im einfachsten Fall definiert eine solche Transferfunktion eine direkte Übertragung der Stimulationsattribute in die Designmatrix, das heißt die Onsetzeit und die Dauer werden als Attribute für das Event verwendet.

```

<transferFunctions>
  <transferFunction transferFunctionID="tranF1" timeOffset="0"
    durationScaleFactor="0" parametricScaleFactor="1"/>
</transferFunctions>

```

Auflistung 3.7: Der EDL-Ausschnitt zeigt die Beispielformatierung einer Transferfunktion für die Erzeugung eines Eintrags in der Designmatrix aus einem Event der Stimulation.

Eine solche Transferfunktion kann in einem Medienobjekt zur Zuordnung zu einem dynamischen Regressor verwendet werden. Dazu wird die ID des Regressors und der Transferfunktion im Element *regressorAssignment* referenziert. Das Medienobjekt im Beispiel 3.8 ist damit durch den *Constraint col* eingeschränkt. Das bedeutet in diesem Kontext, dass das Event nur präsentiert wird, wenn der Proband in der Mitte des Bildschirms fixiert. Zum Zeitpunkt des tatsächlichen Events wird dann in der Designmatrix im Regressor *REG_NEGDYN* ein Eintrag hinzugefügt, der mit der Transferfunktion *tranF1* die Onsetzeit, Dauer und den Skalierungsfaktor des Events berechnet. Das Hinzufügen des Eintrags löst eine Aktualisierung der Designmatrix aus, damit diese sofort für die Datenauswertung verwendet werden kann.

```

<mediaObject moID="BildID1" name="1019" type="IMAGE" useConstraint="col" >
  <contentImage>
    <imageFile>pics/1019.jpg</imageFile>
    <posX>0</posX>
    <posY>0</posY>
  </contentImage>
</mediaObject>

```

```
</contentImage>  
<regressorAssignment assignToRegressor="REG_NEGDYN" useTransferFunction="tranF1"/>  
</mediaObject>
```

Auflistung 3.8: Das Beispiel aus einem EDL-Dokument zeigt die Verwendung einer Transferfunktion tranF1 und einer Regressorzuweisung als Element in einem Medienobjekt. Die Präsentation dieses Medienobjekts in einem Stimulusevent ist bedingt durch den Constraint col. Nach der Präsentation wird mit Hilfe der Transferfunktion tranF1 ein Event erzeugt, der in dem Regressor REG_NEGDYN eingefügt wird und die Designmatrix für die Auswertung aktualisiert.

3.3.7.2 Aktualisierung der Designmatrix in Echtzeit

Mit Hilfe der Transferfunktionen wird zum Zeitpunkt einer Änderung im Paradigma mit einer Notification die Neuberechnung der Designmatrix ausgelöst. Diese Berechnung beinhaltet die Neuberechnung der Faltungsoperation, die parallelisiert ausgeführt wird. Die Berechnung benötigt 15 ms (siehe Abschnitt 3.3.4.3).

3.4 Zusätzliche Entwicklungen im Rahmen dieser Arbeit

Neben dem beschriebenen Softwarepaket BART sind noch drei weitere Projekte im Rahmen der vorliegenden Arbeit entstanden, die einen wichtigen Beitrag für die Arbeitsweise des Softwarepakets BART leisten. Daher werden diese in den folgenden Abschnitten kurz erläutert.

3.4.1 EDLEditor

Aufgabenstellung: Wie bereits beschrieben ist die Sprache EDL (siehe Abschnitt 3.2.3) sehr umfangreich und beinhaltet neben der strukturierten Beschreibung eines fMRT-Experiments auch eine Wissensrepräsentation, um Abhängigkeiten der Parameter untereinander zu prüfen. Das Erstellen von EDL-Dokumenten mit einem unspezifischen Editor ist allerdings sehr aufwendig und fehleranfällig, da alle XML-Beschreibungselemente manuell eingegeben werden müssten. Dabei wäre ein zielgerichtetes Arbeiten nur dann möglich, wenn der Autor eines EDL-Dokumentes bereits sehr gut mit dem EDL-Schema vertraut ist. Außerdem müsste sowohl die Wohlgeformtheit des Dokuments als auch die Überprüfung der Regeln gleichzeitig zusätzlich getestet werden. Um das Erstellen von EDL-Dokumenten zu vereinfachen, wurde daher ein graphischer Editor für das einfache Erstellen wohlgeformter und gültiger EDL-Dokumente implementiert.

EDLEditor: Für die Erstellung der beschriebenen EDL-Konfigurationsdateien für Echtzeit-fMRT-Experimente mit der Software BART hat Oliver Zscheyge im Rahmen seiner Bachelorarbeit einen Editor für die EDL erarbeitet und implementiert (Zscheyge 2010). In diesem Editor werden alle Elemente und Attribute des EDL-Schemas abgebildet. Damit ist der Editor flexibel gegenüber Änderungen, da das gesamte Schema automatisch im Editor übernommen wird. Zusätzlich prüft der Editor zur Eingabezeit alle definierten Regeln des Schemas und gibt im Fehlerfall eine Rückmeldung über bestehende Konflikte. Der Editor bietet eine vereinfachte Eingabemöglichkeit für Stimulationsprotokolle und kann diese auch testen. Weiterhin können die durch das Paradigma definierte Designmatrix, die Orthogonalität und die für das Analysemodell verwendeten HRF-Funktionen angezeigt werden. Die Abbildung 3.11 zeigt diese Funktionalitäten an einem Beispiel. Die Sprache EDL ist unabhängig vom Softwarepaket BART und wird auch im Softwarepaket rtExplorer verwendet. Der Editor wurde plattformunabhängig implementiert, damit die Konfiguration für ein Echtzeit-fMRT-Experiment unabhängig vom Betriebssystemen erzeugt werden kann.

3.4.2 Echtzeitexport

Aufgabenstellung: Eine essentielle Komponente für die Echtzeitauswertung der fMRT-Daten ist die Möglichkeit während der MR-Messung auf die Daten zugreifen zu können. Der hier verwendete MR-Scanner der Firma Siemens bietet einen Export der Daten zur Laufzeit der Messung an. Dieser kann mit der hier vorgestellten Software zusammen verwendet werden. Allerdings hat dieser Export Nachteile bei der Übertragungszeit der Daten. Daher wurde in Zusammenarbeit mit Torsten Schlumm von der NMR-Arbeitsgruppe des Max-Planck-Instituts für Kognitions- und Neurowissenschaften Leipzig eine Eigenentwicklung realisiert.

Zum Verständnis wird der Aufbau des Siemens Scannersystems genauer beschrieben. Wie in Abbildung 3.3 schematisch dargestellt, besteht das hier verwendete Setup des Siemens TIM TRIO Scanners aus drei Systemkomponenten: (1) dem MR-Scanner, der die Rohdaten des MR-Signals an einen Rekonstruktionsrechner (MRIR) weiter sendet. Auf diesem ist ein UNIX-Betriebssystem installiert und die Rekonstruktion der Daten zu MR-Bildern wird auf diesem Rechner berechnet. Der MRIR sendet die rekonstruierten Daten weiter an den (2) Hostrechner. Dieser wiederum hat ein Windows-Betriebssystem und die Aufgabe, die Daten in eine Datenbank zu schreiben und für den Bediener an der (3) Konsole anzuzeigen. Der intern von Siemens zur Verfügung gestellte Export kann die Daten vom Hostrechner auf eine Festplatte kopieren, die in das Netzwerk der Systemkomponenten eingebunden wurde. Nachteilig ist dabei, dass dies ein Datelexport ist. Das bedeutet, dass vollständig generierte Dateien erst auf die eigene Festplatte geschrieben und dann auf den externen Rechner über das Netzwerk kopiert werden. Das Schreiben erfolgt erst nach dem Ablegen der Daten in der internen Datenbank. Die Verbindung des Hostrechners zu einem externen Rechner, der nicht mit dem Windows-Betriebssystem aus-

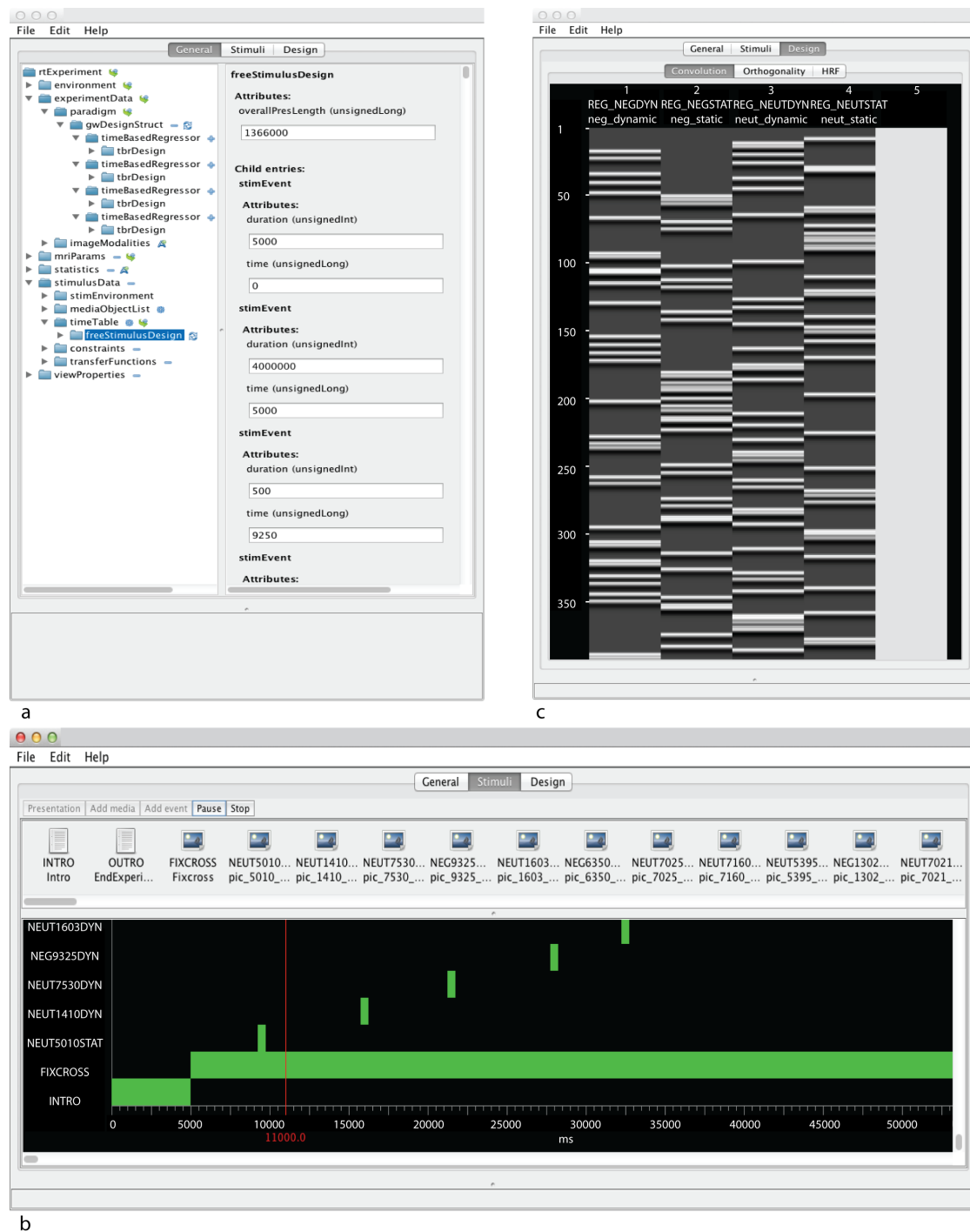


Abbildung 3.11: Screenshots vom Editor für EDL-Dokumente: Übersicht über Struktur des EDL-Dokuments und Eingabe der Parameter mit automatischer Fehlerkontrolle (Regelvalidierung) (a); Testumgebung und vereinfachte graphische Eingabemöglichkeit für die Stimulationsprotokolle (b); Graphische Darstellungen der enthaltenen Designmatrix, Orthogonalität der Designmatrix und der verwendeten HRF-Modellfunktionen (c).

gestattet ist, erfordert zusätzliche Software (hier wurde die Software Samba³⁶ verwendet), deren zeitliches Verhalten nicht beeinflussbar ist. Der externe Rechner, der die Daten weiter verarbeitet, muss das Empfangen der Daten auf seiner Festplatte registrieren. Dazu muss dieser in sehr kurzen zeitlichen Abständen das Dateisystem abfragen, ob neue Daten kopiert wurden, und diese vor der weiteren Verwendung in den Arbeitsspeicher einlesen.

Alle genannten Schritte kosten Zeit und erzeugen Prozessorlast auf den Rechnern. Tests mit dem internen Export haben gezeigt, dass die genannte Konstellation funktioniert, jedoch unregelmäßige Verzögerungen in der Datenübertragung mit sich bringt, deren Ursache nicht genau geklärt werden konnte. Insbesondere ist jedoch der Export über das Dateisystem der verschiedenen Rechner mit einem zeitlichen Mehraufwand für das Schreiben und Lesen der Daten verbunden, der vermieden werden kann.

Eigenimplementierung Echtzeit-Export: Aus diesen Gründen wurde ein eigener Echtzeit-Export implementiert, der die Daten direkt vom Arbeitsspeicher des Rekonstruktionsrechners über das Netzwerk in den Arbeitsspeicher des externen Auswerterechners überträgt. Dies ist in Abbildung 3.3 dargestellt. Die Implementierung erfolgte in dem Siemens Entwicklerframework IDEA³⁷ und kann damit in der Datenverarbeitungskette von Siemens eingesetzt werden. Die Vorteile dieses Exports sind die direkte Datenübertragung aus und in den Arbeitsspeicher, wodurch das Schreiben und Lesen der Daten auf den Festplatten der Rechner entfällt. Es entfällt ebenfalls der Einsatz einer zwischengeschalteten Software zur Übertragung der Daten. Außerdem ist es nicht mehr notwendig, die Festplatte des externen Rechners in das interne Netzwerk zu integrieren. Der Export kann in der Sequenz des Scanners an- und ausgeschaltet werden. Die Abbildung 3.12 zeigt den Vergleich der Exportzeiten jeweils bei der Messung eines Phantoms und eines Probanden mit einer EPI-Sequenz für die TR-Zeiten 1,5 s, 2 s, 3 s und 5 s. Die Eigenimplementierung reduziert die Exportzeit der Daten auf 40 ms bis 60 ms im Vergleich zu Exportzeiten des internen Exports von 150 ms bis 450 ms. Die Unterschiede der beiden Exportvarianten sind in Tabelle 3.4 zusammengefasst. Eine Anmerkung ist an dieser Stelle notwendig: Die periodisch auftretenden negativen Zeitmessungen entstehen dadurch, dass die Aufnahme der Schichten äquidistant auf die TR-Zeit aufgeteilt werden. Dabei kann am Ende einer TR-Zeit eine Pause bis zum Beginn der nächsten Repetition entstehen, die länger dauert als der Export der Daten. Da der Export die Daten bereits zu diesem Zeitpunkt sendet und die Exportzeit aus der Differenz der Empfangszeit der Daten, der Triggerzeit des Scanners und der TR-Zeit berechnet wird, errechnen sich dadurch negative Werte.

³⁶ <http://www.samba.org/>

³⁷ <https://www.mr-idea.com>

	interner Export Siemens	Eigenimplementierung Echtzeitexport
Export von	Festplatte Hostrechner	Arbeitsspeicher Rekonstruktionsrechner
Export nach	Festplatte BART-Rechner	Arbeitsspeicher BART-Rechner
Export über	Netzwerk mit Hilfe zusätzlicher Software	Netzwerk direkt (TCP/IP Protokoll)
Empfang der Daten	Verzeichnis dauerhaft überwachen und bei Bedarf Daten lesen	direktes Warten an Netzwerkempfangsport
Konfiguration	externe Festplatte in Siemens Netzwerk einbinden	Ein- und Ausschalten in der Sequenz
Exportzeit	150-450 ms	40-60 ms

Tabelle 3.4: Die Tabelle fasst die Unterschiede des hier neu entwickelten Echtzeitexports gegenüber dem bereits vorhandenen internen Exports von Siemens zusammen.

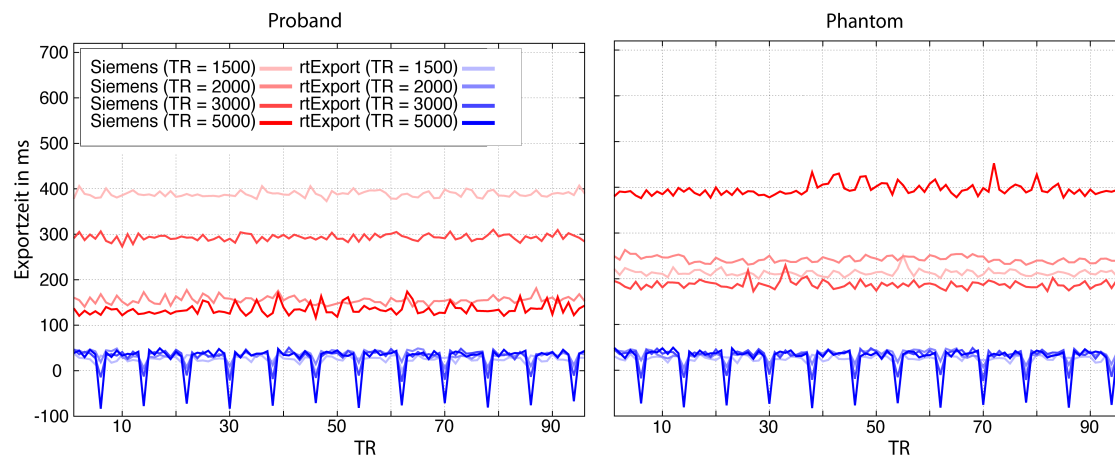


Abbildung 3.12: Die Abbildung zeigt den Vergleich der Exportzeiten für die fMRT-Daten zwischen dem internen Export von Siemens und der Eigenimplementierung für die Messung mit Probanden (links) und Phantom (rechts). Die Eigenimplementierung (blau) exportiert die Daten unabhängig von der TR in ca. 50 ms auf den Auswerterechner (Die negativen Ergebnisse entstehen durch eine Pause am Ende der TR). Der interne Export (rot) benötigt je nach TR und Messobjekt Zeiten von 150 ms bis 450 ms. Außerdem treten unregelmäßige Verzögerungen auf.

3.4.3 Bibliothek *isis*

Aufgabenstellung: Für (f)MRT-Bilddaten gibt es eine Vielzahl von Anwendungsfeldern, die von der klinischen Diagnostik über experimentelle Auswertung bis hin zur physikalischen Methodenentwicklung reichen. Insbesondere in der Forschung werden fortlaufend neue Varianten der Datenauswertung entwickelt, die alle dem gleichen Prinzip folgen. Die Daten müssen eingelesen, im Arbeitsspeicher gehalten und verarbeitet werden, um abschließend die Ergebnisse als Dateien schreiben zu können. Dabei ist allerdings die Verarbeitung der Daten individuell zu lösen. Das Lesen und Schreiben der Daten und der Zugriff auf diese während der Verarbeitung sind konstante Anforderungen. Obwohl seit längerem versucht wird, Standardformate für solche Daten durchzusetzen, sind die vorhandenen Standards DICOM (*Digital Imaging in Communication and Medicine*) und Nifti (*Neuroimaging Informatics Technology Initiative*) noch nicht als endgültige Versionen zu betrachten. Verschiedene Hersteller von MR-Scannern bzw. Softwareentwickler von verschiedenen existierenden Softwarepaketen haben eigene Interpretationen in die Standards eingebracht. Diese Konflikte werden sich eventuell durch weitere Entwicklungen lösen, nichtsdestotrotz sind viele spezifische Fachkenntnisse für den korrekten Umgang mit den jeweiligen Formaten notwendig.

Die Basisbibliothek *isis*: In Zusammenarbeit mit Enrico Reimer, Erik Türke und Thomas Pröger aus der Abteilung Neurophysik des Max-Planck-Instituts für Kognitions- und Neurowissenschaften Leipzig wurde eine Basisbibliothek für das Management von fMRT-Daten unter dem Namen *isis* entwickelt. Alle Beteiligten dieser Kooperation hatten zum gleichen Zeitpunkt unterschiedliche Aufgaben mit (f)MRT-Daten zu lösen. Aus der ergebnislosen Suche nach einer allgemeinen Lösung für die genannten Grundanforderungen entstand die Idee der gemeinsamen Erstellung einer Basisbibliothek für ein performantes Datenmanagement. Durch eine Vereinheitlichung der Schnittstellen zum Lesen und Schreiben der verschiedenen Dateiformate ist eine effiziente Lösung entstanden. Damit wurden Redundanzen in der Softwareentwicklung vermieden und individuelle Fehler minimiert. Das Softwarepaket BART war von Anfang an eine Anwenderapplikation der entwickelten Bibliothek *isis* und hat die Anforderungen an die Performanz des Datenhandlings mitbestimmt.

Die Bibliothek *isis* bietet ein effizientes Management der Daten bei der Verwendung in Algorithmen. Das heißt, dass die Zugriffszeiten minimal sind, eine einheitliche Struktur der Daten angelegt wird und alle Eigenschaften der Daten verwaltet werden. Die gängigen Formate (DICOM, nifti, vista) können gelesen und geschrieben werden. Die Datenformate sind durch ein Pluginkonzept für die Formate beliebig erweiterbar. Das Lesen der Daten des Echtzeitexports für die Software BART konnte somit hinzugefügt werden. Außerdem besitzt die Basisbibliothek Adapter zur Verwendung externer Softwarepakete, wie z. B. ITK. Die Bibliothek steht unter der freien Softwarelizenz GPL für die Verwendung zur Verfügung und die Quellen und Dokumentation finden sich unter <https://github.com/isis-group/isis>.

3.5 Von der Softwareentwicklung zur Anwendung in fMRT-Experimenten

Das in dieser Arbeit entwickelte Softwaresystem BART mit den vorgestellten Modulen zur adaptiven Stimulation der Probanden und Echtzeitauswertung wurde am Max-Planck-Institut für Kognitions- und Neurowissenschaften Leipzig an einem Siemens TIM TRIO MR-Scanner aufgebaut und im experimentellen Teil zur Durchführung von fMRT-Experimenten verwendet. Die im Folgenden beschriebenen Experimente validieren die Funktionalität des neu entwickelten Softwaresystems in Bezug auf die adaptiven Paradigmen, die Echtzeitauswertung und die Kombination dieser beiden Module. Die dazu durchgeführten Experimente werden im folgenden Kapitel methodisch erläutert und ihre Ergebnisse präsentiert.

4 Methoden und Ergebnisse: fMRT Experimente mit dem Softwarepaket BART

Dieser Abschnitt umfasst den empirischen Teil der vorliegenden Arbeit. In diesem wurde das neu entwickelte und in Kapitel 3 detailliert erläuterte Softwarepaket zur Durchführung von fMRT-Experimenten verwendet. Um die Funktionsweise der beiden Hauptmodule und deren Kombination zu erklären, werden im Folgenden drei Experimente beschrieben. Für das bessere Verständnis wird zuerst eine Metaanalyse zu fMRT-Experimenten im Kontext von Emotionsverarbeitung vorgestellt. Anschließend wird ausführlich die hier durchgeführte Studie beschrieben, die ein Paradigma mit konventioneller und adaptiver Stimulation vergleicht. Die Ergebnisse belegen die Vorteile durch die Adaption. Im folgenden Abschnitt werden experimentelle Ergebnisse der hier implementierten Echtzeitanalyse mit den Ergebnissen eines konventionellen Softwarepakets verglichen. Zuletzt werden in diesem Kapitel exemplarisch die Ergebnisse eines Experiments mit der Kombination der beiden Module, adaptive Stimulation und Echtzeitverarbeitung, gegenüber einer konventionellen Auswertung validiert.

4.1 Emotionale Netzwerke in fMRT-Experimenten

Die vorliegende Arbeit hat bisher ein Konzept für ein Softwaresystems für Echtzeit-fMRT und adaptive Paradigmen beschrieben. Im experimentellen Teil wird dieses Softwarepaket angewendet, um ein häufig eingesetztes Paradigma aus der Emotionsforschung mit einem neuen, adaptiven Paradigma zu optimieren. Daher werden hier eine Metaanalyse zu diesem Anwendungsfeld vorgestellt und wichtige Begriffe eingeführt. Für Experimente mit Emotionen werden am häufigsten visuelle Stimuli eingesetzt, die verschiedene Emotionen bei den Probanden auslösen sollen. Die Bilder zeigen entweder Gesichter mit unterschiedlichen Ausdrücken oder Szenen, die bestimmten Emotionen zugeordnet sind. Die Unterteilung erfolgt in negative Emotionen (Trauer, Wut, Ekel und Angst) mit Bildern von u.a. Leichen, Gewalt oder Spinnen und positive Emotionen (Lachen, Feiern, freudige Ereignisse) mit Bildern von Babys, Tierkindern bis hin zu erotische Aufnahmen. Als Kontrollbedingung für die Berechnung der Kontraste zwischen den

Bedingungen werden neutrale Bilder verwendet, die alltägliche Dinge zeigen wie Landschaften, Bücher oder Haushaltsgegenstände. Die Qualität und Eigenschaften der Bilder müssen dabei vergleichbar sind. Eine beliebte Quelle für solche Bilder ist ein standardisiertes Bildarchiv, das sogenannte International Affective Picture System (IAPS³⁸). Diese Bilderdatenbank besteht aus mehreren hundert Bildern und wird seit fast zwei Jahrzehnten von Lang et al. (2008) bereitgestellt und aktualisiert. Die Vorteile sind die dazu erhältlichen Informationen über wichtige Bildeigenschaften und eine allgemeingültige Bewertung der Bildinhalte. Die zwei wichtigsten Parameter für diese Bewertung sind: (1) die Wertigkeit (Valenz), das heißt in welche der Kategorien 'negativ', 'neutral' oder 'positiv' das Bild eingeordnet wird und (2) die skalare Bewertung der dadurch ausgelösten Erregung beim Probanden unabhängig von der Kategorie. Die Bewertungsskalen dazu finden sich im Anhang A.3.1.

Eine Vielzahl an Emotionsstudien mit fMRT wurde bereits durchgeführt und eine Metaanalyse dazu, die 154 Experimenten aus den Jahren 1995 bis 2009 einbezieht, wurde von Sabatinelli et al. (2011) vorgestellt. Die Vielschichtigkeit emotionaler Verarbeitung im menschlichen Gehirn ist noch nicht vollständig geklärt. Bisher existieren dazu mehrere widersprüchliche Theorien. Diese sind jedoch nicht Bestandteil dieser Arbeit. Die in Abbildung 4.1 dargestellten Ergebnisse der Metaanalyse sind ausreichend für das Verständnis des nachfolgend durchgeführten Experiments. Dabei wurden die ausgewählten Experimente je nach Stimuli, emotionale Szenen (rot) oder Gesichter (blau), eingeteilt. Hirnareale mit erhöhter neuronaler Aktivität sind dabei der mediale präfrontale Kortex (mPFC), der orbitofrontale Kortex (OFC), der inferiore frontale Gyrus (IFG), die Insula, der mittlere frontale Gyrus (MFG), der anteriore cinguläre Kortex (ACC), die Amygdala, der mediale dorsale Nucleus des Thalamus (MDNT), der parahippocampale Gyrus, das Pulvinar, der fusiforme Gyrus (FFG), der laterale okzipitale Kortex (LOC) und der posteriore fusiforme Gyrus. Dabei sind die posterioren Regionen wie der FFG und der LOC für die Wahrnehmung von Objekten und Gesichtern verantwortlich, die inferioren Regionen des limbischen Systems, wie IFG und Amygdala sind dagegen bei der Wahrnehmung, Unterscheidung und Bewertung von Emotionen aktiv.

³⁸ <http://csea.phhp.ufl.edu/Media.html>

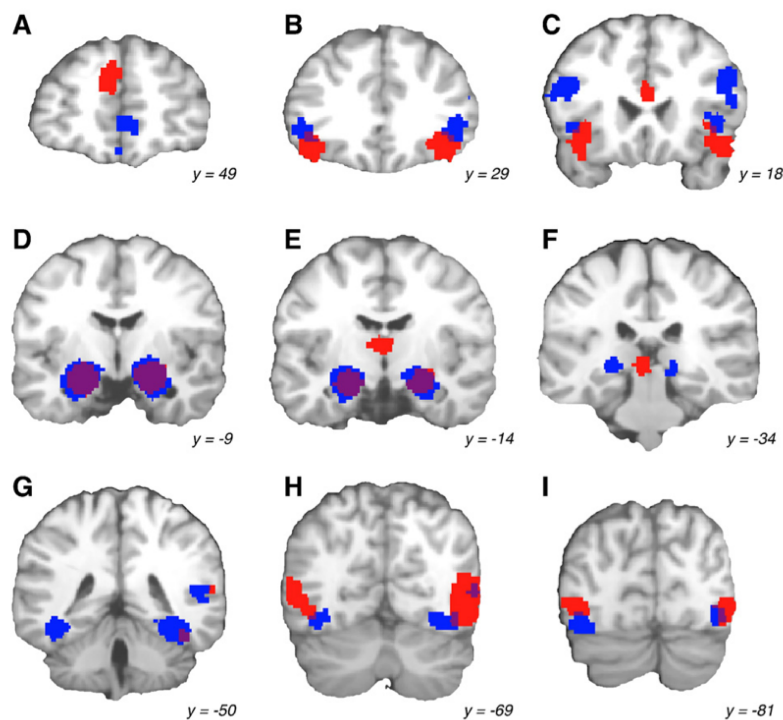


Abbildung 4.1: Die Ergebnisse einer Metaanalyse von Sabatinelli et al. (2011) basierend auf 154 fMRT-Experimenten unter Verwendung von emotionalen Stimuli zeigen Hirnareale die bei der Verarbeitung von Gesichtern mit emotionalen Ausdrücken (blau) oder emotionaler Szenen (rot) stärker aktiv sind. Die beteiligten Hirnareale sind (A) der mediale präfrontale Kortex (mPFC); (B) der orbitofrontale Kortex (OFC) und inferiore frontale Gyrus (IFG) bilateral; (C) der mittlere frontale Gyrus (MFG) bilateral, der anteriore cinguläre Kortex (ACC); (D) die Amygdala bilateral; (E) der mediale dorsale Nucleus des Thalamus (MD-NT); (F) der parahippocampale Gyrus bilateral, das Pulvinar; (G) der fusiforme Gyrus (FFG) bilateral; (H) der laterale okzipitale Kortex (LOC) bilateral und (I) der posteriore fusiforme Gyrus bilateral.

4.2 Experiment I: fMRT-Experiment mit adaptivem Paradigma für emotionale Stimuli

Dieser Abschnitt beschreibt den Hauptteil der empirischen Arbeit. In diesem Experiment wird die Erweiterung eines emotionalen Paradigmas mit der Adaption durch Eyetrackerinformationen vorgestellt. Dazu wurden 17 Probanden Bilder aus der IAPS-Datenbank konventionell und adaptiert präsentiert. Details zu diesem Experiment und der Vergleich der Ergebnisse sind im Folgenden beschrieben. Die hier vorgestellten Daten sind in englischer Sprache in einer Fachzeitschrift publiziert (Hellrung et al. 2015).

4.2.1 Pilotexperiment

Die Idee zu der Studie entstand während der Vorbereitung eines Experiments, in dem ebenfalls IAPS-Bilder präsentiert werden sollten. Dabei wurde ein heterogenes Verhalten der Probanden in Bezug auf ihre Augenbewegungen beobachtet. Daraufhin wurde ein Pilotexperiment durchgeführt, bei dem mit einem Eyetracker die Augenbewegungen von 7 Teilnehmern (Altersdurchschnitt: $25,4 \pm 2,9$ Jahre, 3 Männer) während der Durchführung des Experiments beobachtet wurde. Die Probanden waren explizit instruiert, ihren Blick zur Mitte des Bildschirms auf ein Fixationskreuz zu richten, das zwischen den Bildpräsentationen angezeigt wurde. Außerdem wurden die Probanden angewiesen, per Tastendruck anzugeben, ob ein negatives oder neutrales Bild wahrgenommen wurde. Die Analyse der Eyetrackerdaten ergab, dass die Probanden trotz der expliziten Anweisung in durchschnittlich 22 ± 13 % der Bildpräsentationen ihren Blick nicht auf die Mitte des Bildschirms gerichtet hatten. Außerdem ergab die Analyse der Antwortzeiten, dass die Probanden für die fixierten Bilder signifikant schneller geantwortet hatten (fixierte Bilder: $1,24 \pm 0,26$ s; nicht-fixierte Bilder: $1,34 \pm 0,31$ s; $p < 0,048$). Aus diesem heterogenen Verhalten der Probanden wurde das Konzept eines adaptiven Paradigmas abgeleitet, das die Stimulation in Bezug auf die Augenbewegungen der Probanden adaptiert.

4.2.2 Hypothesen

Aus den Erkenntnissen des Pilotexperiments wurden vier Hypothesen abgeleitet. Zur Überprüfung dieser Hypothesen werden Verhaltensparameter und fMRT-Daten verglichen, die während der adaptiven und konventionellen Stimulation von Probanden mit emotionalen Bildern aus der IAPS-Datenbank ermittelt wurden. Folgende Hypothesen wurden überprüft:

- Adaptive Präsentation der Stimuli führt zu weniger Augenbewegungen bei den Probanden.
- Adaptive Präsentation der Stimuli führt zu einer besseren Erkennung des präsentierten Inhalts.
- Adaptive Präsentation der Stimuli führt zu einem homogeneren Verhalten der Probanden.
- Adaptive Präsentation der Stimuli führt zu einem größeren Anstieg in der Signalstärke in Hirnarealen, die bei der Emotionsverarbeitung relevant sind.

4.2.3 Material und Methoden

4.2.3.1 Probanden

An dem Experiment haben insgesamt 21 gesunde Probanden, die weder Brille noch Kontaktlinsen benötigen, teilgenommen. Bei 4 Probanden wurde das MR-Experiment abgebrochen, da aufgrund von Kopfbewegungen der Eyetracker kein Signal erfassen konnte. Dadurch konnte das Experiment nicht bis zum Ende gebracht werden. Es wurden somit Datensätze von insgesamt 17 Probanden ausgewertet (davon 10 männlich; Altersdurchschnitt $26,6 \pm 4,2$ Jahre). Die Probanden haben vor dem Experiment explizit in die Teilnahme eingewilligt, nachdem ihnen vergleichbare, aber nicht im Experiment verwendete, negative IAPS-Bilder gezeigt wurden. Die Probanden hatten mindestens 3 Monate vor dieser Studie an keinem Experiment teilgenommen, bei dem dasselbe Stimulusmaterial aus der IAPS-Datenbank verwendet wurde. Die Studie wurde durch die Ethikkommission der Medizinischen Fakultät der Universität Leipzig in Anlehnung an die Deklaration von Helsinki genehmigt.

4.2.3.2 Stimulusmaterial

Als Stimulusmaterial wurden 60 negative und 60 neutrale Bilder in einer modifizierten Form aus der IAPS-Datenbank verwendet. Die ausgewählten Bilder wurden jedem Probanden zweimal präsentiert, jeweils einmal in der adaptiven und der nicht-adaptiven Bedingung. Ein Abgleich der Bildmerkmale- und inhalte zwischen diesen beiden Bedingungen war aus diesem Grund nicht notwendig. Die Modifikation der originalen IAPS-Bilder bestand darin, mit einem Bildbearbeitungsprogramm den für die Erfassung des Bildes relevanten Teil in der Bildmitte zu zentrieren, wenn notwendig irrelevante Teile zu entfernen und einen kreisrunden Ausschnitt

der Bildmitte auf einem grauem Hintergrund zu platzieren. Diese Bildmodifikation wurde im Rahmen einer Studie am Max-Planck-Institut für Kognitions- und Neurowissenschaften Leipzig erarbeitet und separat an 40 Probanden validiert. Die Modifikation hat die Bewertung von Probanden für die Valenz und durch die Bilder ausgelöste Erregung nicht verändert. Der Korrelationskoeffizient nach Pearson beträgt dabei für die Valenz: $r = 0,99$ und die Erregung: $r = 0,95$. Die Daten werden zur Zeit von Hellrung et al. zur Publikation vorbereitet.

Die ausgewählten Bilder wurden zwischen negativen und neutralen Bildern auf ihre Bildeigenschaften Farbe, Kontrast und auf Vergleichbarkeit abgeglichen. Die Bilder wurden pseudorandomisiert zwischen einerseits negativ und neutral und andererseits adaptiert und nicht-adaptiert präsentiert. Bei emotionalen Stimuli ist zu beachten, dass die Reaktion bei wiederholter Präsentation unterschiedlich stark ausfallen und die Erstpräsentation eine wesentlich stärkere emotionale Reaktion auslösen kann. Da alle Bilder zweimal präsentiert wurden, war ein Abgleich notwendig bezüglich der jeweiligen Valenz der Erstpräsentation der negativen Stimuli zwischen adaptierten und nicht-adaptierten Stimuli. Daher wurden 1000 Pseudorandomisierungen der Bilder berechnet und für jede dieser Randomisierungen die Differenz für die Valenz zwischen adaptierten und nicht-adaptierter Bedingung bestimmt. Daraus wurden die Randomisierungen ausgewählt, bei denen die Differenz $< 0,01$ und die Standardabweichung $< 0,1$ betrug. Für jeden Probanden wurde eine individuelle Randomisierung ausgewählt.

4.2.3.3 Paradigma

Das Paradigma beinhaltete die Präsentation der insgesamt 240 Stimuli über einen Beamer, der die Bilder in der Bildschirmauflösung von 800 x 600 Pixel auf die Leinwand im Scannerraum projiziert hat. Jedes Bild wurde für 500 ms gezeigt und zwischen den Bildern wurde ein Fixationskreuz in der Mitte des Bildschirms projiziert. Das Intervall zwischen den Bildpräsentation (Interstimulusintervall) lag für die nicht-adaptiven Stimuli zwischen 4,25 s und 6,25 s. Bei den adaptiven Stimuli wurde ein Mindestintervall 4 s bis 6 s festgelegt, das jedoch durch eventuelle Adaptionen entsprechend verlängert wurde. Das Paradigma ist in Abbildung 4.2 schematisch dargestellt.

Für die adaptiven Stimuli wurde zur Laufzeit des Experiments mit Hilfe des Eyetrackers geprüft, ob der Proband zum Zeitpunkt der eigentlichen Bildpräsentation den Blick in der Mitte des Bildschirms fixiert. Nur in diesem Fall wurde das Bild angezeigt. Andernfalls wurde weiterhin das Fixationskreuz gezeigt und der Stimuli und alle noch folgenden Stimuli um 8 ms in ihrer Präsentationszeit verschoben (dies entspricht der Sampling Rate von 120 Hz des Eyetrackers). Nach dieser Zeit wurde erneut geprüft, ob der Proband fixiert und die entsprechende Aktion ausgelöst. Die nicht-adaptiven Stimuli wurden völlig unabhängig vom Verhalten des Probanden immer zum definierten Zeitpunkt angezeigt.

Zur Überprüfung der Fixation wurden die Eyetrackerdaten während der Laufzeit des Experiments ausgewertet. Um eine Fixation zu identifizieren wurde ein Algorithmus in Anlehnung an Salvucci et al. (2000) implementiert, der als *Dispersion-Threshold Identification* bezeichnet wird. Dabei wird eine Fixation anhand von zwei Parametern - Zeit und Abstand - bestimmt. Das heißt, dass für einen definierten Zeitraum der Abstand zwischen aufeinanderfolgenden Blickpositionen nicht überschritten werden darf. Für die vorliegende Arbeit wurde der Zeitraum mit 200 ms und für den Abstand ein Winkel von 1° festgelegt. Zusätzlich zu diesem Kriterium wurde festgelegt, dass eine ermittelte Fixation nur gültig ist, wenn diese weniger als 2° von der Bildschirmmitte entfernt lag. Dieses Kriterium wurde gewählt, da die verwendeten, modifizierten IAPS-Bilder eine Größe von ungefähr 3° besitzen und der wichtigste Bildinhalt in der Mitte zentriert wurde.

Die Probanden zeigten während des Experiments im Scanner durch Drücken einer Tastbox an, ob sie ein negatives oder neutrales Bild wahrgenommen haben. Im Anschluss an das fMRT-Experiment haben alle Probanden in einem Computerexperiment die Bilder noch einmal bezüglich Valenz und ausgelöster Erregung anhand der 9-Punkte Skala ausführlich bewertet (siehe im Anhang Abbildungen A.4 und A.3).

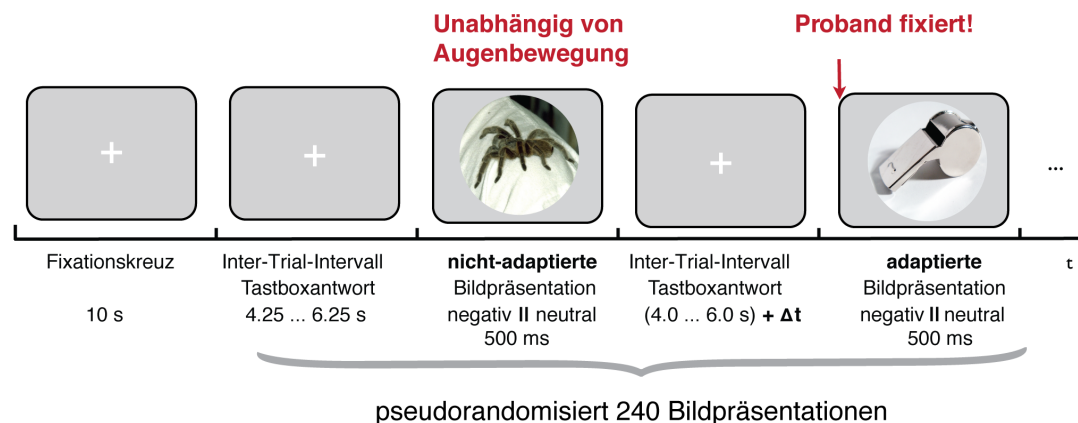


Abbildung 4.2: Paradigma: Die schematische Darstellung des im Experiment verwendeten Paradigmas, bei dem 60 negative und 60 neutrale Bilder jeweils einmal adaptiv und nicht-adaptiv für 500 ms präsentiert wurden. Bei den adaptierten Stimuli wurde während des Experiments die aktuelle Blickrichtung des Probanden ermittelt und der Stimulus nur präsentiert, wenn dieser die Bildschirmmitte fixiert hat.

4.2.3.4 Datenerfassung

Das Experiment wurde an einem Siemens TIM TRIO MR Scanner durchgeführt. Die Probanden wurden in der 12-Kanal Kopfspule mit einem Schaumstoffkissen und zusätzlichen Polstern fest fixiert um Bewegungsartefakte zu minimieren. Für die funktionellen Messungen wurde eine EPI-

Sequenz in Anlehnung an Robinson et al. (2004) zur verbesserten Signalerfassung in Hirnarealen des limbischen Systems, wie z.B. der Amygdala, mit folgenden Parametern definiert:

- 34 axiale Schichten mit einer Auflösung von je 64 x 64 Voxeln,
- TR-Zeit: 2 s; Echozeit: 28 ms,
- Schichtdicke: 2,5 mm und 10 % Abstand,
- Anzahl der gemessenen Volumen: individuell,
- Ausrichtung: ACPC-Ausrichtung + 5°; Flipwinkel: 90°; Bandbreite 1954 Hz; Field-of-View: 192.

Die Anzahl der Messungen war auf 2000 vordefiniert. Das Experiment wurde dann individuell nach der Präsentation aller Stimuli des Paradigmas gestoppt.

Simultan zu den MR-Daten wurden ebenfalls die Augenbewegungen der Probanden erfasst. Dazu wurde ein Eyetracker (ASL EyeTrac 6 Long Range Optics, Applied Science Laboratories, Bedford, USA) verwendet, dessen Infrarotkamera am Kopfende des MR Scanners positioniert wurde. Diese leitete infrarotes Licht über einen Spiegel in das Auge des Probanden und hat die daraus ermittelten Werte für den Pupillendurchmesser und Korneareflex an einen Controller des Eyetrackers gesendet. Dieser Controller wiederum wurde über eine USB-Verbindung direkt mit dem Auswerterechner verbunden auf dem die Eyetrackerdaten von BART ausgewertet wurden.

4.2.3.5 Datenauswertung

FMRT-Daten: Für die Auswertung der fMRT-Daten dieses Experiments wurde das Softwarepaket SPM in der Version 8 (SPM8)³⁹ verwendet. Diese Software ist keine eigenständige Applikation, sondern eine Toolbox die auf der MATLAB-Umgebung⁴⁰ basiert und aufgrund der umfangreichen enthaltenen Methodik sehr häufig für die Auswertung von fMRT-Daten verwendet wird (Friston et al. 2007). Die Daten wurden mit SPM8 vorverarbeitet, indem eine Korrektur von Magnetfeldinhomogenitäten (*Fieldmap Korrektur*), eine Schichtzeitkorrektur (*slice time correction*) und eine Bewegungskorrektur berechnet wurden. Außerdem wurden die Daten zur Berechnung der Gruppenstatistik auf den Referenzraum MNI normalisiert (Collins et al. 1994). Zusätzlich wurden die funktionellen Daten mit einem Gaußfilter von 8 mm Halbwertsbreite⁴¹ geglättet und mit einem Hochpass von 128 Hz gefiltert.

Für die anschließende statistische Auswertung der Daten wurden ein vierstufiger Ansatz mit folgenden Teilschritten gewählt:

- Für jeden Probandendatensatz wurde eine GLM-Analyse mit vier Regressoren als 2 x 2 Design (Emotion (negativ, neutral) und Art der Präsentation (adaptiert, nicht-adaptiert))

³⁹ <http://www.fil.ion.ucl.ac.uk/spm/>

⁴⁰ <http://www.mathworks.de/products/matlab/>

⁴¹ Full Width at Half Maximum (FWHM)

berechnet. Zusätzlich wurden die 6 Bewegungsparameter der Bewegungskorrektur aus SPM8 (3 Rotation, 3 Translation) als Kovariate in die Designmatrix eingefügt.

- Für die Gruppenstatistik über alle Probanden wurde ein *flexible factorial design* verwendet, wie es für das Softwarepaket SPM8 zur Verfügung steht. Damit wurden die Haupteffekte der Emotion und der Adaption bestimmt. Ein Schwellwert von $p < 0,001$ wurde als signifikant betrachtet.
- Für die Quantifizierung der Effekte durch die Adaption wurde für die Hirnareale, die für den Haupteffekt der Emotion gefunden wurden, eine ROI-Analyse mit der Software *rfxplot* für SPM8 berechnet (Gläscher 2009). Dazu wurde für die ermittelten Koordinaten im Umkreis von 3 mm am individuellen Maximum die durchschnittliche Effektstärke berechnet. Anschließend wurden die Ergebnisse in die beiden Gruppen der '*Low-Compliant*' und der '*High-Compliant*' unterteilt, um innerhalb der Gruppen den Einfluss der Adaption zu vergleichen.
- Für einen Vergleich der adaptierten Stimuli zwischen den beiden Probandengruppen wurde der Zweistichproben-t-Test aus SPM8 verwendet.

Eyetrackerdaten: Für die post-hoc Analyse der Eyetrackerdaten wurde der gleiche Algorithmus, wie in 4.2.3.3, beschrieben in der Skriptsprache Perl implementiert. Mit dieser Auswertung wurden die Performanz der Probanden bei nicht-adaptierten Bildpräsentationen quantifiziert und die Adaption zur Laufzeit validiert.

4.2.4 Ergebnisse

4.2.4.1 Behaviorale Ergebnisse

Alle Probanden bewerteten die präsentierten IAPS-Bilder erwartungsgemäß mit großer Übereinstimmung zu den vorhandenen Bewertungen aus einer Validierungsgruppe. Der Korrelationskoeffizient nach Pearson für die Valenz ist mit $r = 0,98$ ($p < 0,001$) und für die ausgelöste Erregung $r = 0,96$ ($p < 0,001$) sehr hoch (siehe Abbildung 4.3c).

Die post-hoc Auswertung der Eyetrackerdaten aller nicht-adaptierten Stimuli zeigt, dass die Probanden durchschnittlich nur bei $67 \pm 22\%$ aller Bildpräsentation den Blick auf die Mitte des Bildschirms fixierten. Dies bestätigt die Ergebnisse des Pilotexperiments. Aufgrund der großen Standardabweichung für die Fixationsleistung wurden die Probanden durch einen Mediansplit für die Fixationen in zwei Gruppen unterteilt. Für diese ergaben sich unterschiedliche Leistungen in der Fixation bei der nicht-adaptierten Bildpräsentation. Die '*High-Compliant*' fixierten im Durchschnitt in $83,0 \pm 6,2\%$ aller nicht-adaptierten Stimuli, die '*Low-Compliant*' hingegen nur in $49 \pm 20\%$ (siehe Abbildung 4.3a).

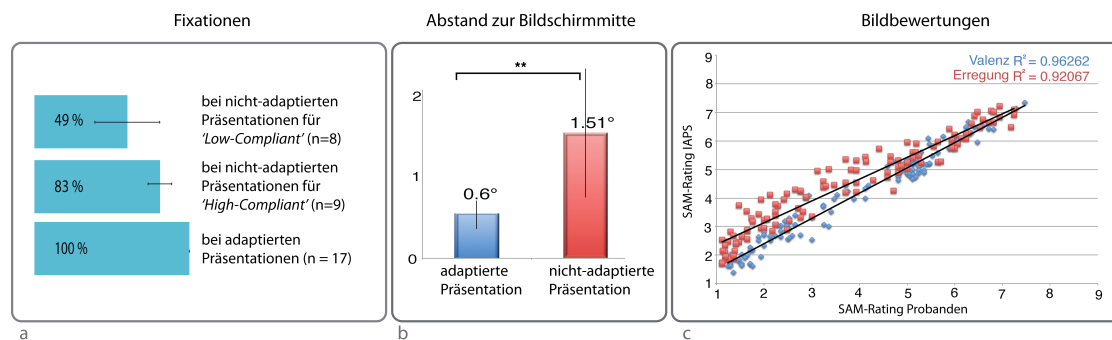


Abbildung 4.3: Die Abbildung zeigt die Verhaltensdaten der Probanden: aufgrund der Fixationsleistung wurden die Probanden mit einem Mediansplit in 'High-Compliant' und 'Low-Compliant' unterteilt (a); für die adaptierte Bildpräsentation war der Abstand zur Bildschirmmitte geringer im Vergleich zur nicht-adaptierten Präsentation (b); die Bewertung der Bilder durch die Probanden stimmt sehr gut mit den vorhandenen Validierungsdaten für die IAPS-Bilder überein (c).

Für die Augenbewegungen wurde post-hoc die Blickrichtung zum Zeitpunkt der Bildpräsentationen kontrolliert. Es zeigte sich, dass der Blickwinkel der Probanden für die nicht adaptierten Stimuli durchschnittlich $1,5^\circ$ von der Bildschirmmitte entfernt lag. Für die adaptierten Stimuli dagegen lag dieser Abstand bei nur $0,6^\circ$ (siehe Abbildung 4.3b).

Das Pilotexperiment hatte eine schnellere Reaktionszeit für adaptierte Bildpräsentationen vermuten lassen. Dieses Ergebnis konnte allerdings weder für die gesamte noch für die Teilgruppen bestätigt werden ($p = 0,47$). Allerdings ergab sich für die 'Low-Compliant' eine signifikante Verbesserung in Bezug auf den Anteil korrekter Antworten. Die Probanden konnten den Inhalt der Bilder besser als negativ oder neutral einordnen. Für die adaptierten Präsentationen wurden $92,0 \pm 2,6\%$ korrekte Antworten gegeben, jedoch nur $88,0 \pm 3,5\%$ der nicht-adaptierten Antworten waren korrekt ($p < 0,04$). Für die 'High-Compliant' wurde kein Unterschied beobachtet.

4.2.4.2 fMRT-Ergebnisse zu adaptiertem Paradigma

Die vier Regressoren der Designmatrix, welche der Berechnung der GLM-Analyse zugrunde liegen, sind wie folgt abgekürzt: negativ adaptiert (NGA); negativ nicht-adaptiert (NGNA); neutral adaptiert (NTA); neutral nicht-adaptiert (NTNA). In vier Auswertungsschritten werden hier die allgemeinen Effekte der Adaption und der Einfluss auf die Ergebnisse des emotionalen Paradigmas vorgestellt.

Haupteffekt der Emotion: Im ersten Schritt wurden die Unterschiede in der neuronalen Aktivität bezüglich negativer und neutraler Bildinhalte über alle Probanden untersucht (d.h. der Kontrast gebildet aus $([NGA,NGNA] - [NTA,NTNA])$). Die Ergebnisse der Gruppenstatistik über alle

Probanden, wie in Abbildung 4.4 dargestellt, zeigt eine erhöhte Aktivität in unterschiedlichen Hirnarealen, u.a. im medialen präfrontalen Kortex, in limbischen Regionen wie der Amygdala und dem FFG. Die exakten Koordinaten in Bezug auf den MNI-Atlas und die statistischen Prüf-werte (t-Werte und normierte z-Werte) sind in Tabelle 4.1 zusammengefasst. Die gefundenen Areale stimmen mit denen in der vorhandenen Übersichtsliteratur zu emotionalen Paradigmen überein, wie z.B. der in Abschnitt 4.1 vorgestellten Metaanalyse.

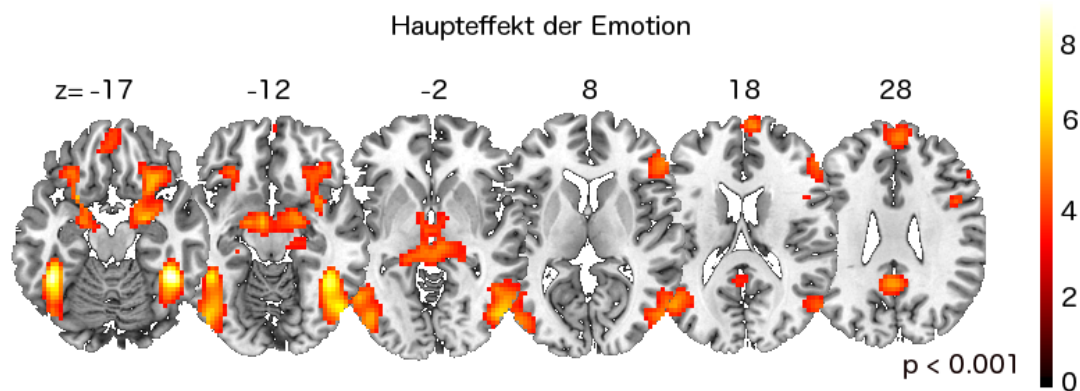


Abbildung 4.4: Die Abbildung zeigt die farblich kodierten Prüf-werte (t-Werte) der GLM-Analyse für den Haupteffekt der Emotion (negativ > neutral) für den eine erhöhte neuronale Aktivität im medialen präfrontalen Kortex (mPFC), in limbischen Regionen wie der Amygdala und dem Fusiformen Gyrus (FFG) sichtbar wird.

Haupteffekt der Adaption: Im zweiten Schritt wurde überprüft, ob die Adaption einen Einfluss auf die neuronale Aktivität hat. Dazu wurden die Regressoren mit [NGA,NTA]-[NGNA,NTNA] kontrastiert. Das heißt es wurde die Differenz zwischen allen adaptierten und konventionellen Bildpräsentationen berechnet. Dabei wurde die beschriebene Gruppenunterteilung der Probanden berücksichtigt. Für die 'High-Compliant', die eine sehr gute Fixationsleistung gezeigt hatten, wurden für diesen Kontrast nur geringe, nicht signifikante Unterschiede gefunden. Für die 'Low-Compliant' dagegen zeigt dieser Kontrast signifikante Unterschiede für Hirnareale vom visuellen Kortex (Brodmann Areal 18) über mediale Areale, wie dem MDNT und dem Pulvinar bis hin zu frontalen Arealen, wie dem OFC und dem IFG, die für die Emotionsverarbeitung zuständig sind (siehe Abbildung 4.5).

Effekt der Adaption in emotionalen Regionen: Für eine detaillierte Aussage über den Einfluss der Adaption auf das zugrundeliegende Paradigma mit den emotionalen Stimuli wurde in dieser zusätzlichen Analyse ein Vergleich der Effektstärken für die relevanten Hirnareale bestimmt. Dazu wurde an den MNI-Koordinaten des Haupteffekts der Emotion eine ROI-Analyse berechnet (siehe Abschnitt 4.2.3.5). Dabei wurde für Regionen der Emotionsverarbeitung ein signifikanter Anstieg der Effektstärken durch die Adaption gefunden. Exemplarisch zeigt die

Hirnareal	MNI Koordinaten			Kontrast	
	in mm			Negativ > Neutral	
	x	y	z	t-Werte	z-Werte
Medial Prefrontaler Kortex	6	53	31	4,82	4,44
Orbitofrontaler Kortex R	30	23	-17	5,48	4,94
Orbitofrontaler Kortex L	-27	29	-14	5,03	4,6
Inferior Frontaler Gyrus R	45	8	31	4,39	4,09
Insula R	33	8	-11	4,86	4,46
Insula L	-24	11	-17	5,52	4,97
Amygdala R	21	-4	-14	5,65	5,07
Amygdala L	-18	-7	-14	5,04	4,61
MDN Thalamus R	-3	-13	-2	4,23	3,96
MDN Thalamus L	3	-13	-2	4,05	3,8
Pulvinar R	-15	-34	1	4,46	4,15
Pulvinar L	15	-31	1	3,94	3,71
Fusiform Gyrus R	42	-46	-17	8,39	6,86
Fusiform Gyrus L	-42	-46	-20	8,29	6,81
Lateraler Occipitaler Kortex R	51	-67	-8	6,84	5,9
Lateraler Occipitaler Kortex L	-45	-73	-11	5,71	5,11
Mittlerer Temporaler Gyrus R	57	-67	16	5,03	4,6
Mittlerer Temporaler Gyrus L	-48	-67	19	5,07	4,63
Precuneus	0	-52	28	5,06	4,62

Tabelle 4.1: Die Tabelle fasst die Ergebnisse der Gruppenstatistik über alle Probanden in MNI-Koordinaten der gefundenen Hirnareale und dazugehörige statistische Prüfwerte (t-Wert und dessen Normierung als z-Wert) für den Haupteffekt der Emotion (negativ > neutral) zusammen.

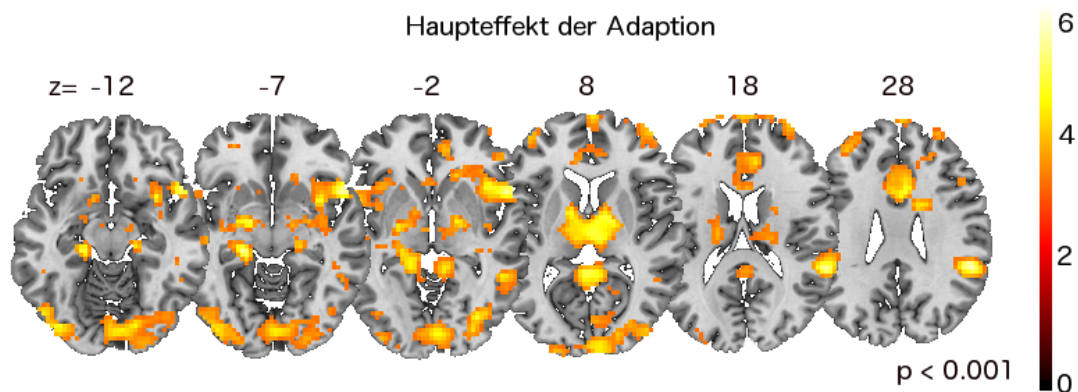


Abbildung 4.5: Die Abbildung zeigt die Ergebnisse der GLM-Analyse für den Haupteffekt der Adaption (*adaptiert > nicht-adaptiert*) für die 'Low-Compliant'. Das Ergebnis zeigt eine erhöhte neuronale Aktivität vom visuellen Kortex, über mediale bis zu frontalen Arealen, die für die Emotionsverarbeitung relevant sind.

Abbildung 4.6 die Ergebnisse für das Pulvinar und die Insula graphisch. Die Unterschiede in den Effektstärken für den MDNT, OFC, LOC und IFG sind in Tabelle 4.2 aufgelistet.

Gruppenvergleich adaptierte Stimulation: Die bisherigen Ergebnisse zeigen den Einfluss der Adaption insgesamt und die höheren Effektstärken in emotionsbezogenen Hirnarealen für die 'Low-Compliant'. Abschließend wurde ein Zweistichproben-t-Tests zum Vergleich der adaptierten Bildpräsentationen zwischen den beiden Gruppen berechnet. In diesem Ergebnis konnte kein Unterschied zwischen den Gruppen festgestellt werden. Die adaptierten Stimuli konnten somit auch bei den Probanden, die bei konventionellen Stimuli eine schlechtere Leistung erzielt hatten, den neuronalen Aktivitätsunterschied in allen relevanten Hirnareale auf das Niveau der 'High-Compliant' anheben.

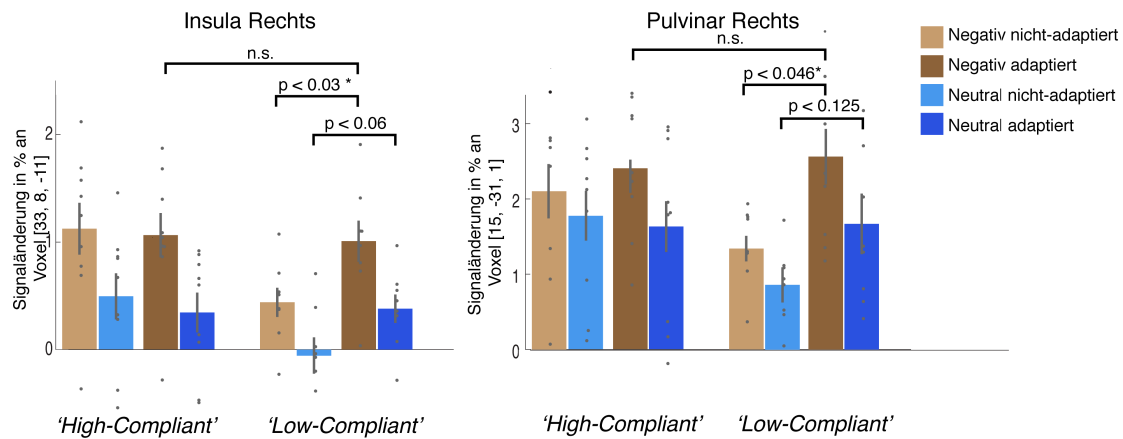


Abbildung 4.6: Die Abbildung zeigt exemplarisch für das rechte Pulvinar und die Insula den Effekt der Adaption auf die erzielten Effektstärken. Für die negativen Stimuli zeigt sich ein signifikanter Anstieg der Effektstärken durch die Adaption für die 'Low-Compliant' auf das Niveau der 'High-Compliant'. Für die neutralen Stimuli wurde ebenfalls ein Anstieg der Effektstärken beobachtet.

Hirnareal	MNI Koordinaten in mm			(Adaptiert > Nicht-adaptiert) negativ	(Adaptiert > Nicht-adaptiert) neutral
	x	y	z	p <	p <
Insula R	33	8	-11	0,0295*	0,0611
Pulvinar L	-15	-34	1	0,0735	0,104
Pulvinar R	15	-31	1	0,046*	0,125
Orbitofrontaler Kortex R	30	23	-17	0,1617	0,238
Orbitofrontaler Kortex L	-27	29	-14	0,36	0,14
Lateral Occipitaler Kortex L	-43	-73	-11	0,19	0,25
Lateral Occipitaler Kortex R	51	-67	-8	0,22	0,175
Inferior Frontaler Gyrus	33	17	-14	0,067	0,178
MDN Thalamus L	-3	-13	-2	0,242	0,033
MDN Thalamus R	3	-13	-2	0,15	0,042

Tabelle 4.2: Die Tabelle fasst die Unterschiede in den Effektstärken für adaptive und konventionelle Stimulation in Bezug auf die MNI-Koordinaten des Haupteffekts der Emotion zusammen.

Zusätzlich zu den vorgestellten Ergebnissen ist der Einfluss der Adaption auf die zeitlichen Intervalle zwischen den Bildpräsentationen (ITIs) zu erwähnen. Die Präsentation der Bilder wurde durch das Warten auf eine Fixation um durchschnittlich $2,3 \pm 2,9$ s verschoben. Dieser Unterschied in Bezug auf die Gruppen ergibt sich zu $0,26 \pm 0,30$ s für die 'High-Compliant' und $4,6 \pm 2,8$ s für die 'Low-Compliant'. Um zu prüfen, ob dieser Unterschied einen Einfluss auf die Ergebnisse hat, wurde in einer hier nicht präsentierten Analyse die ITIs als Kovariate in der GLM-Analyse modelliert. Die Ergebnisse haben zu keinen quantitativen Änderungen der Ergebnisse geführt, d.h. dieser Unterschied ist völlig unerheblich, und daher wird diese zusätzliche Analyse hier nicht weiter erläutert.

4.3 Experiment II: fMRT-Experiment zur Echtzeit-Auswertung

Für die Validierung der in BART implementierten GLM-Analyse wurde ein fMRT-Experiment durchgeführt, das als *Fingertapping* bezeichnet wird und über dessen Ergebnisse bereits zahlreich in der Literatur berichtet wurde (Boecker et al. 1994; Baraldi et al. 1999). Auch bei der Validierung von neu entwickelten Auswertemethoden wird das Fingertapping oft zum Vergleich der Ergebnisse herangezogen (Kansaku et al. 2005; Yoo et al. 2002). Dabei wird das gleichmäßige Bewegen der Finger für eine bestimmten Zeit (hier 10 s) mit einer Ruhephase (hier 20 s) verglichen. Das Fingertapping führt zu einer erhöhten neuronalen Aktivität in motorischen Arealen des Gehirns, zu denen u.a. das primär-motorische Areal (M1), prämotorisches Areal (PMA) und supplementär-motorische Areale (SMA und prä-SMA) gehören. Wie in Abschnitt 3.3.4.4 bereits technisch vorgestellt, wurde für BART ein parallelisierter GLM-Algorithmus neu implementiert. In diesem Abschnitt werden die Ergebnisse der Echtzeitauswertung mit SPM8-Ergebnissen verglichen.

Die Daten aus dem Fingertapping wurden mit SPM8 und BART ausgewertet. Mit SPM8 wurde eine Bewegungskorrektur durchgeführt und die Daten mit einem 8 mm Gaußfilter geglättet. In der GLM-Analyse von SPM wird intern zusätzlich die Hirnstruktur maskiert um Signalrauschen in den Randbereichen zu unterdrücken. Für den Vergleich der Aktivitätsmuster wurde der Kontrast 'Bewegung > Pause' berechnet. Für BART wurden die Daten verwendet, die bereits in der internen Siemens Datenverarbeitung bewegungskorrigiert wurden. Im Simulationsmodus von BART wurden die Daten zusätzlich ebenfalls mit einem 8 mm Gaußfilter geglättet und anschließend eine GLM-Analyse ohne eine Maskierung der Hirnstruktur berechnet. Für den Vergleich der Ergebnisse wurden die Datenvorverarbeitung (VV) und -analyse (GLM) in drei Varianten ausgeführt:

- VV-BART/GLM-BART: Die Vorverarbeitung und Analyse wurde mit BART berechnet.
- VV-SPM8/GLM-BART: Die vorverarbeiteten Daten aus SPM8 wurden mit BART analysiert.

- VV-SPM8/GLM-SPM8: Die Vorverarbeitung und Analyse wurde mit SPM8 berechnet.

Die Ergebnisse sind in Abbildung 4.7 graphisch dargestellt. Die aktivierten Areale entsprechen den in der Literatur beschriebenen Hirnarealen für das Fingertapping. Zusätzlich wurde zum Vergleich der Ergebnisse das Ähnlichkeitsmaß der Mutual Information berechnet. Die Ähnlichkeit ist in Tabelle 4.3 zusammengefasst. Die Mutual Information zeigt eine sehr große Ähnlichkeit zwischen den Ergebnissen aus den unterschiedlichen Softwarepaketen.

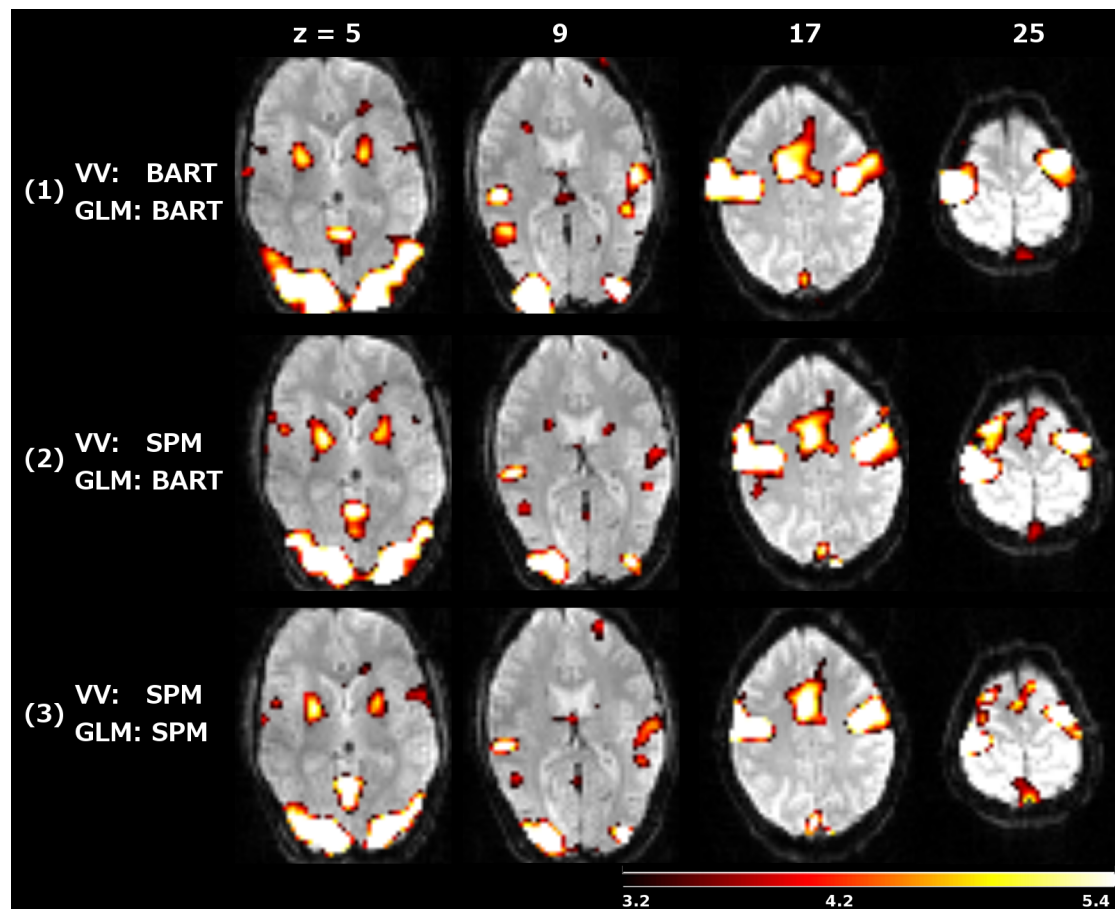


Abbildung 4.7: fMRT-Experiment II: Vergleich der Datenauswertungen für ein Paradigma mit Fingertapping. Hierbei wurde für die Vorverarbeitung (VV) und die GLM-Analyse (GLM) entweder ausschließlich SPM8 oder BART genutzt bzw. vorverarbeitete Daten von SPM8 im Paket BART analysiert. Die Übereinstimmung der gezeigten Ergebnisse ist in Tabelle 4.3 anhand der Mutual Information quantifiziert. Die Unterschiede in den Ergebnissen sind vor allem auf die umfangreichere Vorverarbeitung in SPM8 zurückzuführen.

Mutual Information zwischen (Zeile),(Zeile)	in Spalte (z =)				3D (alle Schichten)
	1 (5)	2 (9)	3 (17)	4 (25)	
(1),(2)	0,9186	0,9220	0,8913	0,9493	0,6781
(2),(3)	0,9533	0,9519	0,9698	0,9868	0,7609
(1),(3)	0,9191	0,9262	0,8885	0,9542	0,6874

Tabelle 4.3: Mutual Information in Bezug auf Abbildung 4.7 als Maß für Übereinstimmung zwischen den Auswertergebnissen mit der Software SPM8 und BART. Die Werte zeigen eine sehr hohe Übereinstimmung zwischen den Ergebnissen.

4.4 Experiment III: fMRT-Experiment mit adaptivem Paradigma und Echtzeit-Auswertung

In diesem abschließenden Experiment wurden die beiden Module des Softwarepakets BART, die adaptive Stimulation und die Echtzeitauswertung der Daten, parallel ausgeführt. Das Experiment wurde mit dem gleichen Paradigma wie in Abschnitt 4.2 durchgeführt. Das heißt, die Bildpräsentation wurde an die Fixation angepasst. Die adaptierte Stimulation wurde direkt mittels der Transferfunktionen in die Designmatrix übertragen, damit die GLM-Analyse direkt berechnet werden konnte. Die technischen Ergebnisse (Funktionalität und Laufzeit) dazu wurden in Abschnitt 3.3.7 beschrieben. Hier werden die Ergebnisse der GLM-Analyse für den Kontrast ‘negativ > neutral’ wiederum im Vergleich zu einer Auswertung mit SPM8 verglichen (siehe Abbildung 4.8). Die Hirnareale, die dabei eine erhöhte Aktivität zeigen, stimmen mit denen in der Literatur beschriebenen überein (siehe Abschnitt 4.1). Zusätzlich wurde auch hier die Mutual Information als Ähnlichkeitsmaß berechnet, die in Tabelle 4.4 zusammengefasst ist und eine hohe Ähnlichkeit zwischen den Ergebnissen zeigt. Die deutlichsten Unterschiede sind im Randbereich der Hirnstrukturen, im Bereich des Schädelknochens, zu sehen. Die GLM-Analyse in SPM verwendet eine Maskierung der Hirnstrukturen um das starke Signalrauschen in diesem Bereich besser zu unterdrücken.

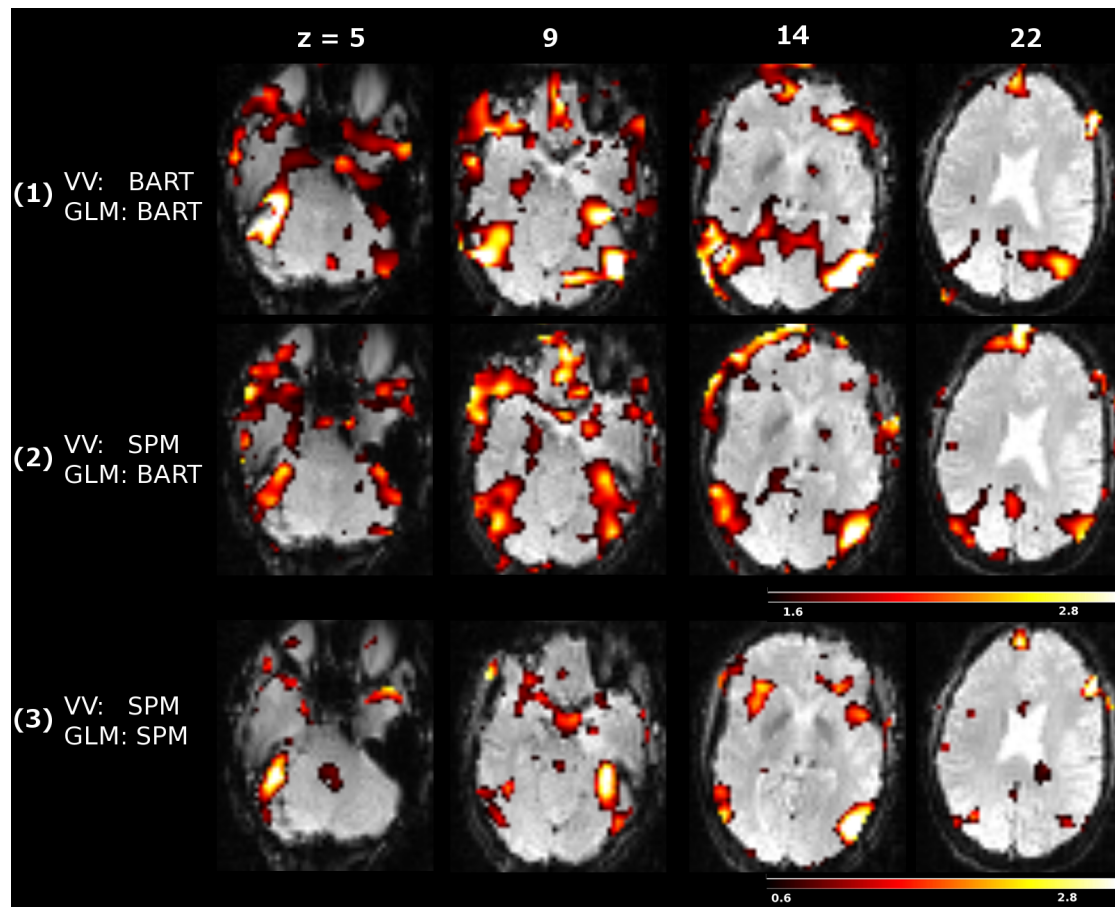


Abbildung 4.8: fMRT-Experiment III: Vergleich der Datenauswertungen für das adaptive Paradigma in Kombination mit der Echtzeitauswertung. Für die Vorverarbeitung (VV) und die GLM-Analyse (GLM) wurden auch hier SPM8 und BART genutzt. Die Übereinstimmung der gezeigten Ergebnisse ist in Tabelle 4.4 anhand der Mutual Information quantifiziert. Die Unterschiede in den Ergebnissen sind vor allem auf die implizite Maskierung des Gehirns in SPM8 zurückzuführen.

Mutual Information zwischen (Zeile),(Zeile)	in Spalte (z =)				3D (alle Schichten)
	1 (5)	2 (10)	3 (14)	4 (22)	
(1),(2)	0,8162	0,8199	0,8177	0,7645	0,5184
(1),(3)	0,9058	0,8896	0,8434	0,8562	0,6157
(2),(3)	0,8461	0,8267	0,7230	0,6894	0,5787

Tabelle 4.4: Mutual Information in Bezug auf Abbildung 4.8 zur Vergleichbarkeit der Auswertergebnisse von SPM8 und BART. Die Werte zeigen eine hohe Übereinstimmung zwischen den Ergebnissen.

5 Diskussion

In der vorliegenden Arbeit wurden Softwarekonzepte für eine universelle methodische Gestaltung von Echtzeit-fMRT-Experimenten erarbeitet. Diese Konzepte wurden in dem neuen Softwarepaket BART implementiert, das bei der flexiblen Durchführung von adaptiven Paradigmen für fMRT- und Echtzeit-fMRT-Experimente Verwendung findet. Dazu wurde ein vereinheitlichtes Framework vorgestellt, mit dem verschiedene externe Parameter eingebunden werden können, die das Paradigma beeinflussen. Die Vorteile von adaptierten Experimenten wurden exemplarisch in einer fMRT-Studie untersucht. Zusätzlich dazu wurden die Implementierung des Echtzeit-GLM-Algorithmus und die Kombination der beiden Komponenten experimentell validiert.

5.1 Diskussion Software

Allgemein: Das vorgestellte Softwarepaket ist mit Methoden und Techniken der objektorientierten Programmierung implementiert, die Muster anbieten, um die gegebene Komplexität der Aufgabe zu beherrschen und die gewünschte Flexibilität zu erreichen. Das Gesamtpaket gliedert die unterschiedlichen Teilaufgaben in einzelne Pakete, um die Gesamtaufgabe übersichtlicher zu gliedern. Dadurch sind die Teilaufgaben durch Fassaden klar voneinander abgegrenzt und die jeweils ausreichend komplexen Inhalte können unabhängig voneinander bearbeitet werden. Dies reduziert die Fehleranfälligkeit und verbessert die Testbarkeit der Software.

Für alle Klassen wurden Softwaretests (Unittests) implementiert. Auch wenn die Implementierung mit einem deutlich erhöhten Aufwand verbunden war, erhöht dies enorm die Laufzeitstabilität und verifiziert die korrekte Funktionalität für alle künftigen Entwicklungen. Das vorgestellte Softwarepaket wurde entwickelt für die Hardware Mac Pro. Der Aufwand einer plattformunabhängigen Version war im Rahmen dieser Arbeit nicht möglich. Die gewählte Hardware erfüllt die geforderten Kriterien optimal.

Die Software wird mit Hilfe der Beschreibungssprache EDL konfiguriert. Diese beschreibt alle notwendigen Parameter vollständig und in einer übersichtlichen Struktur. Zusätzlich beinhaltet sie eine regelbasierte Wissensrepräsentation zur Überprüfung von Abhängigkeiten zwischen diesen Parametern (Hollmann et al. 2008). Damit ist eine übersichtliche und fehlerfreie Konfiguration des Gesamtsystems möglich. Durch die Erweiterung von EDL mit neuen abstrakten

Konstrukten war es möglich ein flexibles Konzept für die adaptive Stimulation zu konzipieren. BART ist eine Open-Source-Lösung wobei der Quellcode auf den Servern des Hosting-Dienstes *github* zu finden ist: <https://github.com/bart-group/BART>.

Einordnung zum Stand der Technik: Bereits vorhandene Softwaresysteme haben keine ausreichende Basis für die Weiterentwicklung eines vorhandenen Systems geboten. Das kommerzielle System Turbo Brain Voyager, wie in Weiskopf (2012); Sorger et al. (2007); Sitaram (2007) verwendet, ist gut geeignet für die Durchführung von Neurofeedbackexperimenten, bietet jedoch keine Möglichkeit für die Eigenentwicklung neuer Methoden. Die Systeme CIGAL von Voyvodic et al. (2011) und I/OVA von Smyser et al. (2001) beschreiben Ansätze zur simultanen Erfassung von verschiedenen physiologischen Parametern. Diese können somit als Faktoren in einer post-hoc Analyse berücksichtigt werden. Die Systeme bieten allerdings keine Adaption an diese Parameter während der Laufzeit an. Die Software rtExplorer, die am Max-Planck-Institut für Kognitions- und Neurowissenschaften für Neurofeedbackexperimente verwendet wird, ist vom Gesamtkonzept nicht für adaptive Stimulation vorgesehen. Die in der Literatur beschriebenen Anwendungen von adaptiven Paradigmen mit kommerzieller Stimulationssoftware präsentieren jeweils individuelle Lösungen für einzelne Anwendungsfälle (Wilms et al. 2010; Becker et al. 2011). Dabei ist neben eigenen Programmiersprachen der kommerziellen Softwarepakete auch die notwendige Unterstützung für die Einbindung externer Parameter durch die Gerätehersteller nachteilig. Die hier vorgestellte Arbeit ist der erste Ansatz, der eine standardisierte Beschreibung von adaptivem Verhalten für fMRT-Paradigmen definiert. Dieser Ansatz wurde in einem flexiblen Framework implementiert, das konsistent zu benutzen und wiederzuverwenden ist. Zusätzlich wurde dies erstmalig an eine Echtzeitauswertung der Daten gekoppelt. Die Datenauswertung ist so konzipiert, dass eine flexible Methodenauswahl für die Datenverarbeitung möglich ist.

Integrierte Stimulationsumgebung: Da das Stimulationsmodul in das Gesamtpaket für die Durchführung von Echtzeit-fMRT Experimenten integriert wurde, wird nur ein zusätzlicher Rechner zum MR-Setup benötigt. Dies ist von Vorteil gegenüber bestehenden Systemen, die eine Verbindung mehrerer Komponenten erfordern. Außerdem werden dadurch Zeitverluste durch die Kommunikation zwischen verschiedenen Softwarekomponenten über ein Netzwerk vermieden. Gleichzeitig ist auch die Fehleranfälligkeit geringer. Momentan ist mit BART eine visuelle oder auditive Stimulation der Probanden möglich. Diese Funktionalität kann jedoch auf andere sensorische Reize erweitert werden. Die Funktionalität des Stimulationsmoduls wurde durch Benchmarktests verifiziert, die unter maximaler Prozessorlast eine Laufzeitgenauigkeit von 2 ms ergeben haben.

Universelles Framework für adaptive Paradigmen: Basierend auf dieser integrierten Stimulationsumgebung und der Konfigurationssprache EDL wurde ein Konzept erarbeitet und implementiert, mit dem sich nicht nur konventionelle Paradigmen, sondern auch zur Laufzeit adap-

tierbare Paradigmen realisieren lassen. Dazu wurden neue Konstrukte in EDL definiert, die eine Adaption abstrakt beschreiben ohne dass spezifisches Wissen über die Komponenten oder Variablen erforderlich ist. Die Auswertung der konkreten Variablen erfolgt in Plugins, die erst zur Laufzeit an das System gekoppelt werden. Damit sind die konkreten Geräte und die aus diesen ermittelten Parameter unabhängig vom Gesamtsystem und beliebig austauschbar. Die Gesamtfunktionalität für die Adaption kann damit direkt erweitert werden. Zu beachten ist, dass die Plugins in einer nativen Programmiersprache implementiert werden müssen. Bisher können keine Plugins von Skriptsprachen, wie Python oder Matlab, eingebunden werden. Die Auswertung der externen Parameter erfolgt zur Laufzeit der Stimulation und ist durch eine binäre Logik unabhängig von den spezifischen Parametern. Das interne Verhalten des Stimulationsmoduls ist unabhängig von den externen Plugins und muss daher für die Erarbeitung eines neuen Paradigmas nicht geändert werden. Es ist zu beachten, dass die Adaption eines Paradigmas hier immer in einer binären Logik abgebildet werden muss. Dies ist für sehr komplexe Paradigmen eventuell nicht möglich. Für das adaptive Verhalten der Stimulationsumgebung wurden ebenfalls Benchmarktests durchgeführt, die eine ausreichende Laufzeitstabilität von 8 ms ergeben haben. Mit diesem vereinheitlichten Konzept zum Einbinden externer Parameter, die das Verhalten oder physiologische Zustände eines Probanden in die Stimulation einzubeziehen, wird die Stimulation zur Laufzeit optimiert. Daraus resultieren einerseits zuverlässigere Ergebnisse für fMRT-Daten, insbesondere bei Methoden die implizit eine geringere statistische Teststärke haben (z. B. Echtzeit-fMRT). Andererseits kann dies auch zu einer Zeit- und Kostenersparnis führen, was z. B. bei Patientmessungen von besonderer Bedeutung ist. Zusätzlich zur Adaption eines Paradigmas wurden auch neue Konstrukte entwickelt mit denen die aktualisierten Daten direkt in die notwendigen Informationen für die Auswertung transferiert werden. Damit ist erstmalig die Kombination dieser beiden Methoden möglich. Die dafür notwendigen Berechnungen wurden ebenfalls so optimiert, dass die Aktualisierung in wenigen Millisekunden erfolgt, womit die gegebenen Zeitlimits erfüllt werden.

Flexible Integration Datenverarbeitungsalgorithmen: Für die Datenverarbeitung wurden ebenfalls Softwarekonzepte gewählt, die einen flexiblen Austausch der Algorithmen ermöglichen. Damit können neue Algorithmen für die Vorverarbeitung und Analyse direkt integriert werden. Die Entwicklung kann somit unabhängig vom Gesamtsystem erfolgen und ohne Kenntnisse über die internen Kommunikationsstrukturen oder das Datenmanagement. Die Gesamtkonzeption mit der Programmiersprache Objective-C und dem Cocoa Framework ermöglicht es, dass die eingebundenen Algorithmen auch in den Sprachen C oder C++ geschrieben werden können. Zukünftige Entwicklungen können damit sehr flexibel gestaltet werden.

Datenverarbeitung Algorithmen: Für die Echtzeit-fMRT-Auswertung wurde ein parallelisierter GLM-Algorithmus neu implementiert und getestet. Dieser kann die Daten sowohl im *sliding-window*- als auch im *growing-window*-Verfahren auswerten. Die Laufzeit des Algorithmus wurde durch die Parallelisierung auf ein Zehntel der konventionellen Berechnung verkürzt und ist

damit ausreichend schnell auch bei längeren Zeitreihen. Es ist zu beachten, dass die Auswertung im *growing window*-Verfahren mit einer ansteigenden Laufzeit verbunden ist, da hierbei alle bereits gemessenen Zeitschritte analysiert werden. Dies ist insbesondere bei Echtzeit-fMRT als unkritisch zu betrachten, da die Messungen meist auf eine kurze Zeit begrenzt und auf mehrere Sessions verteilt werden. Damit kann die erforderliche sehr hohe Konzentration für die Aufgaben durch kurze Pausen verbessert werden. Für sehr lange Sessions ist dagegen das *sliding-window*-Verfahren vorteilhaft. Einerseits werden damit zeitlich bedingte Signaländerungen besser eliminiert und andererseits wird die Laufzeit immer konstant gehalten. Weiterhin ist es denkbar andere Algorithmen für die GLM-Auswertung zu implementieren. In der Software Turbo Brain Voyager wird z. B. ein Algorithmus verwendet, der die Schätzungen für die Koeffizienten mit dem rekursiven Verfahren der kleinsten Fehlerquadrate in jedem neuen Zeitschritt aktualisiert (Sitaram et al. 2008). Eine solche mögliche Implementierung könnte in BART integriert werden.

Die Vorverarbeitung der Daten ist momentan Bestandteil der Weiterentwicklung von BART. Das allgemeine Konzept dafür wird ebenfalls für einen flexiblen Austausch der Algorithmen aufgebaut. Es wurden bereits echtzeitfähige Algorithmen für die Bewegungskorrektur und die Registrierung/Normalisierung der fMRT-Daten implementiert. Diese verwenden u. a. Routinen aus ITK und werden ebenfalls parallel prozessiert. Da der Schwerpunkt dieser Arbeit auf die adaptiven Paradigmen ausgerichtet war, wurden diese Algorithmen noch nicht vollständig in das Gesamtsystem integriert. Für den Vergleich der Ergebnisse der Datenverarbeitung wurde daher für BART eine von Siemens bereitgestellte Bewegungskorrektur und die Vorverarbeitung von SPM8 verwendet. Auch wenn die Unterschiede in dieser Arbeit nicht näher quantifiziert wurden, zeigen die Ergebnisse, dass insbesondere für die Untersuchung von subkortikalen Regionen die Güte der Vorverarbeitung einen starken Einfluss auf die Ergebnisse hat. Dafür müssen die neu implementierten Algorithmen noch getestet werden.

Zusätzlich entwickelte Tools: Die vorgestellten Tools, die zusätzlich zum Kern der vorliegenden Arbeit entwickelt wurden, unterstützen die Funktionalität des Gesamtpakets. Der plattformunabhängige EDLEditor für EDL-Dokumente vereinfacht das Erstellen von Konfigurationsdateien durch eine benutzerfreundliche Oberfläche und vereinfachte Eingabemethoden für die XML-Elemente. Der Editor bietet alle in der XSD definierten Elemente und deren Attribute zur Auswahl an und zeigt die im Schema enthaltene Dokumentation für diese an. Außerdem überprüft der Editor während der Eingabe die Korrektheit und die Abhängigkeiten der Parameter wodurch die Fehleranfälligkeit zur Laufzeit der Software BART reduziert wird. Damit erleichtert dieser Editor die Experimentkonfiguration und die Einarbeitung in die komplexe Thematik, die mit BART bearbeitet wird.

Die Eigenimplementierung eines Exports der fMRT-Daten an den Auswerterechner hat die Stabilität des Gesamtsystems deutlich verbessert und die Geschwindigkeit der Datenübertragung optimiert. Die Übertragung der Daten vom Arbeitsspeicher des MR-Rechners aus dem Siemens

internen Netzwerk in den Arbeitsspeicher des Auswerterechners umgeht die Schwierigkeiten, die mit dem Lesen, Schreiben und Kopieren der Daten einhergeht. Außerdem wurde die Bedienung durch die Integration in die Scannersequenz deutlich vereinfacht, was für einen Routinebetrieb besonders wichtig ist.

Die Basisbibliothek *isis* ist die Grundlage für ein effizientes, internes Datenmanagement für fMRT-Daten. Sie bietet eine Vielzahl an grundlegenden Routinen für die Verwaltung der Daten, die, unabhängig von Echtzeit-fMRT und adaptiven Paradigmen, für das Softwarepaket BART essentiell sind. Zusätzlich bietet *isis* eine vereinheitlichte und erweiterbare Schnittstelle für die Einbindung der verschiedenen Datenformate für fMRT-Daten. Dies ist für die Simulation von Experimenten mit BART und insbesondere für die Unabhängigkeit vom verwendeten Scanner-system wichtig. Ein weiterer Vorteil ist die Einbindung der umfangreichen Algorithmenbibliothek ITK.

5.2 Diskussion fMRT-Experimente mit BART

5.2.1 Diskussion fMRT-Experiment I mit adaptivem Paradigma

Die empirischen Ergebnisse der durchgeführten fMRT-Studie mit einem adaptiven Paradigma, das den Fixationsstatus der Probanden berücksichtigt, zeigen die Vorteile von adaptierten gegenüber konventionellen Paradigmen. Eine erste Erkenntnis ist, dass viele Probanden der expliziten Anweisung, den Bildschirm in der Mitte zu fixieren, nicht gefolgt sind. Bei bis zu 50 % der nicht-adaptierten Bildpräsentationen haben diese ihren Blick nicht auf die Bildschirmmitte gerichtet. Dieser Unterschied im Verhalten der Probanden wurde durch die Adaption der Bildpräsentation aufgehoben. Das Verhalten ist durch die Adaption daher wesentlich homogener und vergleichbarer. Dieses Ergebnis wird bestärkt durch die Auswertung der Blickrichtung zum Zeitpunkt der Bildpräsentation. Für die nicht-adaptierten Bilder ist der Abstand zur Bildschirmmitte signifikant größer. Die Probanden mussten hier ihren Blick erst zur Mitte des Bildschirms bewegen, um den emotionalen Inhalt wahrzunehmen. Die dafür notwendige Zeit ist ein unkontrollierter Einflussparameter in konventionellen Paradigmen. In Bezug auf die weiteren Verhaltensparameter wurde festgestellt, dass kein Unterschied bezüglich der Reaktionszeit besteht, jedoch in Bezug auf die zutreffende Antwort, die gegeben wurde. In der fMRT-Studie haben die 'Low-Compliant' bei den adaptierten Stimuli die Bilder signifikant häufiger korrekt zugeordnet. Das Adaptionsparadigma hat zu einer verbesserten Erkennungsleistung der präsentierten Emotionen geführt. Bei der Bewertung der Bilder in Bezug auf Valenz und ausgelöste emotionale Erregung haben die Probanden die erwartungsgemäße Reaktion auf die emotionalen Stimuli gezeigt. Dies belegt die sehr hohe Übereinstimmung mit den vorhandenen Bewertungen für die IAPS-Bilder.

Die gemessenen fMRT-Daten wurden zuerst auf deren Ergebnisse bezüglich des zugrundeliegenden Paradigmas mit emotionalen Stimuli überprüft. Wie erwartet, entsprechen die detektierten Hirnareale für den Kontrast 'negativ > neutral' den in der Literatur beschriebenen Ergebnissen, zu denen u. a. der mPFC, der OFC, der IFG, die Insula, die Amygdala, der MDNT und das Pulvinar gehören. Diese wurden im Abschnitt 4.1 mit der Metaanalyse von Sabatinelli et al. (2011) vorgestellt. Um die Effekte, die durch die Adaption der Bildpräsentation verursacht werden, darzustellen, wurde anschließend der Kontrast 'adaptiert > nicht-adaptiert' bestimmt. Es ist bekannt, dass die visuelle Aufmerksamkeit die Verarbeitung von emotionalen Stimuli beeinflusst (Pessoa 2004). Auch wenn eine visuelle Fixation nicht garantiert, dass ein Stimuli wie gewünscht wahrgenommen wurde, so ist dies doch ein starker Indikator und notwendige Voraussetzung dazu. Die in Abbildung 4.5 dargestellten neuronalen Aktivierungen für diesen Kontrast stimmen mit den in der Literatur beschriebenen Hirnarealen für die Emotionsverarbeitung gut überein (Pessoa et al. 2010; Vuilleumier 2005).

Um die Vorteile der Adaption zu quantifizieren, wurden in einer weitergehenden Analyse die erzielten Effektstärken für adaptierte und nicht-adaptierte Bilder miteinander verglichen. Wie in Abbildung 4.6 gezeigt, konnten die Effektstärken in mehreren Hirnregionen, die im Haupteffekt der Emotion ermittelt wurden, durch die Adaption der Stimuli signifikant verbessert werden. Insgesamt zeigte sich, dass die Effektstärken für die '*Low-Compliant*' auf das Niveau der '*High-Compliant*' angehoben werden konnten. Das bedeutet, dass erst durch die Adaption zur Laufzeit des Paradigmas das Verhalten der Probanden und die erzielten neuronalen Aktivierungen vergleichbar werden. Ein interessantes Ergebnis ist, dass ein signifikanter Einfluss der Adaption auf die Aktivierung im Pulvinar beobachtet wurde. In kürzlich erschienenen Forschungsergebnissen wurde die Rolle dieser Hirnregion für die Emotionsverarbeitung hervorgehoben (Pessoa et al. 2010; Ward et al. 2007). Außerdem wurde diese Region bereits im Kontext von Aufmerksamkeitsexperimenten beschrieben (Arend et al. 2008) und mit dem Arbeitsgedächtnis und der Merkmalerkennung in Verbindung gebracht (Rotshtein et al. 2011). Das hier vorgestellte Paradigma war nicht dazu entwickelt worden, den Einfluss von Aufmerksamkeit auf die Emotionsverarbeitung zu untersuchen. Allerdings könnte die Methodik der adaptiven Paradigmen durchaus geeignet sein, um weitere Experimente diesbezüglich durchzuführen und offene Fragen zu klären. Dazu könnte ein passendes adaptives Paradigma beruhend auf den Arbeiten von Padmala et al. (2010) und Lim et al. (2009) erstellt werden, in denen bereits erste Untersuchungen zum Zusammenhang zwischen kontinuierlichen Fluktuationen im Gehirn und der Wahrnehmung emotionaler Stimuli gemacht wurden.

Abschließend zu diesem Experiment ist anzumerken, dass mit der Adaption längere ITIs für die adaptierten Bilder bei der Probandengruppe der '*Low-Compliant*' verbunden sind. Allerdings konnte in den vorhandenen Daten kein Anhaltspunkt gefunden werden, dass dieser Unterschied einen Einfluss auf die gefundenen Hirnaktivierungen hat. Die Modellierung der ITIs als zusätzlicher Parameter in der GLM-Analyse hat keine quantitativen Unterschiede ergeben. Daher

wurden die Daten in dieser Arbeit nicht näher gezeigt. Für das durchgeführte Experiment wurde der Kompromiss bezüglich der ITIs gemacht, um möglichst gut vergleichbar mit dem Goldstandard für diese Art von Experimenten zu sein. In diesem werden die Bilder in der Mitte des Bildschirms präsentiert. Wenn dies für ein Experiment nicht adäquat ist, z. B. bei zeitkritischen Studien mit Patienten, wäre es alternativ möglich, die Bilder an der Bildschirmposition zu präsentieren, auf die der Proband zu dem gewünschten Zeitpunkt der Stimulation fixiert.

Insgesamt betrachtet zeigen die Ergebnisse dieser Studie, dass die Adaption der Stimuluspräsentation die Zuverlässigkeit und Reproduzierbarkeit von fMRT-Ergebnissen deutlich verbessert. Die Hälfte der Probanden der Studie zeigte aufgrund der Adaption signifikant verbesserte Effektstärken in emotionsbezogenen Hirnarealen. Damit wurde das Niveau der *'Low-Compliant'* insgesamt auf das Niveau der *'High-Compliant'* angehoben, wodurch die statistischen Ergebnisse insgesamt verbessert wurden. Neben dem hier vorgestellten Beispiel für eine Adaption basierend auf Eyetrackerdaten könnte dieses Setup auch bei Erkrankungen, die mit Schwierigkeiten beim Fixieren einhergehen, wie z. B. bei der Huntington-Krankheit oder Autismus, eingesetzt werden.

5.2.2 Diskussion fMRT-Experimente II und III

Der Vergleich der Datenauswertung von SPM8 und BART hat für beide Experimente eine hohe Übereinstimmung im Ähnlichkeitsmaß der Mutual Information ergeben. Dies validiert die korrekte Funktionalität der implementierten GLM-Analyse. Für das *Fingertapping*-Experiment ist die Übereinstimmung besonders hoch, da dieses Paradigma zu einem robusten Anstieg neuronaler Aktivität im motorischen Kortex führt. Für das emotionale Paradigma in Experiment III ist die Übereinstimmung ebenfalls sehr gut. Dabei sind die glsBOLD-Aktivierungen in subkortikalen Hirnregionen, insbesondere dem limbischen System, deutlich geringer als in kortikalen Arealen. Außerdem sind diese Strukturen anatomisch kleiner und liegen teilweise in Randbereichen der weißen Substanz. Daher treten öfter MR-Signalstörungen auf und der Signalrauschabstand ist per Definition geringer. Diese Probleme können durch eine gute Vorverarbeitung reduziert werden. Für BART wurden bisher Algorithmen für die Bewegungskorrektur, die Glättung und die Schichtkorrektur implementiert. Allerdings sind diese noch nicht vollständig in das Gesamtsystem integriert. Außerdem fehlt eine Maskierung der Hirnstrukturen, wie von SPM verwendet, um Signalrauschen in den Randbereichen besser zu unterdrücken. Die hier in BART verwendete Vorverarbeitung ist daher noch nicht optimal und führt zu den Unterschieden in den Ergebnissen.

5.3 Diskussion Adaptive Paradigmen für fMRT

Die in dieser Arbeit durchgeführte Studie zeigt, dass adaptive Paradigmen die bei fMRT-Studien erzielten Messdaten an einem angewandten Beispiel deutlich verbessern. Die dadurch erzielten Antworten der Probanden sind einheitlicher und stellen die tatsächliche Antwort auf den gezeigten Stimulus dar. Dadurch konnte die statistische Aussagekraft der Daten deutlich verbessert werden. Diese erhöhte Zuverlässigkeit der Ergebnisse ist wichtig für wissenschaftliche und klinische Fragestellungen, insbesondere wenn verfügbare Probanden, Messzeit oder Stimulusmaterial begrenzt sind. Daneben kann diese Art von Paradigmen für die Umsetzung realistischerer Szenarien genutzt werden. Dies ist im Kontext von sozialen Interaktionen wichtig und wurde von Wilms et al. (2010) vorgeschlagen. Für die Adaption der Stimulation kann neben externen Parametern, wie den Augenbewegungen oder der Herzrate, auch die Hirnaktivität selbst verwendet werden. Dabei wird die Stimulation anhand eines Zustands des Gehirns, der direkt aus dem EEG oder fMRT ermittelt werden muss, angepasst (Becker et al. 2011; Yoo et al. 2012). Dafür ist allerdings zu beachten, dass die Hirnsignale, die für die Adaption verwendet werden, von den zu untersuchenden Hirnregionen unabhängig sind. Nur damit kann eine zirkuläre Datenmanipulation vermieden werden (Kriegeskorte et al. 2009).

Insgesamt ist für ein adaptives Paradigma zu beachten, ob die Variabilität des gemessenen Verhaltens- oder physiologischen Parameters von Bedeutung für die zu untersuchende Fragestellung ist. Dies wäre z. B. der Fall wenn die Augenbewegungen für die Aufgabe unerlässlich sind. Mit anderen Worten heißt das, dass adaptive Paradigmen hilfreich sind wenn die Variabilität externer Einflussparameter durch die Adaption reduziert wird und dadurch die statistische Teststärke verbessert wird.

6 Zusammenfassung und Ausblick

6.1 Zusammenfassung

Das Forschungsfeld Echtzeit-fMRT als Schnittstelle zwischen Hirn und Computer ist ein junges Forschungsgebiet mit zahlreichen offenen Fragen. Um Antworten darauf zu finden, ist ein besseres Verständnis des menschlichen Gehirns und Verhaltens mit neuen neurowissenschaftlichen Experimenten notwendig. Damit werden neue Möglichkeiten für Therapien bei verschiedenen Erkrankungen oder in der Qualität der Steuerung eines computergestützten externen Systems erarbeitet.

Dies umschreibt eine interdisziplinäre Aufgabe, welche die Verarbeitung von fMRT-Daten in Echtzeit mit einem algorithmisch flexiblen und trotzdem anwendungstechnisch konsistentem Softwarekonzept für methodische Neuentwicklungen zur zuverlässigen Durchführung von neurowissenschaftlichen Experimenten umfasst. Dazu sind Softwaresysteme notwendig, mit denen die Gestaltung von Echtzeit-fMRT-Experimenten mit ihren vielseitigen Anforderungen und Neuentwicklungen effizient möglich ist. Bisherige Ansätze sind jeweils auf einzelne Anwendungsfälle beschränkt. Damit erfordern diese für die Bearbeitung einer neuen Fragestellung eine nahezu vollständige Softwareentwicklung. Weiterhin lassen die bisherigen System nur teilweise eine Integration neu entwickelter Algorithmen zu.

In dieser Arbeit wurde ein neues Softwareframework erarbeitet, das einerseits eine integrierte Gesamtlösung für die Echtzeit-fMRT darstellt und in seinen Teilmodulen eine abstrakte Basis für eine universelle Methodenentwicklung anbietet. Dazu wurde eine flexible Konfiguration des Systems über die XML-basierte Sprache EDL verwendet, um die gewählten Module zur Laufzeit des Systems einzubinden. Zusätzlich ist damit der Austausch der Module für die Datenvorverarbeitung, Auswertung und Feedback möglich. Die Integration neu entwickelter Ansätze ist somit klar definiert durchführbar. Mit diesen Eigenschaften ist die Konsistenz für die Ausführung von Anwendungen in Echtzeit-fMRT-Experimenten gewährleistet durch die Vereinheitlichung der allgemeinen Module. Gleichzeitig ist jedoch auch die Flexibilität für die Verwendung konkreter Algorithmen zur Datenverarbeitung und Auswertung gegeben. Dieser erweiterbarer Ansatz, der alle notwendigen Komponenten in einem Softwarepaket integriert, kombiniert die Vorteile verschiedener Systeme und war bei bisher vorhandenen Ansätzen nicht gegeben.

Besonders hervorzuheben an dem neuen Softwareframework ist die Entwicklung einer abstrakten Schnittstelle für externe Module zusammen mit einem internen abstrakten Konzept zur Adaption der Stimulation. Damit kann die Präsentation der Stimuli, die zum Paradigma des Experiments gehören, abhängig gemacht werden von externen Parametern. Die Abhängigkeiten sind in einem logischen Konstrukt beschrieben und werden in Echtzeit während der Laufzeit des Experiments evaluiert. Mit Hilfe einer 'Wenn-Dann-Entscheidung' können dann unterschiedliche Adaptionen ausgeführt werden. Für die Adaption wurden verschiedene Funktionen definiert. Aus dem adaptierten Paradigma werden ebenfalls in Echtzeit die notwendigen Informationen berechnet, die für die Echtzeitauswertung der Daten notwendig sind. Eine so flexible Abstraktion der Stimulation und die Kombination mit der Echtzeitauswertung ist bisher einzigartig für neurowissenschaftliche Experimente.

Die verschiedenen Funktionalitäten des entwickelten Softwareframeworks wurden im empirischen Teil der vorliegenden Arbeit validiert. In einer Probandenstudie wurde ein Paradigma anhand der Blickrichtung der Probanden adaptiert. Es konnte gezeigt werden, dass die Hälfte der Probanden der expliziten Anweisungen zur Durchführung des Experiments nicht gefolgt ist und deshalb keine zuverlässigen Ergebnisse erzielt hat. Durch die Adaption des Paradigmas konnte für das Verhalten dieser 'Low-Compliant'-Probanden zur Laufzeit des Experiments kontrolliert und somit die Stimulation optimiert werden. Die erzielten Effektstärken in emotionsbezogenen Hirnregionen konnten dadurch auf das Niveau der 'High-Compliant'-Probanden angehoben werden. Die Zuverlässigkeit und Reproduzierbarkeit der fMRT-Ergebnisse wurde somit durch adaptive Paradigmen signifikant verbessert. In zwei weiteren Experimenten wurden exemplarisch die GLM-Analyse und die Kombination der Echtzeitauswertung mit der adaptiven Stimulation validiert. Die Analyseergebnisse wurden mit denen eines häufig verwendeten Softwarepakets für die fMRT-Datenauswertung verglichen. Die Ergebnisse zeigen eine hohe bis sehr hohe Übereinstimmung.

6.2 Ausblick

Basierend auf der vorgestellten Arbeit sollen zahlreiche weitere Arbeiten aufgebaut werden. Dazu gehören vor allem die Entwicklung von neuen Algorithmen sowie die Anwendung bei bisher unmöglichen Fragestellungen. Parallel zu den in dieser Arbeit beschriebenen Algorithmen wurden bereits weitere für die Echtzeit-Vorverarbeitung der fMRT-Daten (Bewegungskorrektur, Registrierung, Normalisierung) implementiert. Diese sollen zeitnah in das Gesamtsystem integriert werden und müssen anschließend validiert werden. Außerdem wurde bereits eine verbesserte graphische Oberfläche zur vereinfachten Bedienung entwickelt. Diese wird ebenfalls vollständig integriert und validiert. Mit diesem erweiterten System können dann Neurofeedbackexperimen-

te durchgeführt werden. Eine schematische Darstellung für das angestrebte Gesamtsystem ist in Abbildung 6.1 abgebildet.

In Bezug auf die adaptiven Paradigmen sollen weitere externe Komponenten integriert werden. Hierbei ist vor allem eine EEG-Einbindung von besonderem Interesse. Ein erstes Paradigma für eine solche Adaption wurde von Becker et al. (2011) mit Hilfe des posterioren α -Rhythmus vorgestellt. Das EEG bietet einen großen Parameterraum für die Erweiterung dieser Idee. Des Weiteren wurden für die Adaption durch andere Parameter, wie die Herzrate oder den Blutdruck, bereits erste Ideen von Kollegen am Max-Planck-Institut für Kognitions- und Neurowissenschaften Leipzig entwickelt, deren Details noch ausgearbeitet werden.

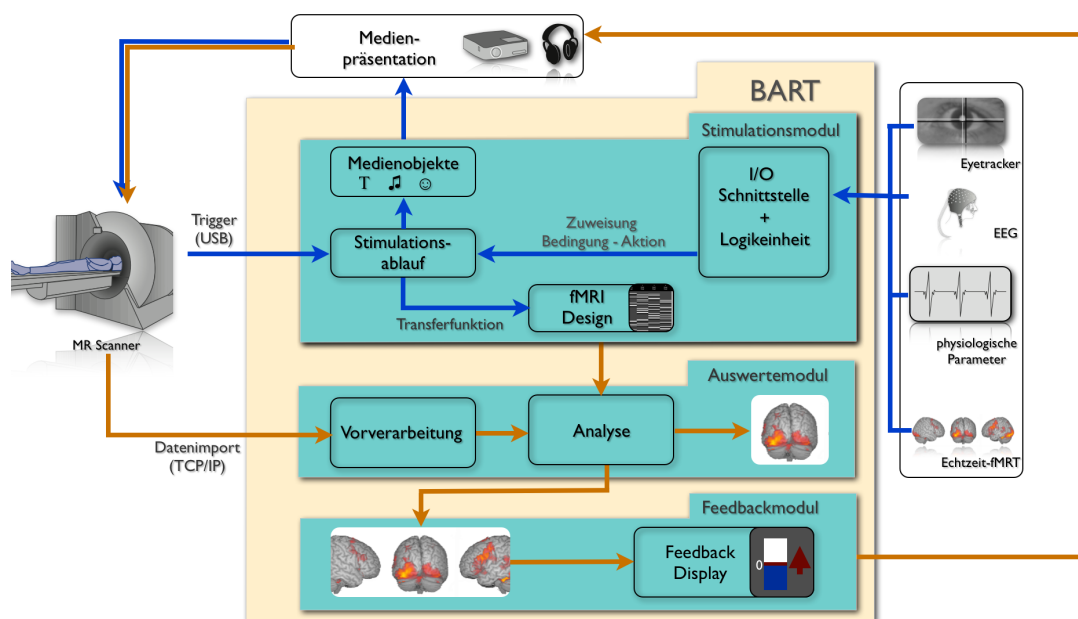


Abbildung 6.1: Der Ausblick auf das Schema des Softwaresystems BART zeigt weitere Plugins, die Parameter zur Adaption des Paradigmas einbinden sollen. Zusätzlich soll das Feedbackmodul vollständig integriert werden. Weiterhin werden zur Zeit neue Algorithmen für die Vorverarbeitung und Analyse entwickelt mit denen die Funktionalität erweitert werden soll.

Für die algorithmische Weiterentwicklung sollen neue Methoden zur Echtzeitanalyse der Daten entwickelt werden. Eine spannende Überlegung ist, die Konnektivitäten zwischen verschiedenen Hirnregionen in Echtzeit zu bestimmen um damit das Neurofeedback zu gestalten. Ein erster Ansatz dazu mit einer funktionellen Konnektivitätsanalyse wurde kürzlich von Koush et al. (2013) präsentiert. Ein vielversprechender Ansatz für die Bestimmung effektiver Konnektivitäten findet sich in Rajapakse et al. (2008). Für diesen Ansatz bleibt zu ermitteln, ob eine Echtzeitberechnung möglich ist. Neben Konnektivitätsanalysen ist die Entwicklung von neuen Klassifikationsalgorithmen ein spannendes Aufgabenfeld. Als Anwendung der vorgestellten Software ist die Durchführung von EEG/fMRT-kombinierten Paradigmen eine spannende Aufgabenstellung (Zotev et al. 2014).

Weiterhin sind zahlreiche Ideen für Neurofeedbackexperimente vorhanden, für die in Zusammenarbeit bereits Pilotexperimente im Kontext von Adipositas (Hollmann et al. 2012) und Emotionsregulation (Okon-Singer et al. 2014) durchgeführt wurden.

A Anhang

A.1 Anhänge zur Software BART

A.1.1 Klassendiagramm Datenmanagement

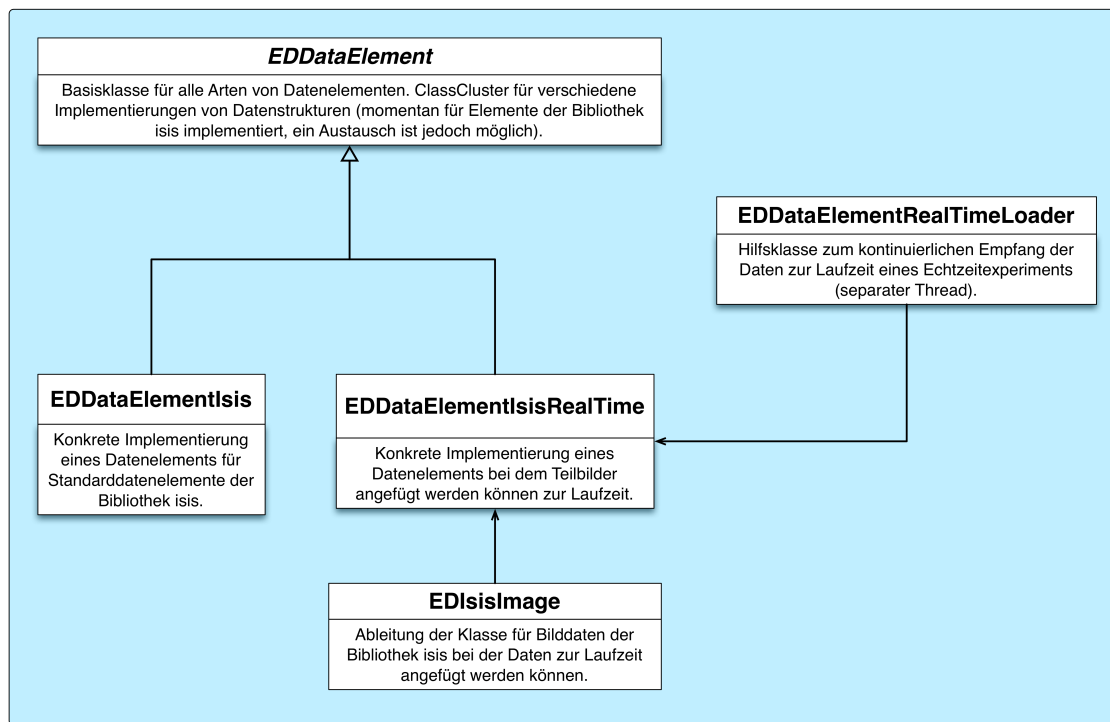


Abbildung A.1: Klassendiagramm Bibliothek Datenmanagement

A.1.2 Klassendiagramm Konfiguration

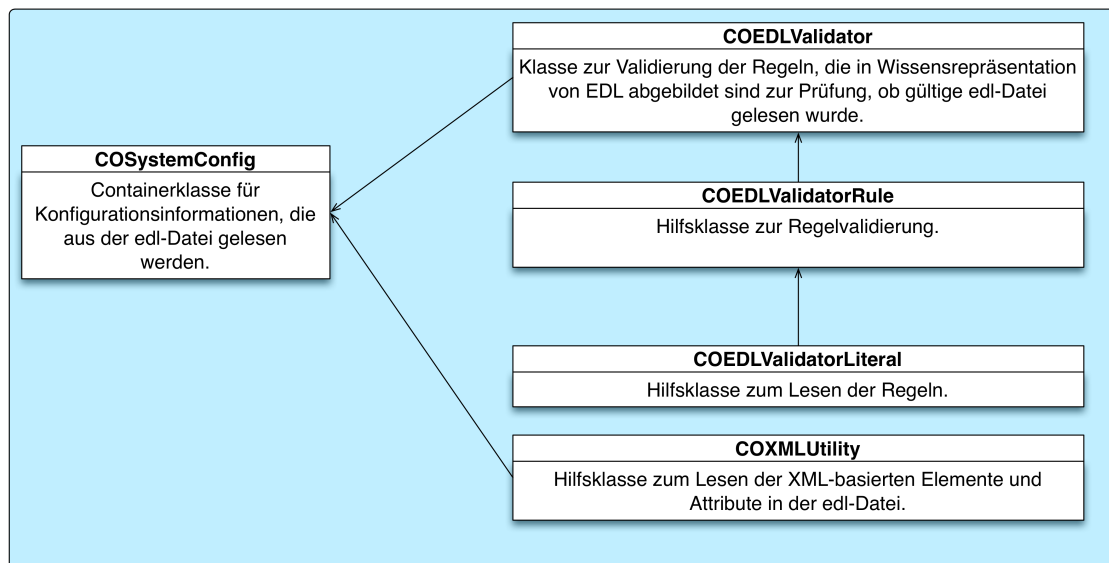


Abbildung A.2: Klassendiagramm Bibliothek Konfiguration

A.2 Anhänge zu Spezifikationserweiterungen in EDL

A.2.1 Vollständige XSD-Spezifikation von Bedingungen (Constraints) für adaptive Paradigmen

```

<xs:element name="constraints">
  <xs:complexType>
    <xs:sequence>
      <xs:element maxOccurs="1" minOccurs="1"
        ref="systemVariables"></xs:element>
      <xs:element maxOccurs="unbounded" minOccurs="1"
        ref="constraint"></xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="constraint">
  <xs:complexType>
    <xs:sequence>
      <xs:element maxOccurs="1" minOccurs="1" name="conditions">
        <xs:complexType>
          <xs:sequence>
            <xs:element maxOccurs="unbounded" minOccurs="1"
              name="condition">
              <xs:complexType>
                <xs:attribute name="systemVariableRef"
                  type="xs:IDREF"
                  use="required"></xs:attribute>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element maxOccurs="1" minOccurs="0" name="stimulusActions_then">
        <xs:complexType>
          <xs:sequence>
            <xs:element maxOccurs="unbounded" minOccurs="1"
              ref="stimulusAction"></xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element maxOccurs="1" minOccurs="0" name="stimulusActions_else">
        <xs:complexType>
          <xs:sequence>
            <xs:element maxOccurs="unbounded" minOccurs="1"
              ref="stimulusAction"></xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="constraintID" type="xs:ID" use="required">
    </xs:attribute>
  </xs:complexType>
</xs:element>
<xs:element name="systemVariables">
  <xs:complexType>

```

```

<xs:sequence>
  <xs:element maxOccurs="unbounded" minOccurs="1"
    name="systemVariable">
    <xs:complexType>
      <xs:attribute name="source"
        type="xs:string"
        use="required"/></xs:attribute>
      <xs:attribute name="systemVariableName"
        type="xs:string"
        use="required"/></xs:attribute>
      <xs:attribute name="systemVariableID" type="xs:ID"
        use="required"/></xs:attribute>
    </xs:complexType>
  </xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="stimulusAction">
  <xs:complexType>
    <xs:sequence>
      <xs:element maxOccurs="unbounded" minOccurs="0"
        name="setMediaObjectParameter">
        <xs:complexType>
          <xs:attribute name="parameterName"
            type="xs:string"
            use="required"/></xs:attribute>
          <xs:attribute name="systemVariableRef"
            type="xs:IDREF"
            use="required"/></xs:attribute>
        </xs:complexType>
      </xs:element>
      <xs:element maxOccurs="1" minOccurs="0"
        name="replaceMediaObject">
        <xs:complexType>
          <xs:attribute name="mediaObjectRef"
            type="xs:IDREF"
            use="required"/></xs:attribute>
        </xs:complexType>
      </xs:element>
      <xs:element maxOccurs="1" minOccurs="0"
        name="insertNewStimulusEvent">
        <xs:complexType>
          <xs:attribute name="duration"
            type="xs:unsignedInt"
            use="required"/></xs:attribute>
          <xs:attribute name="mediaObjectRef"
            type="xs:IDREF"
            use="required"/></xs:attribute>
          <xs:attribute name="shiftFollowingStimulusEvents"
            type="xs:boolean"
            use="required"/></xs:attribute>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

Auflistung A.1: Die Vollständige XSD-Spezifikation von Bedingungen (Constraints) für adaptive Paradigmen

A.2.2 Vollständige XSD-Spezifikation von dynamischen Regressoren für adaptive Paradigmen

```

<xs:element name="paradigm">
  <xs:complexType>
    <xs:sequence>
      <xs:choice>
        <xs:element maxOccurs="1" minOccurs="0"
          ref="gwDesignStruct"></xs:element>
        <xs:element maxOccurs="1" minOccurs="0"
          ref="swDesignStruct"></xs:element>
        <xs:element maxOccurs="1" minOccurs="0"
          ref="dynamicDesignStruct"></xs:element>
      </xs:choice>
      <xs:element maxOccurs="1" minOccurs="0"
        ref="covariateStruct"></xs:element>
    </xs:sequence>
    <xs:attribute name="ignoreScansAtStart"
      type="xs:unsignedInt"
      use="required"></xs:attribute>
  </xs:complexType>
</xs:element>

<xs:element name="dynamicDesignStruct">
  <xs:annotation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element maxOccurs="unbounded" minOccurs="1"
        ref="dynamicTimeBasedRegressor"></xs:element>
    </xs:sequence>
    <xs:attribute name="maxLength"
      type="xs:unsignedLong"
      use="required"></xs:attribute>
  </xs:complexType>
</xs:element>

<xs:element name="dynamicTimeBasedRegressor">
  <xs:complexType>
    <xs:attribute name="regressorID"
      type="xs:ID"
      use="required"></xs:attribute>
    <xs:attribute name="name"
      type="xs:string"
      use="required"></xs:attribute>
    <xs:attribute name="length"
      type="xs:unsignedLong"
      use="required"></xs:attribute>
    <xs:attribute name="useRefFct"
      type="xs:IDREF"
      use="required"></xs:attribute>
    <xs:attribute default="false" name="useRefFctFirstDerivative"

```

```

        type="xs:boolean"
        use="optional"></xs:attribute>
<xs:attribute default="false" name="useRefFctSecondDerivative"
        type="xs:boolean"
        use="optional"></xs:attribute>
<xs:attribute name="scaleHeightToZeroMean"
        type="xs:boolean"
        use="required"></xs:attribute>
</xs:complexType>
</xs:element>

```

Auflistung A.2: Vollständige XSD-Spezifikation von dynamischen Regressoren für adaptive Paradigmen

A.2.3 Vollständige XSD-Spezifikation von Transferfunktionen für adaptive Paradigmen

```

<xs:element name="transferFunctions">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="transferFunction">
        <xs:complexType>
          <xs:attribute name="transferFunctionID"
            type="xs:ID"
            use="required"></xs:attribute>
          <xs:attribute name="timeOffset"
            type="xs:int"
            use="optional"
            default="0" ></xs:attribute>
          <xs:attribute name="durationScaleFactor"
            type="xs:double"
            use="optional"
            default="1"></xs:attribute>
          <xs:attribute name="parametricScaleFactor"
            type="xs:double"
            use="optional"
            default="1"></xs:attribute>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

Auflistung A.3: Vollständige XSD-Spezifikation von Transferfunktionen für adaptive Paradigmen

A.3 Anhänge zum Experiment adaptives Paradigma

A.3.1 SAM Ratings

Mit Hilfe des *Self-Assessment-Manikin* (SAM) werden die IAPS-Bilder bezüglich ihrer Wertigkeit (Valenz) und der durch das Bild ausgelösten Erregung (Bradley et al. 1994) bewertet. In dieser Arbeit haben alle Probanden im Anschluss an das fMRT-Experiment die verwendeten Bilder diesbezüglich bewertet. Dies erfolgte anhand der im folgenden dargestellten Skalen, die zusammen mit den Bildern angezeigt wurden.

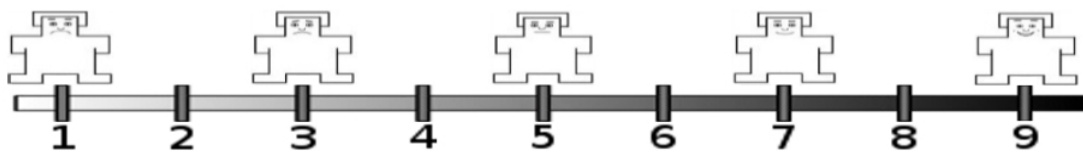


Abbildung A.3: Verwendete Grafik zur Bewertung der Wertigkeit (Valenz) des Bildes durch die Probanden: 1 - sehr negativ; 5 - neutral; 9 - sehr positiv

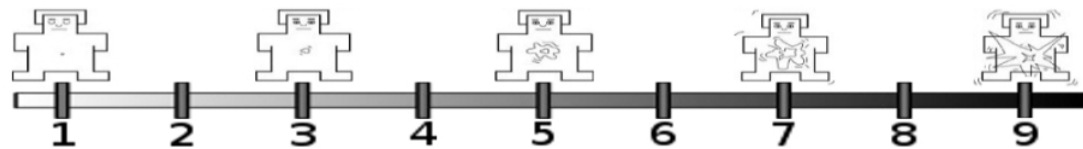


Abbildung A.4: Verwendete Grafik zur Bewertung der Erregung, die vom jeweiligen Bild ausgelöst wurde: 1 - keine Erregung; 5 - mittlere Erregung; 9 - maximale Erregung

A.3.2 Stimulusbilder Übersicht

Die folgende Tabelle beinhaltet die Namen aus der IAPS-Datenbank der im fMRT-Experiment aus Abschnitt 4.2 verwendeten Stimulusbilder. Die Bilder wurden modifiziert, wie im Abschnitt 4.2.3 beschrieben. Das vollständige Stimulusmaterial befindet sich auf dem beigelegten Datenträger.

 Übersicht über verwendete IAPS-Bilder

IAPS-Bilder mit negativer Wertung

1019	1050	1200	1201	1302	1303	1525	1726	1820
1930	1932	1945	2053	2120	2278	2301	2312	2345
2352	2410	2456	2457	2458	2692	2703	2710	2717
2750	2770	2811	2981	3000	3001	3010	3015	3016
3017	3019	3030	3051	3053	3059	3060	3061	3062
3063	3064	3068	3069	3071	3080	3100	3102	3103
3120	3130	3131	3140	3150	3160	3168	3170	3180
3181	3185	3195	3213	3225	3261	3266	3300	3350
3400	3500	3550	6010	6020	6021	6022	6230	6241
6250	6260	6313	6350	6415	6540	6550	6560	8480
8485	9008	9040	9042	9043	9102	9140	9180	9181
9183	9185	9187	9253	9265	9300	9301	9320	9321
9322	9325	9405	9410	9594	9561	2095-1	2375-1	2800-1
2900-1	4664-2	9302-1						

IAPS-Bilder mit neutraler Wertung

1230	1410	1440	1500	1540	1602	1603	1610	1630
1650	1670	1903	2102	2107	2122	2190	2191	2200
2210	2214	2215	2221	2240	2302	2305	2342	2362
2372	2374	2383	2441	2445	2480	2484	2488	2493
2495	2499	2500	2512	2520	2575	2620	2630	2635
2749	2830	2840	2980	5010	5030	5130	5301	5390
5395	5410	5470	5500	5530	5531	5594	5626	5726
5740	5825	5829	5890	5920	6150	7000	7002	7003
7004	7006	7009	7010	7014	7017	7018	7020	7021
7025	7026	7030	7031	7032	7035	7040	7042	7043
7045	7052	7055	7056	7059	7077	7080	7090	7100
7110	7150	7160	7170	7175	7185	7187	7195	7211
7217	7235	7590	10000	10001	10006	10017	10030	10059
10062	10114	6570-2						

Literaturverzeichnis

- Abdelnour, A Farras und Theodore Huppert (Mai 2009). "Real-time imaging of human brain function by near-infrared spectroscopy using an adaptive general linear model." In: *NeuroImage* 46.1, S. 133–143.
- Amaro, Edson und Gareth J Barker (Apr. 2006). "Study design in fMRI: basic principles." In: *Brain and Cognition* 60.3, S. 220–32.
- Andersson, Patrik, Nick F Ramsey, Mathijs Raemaekers, Max A Viergever und Josien P W Pluim (Juli 2012). "Real-time decoding of the direction of covert visuospatial attention." In: *Journal of Neural Engineering* 9.4, S. 1–10.
- Arend, Isabel, Robert Rafal und Robert Ward (Aug. 2008). "Spatial and temporal deficits are regionally dissociable in patients with pulvinar lesions." In: *Brain* 131.8, S. 2140–2152.
- Bagarinao, Epifanio, Kayako Matsuo, Toshiharu Nakai und Shunsuke Sato (Juni 2003). "Estimation of general linear model coefficients for real-time application." In: *NeuroImage* 19.2, S. 422–429.
- Baraldi, Patrizia, Carlo Adolfo Porro, Marco Serafini, Guiseppe Pagnoni, Cristina Murari, Ruggero Corazza und Paolo Nichelli (Juli 1999). "Bilateral representation of sequential finger movements in human cortical areas." In: *Neuroscience Letters* 269.2, S. 95–98.
- Becker, Robert, Matthias Reinacher, Frank Freyer, Arno Villringer und Petra Ritter (Juli 2011). "How Ongoing Neuronal Oscillations Account for Evoked fMRI Variability". In: *Journal of Neuroscience* 31.30, S. 11016–11027.
- Birbaumer, Niels und Leonardo G Cohen (März 2007). "Brain-computer interfaces: communication and restoration of movement in paralysis." In: *The Journal of Physiology* 579.3, S. 621–636.
- Birbaumer, Niels, N Ghanayim, Thilo Hinterberger, I Iversen, B Kotchoubey, Andrea Kübler, J Perelmouter, E Taub und H Flor (März 1999). "A spelling device for the paralysed." In: *Nature* 398.6725, S. 297–298.
- Boecker, H, A Kleinschmidt, M Requardt, W Hänicke, K D Merboldt und J Frahm (Dez. 1994). "Functional cooperativity of human cortical motor areas during self-paced simple finger movements. A high-resolution MRI study." In: *Brain* 117.6, S. 1231–1239.
- Bradley, Margaret M und Peter J Lang (März 1994). "Measuring emotion: The self-assessment manikin and the semantic differential". In: *Journal of Behavior Therapy and Experimental Psychiatry* 25.1, S. 49–59.

- Bray, Signe, Shinsuke Shimojo und John P O'Doherty (Juli 2007). "Direct instrumental conditioning of neural activity using functional magnetic resonance imaging-derived reward feedback." In: *Journal of Neuroscience* 27.28, S. 7498–7507.
- Buck, Erik und Donald A. Yacktmann (2009). *Cocoa Design Patterns*. Hrsg. von Addison-Wesley Longman. 1. Aufl. Amsterdam: Longman, Addison-Wesley, S. 456.
- Caria, Andrea, Ralf Veit, Ranganatha Sitaram, Martin Lotze, Nikolaus Weiskopf, Wolfgang Grodd und Niels Birbaumer (Apr. 2007). "Regulation of anterior insular cortex activity using real-time fMRI." In: *NeuroImage* 35.3, S. 1238–1246.
- Chapin, Heather, Epifanio Bagarinao und Sean Mackey (Juni 2012). "Real-time fMRI applied to pain management." In: *Neuroscience Letters* 520.2, S. 174–81.
- Coben, Robert und James R. Evans (2011). *Neurofeedback and Neuromodulation: Techniques and Applications*. 1. Aufl. Elsevier Inc., S. 431.
- Collins, Louis D, Peter Neelin, Terrence M Peters und Alan C Evans (1994). "Automatic 3D Intersubject Registration of MR Volumetric Data in Standardized Talairach Space". In: *Journal of Computer Assisted Tomography* 18.2, S. 192–205.
- Cox, R W und A Jesmanowicz (Dez. 1999). "Real-time 3D image registration for functional MRI." In: *Magnetic Resonance in Medicine* 42.6, S. 1014–1018.
- Cox, R W, A Jesmanowicz und J S Hyde (Feb. 1995). "Real-time functional magnetic resonance imaging." In: *Magnetic Resonance in Medicine* 33.2, S. 230–236.
- DeCharms, R Christopher, Kalina Christoff, Gary H Glover, John M Pauly, Susan Whitfield und John D E Gabrieli (Jan. 2004). "Learned regulation of spatially localized brain activation using real-time fMRI." In: *NeuroImage* 21.1, S. 436–443.
- DeCharms, R Christopher, Fumiko Maeda, Gary H Glover, David Ludlow, John M Pauly, Deepak Soneji, John D E Gabrieli und Sean C Mackey (Dez. 2005). "Control over brain activation and pain learned by using real-time functional MRI." In: *Proceedings of the National Academy of Sciences of the United States of America* 102.51, S. 18626–18631.
- Erlenkötter, Helmut (2003). *XML - Extensible Markup Language*. 2. Aufl. Hamburg: rororo, S. 288.
- Esposito, Fabrizio, Erich Seifritz, Elia Formisano, Renato Morrone, Tommaso Scarabino, Gioacchino Tedeschi, Sossio Cirillo, Rainer Goebel und Francesco Di Salle (Dez. 2003). "Real-time independent component analysis of fMRI time-series." In: *NeuroImage* 20.4, S. 2209–2224.
- Fazli, Siamac, Jan Mehnert, Jens Steinbrink, Gabriel Curio, Arno Villringer, Klaus-Robert Müller und Benjamin Blankertz (Jan. 2012). "Enhanced performance by a hybrid NIRS-EEG brain computer interface." In: *NeuroImage* 59.1, S. 519–29.
- Feigl, G C, S Safavi-Abbasi, A Gharabaghi, V Gonzalez-Felipe, A El Shawarby, H-J. Freund und M Samii (Juni 2008). "Real-time 3T fMRI data of brain tumour patients for intra-operative localization of primary motor areas." In: *European Journal of Surgical Oncology* 34.6, S. 708–715.

- Fox, Michael D und Marcus E Raichle (Sep. 2007). "Spontaneous fluctuations in brain activity observed with functional magnetic resonance imaging." In: *Nature Reviews Neuroscience* 8.9, S. 700–711.
- Friston, K J, A P Holmes, J B Poline, P J Grasby, S C Williams, R S Frackowiak und R Turner (März 1995). "Analysis of fMRI time-series revisited." In: *NeuroImage* 2.1, S. 45–53.
- Friston, Karl J, John T Ashburner, Stefan J Kiebel, Thomas E Nichols und William D Penny (2007). *Statistical Parametric Mapping*. 1. Aufl. Amsterdam: Elsevier, S. 647.
- Gamma, Erich, Richard Helm und Ralph E. Johnson (1994). *Design Patterns. Elements of Reusable Object-Oriented Software*. 1. Aufl. Amsterdam: Addison-Wesley Longman, S. 395.
- Gellert, W., H. Küstner, M. Hellwich und H. Kästner (1972). *Kleine Enzyklopädie Mathematik*. Frankfurt/M.: Verlag Harri Deutsch, S. 837.
- Gembris, Daniel, John G Taylor, Stefan Schor, Wolfgang Frings, Dieter Suter und Stefan Posse (Feb. 2000). "Functional magnetic resonance imaging in real time (FIRE): sliding-window correlation analysis and reference-vector optimization." In: *Magnetic Resonance in Medicine* 43.2, S. 259–268.
- Gläscher, Jan (Jan. 2009). "Visualization of group inference data in functional neuroimaging." In: *Neuroinformatics* 7.1, S. 73–82.
- Glover, G H (Apr. 1999). "Deconvolution of impulse response in event-related BOLD fMRI." In: *NeuroImage* 9.4, S. 416–429.
- Haacke, Mark E., Robert W. Brown, Michael R. Thompson und Ramesh Venkatesan (1999). *Magnetic resonance imaging: physical principles and sequence design*. 1. Aufl. New York: Wiley, S. 914.
- Haller, Sven, Niels Birbaumer und Ralf Veit (März 2010). "Real-time fMRI feedback training may improve chronic tinnitus." In: *European Radiology* 20.3, S. 696–703.
- Hamilton, J Paul, Gary H Glover, Jung-Jiin Hsu, Rebecca F Johnson und Ian H Gotlib (Jan. 2011). "Modulation of subgenual anterior cingulate cortex activity with real-time neurofeedback." In: *Human Brain Mapping* 32.1, S. 22–31.
- Haus, Karl-Michael, Carla Held, Axel Kowalski, Andreas Krombholz, Manfred Nowak, Edith Schneider, Gert Strauß und Meike Wiedemann (2012). *Praxisbuch Biofeedback und Neurofeedback*. Springer, Berlin, S. 300.
- Hawkinson, Jon E, Amy J Ross, Sudharshan Parthasarathy, David J Scott, Ella a Laramee, Lainie J Posecion, William R Rekshan, Kristen E Sheau, Nkechi D Njaka, Peter J Bayley und R Christopher DeCharms (Juni 2011). "Quantification of Adverse Events Associated with Functional MRI Scanning and with Real-Time fMRI-Based Training." In: *International Journal of Behavioral Medicine* 19, S. 372–381.
- Hellrung, Lydia, Maurice Hollmann, Torsten Schlumm, Oliver Zschoyge, Christian Kalberlah, Elisabeth Roggenhofer, Hadas Okon-Singer, Arno Villringer und Annette Horstmann (2015). "Flexible adaptive paradigms for fMRI using a novel software package 'Brain Analysis in Real-Time (BART)". In: *PLoS ONE* 10.4.

- Hinds, Oliver, Satrajit Ghosh, Todd W Thompson, Julie J Yoo, Susan Whitfield-Gabrieli, Christina Triantafyllou und John D E Gabrieli (Jan. 2011). "Computing moment-to-moment BOLD activation for real-time neurofeedback." In: *NeuroImage* 54.1, S. 361–368.
- Hollmann, M, L Hellrung, B Pleger, H Schlögl, S Kabisch, M Stumvoll, A Villringer und A Horstmann (Mai 2012). "Neural correlates of the volitional regulation of the desire for food." In: *International Journal of Obesity* 36.5, S. 648–55.
- Hollmann, M, T Mönch, S Mulla-Osman, C Tempelmann, J Stadler und J Bernarding (Okt. 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." In: *Journal of Neuroscience Methods* 175.1, S. 154–162.
- Hollmann, Maurice, Jochem W Rieger, Sebastian Baecke, Ralf Lützkendorf, Charles Müller, Daniela Adolf und Johannes Bernarding (Jan. 2011). "Predicting decisions in human social interactions using real-time fMRI and pattern classification." In: *PLoS ONE* 6.10, e25304.
- Johnson, Kevin a, Karen Hartwell, Todd LeMatty, Jeffrey Borckardt, Paul S Morgan, Koushik Govindarajan, Kathleen Brady und Mark S George (Jan. 2012). "Intermittent "real-time" fMRI feedback is superior to continuous presentation for a motor imagery task: a pilot study." In: *Journal of Neuroimaging* 22.1, S. 58–66.
- Johnston, S J, S G Boehm, D Healy, R Goebel und D E J Linden (Jan. 2010). "Neurofeedback: A promising tool for the self-regulation of emotion networks." In: *NeuroImage* 49.1, S. 1066–1072.
- Kansaku, Kenji, Shigeru Muraki, Shinji Umeyama, Yasunori Nishimori, Takanori Kochiyama, Shigeru Yamane und Shigeru Kitazawa (Nov. 2005). "Cortical activity in multiple motor areas during sequential finger movements: an application of independent component analysis." In: *NeuroImage* 28.3, S. 669–681.
- Koush, Yury, Maria Joao Rosa, Fabien Robineau, Klaartje Heinen, Sebastian Rieger, Nikolaus Weiskopf, Patrik Vuilleumier, Dimitri Van De Ville und Frank Scharnowski (Mai 2013). "Connectivity-based neurofeedback: Dynamic causal modeling for real-time fMRI." In: *NeuroImage* 81.1, S. 422–430.
- Kriegeskorte, Nikolaus, W Kyle Simmons, Patrick S F Bellgowan und Chris I Baker (Mai 2009). "Circular analysis in systems neuroscience: the dangers of double dipping." In: *Nature Neuroscience* 12.5, S. 535–40.
- Krüger, Gunnar und Gary H Glover (2001). "Physiological Noise in Oxygenation-Sensitive Magnetic Resonance Imaging". In: *Magnetic Resonance in Medicine* 46, S. 631–637.
- LaConte, Stephen M. (Juni 2010). "Decoding fMRI brain states in real-time". In: *NeuroImage* 56.2, S. 440–454.
- LaConte, Stephen M, Scott J Peltier und Xiaoping P Hu (Okt. 2007). "Real-time fMRI using brain-state classification." In: *Human Brain Mapping* 28.10, S. 1033–1044.
- Lahres, Bernhard und Gregor Rayman (2009). *Objektorientierte Programmierung*. Hrsg. von Galileo Computing. 2. Aufl. Galileo Computing, S. 656.

- Lang, P.J., M.M. Bradley und B.N. Cuthbert (2008). *International affective picture system (IAPS): Affective ratings of pictures and instruction manual*. Techn. Ber. Gainesville, FL: University of Florida.
- Lee, Jong-Hwan, Junghoe Kim und Seung-Schik Yoo (Mai 2012). "Real-time fMRI-based neurofeedback reinforces causality of attention networks." In: *Neuroscience Research* 72.4, S. 347–54.
- Lee, Jong-Hwan, Heather M O'Leary, Hyunwook Park, Ferenc A Jolesz und Seung-Schik Yoo (Feb. 2008). "Atlas-based multichannel monitoring of functional MRI signals in real-time: automated approach." In: *Human Brain Mapping* 29.2, S. 157–166.
- Lee, Jong-Hwan, Jeongwon Ryu, Ferenc a Jolesz, Zang-Hee Cho und Seung-Schik Yoo (Jan. 2009). "Brain-machine interface via real-time fMRI: preliminary study on thought-controlled robotic arm." In: *Neuroscience Letters* 450.1, S. 1–6.
- Leopold, David a und Alexander Maier (Okt. 2012). "Ongoing physiological processes in the cerebral cortex." In: *NeuroImage* 62.4, S. 2190–200.
- Lim, Seung-Lark, Srikanth Padmala und Luiz Pessoa (Sep. 2009). "Segregating the significant from the mundane on a moment-to-moment basis via direct and indirect amygdala contributions." In: *Proceedings of the National Academy of Sciences of the United States of America* 106.39, S. 16841–6.
- Linden, David E J, Isabelle Habes, Stephen J Johnston, Stefanie Linden, Ranjit Tatineni, Leena Subramanian, Bettina Sorger, David Healy und Rainer Goebel (Jan. 2012). "Real-time self-regulation of emotion networks in patients with depression." In: *PLoS ONE* 7.6, e38115.
- Linkenkaer-Hansen, Klaus, Vadim V Nikulin, Satu Palva, Risto J Ilmoniemi und J Matias Palva (Nov. 2004). "Prestimulus oscillations enhance psychophysical performance in humans." In: *The Journal of Neuroscience* 24.45, S. 10186–90.
- Logothetis, Nikos K (Juni 2008). "What we can do and what we cannot do with fMRI." In: *Nature* 453.7197, S. 869–878.
- Lohmann, Gabriele, Karsten Mueller, Volker Bosch, Heiko Mentzel, Sven Hessler, Lin Chen, S Zysset und D.Yves von Cramon (Nov. 2001). "Lipsia - a new software system for the evaluation of functional magnetic resonance images of the human brain". In: *Computerized Medical Imaging and Graphics* 25.6, S. 449–457.
- Mathiak, K und S Posse (Jan. 2001). "Evaluation of motion and realignment for functional magnetic resonance imaging in real time." In: *Magnetic Resonance in Medicine* 45.1, S. 167–171.
- Mathiak, Krystyna a, Yury Koush, Miriam Dyck, Tilman J Gaber, Eliza Alawi, Florian D Zepf, Mikhail Zvyagintsev und Klaus Mathiak (Nov. 2010). "Social reinforcement can regulate localized brain activity." In: *European Archives of Psychiatry and Clinical Neuroscience* 260.Suppl 2, S. 132–136.
- Möller, H. E. (2006). "Grundlagen der MRT". In: *Ganzkörper-MR-Tomographie*. Hrsg. von E.J. Rummeny, P. Reimer und W. Heindel. 2. Aufl. Stuttgart: Thieme Verlag. Kap. 1, S. 1–26.

- Monastra, Vincent J., Steven Lynn, Michael Linden, Joel F. Lubar, John Gruzelier und Theodore J. LaVaque (2005). "Electroencephalographic Biofeedback in the Treatment of Attention-Deficit/Hyperactivity Disorder". In: *Applied Psychophysiology and Biofeedback* 30.2, S. 95–114.
- Mueller, Charles, Michael Luehrs, Sebastian Baecke, Daniela Adolf, Ralf Luetzkendorf, Michael Luchtman und Johannes Bernarding (Juli 2012). "Building virtual reality fMRI paradigms: A framework for presenting immersive virtual environments." In: *Journal of Neuroscience Methods* 209.2, S. 290–298.
- Ogawa, S, T M Lee, a R Kay und D W Tank (Dez. 1990). "Brain magnetic resonance imaging with contrast dependent on blood oxygenation." In: *Proceedings of the National Academy of Sciences of the United States of America* 87.24, S. 9868–72.
- Okon-Singer, Hadas, Jan Mehnert, Jana Hoyer, Lydia Hellrung, Herma Lina Schaare, Juergen Dukart und Arno Villringer (März 2014). "Neural control of vascular reactions: impact of emotion and attention." In: *The Journal of neuroscience : the official journal of the Society for Neuroscience* 34.12, S. 4251–9.
- Oppelt, Arnulf (2005). *Imaging Systems for Medical Diagnosis*. Hrsg. von Siemens Aktiengesellschaft. 1. Aufl. Erlangen: Publics Corporate Publishing, S. 996.
- Padmala, Srikanth und Luiz Pessoa (Mai 2010). "Moment-to-moment fluctuations in fMRI amplitude and interregion coupling are predictive of inhibitory performance." In: *Cognitive, Affective and Behavioral Neurosciences* 10.2, S. 279–97.
- Pessoa, L, M McKenna, E Gutierrez und L G Ungerleider (Aug. 2002a). "Neural processing of emotional faces requires attention." In: *Proceedings of the National Academy of Sciences of the United States of America* 99.17, S. 11458–63.
- Pessoa, Luiz (2004). "Neuroimaging studies of attention and the processing of emotion-laden stimuli". In: *Progress in Brain Research* 144.1995, S. 171–182.
- Pessoa, Luiz und Ralph Adolphs (Nov. 2010). "Emotion processing and the amygdala: from a 'low road' to 'many roads' of evaluating biological significance." In: *Nature Reviews Neuroscience* 11.11, S. 773–83.
- Pessoa, Luiz, Sabine Kastner und Leslie G Ungerleider (Dez. 2002b). "Attentional control of the processing of neutral and emotional stimuli". In: *Cognitive Brain Research* 15.1, S. 31–45.
- Phan, K Luan, Daniel A Fitzgerald, Kunxiu Gao, Gregory J Moore, Manuel E Tancer und Stefan Posse (März 2004). "Real-time fMRI of cortico-limbic brain activity during emotional processing." In: *Neuroreport* 15.3, S. 527–532.
- Phillips, Raquel, Vadim Zotev, Kimberly Young, Wayne C Drevets und Jerzy Bodurka (Jan. 2012). "Self-regulation of Amygdala Activation with Real-Time fMRI Neurofeedback in MDD Patients". In: *Proceedings of the OHBM 2012*. Bd. 6. 9.
- Posse, S, F Binkofski, F Schneider, D Gembris, W Frings, U Habel, J B Salloum, K Mathiak, S Wiese, V Kiselev, T Graf, B Elghahwagi, M L Grosse-Ruyken und T Eickermann (Jan. 2001). "A new approach to measure single-event related brain activity using real-time fMRI:

- feasibility of sensory, motor, and higher cognitive tasks.” In: *Human Brain Mapping* 12.1, S. 25–41.
- Posse, Stefan, Elena Ackley, Radu Mutihac, Jochen Rick, Matthew Shane, Cristina Murray-Krezan, Maxim Zaitsev und Oliver Speck (Mai 2012). “Enhancement of temporal resolution and BOLD sensitivity in real-time fMRI using multi-slab echo-volumar imaging.” In: *NeuroImage* 61.1, S. 115–130.
- Posse, Stefan, Daniel Fitzgerald, Kunxiu Gao, Ute Habel, David Rosenberg, Gregory J Moore und Frank Schneider (März 2003). “Real-time fMRI of temporolimbic regions detects amygdala activation during single-trial self-induced sadness.” In: *NeuroImage* 18.3, S. 760–768.
- Rajapakse, Jagath C, Yang Wang, Xuebin Zheng und Juan Zhou (Juni 2008). “Probabilistic Framework for Brain Connectivity From Functional MR Images”. In: *IEEE Transactions on Medical Imaging* 27.6, S. 825–833.
- Reynolds, John H und Leonardo Chelazzi (Jan. 2004). “Attentional modulation of visual processing.” In: *Annual Review of Neuroscience* 27, S. 611–47.
- Robinson, S, C Windischberger, A Rauscher und E Moser (Mai 2004). “Optimized 3 T EPI of the amygdalae.” In: *NeuroImage* 22.1, S. 203–10.
- Rockstroh, B, T Elbert, N Birbaumer, P Wolf, A Dürchting-Röth, M Reker, I Daum, W Lutzenberger und J Dichgans (Jan. 1993). “Cortical self-regulation in patients with epilepsies.” In: *Epilepsy Research* 14.1, S. 63–72.
- Rota, Giuseppina, Giacomo Handjaras, Ranganatha Sitaram, Niels Birbaumer und Grzegorz Dogil (Juni 2011). “Reorganization of functional and effective connectivity during real-time fMRI-BCI modulation of prosody processing.” In: *Brain and Language* 117.3, S. 123–32.
- Rota, Giuseppina, Ranganatha Sitaram, Ralf Veit, Michael Erb, Nikolaus Weiskopf, Grzegorz Dogil und Niels Birbaumer (Mai 2009). “Self-regulation of regional cortical activity using real-time fMRI: the right inferior frontal gyrus and linguistic processing.” In: *Human Brain Mapping* 30.5, S. 1605–1614.
- Rotshtein, Pia, David Soto, Alessandro Grecucci, Joy J Geng und Glyn W Humphreys (Mai 2011). “The role of the pulvinar in resolving competition between memory and visual selection: a functional connectivity study.” In: *Neuropsychologia* 49.6, S. 1544–52.
- Sabatinelli, Dean, Erica E Fortune, Qingyang Li, Aisha Siddiqui, Cynthia Krafft, William T Oliver, Stefanie Beck und Joshua Jeffries (Feb. 2011). “Emotional perception: meta-analyses of face and natural scene processing.” In: *NeuroImage* 54.3, S. 2524–33.
- Salvucci, Dario D. und Joseph H. Goldberg (2000). “Identifying fixations and saccades in eye-tracking protocols”. In: *Proceedings of the symposium on Eye tracking research & applications*. New York, USA: ACM Press, S. 71–78.
- Schneider, Frank und Gereon R. Fink (2007). *Funktionelle MRT in Psychiatrie und Neurologie*. 1. Auflage. Berlin: Springer, Berlin, S. 370.
- Sitaram, R, N Weiskopf, A Caria, R Veit, M Erb und N Birbaumer (2008). “fMRI Brain-Computer Interfaces”. In: *IEEE Signal Processing Magazine* 25.1, S. 95–106.

- Sitaram, Ranganatha (2007). “fMRI Brain-Computer Interface: A Tool for Neuroscientific Research and Treatment”. In: *Computational Intelligence and Neuroscience* 39.1, S. 5–11.
- Sitaram, Ranganatha, Ralf Veit, Birte Stevens, Andrea Caria, Christian Gerloff, Niels Birbaumer und Friedhelm Hummel (Sep. 2011). “Acquired Control of Ventral Premotor Cortex Activity by Feedback Training: An Exploratory Real-Time fMRI and TMS Study.” In: *Neurorehabilitation and Neural Repair*.
- Smyser, C, T J Grabowski, R J Frank, J W Haller und L Bolinger (Feb. 2001). “Real-time multiple linear regression for fMRI supported by time-aware acquisition and processing.” In: *Magnetic Resonance in Medicine* 45.2, S. 289–298.
- Sorger, Bettina, Rainer Goebel, Christine Schiltz und Bruno Rossion (Apr. 2007). “Understanding the functional neuroanatomy of acquired prosopagnosia.” In: *NeuroImage* 35.2, S. 836–852.
- Sorger, Bettina, Joel Reithler, Brigitte Dahmen und Rainer Goebel (Juni 2012). “A Real-Time fMRI-Based Spelling Device Immediately Enabling Robust Motor-Independent Communication.” In: *Current Biology* 22.14, S. 1333–1338.
- Subramanian, L., J. V. Hindle, S. Johnston, M. V. Roberts, M. Husain, R. Goebel und D. Linden (Nov. 2011). “Real-Time Functional Magnetic Resonance Imaging Neurofeedback for Treatment of Parkinson’s Disease”. In: *Journal of Neuroscience* 31.45, S. 16309–16317.
- Sudre, Gustavo, Lauri Parkkonen, Elizabeth Bock, Sylvain Baillet, Wei Wang und Douglas J Weber (Jan. 2011). “rtMEG: a real-time software interface for magnetoencephalography.” In: *Computational Intelligence and Neuroscience* 2011, S. 7.
- Sulzer, J., S. Haller, F. Scharnowski, N. Weiskopf, N. Birbaumer, M.L. Blefari, A.B. Bruehl, L.G. Cohen, R. C. DeCharms, R. Gassert, R. Goebel, U. Herwig, S. LaConte, D. Linden, A. Luft, E. Seifritz und R. Sitaram (März 2013). “Real-time fMRI neurofeedback: Progress and challenges”. In: *NeuroImage* 76, S. 386–399.
- Voyvodic, J T (Aug. 1999). “Real-time fMRI paradigm control, physiology, and behavior combined with near real-time statistical analysis.” In: *NeuroImage* 10.2, S. 91–106.
- Voyvodic, James T, Gary H Glover, Douglas Greve und Syam Gadde (2011). “Automated real-time behavioral and physiological data acquisition and display integrated with stimulus presentation for fMRI”. In: *Frontiers in Neuroinformatics* 5.December, S. 1–9.
- Vuilleumier, Patrik (Dez. 2005). “How brains beware: neural mechanisms of emotional attention.” In: *Trends in Cognitive Sciences* 9.12, S. 585–94.
- Ward, Robert, Andrew J Calder, Maxine Parker und Isabel Arend (Apr. 2007). “Emotion recognition following human pulvinar damage.” In: *Neuropsychologia* 45.8, S. 1973–8.
- Weishaupt, Dominik, Victor D. Köchli und Borut Marincek (2006). *Wie funktioniert MRI?* 5. Aufl. Heidelberg: Springer Medizin Verlag, S. 145.
- Weiskopf, Nikolaus (Aug. 2012). “Real-time fMRI and its application to neurofeedback.” In: *NeuroImage* 62.2, S. 682–92.

- Weiskopf, Nikolaus, Uwe Klose, Niels Birbaumer und Klaus Mathiak (Feb. 2005). "Single-shot compensation of image distortions and BOLD contrast optimization using multi-echo EPI for real-time fMRI." In: *NeuroImage* 24.4, S. 1068–1079.
- Weiskopf, Nikolaus, Ranganatha Sitaram, Oliver Josephs, Ralf Veit, Frank Scharnowski, Rainer Goebel, Niels Birbaumer, Ralf Deichmann und Klaus Mathiak (Juli 2007). "Real-time functional magnetic resonance imaging: methods and applications." In: *Magnetic Resonance Imaging* 25.6, S. 989–1003.
- Weiskopf, Nikolaus, Ralf Veit, Michael Erb, Klaus Mathiak, Wolfgang Grodd, Rainer Goebel und Niels Birbaumer (Juli 2003). "Physiological self-regulation of regional brain activity using real-time functional magnetic resonance imaging (fMRI): methodology and exemplary data." In: *NeuroImage* 19.3, S. 577–586.
- Wilms, Marcus, Leonhard Schilbach, Ulrich Pfeiffer, Gary Bente, Gereon R Fink und Kai Vogele (März 2010). "It's in your eyes—using gaze-contingent stimuli to create truly interactive paradigms for social cognitive and affective neuroscience." In: *Social Cognitive and Affective Neuroscience* 5.1, S. 98–107.
- Yoo, Julie J, Oliver Hinds, Noa Ofen, Todd W Thompson, Susan Whitfield-Gabrieli, Christina Triantafyllou und John D E Gabrieli (Jan. 2012). "When the brain is prepared to learn: enhancing human learning using real-time fMRI." In: *NeuroImage* 59.1, S. 846–52.
- Yoo, Seung-Schik, Ty Fairney, Nan-Kuei Chen, Seh-Eun Choo, Lawrence P Panych, HyunWook Park, Soo-Young Lee und Ferenc A Jolesz (Juli 2004). "Brain-computer interface using fMRI: spatial navigation by thoughts." In: *Neuroreport* 15.10, S. 1591–1595.
- Yoo, Seung-Schik und Ferenc A Jolesz (Aug. 2002). "Functional MRI for neurofeedback: feasibility study on a hand motor task." In: *Neuroreport* 13.11, S. 1377–1381.
- Yoo, Seung-Schik, Heather M O'Leary, Ty Fairney, Nan-Kuei Chen, Lawrence P Panych, HyunWook Park und Ferenc A Jolesz (Aug. 2006). "Increasing cortical activity in auditory areas through neurofeedback functional magnetic resonance imaging." In: *Neuroreport* 17.12, S. 1273–1278.
- Zotev, Vadim, Frank Krueger, Raquel Phillips, Ruben P Alvarez, W Kyle Simmons, Patrick Bellgowan, Wayne C Drevets und Jerzy Bodurka (Jan. 2011). "Self-regulation of amygdala activation using real-time FMRI neurofeedback." In: *PLoS ONE* 6.9. Hrsg. von Katharina Domschke.
- Zotev, Vadim, Raquel Phillips, Han Yuan, Masaya Misaki und Jerzy Bodurka (Mai 2014). "Self-Regulation of Human Brain Activity Using Simultaneous Real-Time fMRI and EEG Neurofeedback." In: *NeuroImage* 85.3, S. 985–995.
- Zschoyge, Oliver (2010). *Entwicklung einer plattformunabhängigen Software zur spezifikationsadaptiven Konfiguration von fMRT-Experimenten inklusive logischer Validierung mit einer fMRT-Wissensrepräsentation*. Techn. Ber. Leipzig: Hochschule für Technik, Wirtschaft und Kunst (HTWK) Leipzig.

Abbildungsverzeichnis

1.1	Zielsetzung der vorliegenden Arbeit	3
2.1	Das MR-Phänomen	6
2.2	Beispiel MR-Aufnahmen unterschiedlicher Kontraste	8
2.3	MR-Sequenz für Echo Planar Imaging (EPI)	9
2.4	Der BOLD-Effekt und die HRF	11
2.5	Stimulation und Design in fMRT-Experimenten	14
2.6	Die Verwendung eines Eyetracker bei fMRT-Experimenten	16
2.7	Erläuterung GLM-Analyse	20
2.8	Ablauf konventioneller fMRT-Experimente	20
2.9	Ablauf Echtzeit-fMRT-Experimente	24
2.10	Varianten Neurofeedbackvisualisierungen und externer BCI-Systeme	31
2.11	Beispielablaufplan eines adaptives Paradigma	38
3.1	UML Paketdiagramm BART Software	60
3.2	UML Klassendiagramm Hauptapplikation BART	61
3.3	Schema Echtzeit-fMRT Aufbau	62
3.4	Schema BART Software Echtzeitauswertung	64
3.5	UML Klassendiagramm Analyse	66
3.6	Schema BART Software adaptive Stimulation	68
3.7	Screenshots des Stimulationsmoduls BART	70
3.8	UML Klassendiagramm Bibliothek Design und Stimulation	71
3.9	UML Klassendiagramm Bibliothek Serielle Schnittstelle	74
3.10	Schema BART Software Gesamtaufbau: Adaptive Paradigmen und Echtzeitauswertung	78
3.11	Screenshots vom Editor für EDL-Dokumente	82
3.12	Vergleich Exportzeiten interner Export von Siemens und Eigenimplementierung	84
4.1	Grundlagen emotionsrelatierte Hirnareale	89
4.2	fMRT-Experiment I: Schema des adaptiven Paradigmas	93
4.3	fMRT-Experiment I: Ergebnisse Verhaltensparameter adaptives Paradigma	96

4.4	fMRT-Experiment I: Ergebnisse GLM-Analyse adaptives Paradigma: Haupteffekt der Emotion	97
4.5	fMRT-Experiment I: Ergebnisse GLM-Analyse adaptives Paradigma: Haupteffekt der Adaption	99
4.6	fMRT-Experiment I: Ergebnisse für die Effektstärken in adaptivem Paradigma .	100
4.7	fMRT-Experiment II: Vergleich der Datenauswertung SPM8 vs. BART für Fingertapping	102
4.8	fMRT-Experiment III: Vergleich der Datenauswertung SPM8 vs. BART für adaptives Paradigma	104
6.1	Schema BART Software Ausblick	115
A.1	UML Klassendiagramm Bibliothek Datenmanagement	117
A.2	UML Klassendiagramm Bibliothek Konfiguration	118
A.3	SAM Grafik zur Wertung der Wertigkeit eines Bildes	123
A.4	SAM Grafik zur Wertung der Erregung eines Bildes	123

Tabellenverzeichnis

2.1	Vergleich von BCI-Modalitäten	23
3.1	Eigenschaften vorhandener Softwaresysteme	43
3.2	Rechenzeiten GLM-Analyse bei parallelisierter Berechnung	67
3.3	Benchmarktest Stimmulationsmodul	77
3.4	Zusammenfassung Unterschiede interner Export und neu entwickelter Echtzeit-Export	84
4.1	Tabelle MNI Koordinaten Haupteffekt Emotion	98
4.2	Tabelle adaptives Paradigma: Unterschiede der Effektstärken für adaptive und konventionelle Stimulation	100
4.3	Tabelle Mutual Information Tapping	103
4.4	Tabelle Mutual Information adaptives Paradigma mit Echtzeitauswertung . . .	104

Auflistungsverzeichnis

3.1	Minimalbeispiel zur Veranschaulichung des Konzeptes von XML	50
3.2	Exemplarisches Beispiel der Medienobjekte aus EDL-Schemadefinition	52
3.3	Exemplarisches Beispiel der Verwendung von Medienobjekten in konkretem EDL-Dokument	53
3.4	EDL-Beispiel für adaptive Paradigmen: Constraint	73
3.5	EDL-Beispiel für adaptive Paradigmen: Medienobjekt mit Constraint	73
3.6	EDL-Beispiel für adaptive Paradigmen: Dynamische Regressoren	78
3.7	EDL-Beispiel für adaptive Paradigmen: Transferfunktionen	79
3.8	EDL-Beispiel für adaptive Paradigmen: Medienobjekt mit Constraint und Trans- ferfunktion	79
A.1	XSD-Spezifikation für adaptive Paradigmen: Constraints	119
A.2	XSD-Spezifikation für adaptive Paradigmen: Dynamische Regressoren	121
A.3	XSD-Spezifikation für adaptive Paradigmen: Transferfunktionen	122

Glossar

Adaptives Paradigma

Der Begriff bezeichnet hier ein Paradigma, dessen Stimuli zur Laufzeit anhand externer Parameter adaptiert werden können.

BART

BART ist das in dieser Arbeit neu entwickelte Softwareframework, in dem die vorgestellten Konzepte und Algorithmen implementiert wurden und mit dem vorgestellten Experimente durchgeführt wurden.

BCI

BCIs dienen dazu, gemessene Hirnsignale an einen Computer zu übertragen um diese dort zu verarbeiten. Außerdem können daraus abgeleitete Informationen an den Probanden in Form eines Neurofeedbacks zurückgegeben werden.

BOLD-Effekt

Aufgrund des BOLD-Effekts kann indirekt neuronale Aktivität in Hirnregionen mit fMRT detektiert werden.

Designmatrix

In der Designmatrix werden die verschiedenen Kategorien (Bedingungen) von Stimuli und deren Präsentationszeit und -dauer beschrieben. Die Designmatrix ist das Modell für eine die GLM-Auswertung.

EDL

EDL ist eine Beschreibungssprache für (Echtzeit)-fMRT-Experimente, die von Maurice Hollmann entwickelt wurde. Diese Sprache enthält neben einer strukturierten Zusammenfassung aller notwendigen Parameter auch eine Wissensrepräsentation von Regeln um die Abhängigkeiten von Parametern zu beschreiben. EDL wurde in dieser Arbeit erweitert, um adaptive Paradigmen abstrakt abbilden zu können.

EEG

Die Elektroenzephalographie ist eine nicht-invasive Bildgebungsmethode für neuronale Aktivität, die Aktionspotentiale mit Hilfe von Elektroden an der Schädeloberfläche ableitet.

EPI

Eine MR-Pulssequenz, die eine sehr schnelle Datenerfassung vom Gehirn ermöglicht, wodurch der BOLD-Effekt im gesamten Hirnvolumen erfasst werden kann.

fMRT

Die funktionelle Magnetresonanztomographie ist eine nicht-invasive Bildgebungsmethode mit der indirekt neuronale Aktivität im gesamten Gehirn lokalisiert werden kann.

Framework

Ein Framework bezeichnet in der Softwaretechnik ein eine Rahmenstruktur für die Entwicklung eines Systems, welche einzelne Bausteine für die Entwickler zur Verfügung stellt. Ein Framework an sich ist noch keine funktionierende Software..

GCD

GCD ist eine Softwaretechnologie, die von der Firma Apple entwickelt wurde, um die Parallelisierung von Software auf mehreren Prozessorkernen durch Abstraktion zu unterstützen.

GLM

Unter dem Begriff Allgemeines (Generalisiertes) Lineares Modell werden ganz allgemein eine Vielzahl statistischer Testverfahren zusammengefasst, zu denen u. a. Mittelwertvergleiche, varianzanalytische Verfahren, Korrelations- und Regressionsrechnungen gehören.

HRF

Die hämodynamische Antwortfunktion beschreibt die Modellannahme für die erwartete Signalveränderung in Hirnarealen, die als Antwort auf eine Reizapplikation entsprechend dem BOLD-Effekt auftritt.

isis

Diese Basisbibliothek für die Verwaltung von (f)MRT-Daten wurde parallel zur vorliegenden mitentwickelt und wird für das interne Datenmanagement in BART verwendet.

ITI

ITI ist der zeitliche Abstand zwischen der Präsentation von Stimuli in einem Paradigma. Häufig wird diese Zeit innerhalb eines Paradigmas variiert, z. B. ein Intervall zwischen 2 und 6 Sekunden, damit der Zeitpunkt des nächsten Stimulus für den Probanden nicht genau vorhersagbar ist.

ITK

ITK ist eine umfangreiche Open-Source Bibliothek, die optimierte Algorithmen für die Registrierung und Segmentierung medizinischer Bilddaten zur Verfügung stellt.

Konventionelles Paradigma

Der Begriff bezeichnet hier ein Paradigma, dessen Stimulation nach einem fest vordefinierten Schema präsentiert wird.

MNI

MNI beschreibt eine Standardvorlage für (f)MRT-Daten. Wenn gemessene Daten auf diese Standardvorlage registriert werden, können die Koordinaten der ermittelten Ergebnisse (Voxel) dann mit anderen Studien verglichen werden, die ebenfalls diese Vorlage verwenden.

Neurofeedback

Möglichkeit zur operanten Konditionieren von Probanden durch gezieltes Training zum Erlernen der Kontrolle über Hirnregionen.

ROI

In dieser Arbeit wird damit die Auswahl einer Hirnregion beschrieben, die in dem Kontext von besonderem Interesse ist.

rtExplorer

Der rtExplorer ist eine Matlab-basiertes Softwaresystem für Neurofeedbackexperimente, das von Maurice Hollmann entwickelt wurde.

SPM8

SPM ist ein häufig verwendetes Softwarepaket, das zahlreiche Methoden für die Vorverarbeitung und Auswertung von fMRT-Daten zur Verfügung stellt und in dieser Arbeit in der Version 8 verwendet wird.

TR

Die TR ist die Wiederholzeit für die Schichtanregung in einer MR-Pulssequenz, die bei funktionellen Messungen meist um die 2 s beträgt.

XML

XML ist eine Metasprache zur Beschreibung von hierarchisch strukturierten Daten.

XSD

Eine XSD ist eine Schemadefinition für XML-Dokumente.

Abkürzungsverzeichnis

ACC	Anteriorer cingulärer Kortex.
BCI	Brain Computer Interface.
BOLD	Blood oxygenation level dependent Effekt.
EDL	Experiment Description Language.
EEG	Elektroenzephalographie.
EPI	Echo Planar Imaging.
FFG	Fusiformer Gyrus.
fMRT	Funktionelle Magnetresonanztomographie.
fNIRS	Funktionelle Nah-Infrarot-Spektroskopie.
GCD	Grand Central Dispatcher.
GLM	Allgemeines (Generalisiertes) Lineares Modell.
HRF	Hemodynamic Response Function.
IFG	Inferiorer frontaler Gyrus.
ITI	Inter-Trial-Intervall.
ITK	Insight Registration and Segmentation Toolkit.
LOC	Lateraler okzipitaler Kortex.
MDNT	Medialer dorsaler Nucleus des Thalamus.
MEG	Magnetenzephalographie.
MFG	Mittlerer frontaler Gyrus.
MNI	Montreal Neurological Institute.
mPFC	Medialer präfrontaler Kortex.
OFC	Orbitofrontaler Kortex.

ROI	Region of interest.
RVM	Relevance Vector Machine.
sACC	Subgenualer anteriorer cingulärer Kortex.
SPM	Statistical Parametric Mapping.
SVM	Support Vector Machine.
TR	Time of Repetition.
VLDFC	Ventrolater präfrontaler Kortex.
XML	Extensible Markup Language.
XSD	XML Scheme Definition.

Bibliographische Angaben

Hellrung, Lydia:

Softwareframework zur universellen Methodenentwicklung für ein fMRT-BCI: Adaptive Paradigmen und Echtzeitdatenanalyse

Universität Leipzig, Fakultät für Mathematik und Informatik, Dissertation

143 Seiten, 37 Abbildungen, 9 Tabellen, 11 Auflistungen mit Quellcodebeispielen

Selbständigkeitserklärung

Selbständigkeitserklärung

Hiermit erkläre ich, die vorliegende Dissertation selbständig und ohne unzulässige fremde Hilfe angefertigt zu haben. Ich habe keine anderen als die angeführten Quellen und Hilfsmittel benutzt und sämtliche Textstellen, die wörtlich oder sinngemäß aus veröffentlichten oder unveröffentlichten Schriften entnommen wurden, und alle Angaben, die auf mündlichen Auskünften beruhen, als solche kenntlich gemacht. Ebenfalls sind alle von anderen Personen bereitgestellten Materialien oder erbrachten Dienstleistungen als solche gekennzeichnet.

Leipzig, 28. April 2015

.....
(Ort, Datum)

.....
(Unterschrift) Lydia Hellrung