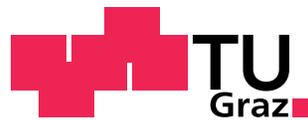


Günter Moitzi

Spielbasiertes funktionales Gehirnmonitoring

Masterarbeit



Institut für Semantische Datenanalyse / Knowledge Discovery Laboratory for Brain-Computer
Interfaces

Technische Universität Graz
Krenngasse 37, A - 8010 Graz
Vorstand: Assoc. Prof. Dipl.-Ing. Dr.techn. Gernot Müller-Putz

Betreuer: Ass.Prof. Dipl.-Ing. Dr.techn. Reinhold Scherer

Begutachter: Ass.Prof. Dipl.-Ing. Dr.techn. Reinhold Scherer

Graz September 2012

resultierende Arbeiten:

R.Scherer, G.Moitzi, I.Daly, G.Müller-Putz (2012) Games for functional brain mapping, *Transactions on Computational Intelligence and AI in Games*

Beschluss der Curricula Kommission für Bachelor-, Master- und Diplomstudien vom
10.11.2008. Genehmigung des Senats am 1.12.2008

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am

.....

(Unterschrift)

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

.....

(date)

.....

(signature)

Zusammenfassung:

Jeder der schon einmal an einer EEG (Elektroenzephalografie) Messung teilgenommen hat, hat zumindest eine Aufgabe durchführen müssen um damit Aktivierung in bestimmten Hirnregionen auszulösen. Spätestens während der Ausführung der Aufgabe fällt auf, dass diese sehr schnell langweilig wird da es sich meistens um eine sehr monotone Aktivität handelt. In der folgenden Arbeit wird gezeigt das Paradigmen, die für die Durchführung von EEG Messungen verwendet werden, mit dem selben Ergebnis, auch in anderer Form eingesetzt werden können. Dafür wird ein Spiel programmiert, dass den Probanden aufgrund des Ablaufs dazu animiert Bewegungen auszuführen um die benötigten Aktivierungen im Gehirn zu erhalten. Während des Spielablaufs muss der Proband versuchen entweder mit einer seiner Hände oder den Knie Objekte zu treffen die sich in Richtung der Spielfigur bewegen. Parallel zum Spiel wird am Benutzer eine EEG Messung durchgeführt. Die aus dieser Messung gewonnenen Daten werden mit den Bewegungsdaten aus dem Spiel synchronisiert. Dadurch ist es möglich die Zeitpunkte mit Bewegung (Events) des Probanden aus den EEG Daten zu Filtern. Abschließend werden mit den gefilterten Events ERD/ERS Maps (Event Related Synchronisation/Event Related Desynchronisation) berechnet. Mit Hilfe dieser Maps wird gezeigt, dass die erhaltenen Daten korrekt sind.

Schlüsselwörter: Microsoft Kinect, Benutzer tracking, Grundlagen Spielprogrammierung, Daten Synchronisation, ERD/ERS Maps

Abstract:

Everybody who has participated to a EEG (electroencephalography) measurement has done at least one task to activate particular brain areas. By executing the task its erverybody will notice that it is very boring to fulfill it because most times the activity is very monotonously. The following work shows that paradigms, which are applied for EEG measurement, can be used in a different order with the same achivement. To accomplish this, a game is implemented, to animate a subject, as a result of the game design, to make movements which trigger an activation in the brain. While playing the game the user tries to hit objects, which are moveing in game charakter direction, either with one of its hands or with the knees. Parallel to the playing of the game an EEG measurement takes place. The data from the measurement is synchronized with the movement data from the game. This allows to filter the periods of movement (events) from the proband in the EEG data. At the end, the filteres events, are used to calculate ERD/ERS (Event Related Synchronisation/Event Related Desynchronisation) maps. This maps help to proof the correctness of the obtained data.

Keywords: Microsoft Kinect, user traking, game development basics, data synchronisation, ERD/ERS maps

Danksagung

An dieser Stelle möchte ich mich bei allen Mitarbeitern des Instituts für Semantische Datenanalyse/Knowledge Discovery bedanken. Vor allen bei meinem Betreuer, Ass.Prof. Dipl.-Ing. Dr.techn. Reinhold Scherer, der immer viele gute Ratschläge für die Lösung meiner Probleme hatte. Auch an Dipl.-Ing. Dr.techn. Denis Kalkofen vom Institut für Maschinelles Sehen und Darstellen, ein Danke. Er hat mich bei der Durchführung von anderen Projekten unterstützt und dabei das Wissen vermittelt, das für den erfolgreichen Abschluss dieser Arbeit sehr wertvoll war.

Ein weiteres Danke geht an meine Eltern die mich während der Dauer meines Studiums in allen Lebenslagen auf verschiedene Arten unterstützt haben.

Zum Schluss noch ein Danke an meine Studienkollegen Lackner Günter, Martin Großegger, Schwarz Andreas und Thomas Wiesner die mich über lange Wege durch alle Höhen und Tiefen des Studiums begleitet haben.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Ziele der Arbeit	3
2	Methoden	4
2.1	Grundlagen	4
2.1.1	Kinect	4
2.1.2	NITE und OpenNI	5
2.1.3	Basistechniken für die Entwicklung eines Spiels	7
2.1.4	TOBI Signal Server	10
2.1.5	Signal Verstärker	11
2.1.6	Neuroanatomische Grundlagen	11
2.1.7	Elektroenzephalogramm und Elektrocorticogramm	13
2.1.8	Epilepsie	15
2.1.9	Event Related Synchronisation und Desynchronisation Maps	18
2.2	Das Spiel	19
2.2.1	Framework	19
2.2.2	Spiel Aufbau und Ablauf	20
2.2.3	Komponenten	23
2.2.4	Generierte Daten	31
2.3	Messung	33
2.3.1	Messaufbau	34
2.3.2	Elektrodenmontage	35
2.3.3	Simulink Model	35
2.3.4	Paradigma	36
2.3.5	Messdaten und Event Extraktion	37
3	Ergebnisse	39

4 Diskussion	43
4.1 Kinect und openNI	43
4.2 Spiel	44
4.3 Messung und Ergebnisse	46
4.4 Schlussfolgerung	47
Literatur	48
5 Anhang	51

1 Einleitung

Der erste Abschnitt dieser Arbeit wird verwendet um die Motivation, die zu dieser Arbeit geführt hat, wiederzugeben. Weiters wird im Punkt 1.2 Ziele der Arbeit eine Auflistung angeführt, in welcher die zu erfüllenden Aufgaben kurz beschrieben sind.

1.1 Motivation

Die, bis zum aktuellen Stand der Technik, durchgeführte Erforschung der Funktion des menschlichen Gehirns und der Aufgabe der einzelnen Hirnareale ermöglichen es viele Krankheitsbilder besser zu erkennen, zu lokalisieren und zu behandeln [15]. Dafür wurden Methoden mit entsprechenden Geräten, wie zum Beispiel funktionelle Magnetresonanz (fMRI) [6], nahinfrarot Spektroskopie (NIRS) [6], Elektrokokortikografie (ECoG) [6] oder Elektroenzelografie (EEG) [12] entwickelt. Damit ist es möglich Gehirnaktivität zu messen und aufzuzeichnen. Mit Hilfe dieser Daten können zwischen dem Verhalten und den dadurch hervorgerufenen Veränderungen in den Hirnströmen Zusammenhänge gefunden werden [7].

Durch dieses Wissen konnten neue Arten für die Behandlung von Erkrankungen des Gehirns entwickelt und getestet werden. Weiters ist es möglich den Effekt der Behandlung im weiteren Verlauf zu erforschen. Das Ergebnis dieser Forschung sind Methoden, die für die Vorbereitung von Operationen am Gehirn verwendet werden können, da es möglich ist betroffene Areale sehr genau einzugrenzen. Zusätzlich ist es durch das Wissen, welchen Zweck ein Areal erfüllt, möglich, vor dem Eingriff sehr genau festzustellen mit welchen Einschränkungen beim Patient nach dem Eingriff gerechnet werden muss bzw. ob es sinnvoll ist einen solchen durchzuführen [12].

Ein weiteres Einsatzgebiet der für die oben angeführten Methoden ist die Erforschung und Bestimmung von Epilepsietypen. Eine genauere Beschreibung wie diese durchgeführt wird, wird im Kapitel 2.7 Epilepsie geliefert.

Da sich immer mehr Unfälle mit einer Schädigung des Gehirns ereignen oder durch z.B. einen Schlaganfall starke Beeinträchtigungen im Gehirn auftreten können, werden auch in der Rehabilitation viele der genannten Techniken eingesetzt [19]. Es ist damit möglich, Verbesserungen in den betroffenen Regionen durch die Behandlung festzustellen.

Alle eingesetzten Methoden haben Vor- und Nachteile bezüglich der anfallenden Kosten, zeitlichen Aufwand für die Durchführung der Messung und die Ergebnisse unterscheiden sich in zeitlicher und räumlicher Auflösung. In den meisten Fällen wird bei einer Messung der Proband instruiert, welche Aufgabe er im Test zu erfüllen hat. Es kann zum Beispiel eine einfache Bewegung sein, die ausgeführt werden muss wenn ein Signal auf einem Monitor erscheint. Die Daten werden anschließend durch Verwendung von verschiedenen mathematischen Algorithmen ausgewertet und dadurch Veränderungen in den Potentialen des Gehirns festgestellt.

Das Problem ist, dass viele der zur Zeit eingesetzten Tests sehr langweilig für die Probanden sind. Vor allem bei Kindern, da diese meistens eine kürzere Aufmerksamkeitsspanne haben, sind diese Paradigmen nur bedingt einsetzbar. Normalerweise wird, wie oben beschrieben, nur eine Bewegung oder ein Sprachmuster über einen längeren Zeitraum ausgeführt. Das führt nach kurzer Zeit bei vielen Probanden zu einer Abnahme der Aufmerksamkeit, was oft dazu führt, dass die Aufgabe nicht mehr korrekt ausgeführt wird. Ergebnisse aus diesen Daten sind dann nicht mehr brauchbar, da sie nicht richtig interpretiert werden können, weil nicht sicher ist, ob die entstandenen Muster fehlerhaft sind. Es folgt daraus die Überlegung, ob es nicht möglich ist, die gängigen Paradigmen so abzuändern oder aufzubereiten, dass für den Probanden während der Testphase die Motivation zur richtigen Durchführung aufrecht bleibt.

Fast in jedem Haushalt findet man entweder eine Spiel-Konsole oder einen Computer. Man kann daher davon ausgehen, dass die meisten Personen in irgendeiner Form Erfahrung mit Computerspielen gemacht haben. Viele dieser Spiele erzeugen zumindest die Motivation sich mehrere Stunden damit zu beschäftigen [27]. Die gestellten Aufgaben können dabei durchaus einfach sein und wenig Abwechslung bieten. Die richtige optische Umsetzung und die Herstellung einer Wettkampfsituation genügen um Personen dazu zu bringen auf den Moment zu warten in dem sie einen Knopf drücken müssen. Für die Rehabilitation wurden, vor allem seit dem Erscheinen der Kinect (siehe Abschnitt 2.1.1), mehrere Anwendungen erstellt [37]. Die in den letzten Jahren entwickelten Techniken, um Bewegungen vom Benutzer direkt in das Spiel einfließen zu lassen, sollten es ermöglichen, Paradigmen die irgendwelche Arten von Bewegungen erfordern als Computerspiel umzusetzen.

1.2 Ziele der Arbeit

Die Herausforderung ist, ein Spiel so zu gestalten, dass ein oder mehrere Paradigmen eingesetzt werden und diese nicht vermischt werden. Es sollte also immer nur eine klar definierte Handlung erforderlich sein um die Aufgaben zu bewältigen. Zusätzlich sollte das Framework des Spiels so gestaltet sein, dass es für nachfolgende Arbeiten einfach erweitert werden kann. Weiters sollen die erzeugten Bewegungsdaten mit den EEG Daten zusammengeführt werden, um nach der Testphase eine Auswertung zu ermöglichen. Zusätzlich wird anhand von Messungen an Probanden gezeigt werden, dass die erhaltenen Daten brauchbare Ergebnisse liefern. Dafür muss eine Möglichkeit gefunden werden, mit der Information über die Bewegung, die benötigten Teilbereiche aus dem EEG Signal zu filtern. Folgend werden die zu erfüllenden Anforderungen punktuell angeführt:

- Kinect Daten aufbreiten und in eine Simulation integrieren
- Erstellen eines erweiterbaren Frameworks. Es soll die Möglichkeit geboten sein, in weiterführenden Arbeiten neue Paradigmen einzufügen.
- Erstellen einer Testumgebung (Spiel)
- Erstellen einer Möglichkeit, die durch das Spiel anfallenden Daten, parallel mit dem EEG Signal zu speichern
- Durchführung von Messungen
- Bewegung des Probanden mit dem EEG Signal in Bezug zu bringen
- Zeigen, dass die erhaltenen Daten korrekt sind

2 Methoden

In diesem Kapitel werden zu Beginn die verwendeten Techniken (2.1 Grundlagen) beschrieben. Danach wird erklärt wie das Spiel (2.2 Das Spiel) aufgebaut ist und abschließend wird noch der Messvorgang (2.3 Messung) beschrieben.

2.1 Grundlagen

In diesem Abschnitt wird die verwendete Hardware (2.1 Kinect) und die dazugehörige Software (2.2 NITE und OpenNI) beschrieben. Weiters wird ein kurzer Überblick zum Thema Elektroenzephalografie (EEG) und Brain-Computer-Interface (BCI) gegeben. Ausserdem werden wird noch kurz beschrieben was Epilepsie ist und wie mit EEG eine Klassifizierung vorgenommen werden kann.

2.1.1 Kinect

Die Kinect (Abb. 1) bietet die Möglichkeit die Bewegungen eines Anwenders zu erfassen. Zu Beginn als Projekt Natal bezeichnet wurde die Kinect in Zusammenarbeit mit PrimeSense entwickelt. Zum ersten Mal wurde sie auf der Electronic Entertainment Expo (E3) vorgestellt. Im Dezember 2010 veröffentlichte PrimeSense einen Open Source Treiber für den PC und zusätzlich noch eine motion sensing Middleware (NITE).

Auf der Kinect befindet sich eine RGB-Kamera, ein Infrarot-Sensor und zwei Mikrofone. Die Kamera erlaubt es RGB-Information, mit einer maximalen Auflösung von 640x480 und 30 Herz, über die Szene zu erhalten. Der Infrarotsensor wird eingesetzt um die Tiefe der Szene für jedes Pixel zu bestimmen. Mit Hilfe dieser Daten kann ein Benutzer in der Szene erkannt werden [36]. Durch die Verwendung der Middelware NITE ist es möglich durch eine Initalpose (Abb.13) die Berechnung für die Benutzererkennung durchzuführen und ausserdem kann für die sich ergebende Silouette ein Skelettfittig aktiviert werden. Sobald das Fitting abgeschlossen ist, wird in jedem Frame für jedes sichtbare Gelenk die 3D Position bestimmt. Mit diesen Daten wird die Position des Avatars während der Messung jeden Frame in Echtzeit aktualisiert.



Abbildung 1: Die Abbildung zeigt den verwendeten Sensor (Microsoft Kinect) für die Messung der Szenentiefe und der Aufnahme der RGB-Bilder. Links sieht man der Infrarot Strahler für die Tiefenmessung und in der Mitte befindet sich der Empfänger. Rechts ist die RGB-Kamera montiert. (Quelle:<http://caa.tuwien.ac.at/cv1/teaching/praktika/kinect/index.html>)

2.1.2 NITE und OpenNI

NITE ist die Abkürzung von Natural Interaction Middleware. Mit Hilfe von verschiedenen Algorithmen wird eine Benutzererkennung und Bewegungstracking ermöglicht. NITE ist in das OpenNI SDK integriert. Durch die zur Verfügung gestellten Komponenten ist es möglich, die gewünschten Berechnungen durchzuführen (Benutzererkennung, Hand-/Benutzertracking,...) [20].

OpenNI ist ein plattformübergreifendes Framework (SDK), welches eine Programmierschnittstelle (API) definiert. OpenNI ermöglicht es eine API zu erstellen, die es erlaubt mit Audio- und Video-Schnittstellen zu kommunizieren und zusätzlich eine Middleware einzusetzen, welche die Daten von den Sensoren weiterverarbeitet [23].

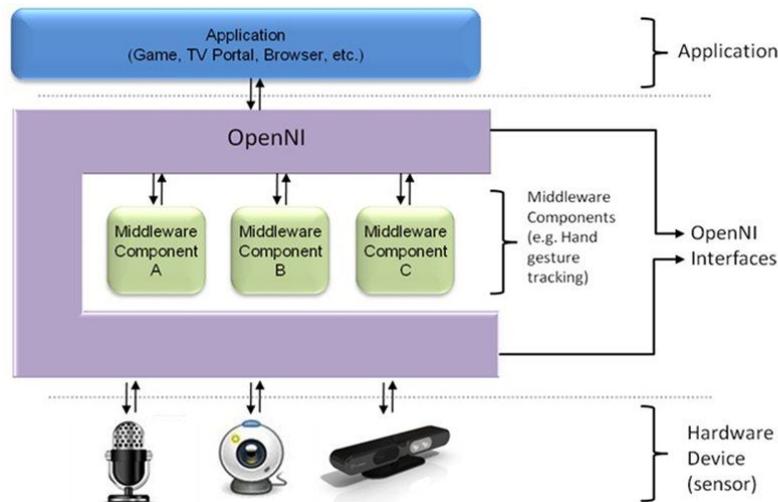


Abbildung 2: Die Abbildung zeigt den grundsätzlichen Zusammenhang zwischen den einzelnen Komponenten. Als unterste Schicht werden Hardware Komponenten verwendet, um den Anwender eine Kommunikation mit dem Programm zu erlauben. Die erzeugten Daten werden über die OpenNI Module gesammelt und an die Middleware(NITE) weitergegeben. Auf das Ergebnis der Berechnungen der Middleware kann danach über weitere Module in der Applikation zugegriffen werden. Es ist aber auch möglich direkt die Daten der einzelnen Hardware Komponenten zu verwenden um z.B. die RGB-Bilder einer Kamera anzuzeigen. (Quelle:openni.org)

Mit dem OpenNI SDK können verschiedene Komponenten sogenannte Module registriert werden. Diese Module werden dann zur Erstellung und Weiterverarbeitung der Daten verwendet. Das OpenNi Framework stellt Module für RGB-/Infrarot-Kameras, Microphone und 3D Sensoren zur Verfügung. Zusätzlich gibt es Komponenten für Skelettberechnung, Handpunktberechnung, Gestenerkennung und Szenenanalyse (Abb. 2). Weitere Informationen über Funktion von OpenNI gibt es in der Dokumentation [20].

Da sich die Kinect normalerweise vor dem Benutzer befindet, das Spiel aber so aufgebaut ist, dass sich der Avatar in die Szene sieht, ergibt sich ein Konflikt in den Koordinatensystemen. Daher müssen die Koordiantensysteme angepasst werden, um eine richtige Darstellung der Bewegung zu erhalten (Kapitel 3.2.2 Animation des Avatars).

2.1.3 Basistechniken für die Entwicklung eines Spiels

Bei der Entwicklung von Spielen müssen verschiedene Komponenten beachtet werden. Zu Beginn ist es wichtig sich im Klaren zu sein, welchen Inhalt das Spiel haben soll. Als nächsten Schritt kann man sich dann einen Überblick verschaffen, welche Techniken eingesetzt werden sollen, um zu dem gewünschten Ergebnis zu kommen. Das Ziel der Arbeit ist es, ein Programm zu schreiben, das den Probanden erlaubt einen Avatar mit seinen Bewegungen zu steuern und zusätzlich soll es möglich sein, mit Objekten zu interagieren. Um diese beiden Punkte zu erfüllen, ist es nötig eine Animation des Avatars und der Objekte zu erzeugen und eine Kollisionserkennung in das Spiel zu integrieren. Um das Aussehen der erzeugten Bilder zu verbessern wird ein Beleuchtungsmodell verwendet und Schatten berechnet. Im folgenden Abschnitt werden Methoden beschrieben wie die oben angeführten Punkte umgesetzt werden können.

Animation

Grundsätzlich versteht man unter einer Animation, dass Erzeugen des Eindrucks von Bewegung durch eine Abfolge von Bildern [16]. Es gibt verschiedene Techniken, die abhängig von dem gewünschten Effekt eingesetzt werden. Die einfachste Methode ist Objekte die am Bildschirm sichtbar sind, oder in weiterer Folge sichtbar werden, für jeden erzeugten Frame um einen bestimmten Wert zu verschieben. Diese Werte können vordefiniert sein oder sich durch Benutzereingaben (z.B. Kinect) ergeben.

Eine komplexere Form ist Image Morphing. Dabei werden zwischen dem Anfangsbild und dem Endbild, anhand von vorgegebenen Parametern, Frames berechnet die einen natürlichen Übergang erzeugen sollen. [35]

Weiters ist es auch möglich durch Verwendung von physikalischen Berechnungen zum Beispiel Feuer oder flüssigen Materialien am Bildschirm darzustellen. Um das Verhalten von Objekten realer erscheinen zu lassen können Parameter wie Elastizität und Verformbarkeit einbezogen werden. Damit kann unter anderem auch die Reaktion und Deformation von Körpern bei einer Kollision dargestellt werden. [30, 9]

Die aufwendigste Form einer Animation ist die Simulation von autonomen Kreaturen die ein spezifisches Verhalten (z.B. Vogelschwarm) haben und mit der Umgebung interagieren

(z.B. andere Kreaturen, statische Objekte). [25]

Mit einer Kombination dieser Techniken ist es möglich virtuelle Welten mit sehr realen Verhältnissen zu erzeugen.

Kollisionserkennung

Sobald in einem Computerprogramm mehrere Objekte miteinander interagieren sollen ist es unerlässlich eine Kollisionserkennung zu implementieren. Nur dadurch ist es dem Computer möglich zu berechnen ob sich zwei oder mehrere Objekte soweit angenähert haben, dass es sich auf das Verhalten oder das Aussehen in irgendeiner Form auswirkt.

Eine einfache Form der Kollisionserkennung ist es, für die Berechnung alle Objekte als Kugel zu betrachten [13]. Mit dieser Variante ergeben sich bei komplexeren Objekten (z.B. animierte Kreaturen) große Genauigkeitsfehler. Um diesen Fehler zu verringern werden Objekte in kleiner Stücke aufgeteilt. Jedes dieser Stücke wird durch eine Oriented Bounding Box (OBB) repräsentiert, welche für die weiteren Berechnungen verwendet wird [29]. Als OBB bezeichnet man den Quader, der das zu repräsentierende Objekt komplett einschließt. Durch die Aufteilung in kleinere Objekte wird die Komplexität der Berechnungen erhöht, aber es ist dadurch auch eine genauere Detektion des Kollisionspunktes möglich. Eine weitere Verfeinerung ergibt sich durch die Verwendung von Axis Aligned Bounding Boxes (AABB). AABB's haben die selbe Funktion wie OBB's, nur werden sie abhängig von der Orientierung des einzuhüllenden Objekts so berechnet, dass sie ein minimales Volumen haben. [32]

Beleuchtung

In der Computergrafik wird durch verschiedene Beleuchtungseffekte der Eindruck von Tiefe erzeugt. Dadurch wird einem Bild mit zwei Dimensionen für den Betrachter eine dritte Dimension hinzugefügt. Mit Hilfe von Beleuchtungsmodellen lassen sich verschiedene Materialien realistisch darstellen und Reflexionen erzeugen. Es werden hier drei bekannte Beleuchtungsmodelle aufgelistet.

Die einfachste Form ein Objekt zu schattieren ist das Flat Shading (Abb.3 zweite von links). Dabei werden abhängig von der Flächennormalen allen Pixel eines Polygons gleich eingefärbt Farbe.

Eine Verbesserung ergibt sich durch das Gouraud Shading [8] bei dem abhängig von der Beleuchtungsintensität der Polygoneckpunkte die Intensität für jedes Pixel im Polygon interpoliert wird (Abb.3 zweite von rechts). Probleme ergeben durch die Interpolation, da dadurch keine guten Glanzpunkte (specular Highlights) realisiert werden können.

Dieses Problem wird durch das Phong Shading [22] behoben (Abb.3 rechts). Es wird für jedes Pixel eine eigene Flächennormale berechnet, wodurch sich eine Verbesserung bei den Glanzpunkten ergibt.

In aktuellen Beleuchtungsmodellen wird oft eine Kombination mit leichten Abwandlungen und Verbesserungen der oben vorgestellten Techniken verwendet.

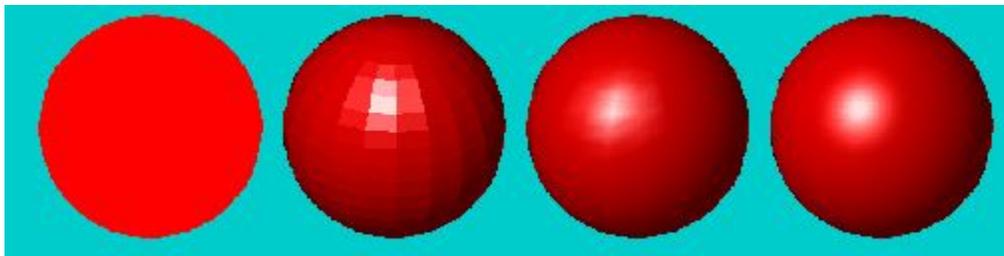


Abbildung 3: Die Abbildung zeigt den Effekt von verschiedenen Beleuchtungsmodellen. Von links nach rechts: ohne Beleuchtung, Flat Shading, Gouraud Shading, Phong Shading. (Quelle: www.mathworks.de)

Schatten

Schatten werden in der Computer Grafik verwendet um den Realismus der erzeugten Bilder zu verstärken. Zusätzlich wird durch das hinzufügen von Schatten die räumliche Position von Objekten besser sichtbar gemacht (Abb. 4). Es gibt verschiedene Techniken Schatten zu erzeugen. Es werden an dieser Stelle nur die für die Arbeit verwendeten und auf dieser Methode aufbauenden Techniken beschrieben.

Beim Shadowmapping wird zuerst von der Position der Lichtquelle in Richtung der Lichtstrahlen gerendert und die sich ergebenden Tiefenwerte der Szene werden gespeichert. Im zweiten Renderdurchgang wird von der Betrachterposition in Blickrichtung gerendert. Dabei wird mit den zuvor erzeugten Tiefenwerten verglichen und festgestellt ob ein Pixel beleuchtet wird. Der Rechenaufwand steigert sich mit der Verwendung von mehreren Lichtquellen um einen Renderdurchgang pro Quelle. Williams [34]

Durch weit entfernte Lichtquellen ergeben sich sehr pixelige Ränder bei den Schatten. Dieser Effekt kann durch die Verwendung von Softshadows verringert werden. Auch hier gibt es wieder verschiedene Methoden. Es können z.B. Variance Shadows verwendet werden. Dabei werden die Pixel an den Rändern der Schatten mit einer Varianzmatrix (Gausmatrix) multipliziert und dadurch geblurt [4]. Eine andere Variante sind Percentage Closer Soft Shadows. Dabei werden die Farbwerte des aktuellen Pixel und Pixel in der näheren Umgebung addiert und ein Mittelwert gebildet. Dadurch ergibt sich ebenfalls eine Penumbra (Halbschatten) [5].

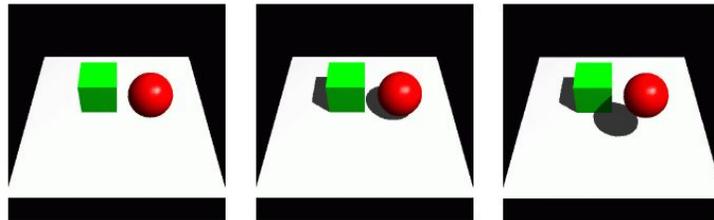


Abbildung 4: Die Abbildung zeigt den visuellen Unterschied, der durch den Einsatz von Schatten entsteht. In der linken Szene ohne Schatten kann keine Aussage über die Lage der Objekte im Raum gemacht werden. Das mittlere und rechte Bild zeigen, dass beide Objekte sehr verschiedene Positionen haben können. Nur durch die Verwendung von Schatten werden diese für den Betrachter erkennbar.

2.1.4 TOBI Signal Server

Der SignalServer ist ein Programm, das Daten von verschiedenen Hardware-Komponenten sammelt und über eine Netzwerkschnittstelle unter Einbindung der TiA Library weiterleitet [2].

Das Programm wurde entwickelt um eine plattformunabhängige Möglichkeit zur Sammlung von Daten zu ermöglichen und diese über eine standardisierte Methode weiterzuleiten.

Damit ist es möglich Daten unabhängig von der verwendeten Hardware und unabhängig vom Hersteller und Betriebssystem immer in derselben Darstellung zu erhalten [2].

TiA (TOBI interface A) ist ein Projekt für ein standardisiertes Interface zur Übertragung von Biosignalen.

2.1.5 Signal Verstärker

Da die zu messenden Potentiale auf der Kopfhaut eine sehr geringe Stärke (5 bis 100 Mikro Volt) haben, wird für die Signalaufnahme ein Biosignalverstärker benötigt. Für die Arbeit wird der g.USBamp (<http://www.gtec.at/Products/Hardware-and-Accessories/g.USBamp-Specs-Features>) eingesetzt. Dieser Verstärkertyp besitzt pro Einheit 16 Eingänge. Um eine Messung mit einer höheren Anzahl von Elektroden durchführen zu können, ist es möglich, mehrere dieser Verstärkereinheiten miteinander zu koppeln. Es existieren bereits vorgefertigte Module für die Verwendung in Simulink [28], wodurch der Einsatz vereinfacht wird. Genauere Informationen können unter [31] gefunden werden.

2.1.6 Neuroanatomische Grundlagen

Im folgenden Abschnitt werden zwei Hirnareale näher erklärt (Abb.5). Das wichtigste für diese Arbeit ist der sensomotorische Cortex. In diesem Areal findet man Aktivierung, wenn eine Person Bewegungen durchführt. Alle Muskeln im Körper werden hier in einem bestimmten Bereich repräsentiert. Es wird angenommen, dass sich bei den meisten Personen die linke Hand im rechten Bereich, die rechte Hand im linken Bereich, und die Beinbewegung im Zentrum des sensomotorischen Cortex befinden. Diese Regionen befinden sich nahe an der Oberfläche des Gehirns und daher können die Potentialunterschiede mit dem EEG gut gemessen werden. Das Areal für die Hände nimmt einen sehr großen Bereich auf beiden Seiten in Anspruch. Die Beinregion ist im Verhältnis sehr klein, wodurch die Aufnahme von Daten entsprechend erschwert werden. Vor der eigentlichen Ausführung einer Bewegung kommt es bereits zu einer Aktivierung in der entsprechenden Gehirnregion. Daher können bereits vor der motorischen Ausführung Potentialunterschiede festgestellt werden [15].

Ein weiteres Areal, welches für die Messdatengenerierung verwendet wird, ist der visuelle Cortex. In diesem Bereich werden alle visuelle Reize verarbeitet. Dabei verlaufen die Fasern der linken Netzhauthälfte beider Augen in den linken Teil und die Fasern der rechten Netzhauthälfte in den rechten Teil des visuellen Cortex [15]. Da zu Beginn der Messungen noch keine Erfahrungswerte für das verwendete Paradigma zur Verfügung standen, wird auch dieser Bereich gemessen. Bei der Auswertung der Daten kann dann festgestellt werden ob und welchen Einfluss der Spielablauf auf diese Region des Gehirns hat.

2.1.7 Elektroenzephalogramm und Electrocorticogramm

Das Elektroenzephalogramm (EEG) ist eine Methode zur Messung der elektrischen Hirnaktivität. Durch diese nicht invasive Methode können Potentiale an der Kopfoberfläche gemessen werden. Das EEG hat eine hohe zeitliche Auflösung (Millisekunden [ms]). Die örtliche Auflösung (Abstand der Elektroden) bei einer üblichen EEG Messung beträgt wenige Zentimeter [12]. Sie erreicht nicht die Qualität von fMRI Messungen (wenige Millimeter) ist aber mit bedeutend weniger Geräteaufwand verbunden.

Wird die Messung direkt an der Hirnoberfläche vorgenommen (invasive Methode) spricht man vom Elektrokortikogramm (ECoG). Mit dieser Messmethode werden größere Amplituden gemessen und eine verbesserte Frequenzwiedergabe erreicht. Zusätzlich kann man durch operativ eingeführte Elektroden Potentialschwankungen tieferer Hirnregionen ableiten. Dadurch ist eine bessere örtliche Auflösung als beim EEG gegeben [12].

Entstehung des Potentials

Für die Erzeugung der postsynaptischen Potentiale sind hauptsächlich die senkrecht zur Oberfläche angeordneten Pyramidenzellen verantwortlich. Diese Feldpotentiale können mit dem EEG/ECoG dargestellt werden. Man unterscheidet, in Abhängigkeit von Depolarisation und Hyperpolarisation einer Neuromembran, zwischen exzitatorischen (erregend) und inhibitorischen (hemmend) postsynaptischen Potentialen (EPSP bzw. IPSP). IPSP erhöhen das Ruhemembranpotential, wodurch das Auslösen des Aktionspotentials erschwert wird. Durch EPSP wird das Membranpotential einer Nervenzelle gesenkt und als Folge davon wird das Auslösen des Aktionspotentials erleichtert.

Für die Darstellung im EEG bedeutet das, dass IPSP die Zahl der positiven Ionen im Extrazellulärraum verringern und EPSP diese erhöhen. Da diese Ladungsänderung im Extrazellulärraum lokal, unter der jeweiligen Synapse, auftritt erscheint bei IPSP der Bereich elektrisch positiver und bei EPSP elektrisch negativer. Es entsteht ein elektrischer Dipol, welcher aufgrund der Beweglichkeit der Ladungen einen gegenläufigen extrazellulären und intrazellulären Ionenfluß zur Folge hat.

Nervenzellen bildet bis zu 10000 Synapsen mit anderen Nervenzellen. Bei gleichzeitiger Aktivierung erzeugt die Summe aller Dipole der einzelnen Synapsen einen Summendipol. Für die Erzeugung der mit dem EEG/EcoG messbaren Potentialen an der Oberfläche des Kopfes bzw. des Cortex ist die Aktivierung vieler parallel angeordneter Nervenzellen notwendig. Obwohl IPSP und EPSP entgegengesetzte elektrischen Dipole erzeugen, können sie in einer Oberflächenableitung zu gleichartigen Potentialschwankungen führen. Abb.7 zeigt eine schematische Darstellung der Polarität der elektrischen Dipole einer kortikalen Pyramidenzelle in Abhängigkeit von ihrer synaptischen Erregung und die daraus resultierende Spannungsschwankung. Abbildung7(a) Zeigt das EPSP im Bereich der apikalen Dendriten einen relativ starken negativen Ausschlag im EEG verursachen und am Soma des Neurons zu einem schwächeren positiven Ausschlag führen (b), (c) und (d) zeigen die genau gegenteiligen Auswirkungen von IPSP in den selben Regionen. Fall (c) tritt nicht in den oberen Cortexschichten auf. Dadurch ergibt sich das EPSP an apikalen Dendriten und IPSP am Soma einer Nervenzelle ein negatives Potential erzeugen. IPSP an den Dendriten und EPSP führen zu einem positiven Potential bzw. Ausschlag im EEG [12, 19].

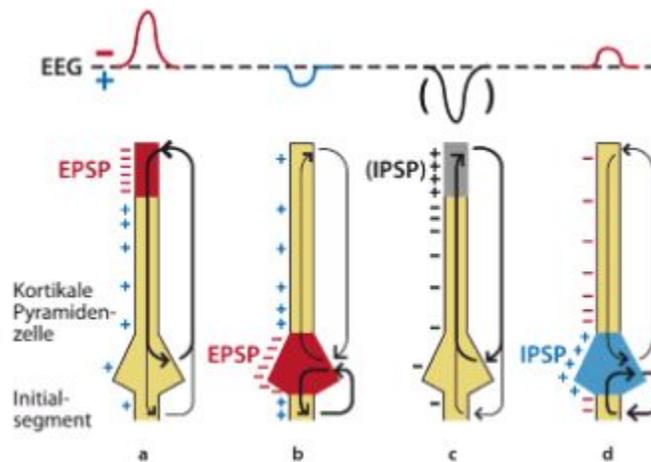


Abbildung 7: Die Abbildung zeigt die Auswirkung von postsynaptischen Potentialen auf das EEG. (Quelle [15])

Funktionelles Gehirn monitoring

Funktionelles Gehirn monitoring (functional brain monitoring) wird in mehreren Bereichen der Medizin eingesetzt. Es wird verwendet um die Veränderung der Aktivität im Gehirn von Epileptikern während eines Anfalls zu messen. Bei operativen Eingriffen kann damit die Narkosetiefe bestimmt werden. Es dient als Hilfe bei der Feststellung ob ein Hirntod vorliegt und wie tief ein Koma ist. In der Schlafmedizin können aufgrund der gemessenen Daten Rückschlüsse über verschiedene Vorgänge während des Schlafs gewonnen werden. Ein weiteres großes Einsatzgebiet ist die Überwachung von Patienten während der Rehabilitation. Durch die Messung kann gezeigt werden, wie viel Erfolg mit den angewandten Rehabilitationsmethoden für den jeweiligen Patient erzielt wird. Falls nötig kann aufgrund der Daten die Behandlung so angepasst werden, dass bessere Ergebnisse erzielt werden können [11]. In der Arbeit werden bei den Messungen Daten aufgezeichnet, die bei der Auswertung zeigen sollen, dass das verwendete Paradigma die gewünschten Ergebnisse liefert.

2.1.8 Epilepsie

Im folgende Abschnitt wird ein Überblick über die Gehirnstörung Epilepsie geliefert und beschrieben, welche Arten unterschieden werden können. Weiters werden Therapiemethoden angeführt und gezeigt, wieso es in manchen Fällen notwendig ist, einen chirurgischen Eingriff durchzuführen. Epilepsie ist eine häufig chronisch auftretende Erkrankung unter der 1% der Weltbevölkerung leidet. Als einen epileptischen Anfall bezeichnet man einen spontan auftretenden Krampfanfall. Dieser wird durch eine exzessive neuronale Entladung hervorgerufen. Je nach Ursprung und Ausbreitung der Entladungen werden verschiedene Epilepsiearten unterschieden [6, 12]. Für die Diagnostik und Therapieplanung einer Epilepsie ist es wichtig den epileptischen Fokus genau zu bestimmen (siehe Epilepsiearten und Diagnose). Dabei werden neben oberflächlichen und invasiven EEG-Ableitungen auch bildgebende Verfahren wie Magnet-Resonanz-Tomographie (MRT), Positronen-Emissions-Tomographie (PET) und Single-photon emission computed tomography (SPECT) verwendet. Weiters finden auch neuropsychologische Verfahren Anwendung in der Bestimmung und Lokalisation des epileptischen Fokus. Dabei wird davon ausgegangen, dass verschiedene Muskelgruppen in verschiede-

nen Hirnregionen eine Aktivierung erzeugen. Fehlfunktionen von einzelnen Regionen können auf diese Art detektiert werden. (Epilepsitherapien 2.7.2). [6, 12]

Epilepsiearten und Diagnose

Die verschiedenen Verlaufsformen werden nach der Definition der Internationalen Liga gegen Epilepsie (ILAE) eingeteilt. Dazu wird eine grobe Unterteilung in fokale Epilepsien (nur eine Hirnregion einer Hirnhälfte ist betroffen) und generalisierte Epilepsien (beide Hirnhälften sind betroffen) durchgeführt. Die jeweiligen Übergruppen werden weiter in symptomatische, ideopatische und kryptogene Epilepsien unterteilt. Als symptomatisch werden Epilepsien bezeichnet, die aufgrund einer nachweisbaren Störung auftreten (z.B. Tumor, Missbildung, Durchblutungsstörung). Ideopatische Epilepsien werden durch eine genetische Disposition (Veranlagung) hervorgerufen. Ideopatische Epilepsien treten vorwiegend im Kindes- und Jugendalter auf. Als kryptogene Epilepsie wird die Form bezeichnet bei der beim Patienten die Ursache noch nicht aufgedeckt werden konnte, aber aufgrund des bisherigen Verlauf von einer organischen Störung ausgegangen werden kann. [12, 6]

Bei der Diagnose wird zuerst eine ausführliche Anamnese (Krankengeschichte) erstellt. In der Regel erlebt der Betroffene den Anfall nicht bewusst. Es werden daher auch Angaben von Personen, die während des Anfalls anwesend waren, miteinbezogen. Es werden verschiedene Diagnoseverfahren eingesetzt. Zu Beginn wird eine Befragung des Patienten und der Angehörigen durchgeführt. Danach können Blutanalysen, EEG, oder bildgebende Verfahren wie CT, MRT eingesetzt werden um weitere Informationen über den Zustand des Patienten zu gewinnen [12].

Detektion von Epilepsieherden mit EEG

Bei Patienten mit Epilepsie zeigen sich in den EEG-Ableitungen neben den normal auftretenden Wellen auch unregelmäßige Strukturen. Typische Formen in EEG-Ableitungen sind spikes, spike-slow-wave Komplexe, sharp waves und sharp-slow-wave Komplexe [12]. Diese weisen auf veränderte Hirnaktivität hin und sind während eines Anfalls deutlich erkennbar.

Vor allem bei Patienten mit fokalen Anfällen sind diese Veränderungen bestimmten Hirnregionen zuzuordnen. Daher kann dadurch festgestellt werden, welche Region als Auslöser in Frage kommt. Da es die Lokalisation des Epilepsieherdes erleichtert, wird versucht Daten während eines Anfalls zu erhalten. Für diesen Zweck wird ein Langzeit EEG erstellt. Dabei werden die Hirnströme über einen Zeitraum von mehreren Stunden aufgezeichnet. Zusätzlich zu den EEG Daten kann eine Videoaufzeichnung hilfreich sein. Damit ist es möglich, den Veränderungen in den Hirnströmen, bestimmte Bewegungen zuzuordnen. Es gibt Patienten bei denen die Ableitung von der Kopfoberfläche nicht ausreichend genaue Eingrenzungen der Hirnregionen ermöglicht. In solchen Fällen werden Elektroden, durch einen operativen Eingriff bei dem der Schädelknochen entfernt wird, direkt auf das Gehirn aufgebracht [12].

Epilepsitherapien

Es werden verschiedene Therapieformen unterschieden. Die Behandlung während des Anfalls bezeichnet man als Akuttherapie. Dabei werden Maßnahmen getroffen, die dem Epileptiker helfen den Anfall ohne zusätzliche Verletzung zu überstehen (z.B.: richtige Lagerung, spitze Gegenstände entfernen, gegebenenfalls aus einer Gefahrenzone entfernen). Da Epilepsie eine chronisch verlaufende Krankheit ist, benötigt der Patient in der Regel eine dauerhafte Therapie bzw. Langzeittherapie durch Einnahme eines Antiepileptikums. Die Hälfte der Patienten erreicht durch eine Therapie eine vollständige Anfallfreiheit. Bei 20-30 Prozent wird zumindest eine Verringerung der Anfallsschwere und Häufigkeit erzielt. Die restlichen 10-20 Prozent erfahren keine nennenswerte Besserung [6]. In solchen Fällen besteht die Möglichkeit, dass eine längere Behandlung bzw. ein Medikamentenwechsel eine Wirkung zeigen. Ansonsten ist eine operative Therapie in Betracht zu ziehen. Operiert können nur Patienten mit einer fokalen Epilepsie werden, da die Erkrankung von einem eingrenzbaeren Herd ausgeht. Ein chirurgischer Eingriff setzt eine sorgfältige EEG-Diagnostik voraus, um die Hirnregion sicher eingrenzen zu können. Unter Umständen ist das Aufbringen von Elektroden auf der Hirnoberfläche notwendig (siehe Detektion von Epilepsieherden mit EEG). Ein epilepsiechirurgischer Eingriff hat zum Ziel das Hirngewebe, das den Anfall auslöst, zu entfernen. Eine vollständige Entfernung ist nicht immer möglich, wenn betroffene Regionen für wichtige Funktionen (z.B.

Sprache, Bewegung) zuständig sind. Trotzdem wird durch eine teilweise Entfernung eine Erleichterung für den Patienten erreicht [12].

2.1.9 Event Related Synchronisation und Desynchronisation Maps

Für die Darstellung der Auswirkung der Bewegungen im Gehirn werden Event Related Synchronisation und Desynchronisation Maps (ERD/ERS Maps) verwendet. Damit können die transienten Veränderungen der Amplituden in den einzelnen gemessenen Regionen gezeigt werden [1].

Es ist möglich die Ergebnisse topografisch so anzuordnen, dass eine klare Vorstellung geschaffen wird, in welchen Regionen für die dargestellte Bewegung eine Synchronisation bzw. Desynchronisation stattfindet [15].

Für die Berechnung der ERD/ERS Maps werden Events benötigt, in welchen eine Veränderung in den Potentialen des Gehirns aufgrund einer Aktivität stattfindet. Wie diese Events aus den gemessenen Daten erzeugt werden, wird im Abschnitt EEG Messung im Punkt Messdaten und Event Extraktion genauer beschrieben.

2.2 Das Spiel

Dieses Kapitel wird dafür verwendet, das Spiel zu erklären. Zu Beginn werden, die für die Erstellung verwendeten Techniken beschrieben. Punkt 2.1 zeigt einen Gesamtüberblick mit den einzelnen Komponenten. Es wird beschrieben wie auf die von der Kinect erzeugten Daten zugegriffen werden kann und welche Information damit gewonnen wird (2.1.1 Kinect Daten). Folgend wird erklärt wie die Objekte erzeugt werden (2.1.3 Animation der Objekte) und wie mit den Kinect Daten der Avatar animiert wird (2.1.2 Animation des Avatars). Weiters wird noch erklärt mit welcher Technik die Schatten generiert werden (2.1.4 Schatten) und welche Berechnungen für die Kollisionserkennung (2.1.5 Kollisionserkennung) durchzuführen sind. Danach wird gezeigt, welche Datenstruktur verwendet wird, um eine Anbindung an den Signal Server durchzuführen (2.1.6 Anbindung an den Signal Server). Am Ende des Kapitels wird der Spielablauf während der Messungen erklärt (2.2 Spielablauf) und welche Daten erzeugt werden (2.3 Generierte Daten).

2.2.1 Framework

Eine der Hauptaufgaben der Arbeit besteht darin, ein Framework zu erstellen, in das einfache, weitere Module für verschiedene Anwendungen integriert werden können. Das Spiel besteht aus zwei Komponenten. Eine ist der Hauptthread, welcher für das Anzeigen der verschiedenen Fenster, die Positionierung und das Darstellen von deren Inhalt zuständig ist. Weiters kümmert er sich um die Aktualisierung der Daten, die von der Kinect erzeugt werden. Der Zugriff auf die Daten erfolgt über das openNI SDK. Der Hauptthread führt beim Programmstart alle notwendigen Initialisierungen durch und startet einen zweiten Thread. Alle Änderungen die das Aussehen des Fensters, in dem das Programm läuft betreffen, können hier beeinflusst werden. Zusätzlich können auch falls benötigt, verschiedene weitere Fenster über diesen Thread geöffnet werden.

Die zweite Komponente des Spiels, der vom Hauptthread gestartete Thread, ist für die Ablaufsteuerung, die Animationen und das Rendern der Grafik zuständig. Es werden hier die Daten erzeugt, welche an den Signal Server gesendet werden. Um ein anderes Paradigma zu verwenden, muss nur an der Stelle im Code, welche die verschiedenen Funktionen für das Rendering aufruft, eine entsprechende Veränderung vorgenommen werden.

2.2.2 Spielaufbau und Ablauf

Es soll ein Ablauf gefunden werden, mit dem isolierte Bewegungen erzeugt werden, anhand derer gezeigt werden kann, dass auch auf spielerische Art die benötigten Daten gesammelt werden können, die nach der Auswertung aussagekräftig sind. Es wird ein einfaches Paradigma verwendet, welches dem regelmäßigen Anheben einer Hand oder eines Beines ähnlich ist. Das Ziel des Spiels ist es den Benutzer zu animieren die für eine Untersuchung benötigten Bewegungen durchzuführen. Zusätzlich sollten nicht zu viele verschiedene Bewegungen (beide Hände, Muskelspannungen durch ruckartige Bewegungen) parallel bzw. zu schnell nacheinander erfolgen. Um dies zu erreichen wurden mehrere Parameter im Vorhinein festgelegt. Zwischen den Bewegungen sollte eine möglichst entspannte Position eingenommen werden. Es soll zwischen den berührbaren Objekten eine bestimmte Zeitspanne (mindestens 3 Sekunden) vergehen um eine durchgehende Aktivierung im EEG zu vermeiden. Bewegungen sollen nicht vor dem Körper vorbeiführen um Verdrehungen des Körpers und den damit verbundenen Muskelaktivierungen zu verhindern. Objekte die mit den Knie berührt werden können, sollen maximal auf Höhe der Knie erscheinen um ein Anheben der Beine und der damit verbundenen Anspannung der Bauch/Rückenmuskulatur zu vermeiden (Abb. 9).

Um zwischen den Bewegungen eine entspannte Haltung bei Probanden zu erreichen, erscheinen nur neue Objekte, wenn die Handflächen des Benutzers eine über die Skelett Parameter festgelegte Höhe unterschreiten (Abb.8). Als guter Schwellwert dafür hat sich die Skelett Hüfthöhe herausgestellt.

Da sich, wie oben erwähnt, Bewegungen nicht über die Körpermitte erstrecken sollen, werden für linke und rechte Hand eigene Objekte definiert (Abb.10, Abb.11). Diese erscheinen in einem bestimmten Feld in welchem jeder Punkt, ohne ein komplettes Strecken der Arme erreicht werden kann. Dadurch wird ein zu starkes Anspannen der Schultermuskulatur, sowie eine Bewegung des gesamten Oberkörpers vermieden. Weiters sollen die Objekte nicht zu tief erscheinen, um ein Vorbeugen zu verhindern. Objekte tauchen für die linke Hand im linken Viertel über Hüfthöhe und knapp über der berechneten Skelett Kopfhöhe auf. Für die rechte Hand tauchen Objekte im rechten oberen Viertel mit den selben Höhenbeschränkungen wie für die linke Hand. Objekte die mit den Knie berührt werden können, erscheinen immer in der Höhe die in den Skelett Daten als Knie erkannten Bereich. Diese Objekte variieren nur

in der horizontalen Achse.

Damit die Information welches Objekt als nächstes berührbar wird nicht zu vorhersehbar ist, werden laufend neue, neutrale Objekte erzeugt (Abb.8). Diese ändern erst nach einiger Zeit ihre Farbe und werden dadurch als nächstes Zielobjekt erkannt (Abb.9, Abb.10, Abb. 11).

Nach der Veränderung sind noch zwischen drei und vier Sekunden Zeit bis das Objekt sich am Benutzer vorbei bewegt. Dadurch ergibt sich ein Ablauf der sich wie folgt beschreiben lässt. In der entspannte Grundhaltung tauchen Objekte auf. Nach Überschreitung der Ruhephase ändert ein neutrales Objekt sich in genügend großem Abstand in ein berührbares Objekt. Dadurch ist es dem Spieler möglich die Hand oder Knie in die richtige Position zu bringen um eine Kollision zu erzeugen. Wenn der Proband eine Kollision von dem Avatar und einem Objekt schafft, wird ein kurzes akustisches Signal ausgelöst. Das getroffene Objekt bewegt sich abhängig von der Trefferposition in eine neue Richtung (siehe Kollisionserkennung). Da sich danach keine berührbaren Objekte mehr am Bildschirm befinden und auch keine neuen Objekte erscheinen, muss der Spieler wieder in die neutrale Position wechseln, um Objekte auf dem Bildschirm zu erzeugen.

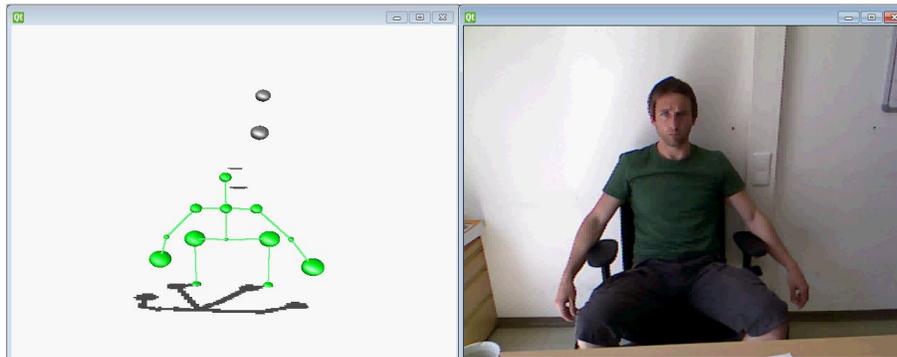


Abbildung 8: Benutzer befindet sich in neutraler Haltung um Objekte auf dem Bildschirm erscheinen zu lassen. Die neutrale Haltung wird durch die Positionierung der durch das Skelettfitting erhaltenen Handmittelpunkte unterhalb des berechneten Hüftpunktes erkannt.

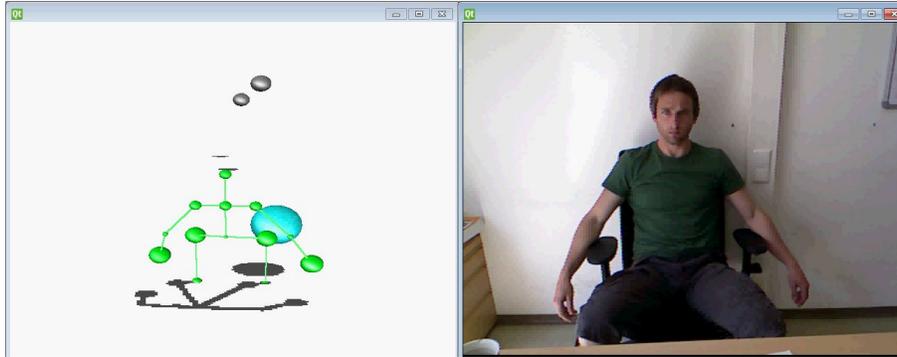


Abbildung 9: Ein cyan-farbiges Objekt befindet sich auf dem Bildschirm. Die Farbe signalisiert dem Benutzer, dass es sich um ein Objekt für die Knie handelt. Das Objekt befindet sich bereits auf der Höhe der Knie. Um eine Kollision zwischen Avatar und Knie zu erzeugen und um eine erkennbare Bewegung machen, genügt es in diesem Fall, das rechte Knie leicht abzusenken.

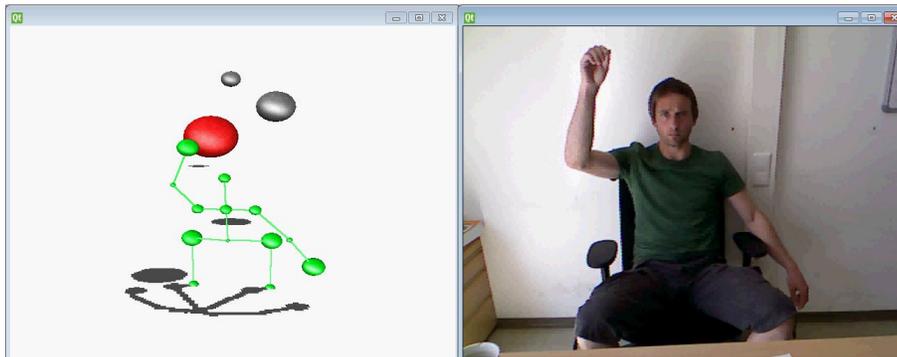


Abbildung 10: Ein rotes Objekt befindet sich auf dem Bildschirm. Die Farbe signalisiert dem Benutzer, dass es sich um ein Objekt für die linke Hand handelt. Rote Objekte erscheinen immer links und über der Höhe des Hüftpunktes des Avatars.



Abbildung 11: Ein blaues Objekt befindet sich auf dem Bildschirm. Die Farbe signalisiert dem Benutzer, dass es sich um ein Objekt für die rechte Hand handelt. Blaue Objekte erscheinen immer rechts und über der Höhe des Hüftpunktes des Avatars.

2.2.3 Komponenten

Dieser Abschnitt liefert einen Überblick aus welchen Komponenten sich das Spiel zusammensetzt und wie diese zueinander in Verbindung stehen (Abb.12). Es auch wird beschreiben, welche Techniken eingesetzt worden sind, um die gewünschte Funktionalität beim Spiel zu erhalten.

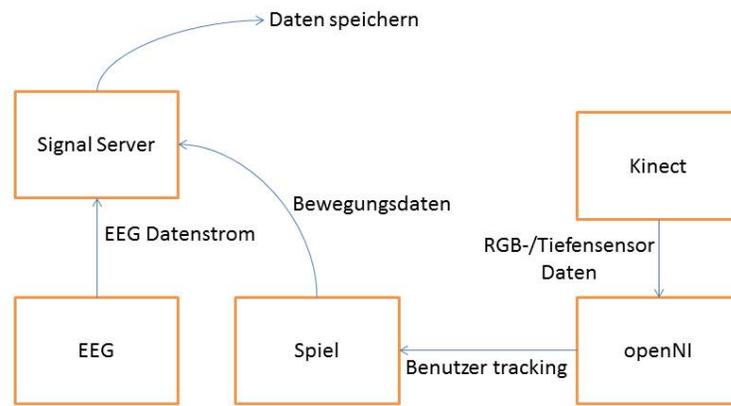


Abbildung 12: Die Abbildung zeigt die einzelnen Komponenten, die für das Spiel benötigt werden. Die Kinect wird verwendet um Information über die Szene zu erhalten. Auf die erzeugten Daten wird mit Hilfe des openNI SDK zugegriffen. Weiters wird die zur Verfügung gestellte Funktion einer Skelettberechnung verwendet, um die Bewegung des Benutzers zu tracken. Die Position und Bewegung des Benutzers werden benutzt um den Avatar in der Szene zu positionieren und zu animieren. Die erzeugten Daten werden über eine Schnittstelle an den Signal Server gesendet, der diese mit den EEG Datenstrom synchronisiert und in eine Datei speichert.

Kinect Daten

Der Zugriff auf die Daten der Kinect erfolgt, wie im Kapitel Grundlagen beschrieben über die Middleware openNI. Um Information über die Bewegungen des Benutzers zu erhalten, wird auf die Szenentiefeninformation zugegriffen. Mit Hilfe der des NITE Framework kann

mit diesen Daten eine Segmentierung des Benutzers aus den Daten erfolgen [23, 20]. Dafür muss zu Beginn des Spiels eine Änderung der Körperhaltung durchgeführt werden, um eine Vorder-/Hintergrund Segmentierung in den Daten zu ermöglichen. Nachdem der Benutzer von System erkannt worden ist, muss dieser eine Initialisierungspose (Abb. 13) einnehmen, um die Skelett Berechnung zu aktivieren. Sobald diese erfolgreich durchgeführt worden ist, werden in jedem Frame für jedes Gelenk 3D Daten erzeugt, welche in weiterer Folge für die Animation des Avatars verwendet werden.

Zusätzlich zum Spielfenster kann ein weiteres Fenster (Abb.8 rechtes Fenster) angezeigt werden, welches den Benutzer darstellt. Dafür werden die RGB-Daten der Kinect verwendet. Dieses Fenster wird zur Zeit für das Erstellen von Videos verwendet, um den grundsätzlichen Ablauf des Spiels zu zeigen. Für den eigentlichen Ablauf des Spiels, beziehungsweise für das Erzeugen von relevanten Daten für die Auswertung, wird es nicht benötigt und ist während der Tests auch deaktiviert.



Abbildung 13: Die Abbildung zeigt ein Grauwertbild, das abhängig von den Werten des Tiefensensors eingefärbt ist. Größere Werte erzeugen eine stärkere Einfärbung. Der schwarze Bereich links und oben im Bild ergibt sich durch den Abgleich von RGB Bild und Tiefensensor Bild. Der blau eingefärbte Teil ist der Bereich der durch die Vorder-/Hintergrundsegmentierung als Benutzer erkannt wird.

Animation des Avatars

Folgend wird beschrieben welche Dinge bei der Animation und Positionierung des Avatars beachtet werden müssen. Wie bereits erwähnt werden für die Animation des Avatars die 3D Koordinaten aus der Skelett Berechnung [18] verwendet. Um eine Verwendung der Daten zu ermöglichen muss beachtet werden, dass die Koordinatensysteme von Kinect und Spiel nicht übereinstimmend sind. Im Spiel liegt die hintere Clippingebene auf der positiven Z-Achse, welche die dargestellte Szene zeigt und die vordere Clippingebene ist kurz hinter dem Avatar festgelegt. In den Tiefenwerten der Kinect liegt der Ursprung beim Standpunkt der Kinect und von diesem ausgehend in Sensorrichtung wird die positive Z-Achse festgelegt (Abb 14). Daher ergibt sich in der Berechnung des Skelettes eine Rotation um 180 Grad. Als Folge davon ergeben sich auch in den Daten für die Gelenke falsche Koordinaten. Für das Spiel wird das Kinect Koordinatensystem an der des Spiels angepasst. Zuerst erfolgt eine Spiegelung an der Y-Achse. Weiters muss auch die Z-Koordinate korrigiert werden. Die Skelett Berechnung liefert auch einen Punkt für den Nackenansatz des Benutzers. Dieser Punkt wird als Bezugspunkt für die Umrechnung der Z-Koordinaten verwendet. Zuerst wird die Differenz zwischen der Z-Koordinate des Gelenks und der des Bezugspunkts ermittelt. Danach wird dieser Wert zur Z-Koordinate des Bezugspunkts addiert. Dadurch ergibt sich eine Spiegelung der Z-Koordinaten in der X-Y-Ebene des Bezugspunktes (Abb. 14). Nach dieser Anpassung des Koordinatensystems werden die Punkte als Modell-Koordinaten gespeichert. Mit diesen Daten wird in weiterer Folge die Position der Gelenke des Avatar bestimmt. Für die Darstellung der Gelenke werden verschieden große Kugeln verwendet und als Verbindung einfache Linien. Diese Art der Visualisierung ermöglicht, wie später beschrieben, eine relativ einfache Kollisionserkennung zwischen den Objekten und Avatar. Die Position der Gelenke wird mit der Arbeitsfrequenz (ca. 30 Hz) der Kinect verändert und an die Bewegung des Benutzers angepasst. Diese Updaterate genügt um eine für den Betrachter fließende Animation zu erzeugen.

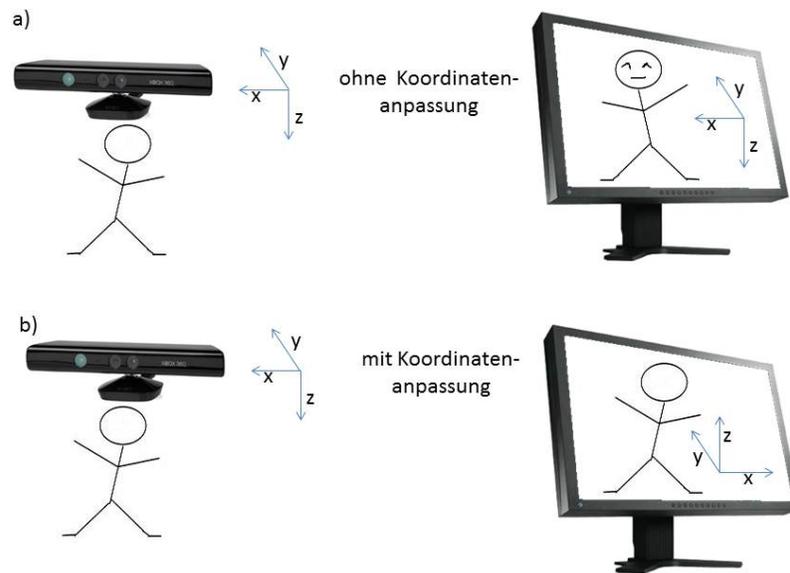


Abbildung 14: Die Abbildung zeigt das Problem, welches sich durch die verschiedenen Koordinatensysteme ergibt. Fall a) zeigt das Ergebnis ohne Koordinatenkorrektur. Da der Benutzer mit dem Gesicht in Richtung Kinect steht (von vorne gesehen wird) ergibt sich im Spiel die Abbildung der Koordinaten ebenfalls von vorne. Das führt aber zu einer Spiegelung an der Y-Achse und zusätzlich zu einer Negativierung der Z-Achse. Durch diese Achsenänderungen ergibt sich ohne Korrektur eine falsche Animation des Avatars. Fall b) zeigt das Ergebnis nach der Korrektur. Die X-Achse wird an der Y-Achse gespiegelt und die Z-Achse an der X-Y-Ebene des Bezugspunkts. Dadurch ergibt sich, dass der Avatar sich entsprechend dem Benutzer bewegt.

Animation der Objekte

Während des Spielverlaufs tauchen immer neue Objekte am Bildschirm auf. Folgend werden die Kriterien beschrieben wie die Positionen der Objekte festgelegt werden. Wie bereits beim Avatar wurde auch für die Darstellung der Objekte eine Kugel gewählt, da sich spätere Berechnungen dadurch vereinfachen. Beim Erstellen eines Objekts werden mehrere Phasen durchlaufen. Da vermieden werden soll, dass sich zu viele Objekte am Bildschirm befinden, wird vor dem Erstellen die aktuelle Anzahl überprüft. Weiters sollen auch nicht mehrere Objekte zugleich oder in zu kurzem Abstand zueinander erscheinen. Daher wird auch der Zeitpunkt des Erscheinens des letzten Objekts überprüft. Sollten beide Kriterien erfüllt sein,

wird ein neues Objekt erstellt. Um die initiale Startposition festzulegen, wird zufällig (Gleichverteilung mit jeweils einem Drittel Wahrscheinlichkeit für jede Objektklasse) ermittelt, ob es sich um ein Objekt für Knie, linke oder rechte Hand handeln soll. Dies ist wichtig, da Objekte für die Knie nicht über der für das Skelett ermittelten Kniehöhe befinden sollen. Wenn das neue Objekt für die Knie bestimmt ist wird für die Y-Koordinate die Kniehöhe verwendet. Die X-Koordinate wird zufällig (Normalverteilung) auf einen Wert, der maximal 30cm links oder rechts des Hüftpunktes liegt, festgelegt. Handelt sich bei dem neuen Objekt um eines für die Hände, wird für die X-Koordinate ein Wert ermittelt der sich maximal 65cm links oder rechts des Nackenansatzes befindet. Die Y-Koordinate wird abhängig von der Hüfthöhe maximal 65cm oberhalb dieser festgelegt. Die Z-Koordinate wird immer auf die hintere Clippingebene gelegt. Zusätzlich werden die Größe und Geschwindigkeit des Objekts festgelegt. Beide Werte können in einem definiertem Bereich (1.5 bis 2 mal die Größe der Handflächen des Avatars) variieren, um eine Änderung im Aussehen der Objekte zu erhalten und vom Benutzer eine Anpassung an die Geschwindigkeit zu verlangen, wodurch ein gewisses Maß an Konzentration gefordert ist. Die Bewegungsrichtung der Objekte ist zu Beginn immer in Richtung X-Y-Ebene des Ursprungs. Als Ausgangswert für die Farbe ist Grau gewählt worden (Abb.8).

Für die Animation der Objekte wird in jedem Frame die Position in Abhängigkeit zur Bewegungsrichtung und Geschwindigkeit verändert [17]. Dadurch ergibt sich zu Beginn der Eindruck, dass sich die Objekte auf den Bildschirm zu bewegen. Nach einer Kollision wird auch die Bewegungsrichtung neu bestimmt und dieser entsprechend bewegt sich das Objekt danach in eine andere Richtung (siehe Kollisionserkennung). Ab einem bestimmten Abstand zum Benutzer ist es möglich, dass sich die Farbe des Objekts verändert. Der Abstand ist so gewählt, dass abhängig von der Objektgeschwindigkeit, noch zwei bis drei Sekunden für den Probanden zur Verfügung stehen, den Avatar in die richtige Position zu bringen. Da auch hier nicht erwünscht ist, dass zu viele Objekte die Farbe ändern, wird wieder ein vordefinierter Zeitwert überprüft. Dafür wird die Zeit des letzten Objekts, welches die Farbe verändert hat, herangezogen. Sollte dieser Test positiv sein wird die Farbe verändert. Dafür wird in Abhängigkeit der Position, wie im Spielablauf beschrieben, die Farbe gewählt.

Um zu bestimmen welches Objekt noch angezeigt werden muss, wird die hintere und vordere

Clippingebene verwendet. Sollte die Z-Koordinate größer als die hintere Clippingebene oder kleiner als die vordere Clippingebene sein, wird das Objekt gelöscht. In jedem anderen Fall wird das Objekt am Bildschirm dargestellt.

Schatten

Ein weiterer wichtiger Punkt in Spielen ist das Erzeugen von Schatten. In diesem Kapitel wird die für die Schattenerzeugung verwendete Technik beschrieben. Schatten werden im Spiel verwendet um, wie im Abschnitt 2.1.3 Basistechniken für die Entwicklung eines Spiels beschrieben, den optischen Eindruck zu verbessern und dem Benutzer eine zusätzliche visuelle Stütze zu liefern (Abb.4). Durch den Einsatz von Schatten ist es einfacher die räumlichen Abstände der Objekte zum Avatar einzuschätzen.

Für die Erzeugung der Schatten wurde ein Shadowmapping Algorithmus [34]implementiert. Bei der verwendeten Technik muss für zumindest eine Lichtquelle in der Szene vorhanden sein. Im Spiel wird aus Performanz-Gründen auch nur eine Lichtquelle verwendet. Als erster Schritt wird die Szene von der Position der Lichtquelle in Richtung der Lichtstrahlen als Tiefenwerttextur gerendert. In einem weiteren Durchgang wird die Szene aus Sicht der Kamera (Benutzer) gerendert.

Im nächsten Schritt wird die Tiefenwerttextur an einen Shader übergeben, der pro Frame für jedes Pixel vergleicht, ob es beleuchtet wird oder im Schatten liegt. Für genauer Informationen wie Shader funktionieren und Schatten erzeugt werden siehe OpenGL Shading Language [26].

Kollisionserkennung

Das Ziel des Spiels ist es Objekte zu treffen. Es ist aus diesem Grund unerlässlich, wie auch in [21, 14] beschrieben, eine Kollisionserkennung zu implementieren. Da alle Objekte, die an einer Kollision beteiligt sein können, kugelförmig sind, wird für die Kollisionserkennung eine vereinfachte OBB Variante (siehe Basistechniken für die Entwicklung eines Spiels) eingesetzt.

Um Kollisionen zu erkennen werden in jedem Frame folgende Punkte durchlaufen. Zu Beginn wird festgestellt ob ein Objekt vorhanden ist, das mit einem Körperteil in Berüh-

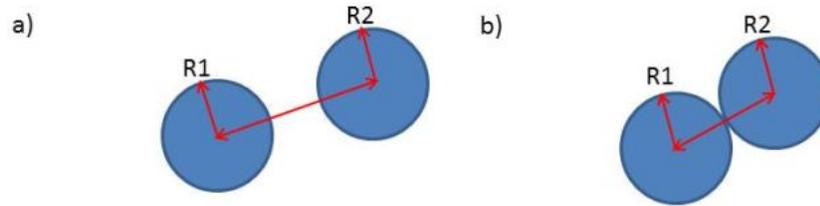


Abbildung 15: Die Abbildung zeigt die beiden Möglichkeiten die sich bei der Detektion einer Kollision ergeben. a) die Summe der Radien der Objekte ergibt einen kleineren Wert als der Abstand der Mittelpunkte der Objekte. Daraus folgt, dass sich die Objekte nicht berühren können und keine Kollision vorhanden ist. b) die Summe der Radien der Objekte ergibt einen Wert als gleich oder größer dem Abstand der Mittelpunkte der Objekte ist. Daraus folgt, dass sich die Objekte berühren können und eine Kollision vorhanden ist.

rung steht. Objekte können nur mit den Händen oder den Knie berührt werden. Es wird der Anstand vom Mittelpunkt eines Objekts bis zum Mittelpunkt eines Körperteils berechnet. Danach wird der Abstand mit der Summe der Radien der beiden Körper verglichen. Sollte diese größer als der Abstand sein, entspricht dies einer Kollision (Abb.15). Da Objekte, wie bereits im Kapitel Spielablauf beschrieben, abhängig von ihrer Farbe nur mit dafür vorgesehenen Körperteilen berührt werden können, wird, im Fall einer möglichen Kollision, als nächstes überprüft ob das Objekt mit dem richtigen Körperteil kollidieren würde. Trifft auch dies zu wird die neue Flugrichtung des Objekts berechnet. Dazu werden die Richtungsvektoren der beiden an der Kollison beteiligten Objekte gebildet. Danach wird der Abprallwinkel in Abhängigkeit zur Bewegungsrichtung gebildet. Abschließend wird eine neue Bewegungsgeschwindigkeit berechnet. Diese ergibt sich aus der Geschwindigkeit des Objekts vor der Kollision und der Bewegungsgeschwindigkeit mit der der Benutzer das Objekt trifft.

Anbindung an den TOBI Signalsever

Während des Spiels werden laufend Daten erzeugt, die später für die Auswertung verwendet werden. Um zu gewährleisten, dass EEG und Spiel parallel und zur richtigen Zeit Daten schreiben, wird der TOBI Signal Server [2] verwendet. Mit dem Signal Server ist es möglich die mit ca. 30 Hz generierten Daten vom Spiel auf die EEG Abtastfrequenz von 512 Hz upzusamplen. Da die EEG Samplingrate im Normalfall kein Vielfaches von der Framerate

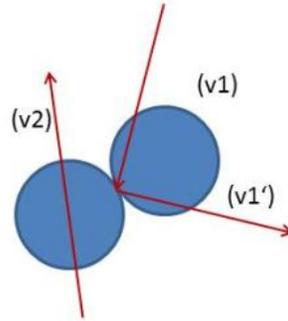


Abbildung 16: Nach der Bildung der beiden Bewegungsvektoren $v1$ und $v2$ wird der Abprallwinkel berechnet und daraus der neue Bewegungsvektor erstellt. Weiters wird die neue Bewegungsgeschwindigkeit in Abhängigkeit der Geschwindigkeit des Objekts vor der Kollision, und der Bewegungsgeschwindigkeit des Benutzers während der Kollision gebildet.

(ausser bei 32Hz) ist, ergibt sich beim Upsampling ein Fehler. Die Framerate der Kinect liegt zwischen 28Hz und 32Hz. Dadurch ergibt sich ein maximaler Fehler von 15,6ms bei einer Framerate von 28Hz durch das Upsampling.

Da es keine vorgefertigte Schnittstelle für das Einlesen von Daten aus einem Programm gibt, wurde eine eigene Kommunikationsmöglichkeit zwischen Signal Serverm und Spiel implementiert. Dafür wird beim Spielstart ein Shared Memory erzeugt. In diesen Shared Memory werden alle Daten geschrieben, die für die spätere Auswertung relevant sind. Zusätzlich wird nach jedem Datenupdate ein Flag gesetzt, dass neue Daten anliegen.

Nachdem das Spiel gestartet ist, kann auch der Signal Server gestartet werden. Der Signal Server greift über die neu erstellte Schnittstelle auf den Shared Memory Bereich zu. Sobald neue Daten von Spiel erzeugt werden, wird auch das Flag auf true gesetzt. Dadurch wird dem Signal Server angezeigt, dass die Daten im Speicher neu sind und dies an der entsprechende Position in den Daten eingefügt werden müssen.

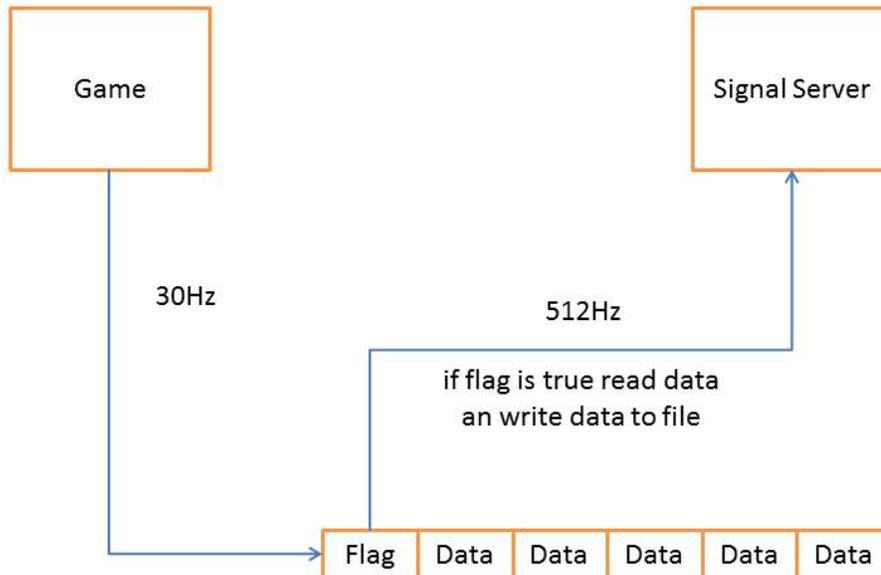


Abbildung 17: Die Abbildung zeigt den Zugriff der beiden Programme auf den Shared Memory. Vom Spiel werden mit ca. 30 Hz Daten in den Speicher geschrieben. Jedesmal nach dem Schreiben wird das Flag auf true gesetzt. Der Signal Server liest mit 512 Hz die Daten im Speicher. Um zu verhindern, dass nicht relevante Daten gespeichert werden, werden die Daten im Speicher nur dann ins File geschrieben, wenn das Flag true ist. Nach dem Schreiben der Daten setzt der Signal Server das Flag auf false.

2.2.4 Generierte Daten

Der Sinn des Spiels ist es, den Benutzer zu animieren die gewünschten Bewegungen auszuführen. Zusätzlich wird aber auch Information erzeugt, die es ermöglicht in der späteren Auswertung der EEG Daten die Bereiche zu finden in denen sich der Benutzer bewegt hat. Dieser Abschnitt erklärt welche Daten generiert werden und wie diese gespeichert werden. Um die Abschnitte zu detektieren in welchen sich der Benutzer bewegt hat, wird für jeden relevanten Körperteil ein Kanal in den Daten reserviert. Sobald der Benutzer einen Körperteil bewegt, wird dies in den Daten die an den Signal Server geschickt werden eingetragen.

Abhängig von der Beschaffenheit der Oberfläche der Bekleidung des Benutzers und von den Lichtverhältnissen im Raum können Reflexionen auftreten. Daher ergeben sich in der Berechnung des Skelettes Schwankungen, die eine nicht vorhandene, falsche Bewegungserkennung auslösen können. Um das zu verhindern werden Kriterien eingeführt, ab wann eine Positionsveränderung auch in die Daten eingeht. Es werden für die Bewegungsgeschwindigkeit Werte zwischen eins und fünf vergeben. Sollte sich einer Handfläche des Avatars über seiner Hüfthöhe befinden wird in den Daten der Wert eins gespeichert um zu kennzeichnen, dass sich der Benutzer nicht in Ruheposition befindet. Um Bewegung zu erkennen, werden die Positionen von allen Gelenken aus den vorherigen Frame mit den aktuellen Positionen verglichen. Abhängig vom Positionsunterschied, um die Geschwindigkeit der Bewegung zu reflektieren, werden bei größeren Positionsunterschieden Werte zwischen zwei (der Proband hat sich innerhalb von zwei Frames um mehr als 10 cm bewegt) und fünf (der Proband hat sich innerhalb von zwei Frames um mehr als 25 cm bewegt) gespeichert. Die Werte werden diskretisiert, um bei der Auswertung der Daten einfacher zwischen gewollter Bewegung und durch Trackingprobleme entstandenen Bewegungen unterscheiden zu können.

Der Benutzer hat nach der Änderung der Objektfarbe genügend Zeit den benötigten Körperteil richtig zu positionieren. Dadurch ist es möglich, dass nach einer kurzen Zeitspanne keine Bewegung mehr notwendig ist. Um trotzdem zu registrieren, dass sich der Körperteil nicht in Ruheposition befindet, wird bei den Handbewegungen überprüft, ob sich der Handmittelpunkt des Avatars über dem Hüftpunkt befindet. Sollte das der Fall sein, wird so lange eine Eins in die Daten geschrieben, bis sich die Hand wieder dem definierten Ruhebereich befindet.

Während dem Verlauf der Messung sitzt der Proband. Daher ist die Detektion der Kniebewegung aufgrund der relativ kurzen Bewegung die notwendig ist um die Objekte zu treffen etwas schwieriger. Die Benutzer wurden während des Tests gebeten beim Auftauchen eines Objekts für die Knie solange eine links-rechts Bewegung auszuführen bis das Objekt getroffen wurde. Daher ergeben sich für die Knie folgende Überprüfungen bevor eine Bewegung auch an den Signal Server geschickt und gespeichert wird. Zuerst wird überprüft ob sich ein Objekt für die Knie am Bildschirm befindet. Sollte dies zutreffen wird weiters überprüft ob die Knieposition (Trackingprobleme) auch möglich ist. Erst nachdem auch diese Überprüfung positiv

abgeschlossen ist, werden die Bewegungsdaten ermittelt und an den Signal Server geschickt. Die durch die oben beschriebenen Techniken erzeugten Daten beinhalten Bewegungsintervalle von mindestens zwei Sekunden, welche wie in Kapitel 4 beschrieben, gefiltert werden können (Abb. 18).

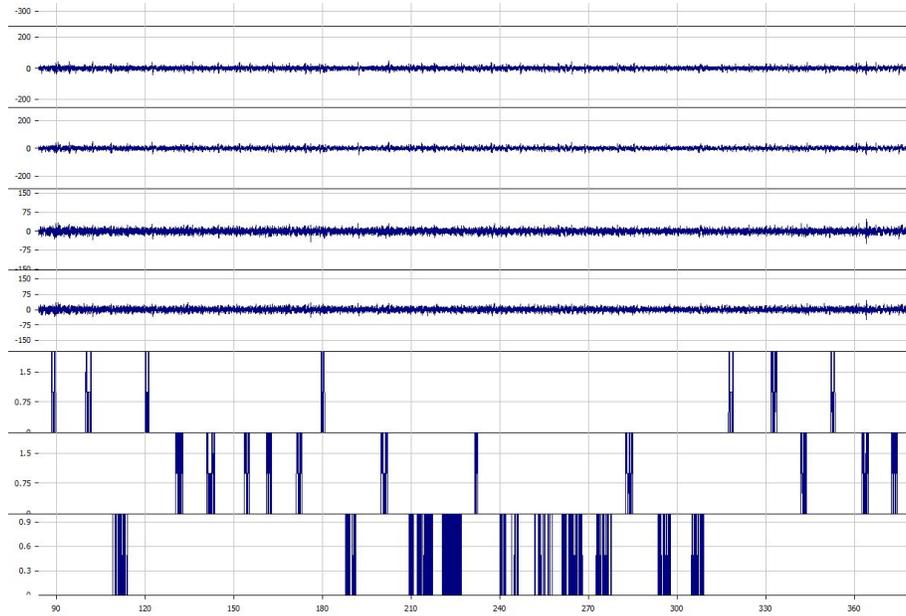


Abbildung 18: In den untersten drei Kanälen befinden sich die Daten, die aufgrund der Bewegung die der Benutzer durchführt, vom Programm an den Signalserver gesendet werden. Die X-Achse stellt den Zeitverlauf dar und auf der Y-Achse sind die Amplituden, abhängig von der Bewegungsgeschwindigkeit, zum jeweiligen Zeitpunkt aufgetragen. Die blauen Bereiche werden später bei der Analyse der Daten zur Generierung der Epochen verwendet. Die Kanäle darüber beinhalten EEG Daten.

2.3 Messung

Dieses Kapitel beschreibt das für die Messung verwendete Simulink Model (4.3 Simulink Model). Weiters wird der Aufbau (4.1 Messaufbau) und die Elektrodenmontage (4.2 Elektrodenmontage) bzw. der Ablauf der Messung beschrieben (4.4 Paradigma). Im letzten Abschnitt werden die Daten, welche sich bei der Messung ergeben genauer betrachtet (4.5 Messergebnis).

2.3.1 Messaufbau

Der für die Messung verwendete Aufbau wird in Abb. 19 gezeigt. Zu Beginn werden am Benutzer alle für die Messung benötigten Elektroden (4.2 Elektrodenmontage) montiert. Diese werden mit dem g.USBamp verbunden. Um die vom Verstärker erzeugten Daten für MATLAB zugänglich zu machen, werden die Daten an den SignalServer weitergeleitet.

Nachdem die Elektrodenmontage am Benutzer erfolgt ist, wird dieser so positioniert, dass er von der Kinect komplett erfasst wird, um die Skelettberechnung zu ermöglichen. Weiters wird die Ausrichtung so vorgenommen, dass sich der Benutzer mit dem Monitor und der Kinect in einer Linie befindet. Damit wird verhindert, dass der Benutzer seinen Kopf drehen muss und sich dadurch die Nackenmuskulatur anspannt was zu zusätzlichen, nicht gewollten, Potentialschwankungen führen würde. Sobald die Ausrichtung erfolgt ist, wird das Spiel gestartet. Nachdem die Benutzererkennung durchgeführt worden ist, werden die erzeugten Daten ebenfalls an den Signalserver geschickt.

Die am Signalserver ankommenden Daten werden dann mit MATLAB wie in Kapitel 4.3 beschrieben weiter verarbeitet.

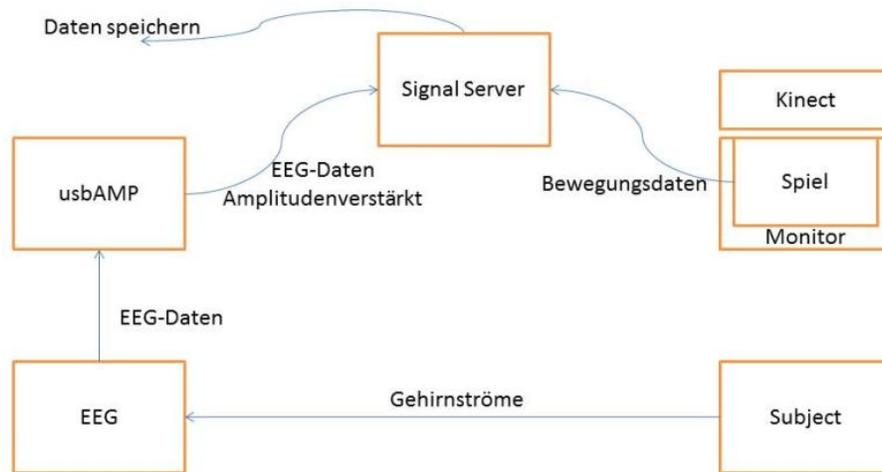


Abbildung 19: Die Abbildung zeigt schematisch den für die Messung durchgeführten Aufbau. Es entstehen zwei Datenströme die über den Signalserver synchronisiert werden. Links wird mittels EEG die Potentialschwankung des Probanden durch eine Verstärkung mit dem g.usbAMP erzeugt. Rechts werden die von der Kinect und dem verwendeten Framework erzeugten Daten ebenfalls an den Signalserver gesendet.

2.3.2 Elektrodenmontage

Für die spätere Auswertung werden während des Versuchs EEG Daten von folgenden Elektrodenpositionen aufgezeichnet. Über dem motorischen Cortex werden die Positionen CP3, CPz, CP4, FC3, FCz, FC4 gemessen. Diese Areale werden gemessen, da in diesen Bereichen eine Aktivierung erwartet werden kann, wenn der Benutzer die linke bzw. rechte Hand oder die Knie bewegt. Weiters wird über den visuellen Cortex O1, O2, PO3, PO4 gemessen, um die Aktivierung beim Erscheinen eines Objekts feststellen zu können. Eine Elektrode wurde am Frontallappen an der Position AFz angebracht um die dort stattfindende Bewegungsvorbereitung in späteren Berechnungen auch darstellen zu können. Die möglichen Elektrodenpositionen der verwendeten Montagehaube werden in Abb. 20 gezeigt.

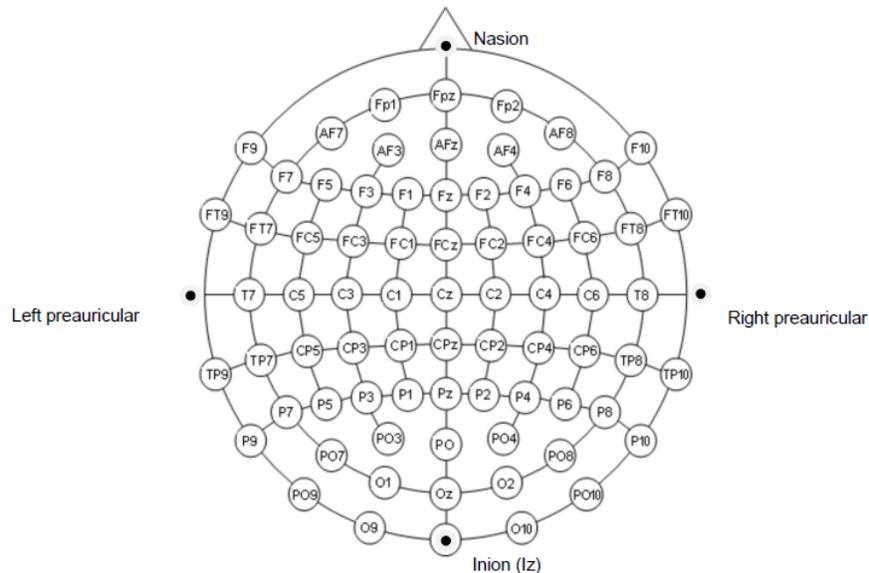


Abbildung 20: In der Abbildung ist eine Darstellung der möglichen Elektrodenpositionen die bei der verwendeten Montagehaube zur Verfügung stehen. Gemessen wurden die Positionen O1, O2, PO3, PO4 am visuellen Cortex, CP3, CPz, CP4, FC3, FCz, FC4 am sensomotorischen Cortex und AFz am Frontallappen. (Quelle: <http://www.gtec.at/>)

2.3.3 Simulink Model

Für die Aufnahme der Daten werden bereits zur Verfügung stehenden Komponenten aus der MATLAB BCI Blocks for Simulink Toolbox verwendet. Damit ist es möglich die Daten, die von Signalserver gesammelt werden auszulesen. Für jede Hardware die im Konfigurationsfi-

le angegeben wird, stehen im Model eigene Ausgänge zur Verfügung (Abb. 21). Die Daten werden in einem Unbuffer Block gesammelt um sie in weiterer Folge mit einem C++ DAQ Saver Block im richtigen Format über den GDF Writer in einer Datei zu speichern. Für die spätere Auswertung werden nur die ersten beiden Datenausgänge des SignalServer Clients benötigt. Der erste Kanal (eeg) beinhaltet alle anfallenden EEG Daten. Der zweite Kanal wird für die Kinect Daten verwendet. Zur Überprüfung der erzeugten Daten ist an die beiden Datenströme ein Anzeigefenster (status) gekoppelt.

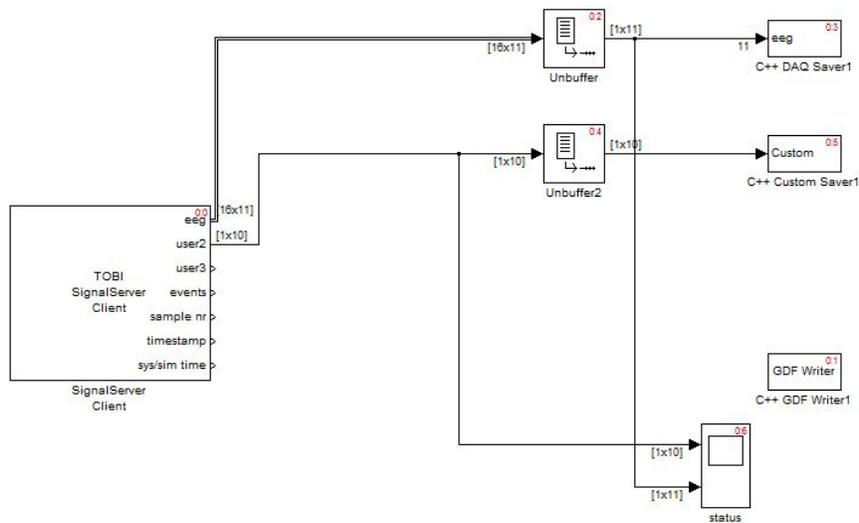


Abbildung 21: Das für die Messung verwendete Simulink Model beinhaltet zwei Datenströme. Der Erste ist das EEG Signal. In den eckigen Klammern wird die Anzahl der Kanäle angezeigt. Der Zweite beinhaltet die Daten die mit Hilfe des Spiels und der Auswertung der Bewegung des Benutzters erzeugt werden.

2.3.4 Paradigma

Um eine klare Unterscheidung zu ermöglichen, welches Objekt mit welchem Körperteil berührt werden kann, werden diese in klar erkennbaren Farben dargestellt (rote Objekte für links, blau Objekte für rechts, cyane Objekte für Knie). Es erscheint zur Laufzeit bei entspannter Haltung alle zwei Sekunden ein neues graues Objekt. Alle Objekte berührbar zu machen würde dazu führen, dass sich der Benutzer zu viel und zu schnell bewegen müsste

und als Folge davon keine klar definierten Bewegungsmuster entstehen. Daher wird eine Zeit vorgegeben die verstreichen muss, bevor das nächste berührbare Objekt auftaucht. Für die durchgeführten Tests wurden neun Sekunden verwendet. Das führte zu Daten die zwischen Anfang und Ende der Bewegung mindesten drei und maximal fünf Sekunden Ruhephase beinhalten. Die Dauer der Bewegung variiert zwischen zwei und drei Sekunden. Als Bewegung wird erkannt, wenn der Benutzer eine erkennbare Positionsänderung zwischen zwei Frames durchführt oder sich seine Handfläche nicht unter dem Schwellwert befindet. Als Ruhephase für Handbewegungen wird der Zeitpunkt verwendet, wenn der Benutzer beide Handfläche unter dem definierten Schwellwert hat und zwischen zwei aufeinander folgenden Frames keine erkennbare Bewegung durchgeführt wird. Um schnelle, ruckartige Bewegung, und die daraus resultierende starke Muskelkontraktion, in normal für die Bewegung nicht benötigten Körperbereichen zu vermeiden, wurden die Probanden angehalten bei Auftauchen eines von einer Hand berührbaren Objekts diese möglichst früh in die richtige Position zu bringen. Zusätzlich wird dadurch erreicht, dass sich die Dauer der Bewegungsphasen auf die oben genannten Werte erhöht. Da die Bewegung der Knie in einem sehr eingeschränkten Bereich stattfindet, wurden die Probanden angehalten beim Erscheinen eines Objekts für die Knie diese etwas links und recht zu bewegen um eine leichter detektierbare Positionsänderung zu erhalten.

Um zu zeigen, dass die beschriebene Methode richtige Ergebnisse liefert wurden fünf gesunde Probanden gemessen. Alle waren Rechtshänder und in einem Alter zwischen 25 und 30 Jahren.

2.3.5 Messdaten und Event Extraktion

Das Ergebniss der Messung ist eine Datei die für jeden gemessenen EEG-Kanal zu jedem Samplezeitpunkt eine Amplitude enthält. Weiters werden alle vom Spiel erzeugten Daten für die Bewegungen des Benutzers, abhängig von der Bewegungsgeschwindigkeit und dem bewegten Körperteil, gespeichert. Die Amplitude der Bewegungsdaten wird in diskrete Stufen unterteilt und abhängig von der Geschwindigkeit vergrößert bzw. verringert um Information über die durchgeführten Bewegungen zu bewahren.

Für Extraktion der Events aus den Bewegungsdaten werden folgende Merkmale verwendet:

Ein Event beginnt immer bei einer steigenden und endet bei einer fallenden Flanke.

Eine steigende Flanke zeigt den Beginn einer Bewegung und eine fallende Flanke einen Bewegungsstop.

Die Dauer der Bewegung muss einen Zeitraum von mindestens zwei Sekunden überschreiten. Dadurch wird vermieden, dass Events durch falsch erkannte Bewegung in die Daten einfließen.

Vor einer steigenden Flanke und nach einer fallenden Flanke muss ein Zeitraum von mindestens drei Sekunden ohne Bewegung sein.

Dieses Kriterium sorgt dafür, dass nur klar definierte Bewegungen verwendet werden. Es wird verhindert, dass Events aus Daten erzeugt werden, welche überschneidende Bewegungen von mehreren Körperteilen oder zu schnell aufeinander folgende Aktionen des Benutzers beinhalten.

Mit den, wie oben beschrieben, extrahierten Events werden im nächsten Schritt, aus den EEG Daten die Bereiche entnommen die Bewegung enthalten. Um auch den Zeitbereich vor der Bewegung später visualisieren zu können, werden alle Events um vier Sekunden vor dem eigentlichen Beginn verschoben. Dadurch werden in den weiteren Berechnungen auch Daten verwendet und angezeigt die noch keine Aktivierung in den entsprechenden Hirnarealen beinhalten.

3 Ergebnisse

In diesem Kapitel werden die Ergebnisse der Berechnungen in Form von ERD/ERS Maps präsentiert. Für die Abbildungen werden die bipolaren Ableitungen von AZz-FCz für die erste Reihe, FC3-CP3, FCz-CPz, FC4-CP4 für die zweite Reihe (von links nach rechts) und PO3-O1, PO4-O2 für die dritte Reihe gebildet. Die horizontale Achse zeigt den Zeitverlauf. Die schwarze durchgängige vertikale Line entspricht dem aus den Berechnungen hervorgehenden Bewegungsbeginn. Zwischen den beiden punktiert dargestellten Linien befindet sich der für die Berechnung festgelegte Referenzbereich. Auf der vertikalen Achse wird die Frequenz aufgetragen.

Die Leistung des Signals wurde im Bereich von 5-30 Hz berechnet. Dieser Bereich ist in 1 Hz Bereiche unterteilt für welche jeweils die Band Leistung berechnet wird. Auf den folgenden Seiten werden ERD/ERS Maps von einem Probanden gezeigt und kurz auf die einfach erkennbaren und in der Literatur beschriebenen Muster hingewiesen.

Abbildung 22 zeigt die Ergebnisse der Berechnung von einem Probanden für die oben genannten Elektrodenpositionen. Dargestellt sind die Events, die sich aus den vom Spiel generierten Daten für die linke Hand ergeben. Man sieht deutlich den in der Literatur [15] beschriebenen Effekt, dass eine Bewegung der linken Hand eine Aktivierung im rechten Bereich des motorischen Cortex hervorruft. Weiters sieht man, dass die aus der Berechnung hervorgehenden Events sich auch mit der Theorie decken, dass vor dem sichtbaren Beginn einer Bewegung bereits eine Aktivierung im entsprechenden Areal vorhanden ist. Dadurch ist der Bewegungsanfang in den Grafiken immer kurz vor dem aus den Daten hervorgehenden eingezeichnet. Es sind wie erwartet hauptsächlich Frequenzbänder in einem Bereich zwischen 10 und 13 Herz [Hz] vorhanden. Ein weiterer beobachtbarer Effekt ist, dass sich das entstehende Potential über einen größeren Bereich im Hirn erstreckt. Es kann in allen gemessenen Positionen des motorischen Cortex eine Aktivierung festgestellt werden.

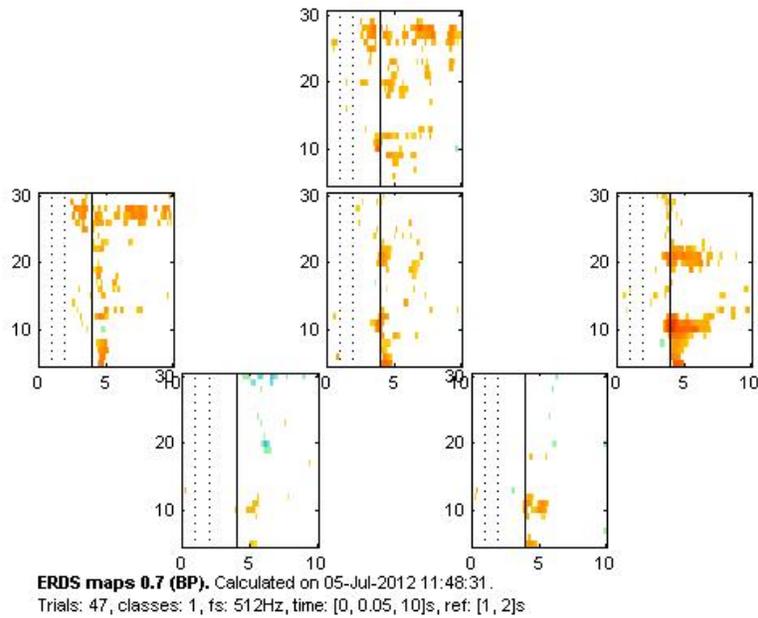


Abbildung 22: ERDS-Maps für die aus den Bewegungsdaten berechneten linke Hand Events. Es ist deutlich eine Aktivierung im Frequenzbereich zwischen 10 und 13 Hz über dem motorischen Cortex feststellbar.

Abbildung 23 zeigt wieder die Ergebnisse der Berechnung für die oben genannten Elektrodenpositionen. In diesem Fall werden aber die Events verwendet, welche die rechte Hand betreffen. In diesem Fall ist auch für die rechte Hand in der rechten Hemisphäre eine stärkere Leistungsveränderung vorhanden was sich nicht mit dem in [15] beschriebenen Effekt, dass auch eine Bewegung der rechten Hand eine contralaterale Aktivierung des motorischen Cortex hervorruft, deckt. Aber auch bei der Bewegung der rechten Hand kann eine Aktivierung kurz vor der Durchführung festgestellt werden. Die Frequenz der Hirnstromwellen befindet sich auch in diesem Fall wie erwartet in einem Bereich zwischen 10 und 13 Herz [Hz].

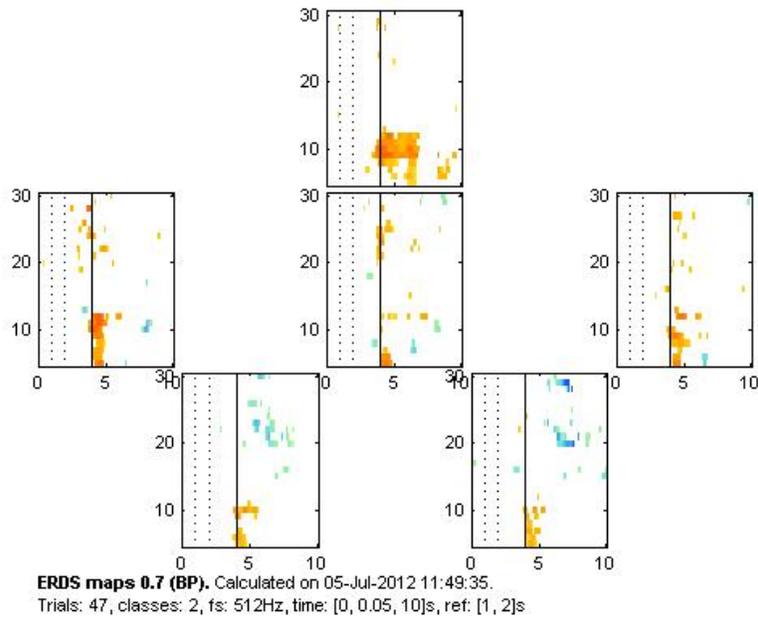


Abbildung 23: ERDS-Maps für die aus den Bewegungsdaten berechneten rechte Hand Events. Es ist deutlich, wie bei der linken Hand, eine Aktivierung im Frequenzbereich zwischen 10 und 13 Hz über dem motorischen Cortex feststellbar.

Abbildung 24 zeigt die Ergebnisse der Berechnung bei der Verwendung der Events die bei der Bewegung der Knie entstanden sind. Die Bewegung der Beine bzw. Knie sollte eine Aktivierung im Bereich von Cz erzeugen. In den Ergebnissen wird diese auch klar ersichtlich dargestellt.

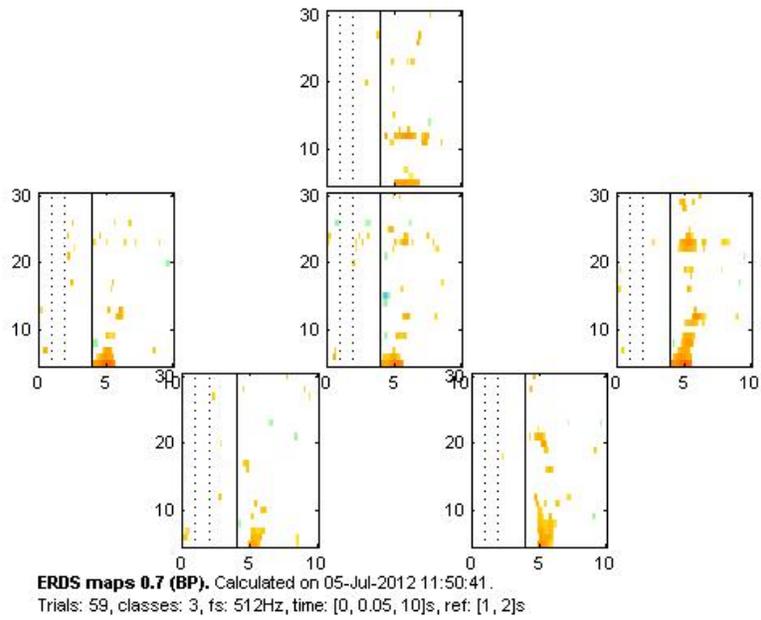


Abbildung 24: ERDS-Maps für die aus den Bewegungsdaten berechneten Knie Events. Aufgrund der schwierigeren Messbedingungen ist das Ergebniss nicht so klar wie bei den Handbewegungen.

4 Diskussion

Im Folgenden werden die verwendeten Hardware und Software Komponenten kritisch betrachtet und auf Probleme der entwickelten Methode eingegangen.

4.1 Kinect und openNI

Um Information über die Bewegung des Benutzers zu erhalten, wird in der Arbeit die Microsoft Kinect in Verbindung mit dem openNI SDK verwendet. Die Kinect wird eingesetzt, da mit diesem Gerät, wie bereits im Kapitel 2 beschrieben, einfach Information über die Szene vor der Kamera gesammelt werden kann. Vor allem die Möglichkeit die Tiefe der Szene zu erhalten, ohne komplizierte Berechnungen durchführen zu müssen, ist ein großer Vorteil gegenüber herkömmlichen Kameras. Die Übertragungsrate von ca. 30 Hz ist durchaus ausreichend, da man ab dieser Frequenz auch bei Computerspielen keine optischen Probleme auftreten (ruckeln) [33]. Das folgt aus der Tatsache, dass das menschliche Auge bzw. das Gehirn ab einer Frequenz von ca. 24 Hz keinen Übergang zwischen den einzelnen berechneten Bildern erkennen kann und die Animation der Objekte flüssig erscheint [33]. Die etwas geringe maximale Auflösung von 640x480 Bildpunkten ist für diese Arbeit nicht weiter von Bedeutung, da während der Spielzeit im Normalfall kein Bild vom Benutzer (siehe z.B. Abb. 10) angezeigt wird. Der Benutzer wird nur in den Abbildungen in der Arbeit angezeigt, um die Vorgänge während der Testphase einfacher erklären zu können.

Der Tiefensensor der Kinect erlaubt eine maximale Auflösung von 320x240. Die so erhaltenen Daten werden dann auf die vom der RGB Kamera verwendete Auflösung hochgerechnet. Die dadurch entstehenden Ungenauigkeiten bei den Übergängen von scharfen Kanten können vernachlässigt werden, da für die Anwendung in dieser Arbeit keine genaue Abgrenzung von Objekten notwendig ist. Für das zur Laufzeit durchgeführte Benutzertracking und Skelettfitting ist die vorhandene Genauigkeit durchaus ausreichend.

In der vorgestellten Arbeit wird für den Zugriff auf die von der Kinect erzeugten Daten das openNI SDK verwendet, wobei es auch möglich gewesen wäre, das von Microsoft angebotene SDK einzusetzen. Grundsätzlich kann mit beiden Paketen das gleiche Ergebnis erzielt

werden. Es gibt fertige Methoden um auf die RGB Daten und Tiefenwert-Daten der Kinect zuzugreifen und weiters ermöglichen beide Systeme ein Benutzertracking mit Skelettfittig. Ein Vorteil des Microsoft SDK ist, dass zu Beginn des Spiels keine Benutzererkennung durch eine Initialisierungspose durchgeführt werden muss. Dadurch muss der Benutzer nicht vor der Anwendung instruiert werden, welche Bewegung er wie durchführen muss, um der Software die Erkennung zu ermöglichen. Es hat sich während der Testphase herausgestellt, dass es für die Probanden durchaus schwierig sein kann, diese Initialisierungspose bei der ersten Anwendung des Spiels richtig auszuführen. Ein Nachteil der Software von Microsoft bei Beginn der Implementierung war, dass es keinen automatisches Mapping zwischen RGB Bild und Tiefenwerten gibt. Dieser nicht durchgeführte Abgleich führt dazu, dass die Position des Benutzers in RGB Bild und im Tiefensensorbild verschoben ist. Dadurch stimmen die aus den Tiefenwerten berechneten Positionen für das Skelett nicht mit dem RGB Bild überein. Da zu Beginn der Arbeit noch nicht ganz klar war, ob ein Avatar verwendet werden würde, oder ob zusätzliche Grafikelemente in das RGB Bild des Benutzers eingefügt werden, hätte dieser Punkt dazu geführt, dass weitere Berechnungen zur Laufzeit ausgeführt werden müssen, wodurch eine mögliche Beeinträchtigung der Framerate entstanden wäre, was vielleicht zu einem Absinken unter die oben angegebene und für einen flüssigen Ablauf benötigte 30 Hz Grenze [33] geführt hätte. Ein weiterer Nachteil des Microsoft SDK war, dass das auch dasTracking und damit verbunden auch die Skelettberechnung bei Beginn der Implementierung noch nicht so stabil war, wie beim openNI SDK. Aufgrund dieser inzwischen behobenen Nachteile wurde in der Arbeit das openNI SDK eingesetzt.

4.2 Spiel

Im Spiel werden viele Basistechniken verwendet die in verbesserter Form auch in den meisten aktuellen Computerspielen und Filmproduktionen Anwendung finden. Das Problem bei den meisten dieser Techniken ist, dass sie bei nicht optimaler Anwendung zu starken Einbußen in der Peformanz der Software führen. Da eine Implementierung entsprechend dem aktuell Möglichen den Rahmen der Arbeit um ein Vielfaches überschritten hätte, sind in den meisten Fällen noch Verbesserungen möglich.

Im Spiel findet man nur einfache Objekte (Kugeln) da diese wie im Kapitel 3 bereits

beschrieben eine einfache Kollisionserkennung ermöglichen. Jede andere Form von Objekten benötigt einen viel komplexeren Aufbau dieser Berechnung wie in z.B. [10] beschrieben. Ein weiterer wichtiger Punkt ist die Berechnung von Lichteffekten und damit verbundenen Schatten. In der derzeitigen Version wird für die Beleuchtung das in OpenGL fertig implementierte Phong Lighting Model verwendet Phong [22]. Es wäre mit dem aktuellen Techniken auch möglich einen Shader (z.B. PerPixel Shader) für die Beleuchtung zu implementieren. Da sich aber in der gerenderten Szene keine Objekte befinden, die sich bei der Verwendung eines komplexeren Beleuchtungsmodells optisch signifikant ändern würden, wird auf diese Möglichkeit verzichtet. Wie schon bei der Beleuchtung kann auch die Berechnung von Schatten sehr aufwendig gestaltet werden [4, 5]. Im Spiel werden die Schatten für eine Lichtquelle die sich oberhalb der Szene befindet berechnet. Das erfordert zwei komplette Renderdurchgänge (1x aus Kamerasicht und 1x von der Lichtposition) in denen die gesamte Szene dargestellt werden muss. Weiters müssen beide erzeugten Szenen miteinander verglichen und anschließend das Ergebnis auf dem Bildschirm angezeigt werden. Werden weitere Lichtquellen miteinbezogen steigert sich der Rechenaufwand um ein Vielfaches (bei einer Lichtquelle $O(n)$, bei m Lichtquellen $O(m*n)$). Für jede Quelle muss ein weiteres Mal die Szene gerendert werden. Weiters besteht die Möglichkeit den erzeugten Schatten realistischer wirken zu lassen, indem man sogenannte Softshadows implementiert. Dabei werden die Ränder aufgehellt und durch die Erzeugung einer Penumbra ein fließender Übergang zwischen Untergrund und Schatten erzeugt [24]. Es wurden keine Softshadows verwendet, da das Ziel der Arbeit war, ein einfach erweiterbares Framework zu erstellen. Die Möglichkeit der Implementierung von zusätzlichen Techniken, die die Optik des Spiel aufwerten, kann in auf dieses Framework aufbauenden Arbeiten durchgeführt werden.

Das Hauptobjekt des Spiels ist der Avatar. Dieser repräsentiert den Spieler und er reagiert auch auf dessen Bewegungen. Für die Darstellung wird ein einfaches Balls and Sticks Modell eingesetzt. Das hat den Vorteil, dass auch hier die Kollisionserkennung relativ einfach ist und weiters bei der Animation keine optisch störenden Artefakte an den Gelenken auftreten können [3]. Ein Problem, dass sich aufgrund des nicht immer stabilen Skelettfittig ergibt besteht darin, dass vor allem bei sitzenden Probanden gelegentlich die Sessecken in die Berechnung miteingeflossen sind und diese dann auch als Kniegelenke aus der Berechnung hervorgegan-

gen sind. Die Probanden wurden zu Beginn darauf hingewiesen sich möglichst am Anfang der Sitzfläche zu positionieren. Im Verlauf des Tests ergab sich aufgrund der Bewegung eine Positionsveränderung und damit verbunden eine Fehlverhalten des Algorithmus. Dieser Fehler resultiert daraus, dass bei der Vorder-/Hintergrundsegmentierung der Sessel als Teil des Probanden miterkannt wird und dadurch auch weiter für den Skelettfittig Algorithmus verwendet wird. Durch diesen nicht behebbaren Fehler wird für die Messung ein Kompromiss eingegangen. Die Probanden werden vor der Messung, wie bereits im Kapitel 2.3 beschrieben, instruiert nur dann eine Bewegung mit einem Körperteil auszuführen wenn auch ein entsprechendes Objekt am Bildschirm ist. Daher wird angenommen, dass der Proband nur dann eine Kniebewegung ausführt wenn diese auch gewünscht ist. Ein weiterer Berechnungsvorgang um zu verhindern, dass nicht gewünschte bzw. nicht vorhandene Bewegungen in die Daten eingehen ist, dass für jeden Frame überprüft wird, ob die aus den Fittigvorgang resultierende Knieposition auch möglich ist.

Um den Ablauf des Spiels nicht vorhersehbar zu machen, werden Objekte immer an verschiedenen Positionen erzeugt. Dafür wird ein einfacher von der Zeit abhängiger Randomizer verwendet. Es hat sich während der Messungen herausgestellt, dass es gelegentlich vorkommt, dass für einen längeren Zeitraum nur eine Klasse von Objekten erscheint. Das führte zu einem klar sichtbaren Motivationsverlust bei den Probanden. Da sich aber am Ende der Messung für jede Klasse eine ähnliche Anzahl an gültigen Events ergibt, besteht kein Problem für die Ergebnisse. Aufgrund des Verhaltens der Probanden, die bereits nach einer geringen Anzahl von Wiederholungen derselben Aufgabe ein klar erkennbares Aufmerksamkeitsdefizit aufwiesen, ist es sinnvoll den Test interessanter zu gestalten um die Motivation während der Messung aufrecht zuerhalten.

4.3 Messung und Ergebnisse

Der in Kapitel vier beschriebene Messvorgang ist relativ einfach durchzuführen. Der zeitaufwändigste Teil ist das Montieren der Elektroden für die Aufzeichnung vom EEG Signal des Probanden. Alle anderen Komponenten benötigen keine zusätzliche Kalibrierung von der die Messung durchführenden Person. Auch der Proband muss keine komplexen Aufgaben erfüllen, wodurch sich die Erklärung vereinfacht. Es muss nur die Pose für die Initialisierung

(Abb.13) durchgeführt werden um das Tracking zu aktivieren. Alle nachfolgenden Aufgaben, während des Tests, lassen sich einfach beschreiben, da der Proband nur einen Arm oder ein Bein bewegen muss.

In Anhang werden die Ergebnisse der Messungen der vier weiteren Probanden gezeigt, welche ebenfalls die in der Literatur beschriebenen Muster beinhalten.

4.4 Schlussfolgerung

Anhand des erstellten Paradigmas und den anschließend durchgeführten Auswertungen der Daten konnte gezeigt werden, dass durch den Einsatz von neuen Methoden brauchbare Ergebnisse erzielt werden können. Die erhaltenen Bilder (ERD/ERS Maps) können für Forschung und medizinische Zwecke durchaus verwendet werden. Ob die Ergebnisse durch die während der Testphase erhaltene Aufmerksamkeit, verbessert werden, kann durch die geringe Anzahl von durchgeführten Messungen nicht eindeutig bestätigt werden. Das verwendete Paradigma ist in dieser Form neu und daher können auch keine Vergleiche mit bereits vorhandenen Daten durchgeführt werden.

In den Daten von allen gemessenen Probanden sind die erwarteten Muster zu finden, daher kann davon ausgegangen werden, dass die Ergebnisse, die mit dieser neuen Methode erzielt werden können, richtig sind. Die Probleme die sich während der Testphase ergaben, sind hauptsächlich auf schlechte Elektrodenmontage zurückzuführen. Auch für nicht mit dem Programm vertrauten Personen, war die Durchführung des Tests, aufgrund der wenigen veränderlichen Parameter, nach einer kurzen Erklärung ohne Schwierigkeiten möglich. Weiters kann aus den nachfolgenden Gesprächen mit den Probanden geschlossen werden, dass diese Methode der EEG Messung für interessanter und dadurch motivierender empfunden worden ist, und es daher sinnvoll ist weiter auf diesem Gebiet Forschung zu betreiben.

Literatur

- [1] B. Graimann, J.E. Huggins, S. L. G. P. (2002). Visualization of significant erd/ers patterns in multichannel eeg and ecog data. *Clin. Neurophysiol*, 113:43–47.
- [2] C. Breitwieser, I. Daly, C. N. (2012). Proposing a standardized protocol for raw biosignal transmission. *IEEE Trans Biomed Eng.*
- [3] Cappozzo, A., Catani, F., Leardini, A., Benedetti, M., and Croce, U. D. (1996). Position and orientation in space of bones during movement: experimental artefacts. *Clinical Biomechanics*, 11(2):90 – 100.
- [4] Donnelly, W. and Lauritzen, A. (2006). Variance shadow maps. In *Proceedings of the 2006 symposium on Interactive 3D graphics and games*, I3D '06, pages 161–165. ACM.
- [5] Fernando, R. (2005). Percentage-closer soft shadows. In *ACM SIGGRAPH 2005 Sketches*, SIGGRAPH '05. ACM.
- [6] Fröscher, Vassella, H. (2004). *Die Epilepsien Grundlagen-Klinik-Behandlung*. Schattauer.
- [7] G. Pfurtscheller, F. L. d. S. (1999). Event-related eeg/meg synchronization and desynchronization: basic principles. *Clin. Neurophysiol*, 110:1842–1857.
- [8] Gouraud, H. (1971). Continuous shading of curved surfaces. *IEEE, Computers*, 20(6):623–629.
- [9] Gourret, J.-P., Thalmann, N. M., and Thalmann, D. (1989). Simulation of object and human skin formations in a grasping task. *SIGGRAPH Comput. Graph.*, 23(3):21–30.
- [10] Govindaraju, N. K., Redon, S., Lin, M. C., and Manocha, D. (2003). Cullide: interactive collision detection between complex models in large environments using graphics hardware. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware*, HWWS '03, pages 25–32, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.
- [11] H. Zhou, H. H. (2007). Human motion tracking for rehabilitation: A survey. *ScienceDirect Biomedical Signal Processing and Control*, page 18.

- [12] Hansen, Z. (2012). *Klinische Elektroenzephalographie*. Springer.
- [13] Hubbard, P. M. (1996). Approximating polyhedra with spheres for time-critical collision detection. *ACM Trans. Graph.*, 15(3):179–210.
- [14] Hubbard, P. (1993). Interactive collision detection. *IEEE Symposium on Research Frontiers in Virtual Reality*, pages 24–31.
- [15] Kandel, E., Schwartz, J., and Jessell, T. (2000). *Principles of Neural Science*. McGraw-Hill, Health Professions Division.
- [16] Lasseter, J. (1987). Principles of traditional animation applied to 3d computer animation. *SIGGRAPH Comput. Graph.*, 21(4):35–44.
- [17] Matthew Moore, J. W. (1988). Collision detection and response for computer animation. *ACM SIGGRAPH Computer Graphics*, 22:289 – 298.
- [18] Michalis Raptis, Darko Kirovski, H. H. (2011). Real-time classification of dance gestures from skeleton animation. *Eurographics/ ACM SIGGRAPH Symposium on Computer Animation*.
- [19] Niedermeyer E., L. D. S. F. H. (2004). *Electroencephalography: Basic Principles, Clinical Applications, and Related Fields*. Lippincott Raven.
- [20] openNI (2012). <http://openni.org/documentation>.
- [21] Palmer, I. J. and Grimsdale, R. L. (1995). Collision detection for animation using sphere-trees. *Computer Graphics Forum*, 14(2):105–116.
- [22] Phong, B. T. (1975). Illumination for computer generated pictures. *Commun. ACM*, 18(6):311–317.
- [23] Primesense (2012). <http://www.primesense.com/nite/>.
- [24] Ren, Z., Wang, R., Snyder, J., Zhou, K., Liu, X., Sun, B., Sloan, P.-P., Bao, H., Peng, Q., and Guo, B. (2006). Real-time soft shadows in dynamic scenes using spherical harmonic exponentiation. *ACM Trans. Graph.*, 25(3):977–986.

- [25] Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. *SIGGRAPH Comput. Graph.*, 21(4):25–34.
- [26] R.J. Rost, J.M. Kessenich, B. L. h. M. M. W. M. B. (2007). *OpenGL Shading Language*.
- [27] Schell, J. (2008). *The Art of Game Design*. Morgan Kaufmann Publishers.
- [28] Simulink (2012). www.mathworks.de/products/simulink.
- [29] Smith A., Kitamura Y., T. H. K. F. (1995). A simple and efficient method for accurate collision detection among deformable polyhedral objects in arbitrary motion. *Virtual Reality Annual International Symposium*, pages 136–145.
- [30] Terzopoulos, D., Platt, J., Barr, A., and Fleischer, K. (1987). Elastically deformable models. *SIGGRAPH Comput. Graph.*, 21(4):205–214.
- [31] usbAmp (2012). <http://www.neurospec.com>.
- [32] van den Bergen, G. (1998). Efficient collision detection of complex deformable models using aabb trees. *J. Graph. Tools*, 2(4):1–13.
- [33] Werler, K.-H. (1975). *Probleme der grafischen Datenverarbeitung*. Akademie-Verlag.
- [34] Williams, L. (1978). Casting curved shadows on curved surfaces. *SIGGRAPH Comput. Graph.*, 12(3):270–274.
- [35] Wolberg, G. (1990). Digital image warping. *IEEE Computer Society Press*.
- [36] Xia, L., Chen, C., and Aggarwal, J. (2011). Human detection using depth information by kinect. pages 15–22.
- [37] Yao-Jen Changa, Shu-Fang Chenb, J.-D. H. (2011). A kinect-based system for physical rehabilitation: A pilot study for young adults with motor disabilities. *Research in Developmental Disabilities*, 6:2566–2570.

5 Anhang

Im folgenden werden die noch nicht gezeigten ERD/ERS Maps der übrigen 4 Probanden abgebildet.

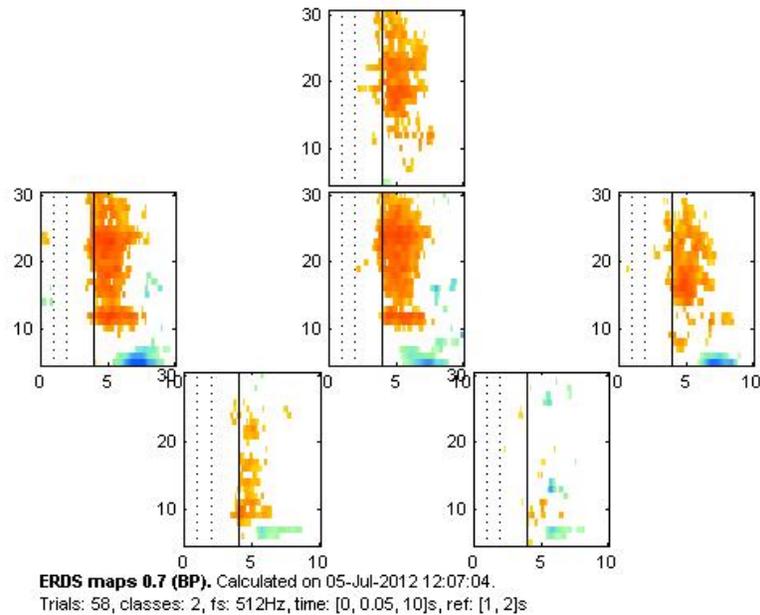


Abbildung 25: ERDS-Maps für die aus den Bewegungsdaten berechneten rechte Hand Events für Proband 1

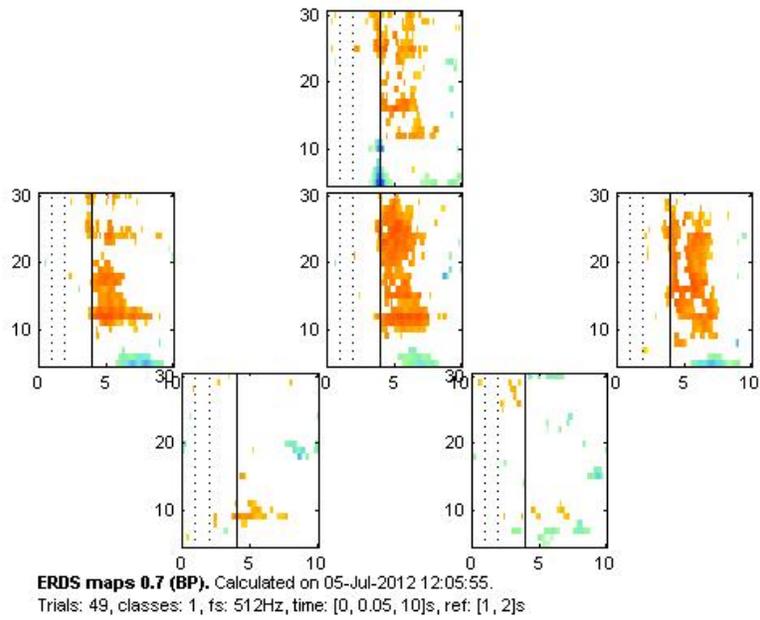


Abbildung 26: ERDS-Maps für die aus den Bewegungsdaten berechneten linke Hand Events für Proband 1

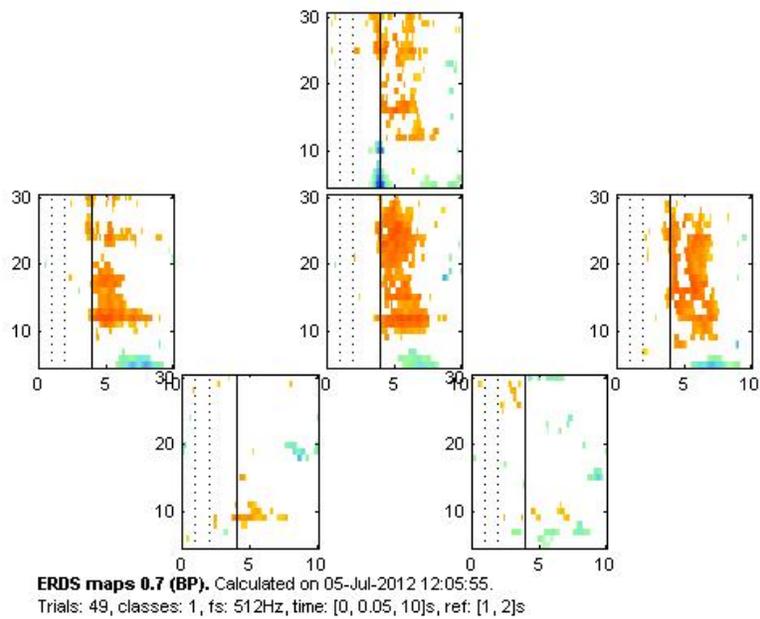


Abbildung 27: ERDS-Maps für die aus den Bewegungsdaten berechneten Knie Events für Proband 1

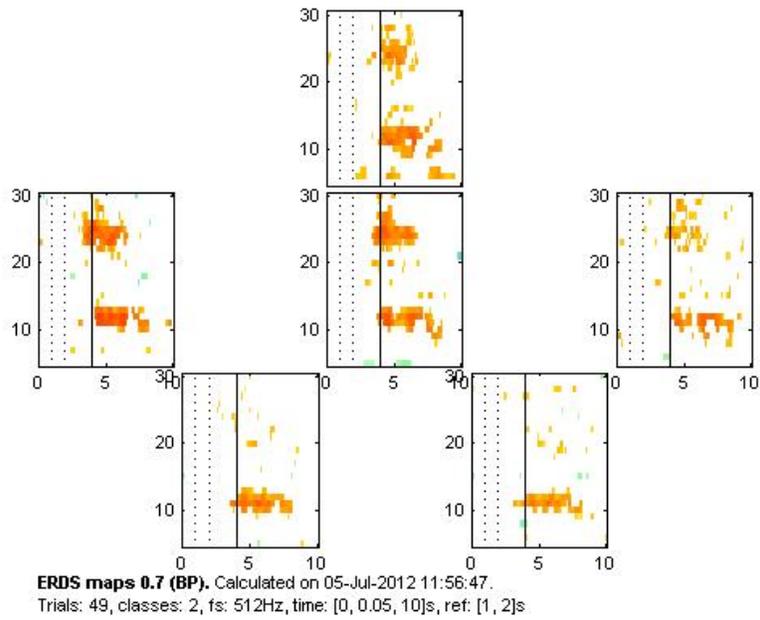


Abbildung 28: ERDS-Maps für die aus den Bewegungsdaten berechneten rechte Hand Events für Proband 2

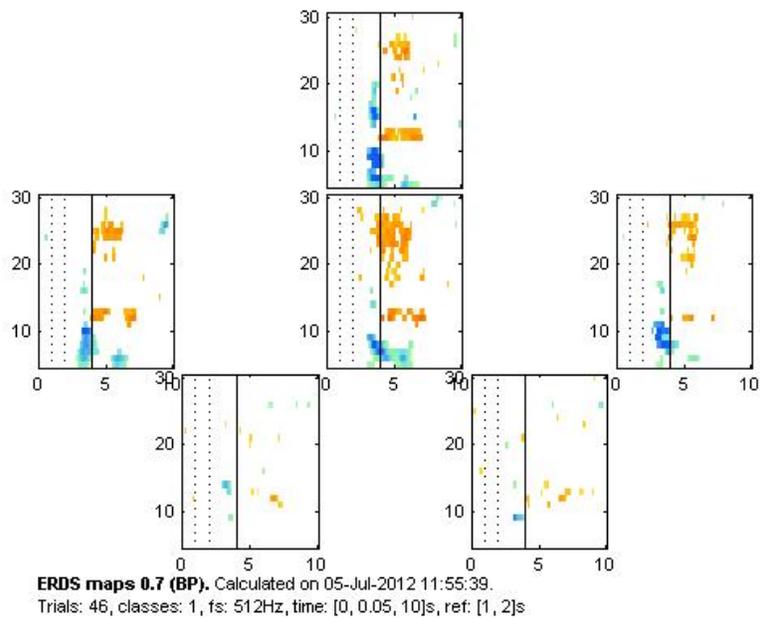


Abbildung 29: ERDS-Maps für die aus den Bewegungsdaten berechneten linke Hand Events für Proband 2

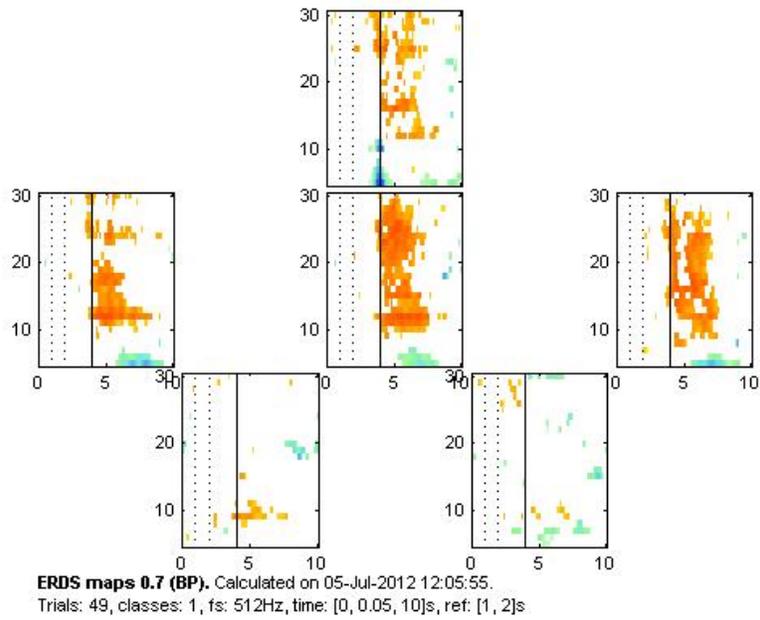


Abbildung 30: ERDS-Maps für die aus den Bewegungsdaten berechneten Knie Events für Proband 2

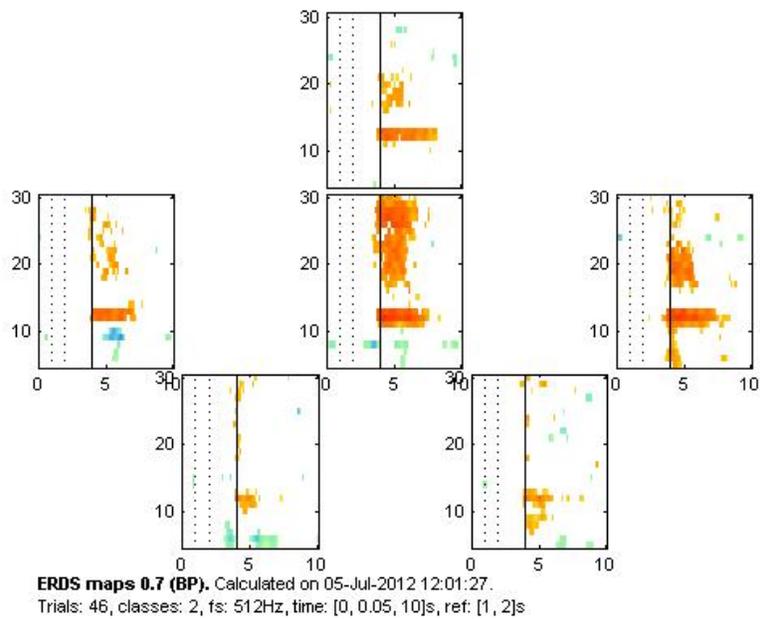


Abbildung 31: ERDS-Maps für die aus den Bewegungsdaten berechneten rechte Hand Events für Proband 3

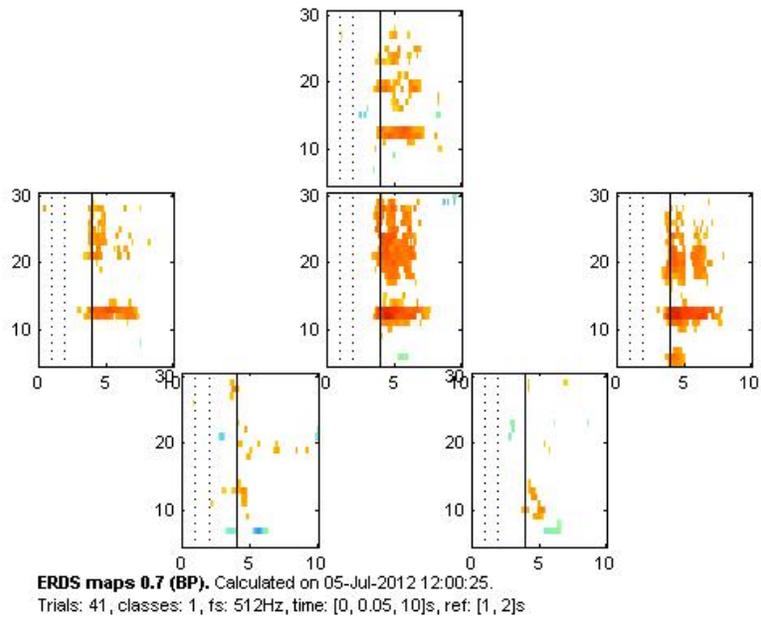


Abbildung 32: ERDS-Maps für die aus den Bewegungsdaten berechneten linke Hand Events für Proband 3

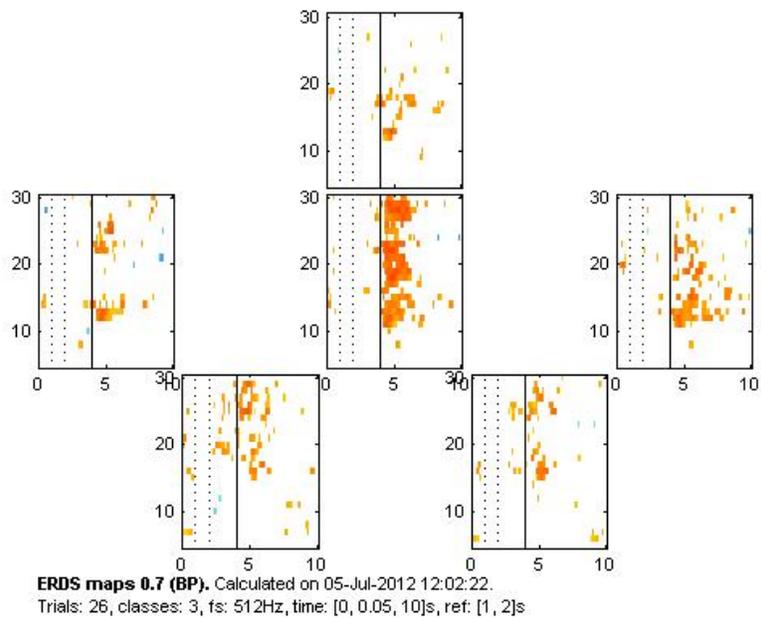


Abbildung 33: ERDS-Maps für die aus den Bewegungsdaten berechneten Knie Events für Proband 3

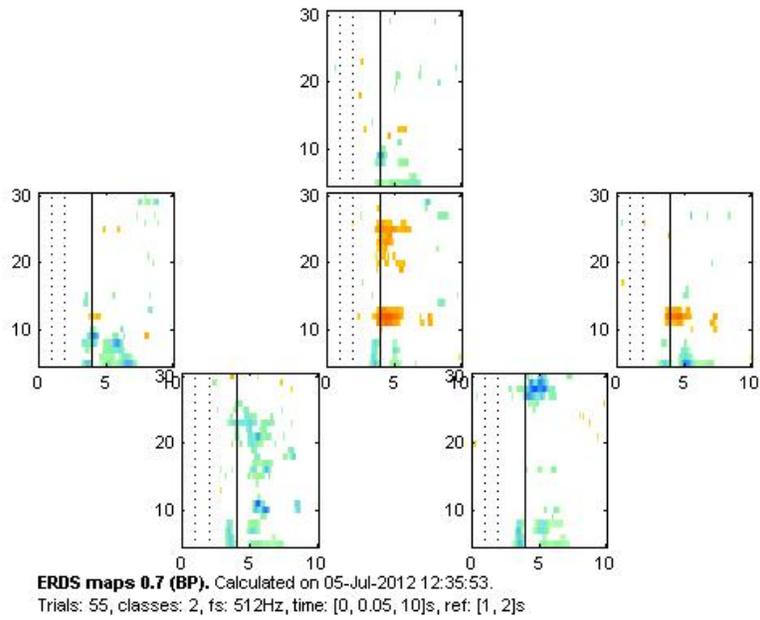


Abbildung 34: ERDS-Maps für die aus den Bewegungsdaten berechneten rechte Hand Events für Proband 4

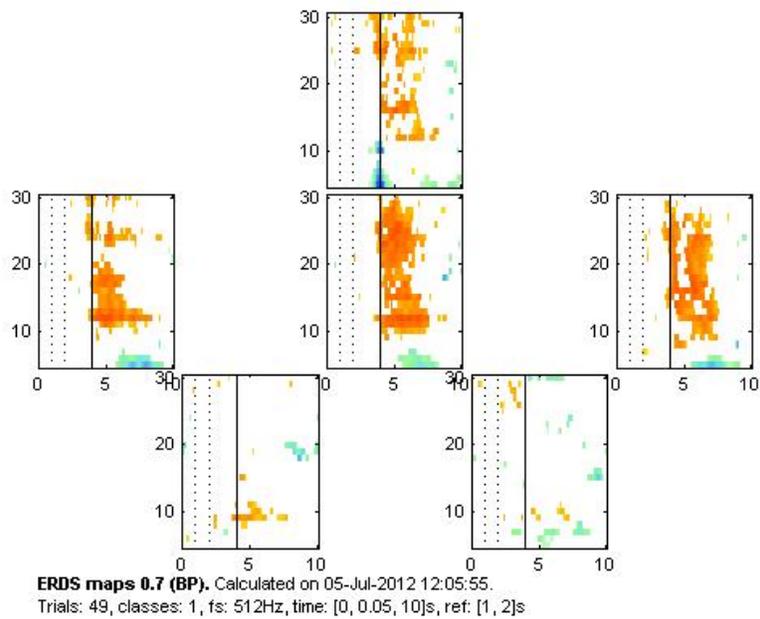


Abbildung 35: ERDS-Maps für die aus den Bewegungsdaten berechneten linke Hand Events für Proband 4

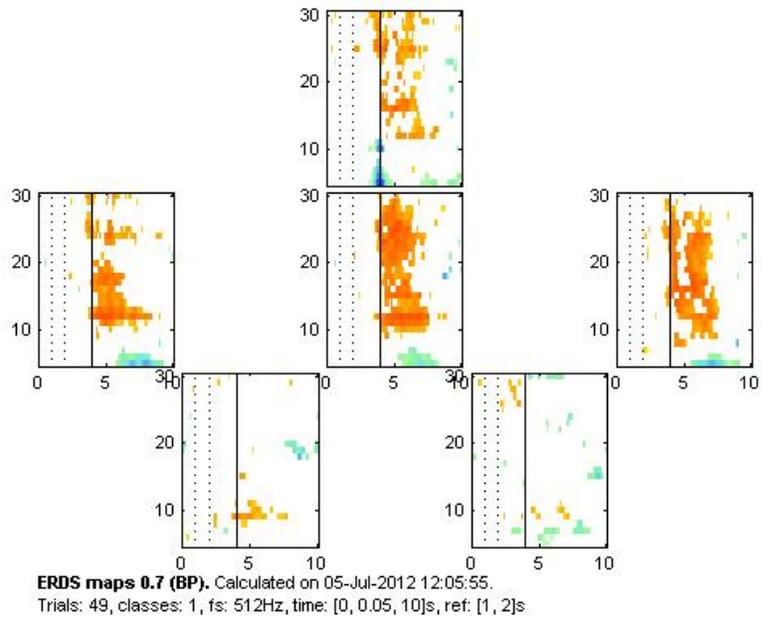


Abbildung 36: ERDS-Maps für die aus den Bewegungsdaten berechneten Knie Events für Proband 4